



**УНИВЕРЗИТЕТ У НОВОМ САДУ  
ТЕХНИЧКИ ФАКУЛТЕТ „МИХАЈЛО ПУПИН“  
ЗРЕЊАНИН**



**РАЗВОЈ АДАПТИБИЛНОГ ДИСТРИБУИРАНОГ  
ИНФОРМАЦИОНОГ СИСТЕМА ЗА ПОДРШКУ УПРАВЉАЊУ  
РЕАЛИЗАЦИЈОМ СОФТВЕРСКИХ ПРОЈЕКТА**

**DEVELOPMENT OF AN ADAPTABLE DISTRIBUTED INFORMATION  
SYSTEM FOR SOFTWARE PROJECT MANAGEMENT SUPPORT**

ДОКТОРСКА ДИСЕРТАЦИЈА

**МЕНТОР**  
Проф. др Биљана Радуловић

**КАНДИДАТ**  
Мр Љубица Кази

ЗРЕЊАНИН, 2015. године

УНИВЕРЗИТЕТ У НОВОМ САДУ  
ТЕХНИЧКИ ФАКУЛТЕТ „МИХАЈЛО ПУПИН“, ЗРЕЊАНИН

**КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА**

Редни број: <b>РБР</b>	
Идентификациони број: <b>ИБР</b>	
Тип документације: <b>ТД</b>	Монографска документација
Тип записа: <b>ТЗ</b>	Текстуални штампани материјал
Врста рада (дипл., маг., докт.): <b>ВР</b>	Докторска дисертација
Име и презиме аутора: <b>АУ</b>	Мр Љубица Кази
Ментор (титула, име, презиме, звање): <b>МН</b>	Проф. др Биљана Радуловић, редовни професор
Наслов рада: <b>НР</b>	Развој адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката
Језик публикације: <b>ЈП</b>	српски (ћирилица)
Језик извода: <b>ЈИ</b>	српски и енглески
Земља публикавања: <b>ЗП</b>	Србија
Уже географско подручје: <b>УГП</b>	Војводина
Година: <b>ГО</b>	2015.
Издавач: <b>ИЗ</b>	ауторски репринт

Место и адреса: <b>МА</b>	Ђуре Ђаковића бб Зрењанин, 23000
Физички опис рада: (број поглавља/ страница/ литературних референци/слика/ табела/прилога/листинга) <b>ФО</b>	9/427/618/205/142/14/17
Научна област: <b>НО</b>	Информационе технологије
Научна дисциплина: <b>НД</b>	Информациони системи, Софтверско инжењерство, Пројектни менаџмент
Предметна одредница/Кључне речи: <b>ПО</b>	Адаптибилни системи, Дистрибуирани информациони системи, Управљање софтверским пројектима, Софтверске метрике
<b>УДК</b>	
Чува се: <b>ЧУ</b>	Библиотека Техничког факултета „Михајло Пупин“, Зрењанин
Важна напомена: <b>ВН</b>	
Извод: <b>ИЗ</b>	<p>У овом раду приказана је анализа резултата истраживања, постојећих техничких решења и искустава из ИТ индустрије у појединачним и комбинованим областима: дистрибуирани развој софтвера и управљање софтверским пројектима у дистрибуираном окружењу, метричка заснованост управљања софтверским пројектима, адатибилност у управљању агилним софтверским пројектима, адаптибилни дистрибуираних информациони системи.</p> <p>Предложен је функционално-технолошки теоријски модел софтверске подршке адаптибилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката. Предложени су приступи у процесу развоја система у оквиру креирања модела софтверских функција и концептуалног модела података. Предложени су метрички модели за евалуацију артефакта у развоју софтвера у области развоја информационих система, као и метрички модели за процену трајања пројекта и мониторинг успеха процеса у дистрибуираном развоју софтвера.</p> <p>Описана је имплементација почетног прототипа система (реализованог као web апликација <a href="http://www.it-project.rs">www.it-project.rs</a>), у оквиру ког су реализоване основне функције система. Извршена су емпиријска истраживања могућности коришћења прототипа система у настави и професионалном усавршавању студената и</p>

	<p>анализа резултата пробног коришћења прототипа од стране ИТ кадрова уз пратеће анкетирање. Извршена је бенчмаркинг анализа најчешће коришћених расположивих алата издвојених анкетирањем. Извршена је анализа резултата емпиријских истраживања у односу на ефикасност примене прототипа. Извршена је анализа резултата примене метричких модела за евалуацију артефакта развоја софтвера, мониторинг успеха процеса и процену трајања пројекта у делу процеса имплементације софтвера. Реализовано је унапређење прототипа у односу на претходно дефинисане захтеве функционално-технолошког теоријског модела система.</p>
<p>Датум прихватања теме од стране Сената: <b>ДП</b></p>	17.12.2009.
<p>Датум одбране: <b>ДО</b></p>	
<p>Чланови комисије: (име и презиме/титула / звање/ назив организације) <b>КО</b></p>	
<p><b>Председник:</b></p>	Проф. др Миодраг Ивковић, редовни професор, Универзитет у Новом Саду, Технички факултет „Михајло Пупин“, Зрењанин
<p><b>Члан:</b></p>	Проф. Др Алемпије Вељовић, редовни професор, Универзитет у Крагујевцу, Факултет техничких наука, Чачак
<p><b>Члан:</b></p>	Проф. Др Бранко Маркоски, ванредни професор, Универзитет у Новом Саду, Технички факултет „Михајло Пупин“, Зрењанин
<p><b>Члан:</b></p>	Доц. Др Весна Макитан, доцент, Универзитет у Новом Саду, Технички факултет „Михајло Пупин“, Зрењанин
<p><b>Члан, ментор:</b></p>	Проф. др Биљана Радуловић, редовни професор, Универзитет у Новом Саду, Технички факултет „Михајло Пупин“, Зрењанин

UNIVERSITY OF NOVI SAD  
TECHNICAL FACULTY „MIHAJLO PUPIN“, ZRENJANIN

**KEY WORD DOCUMENTATION**

Accession number: <b>ANO</b>	
Identification number: <b>INO</b>	
Document type: <b>DT</b>	Monograph documentation
Type of record: <b>TR</b>	Textual printed material
Contents code: <b>CC</b>	Ph.D. Thesis
Author: <b>AU</b>	Ljubica Kazi, M.Sc.
Mentor: <b>MN</b>	Biljana Radulović, Ph.D., full professor
Title: <b>TI</b>	Development of an Adaptable Distributed Information System for Software Project Management Support
Language of text: <b>LT</b>	Serbian (cyrillic letters)
Language of abstract: <b>LA</b>	Serbian and English
Country of publication: <b>CP</b>	Serbia
Locality of publication: <b>LP</b>	Vojvodina
Publication year: <b>PY</b>	2015.
Publisher: <b>PB</b>	Author reprint

Publication place: <b>PP</b>	Zrenjanin, 23000 Djure Djakovića bb
Physical description: (chapters/pages/ references/images/ tables/ appendixes/listings) <b>PD</b>	9/427/618/205/142/14/17
Scientific field: <b>SF</b>	Information Technologies
Scientific discipline: <b>SD</b>	Information Systems, Software Engineering, Project Management
Subject, Key words: <b>S/KW</b>	Adaptable Systems, Distributed information systems, Software project management, Software Metrics
<b>UC</b>	
Holding data: <b>HD</b>	Library of Technical faculty „Mihajlo Pupin“, Zrenjanin
Note: <b>N</b>	
Abstract: <b>AB</b>	<p>This work presents analysis of research results, existing technical solutions and experiences from IT industry in separate and combined fields of: distributed software development, software project management in distributed environment, metric-based software project management, adaptability in management of agile software projects, as well in the field of adaptable distributed information systems.</p> <p>Functional-technological theoretical model of software support to adaptable distributed information system for software projects realization has been proposed. Approaches to creating software functions model and conceptual data model within the process of the system development were proposed. Metric models for evaluation of artefacts created in software development within information system development, as well as metric models for project duration estimation and monitoring of process success in distributed software development were proposed.</p> <p>Implementation of initial prototype of the system (developed as web application <a href="http://www.it-project.rs">www.it-project.rs</a>), that includes basic functions, was described. Empirical research on possibilities for using prototype in educational environment and professional improvement of students, as</p>

	<p>well as analysis of results from using the prototype by IT professionals, with additional questionnaire, has been conducted. Benchmarking analysis of most frequently used available tools, extracted from questionnaire results, has been performed. Analysis of empirical research results has been performed in the context of prototype using efficiency. Analysis of results in metric models application in evaluation of software development artefacts, process success monitoring and project duration estimation, in software implementation process part, has been conducted. Improvement of prototype has been implemented according to requirements defined in previously proposed functional-technological theoretical model of the system.</p>
<p>Accepted on Senate on: <b>ASB</b></p>	<p>December 17, 2009</p>
<p>Defended on: <b>DE</b></p>	
<p>Thesis defended board: (name and surname /degree/title/ faculty) <b>DB</b></p>	
<p><b>President:</b></p>	<p>Miodrag Ivković, Ph.D, full professor, University of Novi Sad, Technical faculty „Mihajlo Pupin“, Zrenjanin</p>
<p><b>Member:</b></p>	<p>Alempije Veljović, Ph.D, full professor, University of Kragujevac, Faculty of Technical Sciences, Cacak</p>
<p><b>Member:</b></p>	<p>Branko Markoski, Ph.D, associate professor, University of Novi Sad, Technical faculty „Mihajlo Pupin“, Zrenjanin</p>
<p><b>Member:</b></p>	<p>Vesna Makitan, Ph.D, assistant professor, University of Novi Sad, Technical faculty „Mihajlo Pupin“, Zrenjanin</p>
<p><b>Member, Menthor:</b></p>	<p>Biljana Radulović, Ph.D., full professor, University of Novi Sad, Technical faculty „Mihajlo Pupin“, Zrenjanin</p>

# ЗАХВАЛНИЦА

Ментору

Проф.др Биљани Радуловић

*на поверењу, стрпљењу, подршци, сугестијама и саветима*

Члановима комисије

Проф. др Миодраг Ивковић, проф. др. Алемпије Вељовић, проф. др Бранко Маркоски, доц. др Весна Макитан

*на колегијалној подршци, сугестијама и саветима*

Професорима и колегама са Факултета

проф. др Момчило Бјелица, проф. др Милан Павловић, проф. др Душан Малбашки, проф. др Душко Летић, проф. др Ивана Берковић, проф. др Петар Хотомски<sup>+</sup>, проф. др Борислав Одаџић, проф. др Драгица Радосав, проф. др Драгана Глушац, проф. др Дијана Каруовић, проф. др Владимир Бртка, проф. др Драган Ђоћкало, доц. др Далибор Добриловић, доц. др Жељко Стојанов, доц. др Марјана Пардањац, Мр Дејан Лацмановић, Мр Елеонора Бртка, Мр Вишња Огњеновић, Мр Ерика Елевен

*на срдечној колегијалној подршци*

Колегама и пријатељима из земље и иностранства

који су запослени у ИТ фирмама, јавним предузећима, образовним институцијама и привредним субјектима

*на сарадњи у емпиријском истраживању и моралној подршци*

Студентима

Техничког факултета „Михајло Пупин“, Зрењанин, Универзитета у Новом Саду

*на сарадњи у емпиријском истраживању и инспирацији*

Породици

Мајци Олгици, супругу Золтану, ћерки Марији, ћерки Клари, сестри Јелени, тетки Зорици, ујаку Ивану, течи Слободану, ујни Олгици, сестри Снежани, брату Влајку, брату Марку, брату Александру

*на стрпљењу, помоћи и моралној подршци*

*Дугујем велику захвалност свима који су ми, у току припреме и реализације докторске дисертације, омогућили и подржали ме у радости истраживања, стваралаштва и пружања, у нади да ће резултати докторске дисертације наћи примену у научној, наставној и стручној пракси.*

*Мр Љубица Кази*



*„Без информационог система, који садржи податке о полазним планираним величинама о реализацији пројекта, и правовремене информације о стању у току реализације пројекта, није могуће покретати потребне управљачке акције, значи није могуће вршити перманентно праћење и контролу и управљати пројектом.“*

*Проф. Др Петар Јовановић*

## САДРЖАЈ

<b>1. УВОДНА РАЗМАТРАЊА</b>	1
<b>1.1. МЕТОДОЛОШКИ ОКВИР РАДА</b>	3
1.1.1. Предмет и проблем истраживања	3
1.1.3. Циљ и задаци	3
1.1.4. Хипотеза и подхипотезе	4
1.1.5. Методе и узорак	5
1.1.6. Очекивани резултати	5
<b>1.2. ОПРАВДАНОСТ И МОТИВАЦИЈА ИСТРАЖИВАЊА</b>	6
1.2.1. Анализа значаја и фактора успеха примене пројектног менаџмента	6
1.2.2. Уводно емпиријско истраживање - анализа проблема реализованих софтверских пројеката	8
<b>1.3. ТЕОРИЈСКЕ ОСНОВЕ</b>	10
<b>1.3.1. Основне области</b>	10
1.3.1.1. Развој информационих система	10
1.3.1.2. Методологија развоја софтвера	17
1.3.1.3. Управљање пројектима	28
<b>1.3.2. Технолошка основа</b>	35
1.3.2.1. Дистрибуирани информациони системи	35
1.3.2.2. Адаптивни и адаптивбилни системи и софтвер	40
1.3.2.3. Системи за управљање пословним правилима, аутоматско резонување и онтологије	47
1.3.2.4. Визуализација података	48
<b>1.3.3. Елементи управљања софтверским пројектима</b>	51
1.3.3.1. Категоризација софтвера и софтверских пројеката	51
1.3.3.2. Дистрибуирани развој софтвера	54
1.3.3.3. Управљање перформансама пословних процеса	56
1.3.3.4. Успех софтверског пројекта, софтверске метрике и контрола софтверског пројекта	58
<b>2. ПРЕГЛЕД ПОСТОЈЕЋИХ ИСТРАЖИВАЊА, ТЕХНОЛОШКИХ РЕШЕЊА И ИСКУСТАВА ИЗ ПРАКСЕ</b>	63
<b>2.1. УПРАВЉАЊЕ СОФТВЕРСКИМ ПРОЈЕКТИМА У ДИСТРИБУИРАНОМ ОКРУЖЕЊУ</b>	63
2.1.1. Анализа истраживања проблема дистрибуираног развоја софтвера и примена SE-PM матрице	63

2.1.2. Анализа истраживања у области управљања софтверским пројектима који се реализују у дистрибуираном окружењу	67
2.1.2. Анализа емпиријских истраживања ИТ индустрије	70
2.1.3. Анализа постојећих технолошких решења	74
<b>2.2. МЕТРИЧКИ ЗАСНОВАНО УПРАВЉАЊЕ СОФТВЕРСКИМ ПРОЈЕКТИМА</b>	79
2.2.1. Модел анализе постојећих истраживања	79
2.2.2. Димензије успеха софтверског пројекта	80
2.2.3. Мерење перформанси софтверског пројекта	96
2.2.4. Мерење карактеристика софтверског производа, артефакта процеса развоја софтвера и информационог система	106
2.2.5. Анализа постојећих технолошких решења	119
<b>2.3. АДАПТИВНОСТ У РЕАЛИЗАЦИЈИ И УПРАВЉАЊУ АГИЛНИМ СОФТВЕРСКИМ ПРОЈЕКТИМА</b>	122
2.3.1. Адаптивност у одржавању софтвера и закони еволуције софтвера	122
2.3.2. Карактеристике агилног приступа развоју софтвера, адаптивност и истраживања мерења агилности	122
2.3.3. Компарација агилних метода развоја софтвера	126
2.3.4. Истраживање улоге инжењерства захтева, моделовања, дизајна и документације у агилном развоју софтвера	132
2.3.4. Дисциплиновани агилни приступ развоју софтвера	134
2.3.5. Адаптивно управљање перформансама у агилном пројектном менаџменту	135
2.3.6. Адаптивност у управљању агилним софтверским пројектима	136
<b>2.4. АДАПТИВНОСТ И АДАПТИБИЛНОСТ ДИСТРИБУИРАНИХ ИНФОРМАЦИОНИХ СИСТЕМА</b>	140
2.4.1. Истраживања у области дистрибуираних информационих система	140
2.4.2. Истраживања у области адаптивних и адаптибилних информационих система	142
2.4.3. Истраживања у области адаптивних и адаптибилних дистрибуираних информационих система	148
2.4.4. Технолошка решења адаптивности и адаптибилности у дистрибуираним информационим системима	150
<b>2.5. ИСТРАЖИВАЊА, ТЕХНИЧКА РЕШЕЊА И ИСКУСТВА У КОМБИНОВАНИМ ОБЛАСТИМА</b>	159
2.5.1. Агилни развој софтвера у дистрибуираном окружењу	159
2.5.2. Агилно управљање софтверским пројектима у дистрибуираном окружењу	160
2.5.3. Методе и технике контроле квалитета и фактори успеха у дистрибуираном агилном развоју софтвера и пројектном менаџменту	161
2.5.4. Примена Balanced Scorecard у ИТ и пројектном менаџменту	163
2.5.5. Информациони системи у пројектном менаџменту	168
2.5.6. Истраживања динамичких онтолошки заснованих web сервиса и софтверских агената и примена web сервиса у пројектом менаџменту	168

<b>3. МОДЕЛ ПРЕДЛОЖЕНОГ РЕШЕЊА</b>	173
<b>3.1. Опис проблема</b>	173
3.1.1. Опис посла	173
3.1.2. Снимак стања	173
3.1.3. Спецификација захтева корисника	173
3.1.3.1. Функционално-технолошки теоријски модел софтверске подршке адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката	174
3.1.4. Анализа документације	175
<b>3.2. Модели у процесу развоја система</b>	178
3.2.1. Теоријско-истраживачка основа за креирање предложених модела развоја система	178
3.2.2. Модел пословних процеса базиран на PRINCE 2 методологији управљања пројектима	179
3.2.5. Креирање модела софтверских функција на основу пресликавања примитивних пословних процеса	183
3.2.6. Складишта података за нормализацију и концептуални модел података	188
3.2.7. Модел компоненти и размештаја у опису архитектуре предложеног система	190
<b>3.3. Метрички модели за евалуацију артефакта развоја софтвера у области развоја информационог система</b>	193
3.3.1. Теоријско-истраживачка основа предложених модела	193
3.3.2. Метрички модел за евалуацију модела пословних процеса	193
3.3.3. Метрички модел за евалуацију модела софтверских функција	194
3.3.4. Метрички модел за евалуацију концептуалног модела података	195
3.3.5. Метрички модел за евалуацију софтверске апликације	197
<b>3.4. Метрички модели за процену трајања и мониторинг успеха процеса у оквиру дистрибуираног агилног софтверског пројекта</b>	200
3.4.1. Теоријско-истраживачка основа предложених модела за процену и мониторинг успеха процеса развоја софтвера	200
3.4.2. Метрички модели за мониторинг успешности процеса у оквиру дистрибуираног агилног софтверског пројекта	201
3.4.3. Метрички модели за модел-базирану процену трајања софтверског пројекта у области развоја информационог система	206
3.4.3.1. Анализа недостатака постојећих модел-базираних приступа процени величине, трајања и трошкова реализације софтверског пројекта у развоју информационог система	206
3.4.3.2. Метрички модел за процену трајања пројекта будућег софтверског решења на основу модела пословних процеса	208

<b>4. ОПИС ИМПЛЕМЕНТАЦИЈЕ ПОЧЕТНОГ ПРОТОТИПА</b>	210
<b>4.1. Основни поступци имплементације прототипа</b>	210
4.1.1. Hosting и администрација web сајта	210
4.1.2. Креирање web апликације у развојном окружењу	212
<b>4.2. Карактеристике имплементираних прототипа</b>	214
4.2.1. Опис карактеристика имплементираних прототипа	214
4.2.2. Имплементирана база података	215
4.2.3. Административни модул	217
4.2.4. Елементи подршке Waterfall приступу	221
4.2.5. Елементи подршке мултипројектном раду, агилном приступу, визуализацији и адаптивности	225
<b>5. ЕМПИРИЈСКА ИСТРАЖИВАЊА</b>	228
<b>5.1. План реализације емпиријских истраживања</b>	228
5.1.1. Методолошке основе емпиријских истраживања	228
5.1.2. План емпиријских истраживања у односу на истраживачка питања	228
5.1.3. План емпиријских истраживања у односу на евалуацију употребљивости развијеног прототипа	229
5.1.4. План емпиријских истраживања у односу на утврђивање статуса потврђености хипотезе и подхипотеза	229
5.1.5. Сумарни приказ планираних техника и метода	231
<b>5.2. Емпиријско истраживање могућности коришћења прототипа у настави и професионалном усавршавању студената</b>	232
5.2.1. Циљеви и методе емпиријског истраживања	232
5.2.2. Узорак истраживања	232
5.2.3. Резултати истраживања	234
5.2.4. Анализа резултата са становишта истраживачких питања и хипотеза	239
<b>5.3. Пробно коришћење прототипа од стране ИТ кадрова и испитивање њихових ставова анкетирањем</b>	240
5.3.1. Методологија истраживања	240
5.3.2. Резултати истраживања	241
5.3.2.1. Приказ примера резултата пробног коришћења прототипа од стране ИТ кадрова	241
5.3.2.2. Анализа биографских података почетног узорка истраживања	243
5.3.2.3. Анализа критеријума и метода за утврђивање репрезентативности узорка и опис метода ELIG-RANK и ELIM-F	246
5.3.2.4. Сумарни приказ одговора у вези најчешће коришћених алата за подршку тимском раду и софтверском пројектном менаџменту	249

5.3.2.5. Сумарни приказ одговора у вези фактора утицаја на успех и неуспех софтверског пројекта	250
5.3.2.6. Сумарни приказ одговора у вези значаја примене и развоја, као и потребних карактеристика информационог система за подршку управљању софтверским пројектима	252
5.3.2.7. Сумарни приказ одговора у вези употребљивости развијеног прототипа, предности, недостатака и могућности унапређења	254
5.3.3. Анализа резултата са становишта истраживачких питања и хипотеза	258
<b>5.4. Бенчмаркинг анализа најчешће коришћених алата издвојених анкетирањем</b>	261
5.4.1. Методологија истраживања	261
5.4.1.1. Појам, карактеристике и истраживања примене методе бенчмаркинга у ИТ и ИТ пројектном менаџменту	261
5.4.1.2. Креирање модела бенчмаркинг анализе	262
5.4.1.3. Припрема узорка истраживања	263
5.4.2. Резултати анализе карактеристика и компарација алата	264
<b>5.5. Анализа резултата емпиријских истраживања у односу на ефикасност примене прототипа</b>	265
5.5.1. Циљ и методе истраживања	265
5.5.2. Испитивање усклађености функционалних карактеристика прототипа са сврхом и циљевима развоја апликације	265
5.5.3. Ставови ИТ кадрова у контексту процене утицаја примене прототипа на укупно време реализације пројекта	266
5.5.4. Испитивање фреквенције истовремених вишеструких активности и адекватности подршке реализованог решења	267
5.5.5. Утицај коришћења прототипа на квалитет реализованих софтверских пројеката	270
5.5.6. Анализа резултата са становишта истраживачких питања и хипотеза	272
<b>5.6. Анализа резултата примене метричких модела за евалуацију артефакта развоја софтвера</b>	273
5.6.1. Узорак евалуације модела као артефакта развоја софтвера у области развоја информационог система	273
5.6.2. Евалуација модела пословних процеса	273
5.6.3. Евалуација модела дизајна софтверских функција	275
5.6.4. Евалуација концептуалног модела података	276
5.6.5. Евалуација софтверске апликације	278
<b>5.7. Анализа резултата примене метричких модела за мониторинг успеха и процену трајања процеса имплементације софтвера у оквиру софтверског пројекта</b>	280
5.7.1. Примена модела за мониторинг успеха процеса развоја софтвера	280
5.7.2. Примена модела за процену трајања софтверског пројекта на основу модела пословних процеса	282
<b>6. УНАПРЕЂЕЊЕ ПРОТОТИПА</b>	283
<b>6.1. Планирање унапређења прототипа</b>	283

6.1.1. Статистички приказ резултата анализе степена реализације захтеваних функционално-технолошких карактеристика у односу на предложени функционално-технолошки теоријски модел	283
6.1.2. Издвајање непокривених функционално-технолошких области постојећим почетним прототипом	283
6.1.3. Анализа захтева непокривених области и план реализације унапређења	284
<b>6.2. Опис елемената имплементације унапређења прототипа</b>	<b>287</b>
6.2.1. Подршка дистрибуираности и адаптивности на промене методологије развоја софтвера - креирање и тестирање ASP.NET Web сервиса уз коришћење XML шифарника	287
6.2.2. Прилагођавање различитим технолошким окружењима - повезивање и коришћење Web сервиса у оквиру реализације различитих технологија корисничких интерфејса	289
6.2.3. Графичка адаптивност и адаптивност слоја података – динамичко креирање компоненти корисничког интерфејса PHP апликације на основу ASP.NET Web сервиса и креирање структуре XML	291
6.2.4. Метричка заснованост управљања – мерење карактеристика концептуалног модела података помоћу ASP.NET web сервиса	293
6.2.5. Динамичко повезивање и извршавање web сервиса ради омогућавања измена у делу акције у IF-THEN правилу	296
<b>7. ЗАКЉУЧНА РАЗМАТРАЊА</b>	<b>299</b>
7.1. Осврт на резултате у односу на методолошки оквир истраживања	299
7.1.1. Резултати у односу на статус потврђености хипотезе и подхипотеза	300
7.2. Анализа доприноса истраживања	301
7.2.1. Научни, практични и друштвени допринос истраживања	301
7.2.2. Анализа доприноса истраживања у оквиру објављених радова	302
7.3. Анализа предности, недостатака и могућности даљег унапређења имплементираних прототипа система	305
7.4. Анализа ограничења реализованих истраживања	306
7.5. Правци даљих истраживања	306
<b>8. РЕФЕРЕНЦЕ</b>	<b>308</b>
8.1. Литература	308
8.2. Списак објављених радова кандидата из области истраживања и цитираност радова	332
<b>9. ПРИЛОЗИ</b>	<b>335</b>
9.1. Списак анализираних софтверских пројеката у уводном емпиријском истраживању	335

9.2. Категоризација радова који анализирају појединачне проблеме дистрибуираног развоја софтвера	337
9.3. Текст анкете коришћен у емпиријском истраживању на српском језику	340
9.4. Текст анкете коришћен у емпиријском истраживању на енглеском језику	342
9.5. Подаци из емпиријског истраживања коришћења прототипа у настави и професионалном усавршавању студената	344
9.6. Два примера софтверских пројеката реализованих уз административну подршку управљању пројектом коришћењем прототипа система	348
9.7. Подаци из емпиријског истраживања пробног коришћења прототипа од стране ИТ кадрова уз пратеће анкетирање	352
9.8. Подаци о алатима из бенчмаркинг анализе	378
9.9. Подаци из анализе ефикасности коришћења прототипа анализом „истовремених вишеструких активности“	388
9.10. Категоризација фактора успеха, неуспеха и ризика у оквиру истраживања софтверских, дистрибуираних и агилних софтверских пројеката	390
9.11. Подаци из примене метричких модела у евалуацији артефакта развоја софтвера у области развоја информационих система	392
9.12. Подаци из примене метричких модела за мониторинг успеха и процену трајања софтверског пројекта	403
9.13. Листинг кода	411
9.13.1. Листинг кључних делова кода почетног прототипа	411
9.13.2. Листинг кода унапређења прототипа	418
9.14. Биографија кандидата	427



## СПИСАК СЛИКА

РБ	Ознака и назив слике	Страна
1.	1.3.1.2.3.1. Waterfall модел животног циклуса софтвера према Cadle&Yeates	21
2.	1.3.1.2.3.2. „b“ модел животног циклуса софтвера према Cadle&Yeates	22
3.	1.3.1.2.3.3. Model V животног циклуса софтвера према Castro Souto	22
4.	1.3.1.2.3.4. Инкрементални модел животног циклуса софтвера према Cadle&Yeates	23
5.	1.3.1.2.3.5. Модел итеративног развоја софтвера према Castro Souto	23
6.	1.3.1.2.3.6. Спирални модел животног циклуса софтвера према Boehm-у	24
7.	1.3.1.2.3.7. Спирални модел животног циклуса софтвера према Cadle&Yeates-у	24
8.	1.3.1.2.3.8. Животни циклус Extreme Programming према Castro Souto	25
9.	1.3.1.2.3.9. Модел Unified Process-а према Castro Souto	25
10.	1.3.1.3.1.1. Основни параметри у управљању пројектима (Гвоздени троугао) према Atkinson-у	28
11.	1.3.1.3.2.1. Основне области управљања пројектима, према PMBOK	29
12.	1.3.1.3.3.1. Структура PRINCE2 методологије, према CCTA	31
13.	1.3.2.1.4.1. Основни концепт примене Web сервиса, према Kreger	38
14.	1.3.2.2.3.1. Два приступа архитектури само-адаптивних система, према Salehie&Tahvildari	44
15.	1.3.2.2.3.2. Хијерархијска уређеност интерних карактеристика само-адаптивног софтвера, према Salehie&Tahvildari	45
16.	1.3.2.2.3.3. Процес самоадаптације софтвера кроз четири кључна подпроцеса, према Salehie&Tahvildari	45
17.	1.3.2.4.1. Редослед поступака визуализације података, према Kaidi	49
18.	1.3.3.3.1. Животни циклус управљања перформансама пословних процеса, према Eckerson-у	56
19.	1.3.3.4.3.1. Активности у контроли пројекта, према Jurison-у	61
20.	1.3.3.4.3.2. Модел радног тока у мерењу и контролосању софтверских пројеката, према Fairley-у	62
21.	2.1.1.1.1.1. Графички приказ покривености ДСД проблема у анализи прегледних радова, на основу примене SE-PM матрице	67
22.	2.1.3.1. Карактеристике система за подршку колаборацији, према Booch&Brown	75
23.	2.1.3.2. Кључни део концептуалног модела CDE (Collaborative development environment) система, према Booch&Brown	75
24.	2.2.1.1. Модел анализе постојећих истраживања у области метрички заснованог управљања софтверским пројектима	79
25.	2.2.2.1.1. Успех информационог система према DeLone & McLean моделу	80
26.	2.2.2.1.2. Успех пројекта = успех пројектног менаџмента и производа, према Westhuizen& Fitzgerald	81
27.	2.2.2.2.1.1. Однос успеха малих и великих софтверских пројеката, према истраживању 2012. године од стране Standish group, према CHAOS извештају из 2013. године	86
28.	2.2.2.4.2.1. Формула за изражавање трошкова измене софтвера, према Sharif et al	94
29.	2.2.2.4.2.2. Утицај подгрупа ризика на перформансе процеса и производа, према Wallace et al	95
30.	2.2.3.3.1. Технике процене софтвера према Boehm-у	97
31.	2.2.3.3.2. Модел за подршку процени софтверског пројекта према Abdel-Hamid & Madnick-у	98
32.	2.2.3.3.1.1. Формула за израчунавање времена испоруке на основу COSMIC методе, према Monsalve et al	99
33.	2.2.3.3.1.2. Формула за израчунавање времена испоруке на основу броја кандидата за случајеве коришћења, према Issa&Rub	100
34.	2.2.3.3.1.3. Формула за израчунавање Use Case Points према Clemmons	100
35.	2.2.3.3.1.4. Формула за израчунавање величине софтвера (изражено у хиљадама линија кода) према Tan et al	101
36.	2.2.3.4.2.1. Формула за сумарно израчунавање резултата мерења софтверских метрика у софтверском пројекту према Demir-у	104
37.	2.2.3.4.2.2. Формула за сумарно израчунавање резултата мерења квалитета активности и особина ентитета у софтверском пројекту према Demir-у	104

38.	2.2.3.4.2.3. Фазе у примени GQM методе, према Solingen& Berghout-y	105
39.	2.2.4.1.1. Модел квантитативног приступа мерењу квалитета софтвера према Cavano&McCall	106
40.	2.2.4.1.2. Формула за израчунавање вредности фактора квалитета сумирањем вредности метрика према Cavano&McCall	106
41.	2.2.4.1.3. Карактеристике и субкарактеристике квалитета софтвера према ISO/IEC 9126	107
42.	2.2.4.1.4. Елементи ISO 9126 стандарда за квалитет софтвера	107
43.	2.2.4.1.5. Основне компоненте квалитета софтвера, према ISO2010	108
44.	2.2.4.1.6. Основне компоненте квалитета софтвера као производа, према ISO2010	108
45.	2.2.4.1.7. Основне компоненте квалитета софтвера у употреби, према ISO2010	109
46.	2.2.4.1.8. Основне компоненте квалитета података, према ISO2010	109
47.	2.2.4.1.9. Фактори квалитета софтвера, према Cavano&McCall	110
48.	2.2.4.1.10. McCall-ов модел са три нивоа погледа на карактеристике квалитета софтвера, , према Sanz et al	111
49.	2.2.4.3.1. Нивои архитектуре информационог система према Whitworth&Zaic	114
50.	2.2.4.3.2. Системска група критеријума у евалуацији информационог система, према Palmius	114
51.	2.2.4.3.3. Технолошка група критеријума у евалуацији информационог система, према Palmius	115
52.	2.2.4.3.4. Информациона група критеријума у евалуацији информационог система, према Palmius	115
53.	2.2.4.3.5. Група критеријума која се односи на појединце (особе, људски фактор) у евалуацији информационог система, према Palmius	116
54.	2.2.4.3.6. Организациона група критеријума у евалуацији информационог система, према Palmius	116
55.	2.2.4.4.1. Кључне квалитативне особине у евалуацији модела класа према Khan et al	117
56.	2.2.5.1.1. Историјски развој софтверске подршке одлучивању према Frolick&Ariyachandra	119
57.	2.2.5.1.2. Структура система за подршку управљању перформансама према Eckerson-y	119
58.	2.2.5.2.1. Архитектура система за евалуацију перформанси дистрибуираних софтверских компоненти, према Gao et al	120
59.	2.3.2.1. Компарација фаза животног циклуса Waterfall приступа и корака у оквиру једног циклуса примене агилне методе у развоју софтвера према Huo et al, 2004	125
60.	2.3.3.1. Историјски преглед метода агилног приступа развоју софтвера према Abrahamsson et al	129
61.	2.3.4.1. Животни циклус у оквиру дисциплиноване агилне испоруке (DAD) према Ambler&Lines	135
62.	2.3.5.1. Компарација традиционалног и адаптивног пројектног менаџмента, према Intaver Institute	136
63.	2.3.6.1. Стратегије пројектног менаџмента на основу комплексности и неизвесности у процесу развоја софтвера, према Wysocki, Fernandez&Fernandez	138
64.	2.3.6.2. Стратегије пројектног менаџмента на основу комплексности и неизвесности у процесу развоја софтвера, према Fernandez&Fernandez	139
65.	2.4.1.2.1. Софтверски елементи дистрибуираних система, према Braubach& Pokahr	141
66.	2.4.2.1.1. Таксономија истраживања у области само-адаптивности софтвера, према Salehie&Tahvildari	142
67.	2.4.2.2.1. Архитектура адаптивног информационог е-Сервис система, према Maurino et al	144
68.	2.4.2.2.2. Формула која представља параметре вредновања квалитета атрибута одређених компоненти на основу дефинисаног интервала прихватљивости вредности према Maurino et al	145
69.	2.4.2.2.3. Архитектура мултиканалног адаптивног web-базираног информационог система (MAIS) система према Carpiello et al	146
70.	2.4.2.2.4. Архитектура контекст-свесних система, према Baldauf et al	147
71.	2.4.2.2.5. Структурни модел организације адаптивног информационог система,	148

	према Matson& DeLoach	
72.	2.5.1.1. Проблеми агилног дистрибуираног развоја софтвера и најчешћа практична решења према Balasubramaniam et al	159
73.	2.5.3.1. Sense and Respond™ инфраструктура Wahyuudin & Tjoa	162
74.	2.5.4.1. Допуна гвозденог троугла управљања пројектима критеријумима који се односе на стратегијске аспекте, према www.projectmanager.com	163
75.	2.5.4.2. Примене Balanced Scorecard и одговарајући циљеви и области мерења у пословном, ИТ стратешком и ИТ развојном контексту, према Grembergen	164
76.	2.5.4.3. Примена Balanced Scorecard у оквиру пројектно орјентисане организације према Alleman	167
77.	2.5.5.1. Развој семантичких web сервиса, према Narayanan&McIlraith	169
78.	2.5.5.2. Архитектура система за резонување применом агената и web сервиса према Badica et al	170
79.	2.5.5.3. Улоге и активности у оквиру система резонувања путем web сервиса и агената према Badica et al	171
80.	3.2.2.1 . Бизнис процес модел за приказ модела процеса тока активности у управљању пројектом према PRINCE 2 и Balanced Scorecard методологији	179
81.	3.2.2.2. Део бизнис процес модела који се односи на улоге корисника, клијента и портфолио менаџера	180
82.	3.2.2.3. Део бизнис процес модела који се односи на улогу пројектног менаџера или тим-лидера	181
83.	3.2.2.4. Део бизнис процес модела који се односи на улогу члана пројектног тима	182
84.	3.2.5.1. Приступ пресликавању примитивних пословних процеса у софтверске функције 1. и 2. приоритета	183
85.	3.2.5.2. Дијаграм случајева коришћења са приказом софтверских функција	186
86.	3.2.5.3. Дијаграм случајева коришћења за профил Корисник	187
87.	3.2.5.4. Дијаграм случајева коришћења за профил Пројектни менаџер	187
88.	3.2.5.5. Дијаграм случајева коришћења за профил Члан тима	188
89.	3.2.6.1. ЕР дијаграм – концептуални модел података	189
90.	3.2.7.1. Дијаграм компоненти у приказу архитектуре предложеног система	190
91.	3.2.7.2. Дијаграм размештаја предложеног система	191
92.	3.2.7.3. Дијаграм размештаја за почетни прототип који садржи основне функције, развијен у првој итерацији у складу са агилним приступом развоја	192
93.	3.4.3.1.1. Формула за израчунавање трајања софтверског пројекта на основу структуре модела пословних процеса	207
94.	3.4.3.2. Формула за израчунавање трајања софтверског пројекта на основу елемената дизајна и могућности технолошке имплементације	209
95.	4.1.1.1. Подаци о регистрацији домена www.it-project.rs, доступни на сајту www.whois.net	210
96.	4.1.1.2. Подаци о хостингу web апликације домена www.it-project.rs	210
97.	4.1.1.3. Административни plesk панел web апликације домена www.it-project.rs	211
98.	4.1.1.4. Опција за рад са базом података у оквиру одељка plesk администрације	211
99.	4.1.1.5. Опције File Manager-а за постављање фајлова апликације на web server	211
100.	4.1.2.1. Списак фајлова прототипа апликације www.it-project.rs	212
101.	4.1.2.2. Форма са списком пројеката у дизајн режиму у оквиру развојног окружења	212
102.	4.1.2.3. Форма са списком радних ставки у оквиру изабраног пројекта, у дизајн режиму у оквиру развојног окружења	213
103.	4.1.2.4. Форма за унос радне ставке, у дизајн режиму у оквиру развојног окружења	213
104.	4.1.2.5. Изворни код за унос података радне ставке, у развојном окружењу	213
105.	4.2.1.1. Почетна страница web апликације домена www.it-project.rs	214

106.	4.2.1.2. Почетна страница web апликације www.it-project.rs за рад са менијем за корисника	214
107.	4.2.1.3. Почетна страница web апликације www.it-project.rs за рад са менијем за администратора	215
108.	4.2.2.1. Структура табеле tbProject у оквиру одељка plesk администрације web апликације www.it-project.rs	215
109.	4.2.2.2. Шема базе података реализоване апликације, приказана комбинованом нотацијом ER дијаграма и релационе шеме	216
110.	4.2.3.1. Ставке менија администрације	217
111.	4.2.3.2. Екран - табеларни приказ институција где раде или студирају учесници пројеката	218
112.	4.2.3.3. Екран - табеларни приказ учесника, уз филтер према радном статусу	218
113.	4.2.3.4. Екран - табеларни приказ улога учесника у пројекту, с описима задужења	219
114.	4.2.3.5. Екран - приказ детаља у области пројектних тема	219
115.	4.2.3.6. Екран - приказ детаља пројекта	220
116.	4.2.3.7. Екран - табеларни приказ пројеката	221
117.	4.2.3.8. Екран- унос података о додељивању учесника пројекту	221
118.	4.2.4.1. Екран - унос података о задатку	222
119.	4.2.4.2. Екран - табеларни приказ података о задацима пројеката	222
120.	4.2.4.3. Екран- унос података о ресурсу пројекта	223
121.	4.2.4.4. Екран - табеларни приказ података о ресурсима пројеката	223
122.	4.2.4.5. Екран -унос података о резултату реализованом у односу на задатак	223
123.	4.2.4.6. Екран - табеларни приказ података о резултатима задатака	224
124.	4.2.4.7. Екран - унос података о верификацији резултата задатка	224
125.	4.2.4.8. Екран - табеларни приказ података о верификацији резултата задатака	225
126.	4.2.5.1. Екран - табеларни приказ пројеката пријављеног учесника	225
127.	4.2.5.2. Екран - табеларни приказ пројеката пријављеног администратора	226
128.	4.2.5.3. Екран -Табеларни приказ ставки у току реализације пројекта	226
129.	4.2.5.4. Екран - Унос нове ставке на списак ставки тока реализације пројекта	227
130.	4.2.5.5. Шифарници у уносу ставке тока пројекта – сегмент, артефакт и фаза	227
131.	5.2.3.1. Приказ дела табеларног приказа пројеката у оквиру коришћења прототипа www.it-project.rs	234
132.	5.2.3.2. Графички приказ броја пројеката према броју месеци трајања	235
133.	5.3.2.1.1. Екран - Списак пројеката за пробни налог апликације	242
134.	5.3.2.1.2. Екран -Табеларни приказ списка радних ставки у оквиру пробног пројекта	242
135.	5.3.2.1.3. Екран - Приказ детаља једне пробне радне ставке, која садржи више подставки	243
136.	5.3.2.2.1. Процентуални графички приказ броја учесника према земљама седишта организације где су запослени	243
137.	5.3.2.3.1. Формула за израчунавање ELIG-RANK	248
138.	5.3.2.4.1. Графички приказ броја одговора у вези коришћења алата за подршку тимском раду	249
139.	5.3.2.4.2. Графички приказ броја одговора у вези коришћења алата за подршку пројектном менаџменту	250
140.	5.3.2.5.1. Графички приказ броја одговора у вези фактора утицаја на успех пројекта	251
141.	5.3.2.5.2. Графички приказ броја одговора у вези фактора утицаја на неуспех пројекта	251
142.	5.3.2.5.3. Графички приказ броја одговора у вези рангирања фактора утицаја на успех пројекта	252
143.	5.3.2.6.1. Графички приказ броја одговора у вези потребе за коришћењем и развојем информационих система за подршку управљању пројектом	253
144.	5.3.2.6.2. Графички приказ оцена значаја појединих карактеристика информационог система за подршку управљању софтверским	253

Љубица Кази - Докторска дисертација

	пројектима	
145.	5.3.2.6.3. Формуле за израчунавање значаја појединих карактеристика информационог система за подршку управљању софтверским пројектима	254
146.	5.3.2.6.4. Графички приказ сумарних података о значају појединих карактеристика информационог система за подршку управљању софтверским пројектима	254
147.	5.3.2.7.1. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа	255
148.	5.3.2.7.2. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа – утицај примене на процесе управљања пројектима и функционалне карактеристике	256
149.	5.3.2.7.3. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа – карактеристике прототипа и утицај на успех пројекта	256
150.	5.3.2.7.4. Графички приказ сумарних података о недостацима реализованог прототипа	258
151.	5.5.5.1. Исечак из табеларног приказа радних ставки у оквиру евиденције рада на пројекту „Алумни“ – унос једне радне ставке за једну активност која се засебно реализовала једина у наведеном тренутку	267
152.	5.5.5.2. Исечак из табеларног приказа радних ставки у оквиру евиденције рада на пројекту „Алумни“ – унос више радних ставки, за сваку активност посебна ставка у оквиру једног датума	268
153.	5.5.5.3. Унос података о више активности у оквиру једног записа (једне радне ставке) на примеру пројекта „Алумни“	268
154.	5.5.5.4. Процентуални приказ појединачних активности и записа и активности које су бележене у оквиру вишеструких/композитних записа	269
155.	5.5.6.1. Графички приказ броја појава фаза у оквиру свих реализованих софтверских пројеката у школској 2013/14 години	270
156.	5.5.6.2. Број појава сегмената у оквиру свих реализованих софтверских пројеката у школској 2013/14 години	271
157.	5.5.6.3. Број појава артефакта у оквиру свих реализованих софтверских пројеката у школској 2013/14 години	271
158.	5.6.1.2.1. Графички приказ сумарних података о примени метричког модела за евалуацију модела пословних процеса – шк. 2013/14	274
159.	5.6.1.2.2. Графички приказ сумарних података о примени метричког модела за евалуацију модела пословних процеса – шк. 2014/15	274
160.	5.6.1.3.1. Графички приказ сумарних података о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2013/14	275
161.	5.6.1.3.2. Графички приказ сумарних података о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2014/15	276
162.	5.6.1.4.1. Графички приказ сумарних података о примени метричког модела за евалуацију концептуалног модела података – шк. 2013/14	277
163.	5.6.1.4.2. Графички приказ сумарних података о примени метричког модела за евалуацију концептуалног модела података – шк. 2013/14	277
164.	5.6.5.1. Графички приказ успешности студентских софтвера према областима	278
165.	5.6.5.2. Графички приказ успешности студентских софтвера према областима уз примену тежинских фактора	279
166.	5.7.1.1. Графички упоредни приказ задатих и реализованих елемената елемената процеса развоја софтвера у оквиру мониторинга пројекта „Алумни“	281
167.	6.1.1.1. Графички приказ процента реализације захтеваних функција у оквиру почетног прототипа	283
168.	6.2.1.1. Изворни код и развојно окружење Web сервиса	287
169.	6.2.1.2. Web сервис након покретања у локалном режиму рада	288
170.	6.2.1.1.3. Унос параметра ради извршавања процедуре из Web сервиса	288
171.	Резултат извршавања web сервиса - XML	289
172.	6.2.2.1. Развојно окружење ASPX web апликације са програмским кодом уз примену web сервиса	289
173.	6.2.2.2. Извршавање пробне ASP.NET апликације за тестирање рада са ASP.NET web сервисом	290
174.	6.2.2.3. Изворни код пробне PHP апликације који укључује рад са ASP.NET web	290

	сервисом у оквиру Notepad ++ развојног окружења	
175.	6.2.2.4. Извршавање пробне PHP апликације за тестирање рада са ASP.NET web сервисом	291
176.	6.2.3.1. Проширење ASP.NET web сервиса додатним функцијама	291
177.	6.2.3.2. Списак расположивих метода/функција Web сервиса након покретања	292
178.	6.2.3.3. Проширење PHP програмског кода ради омогућавања динамичког приказивања контрола на корисничком интерфејсу	292
179.	6.2.3.4. Извршавање PHP апликације са динамичким креирањем елемената корисничког интерфејса на основу података из Web сервиса	293
180.	6.2.4.1. Пример концептуалног модела података	293
181.	6.2.4.2. Изворни код web сервиса за анализу концептуалног модела података	294
182.	6.2.4.3. Покретање web сервиса за анализу концептуалног модела података, уз унос XML stringа CDM модела	294
183.	6.2.4.4. Резултат извршавања web сервиса за анализу концептуалног модела података	295
184.	6.2.5.1. Делови кода PHP апликације који би могли да се параметризују и динамички мењају ради динамичког повезивања ASP.NET web сервиса	295
185.	6.2.5.2. Изворни код библиотеке „Toolkit.Net“ у оквиру развојног окружења Visual Studio.NET	296
186.	6.2.5.3. Укључивање библиотеке „Toolkit.Net.dll“ у оквиру референци пројекта Web сервиса	296
187.	6.2.5.4. Покретање web сервиса који врши израчунавање и враћа вредност	297
188.	6.2.5.5. Покретање web сервиса који покреће други web сервис динамички	297
189.	6.2.5.6. Резултат покретања другог web сервиса динамички од стране првог web сервиса	298
190.	6.2.5.7. Резултат извршавања другог, динамички покренутог web сервиса, у форми XML	298
191.	9.6.1.1. Приказ апликације „Alumni Web“ са web сајта Техничког факултета „Михајло Пупин“ у Зрењанину	348
192.	9.6.1.2. Приказ коришћења прототипа апликације www.it-project.rs за евиденцију података о току реализације пројекта Alumni Web	349
193.	9.6.2.1. Приказ апликације On-line скриптарница са web сајта Техничког факултета „Михајло Пупин“ у Зрењанину	350
194.	9.6.2.2. Приказ коришћења прототипа апликације www.it-project.rs за евиденцију података о току реализације пројекта On-line скриптарница	351
195.	9.8.1.1. Радно окружење алата „Active Collab“	378
196.	9.8.1.2. Радно окружење алата „Basecamp“	378
197.	9.8.1.3. Радно окружење алата „dotProject“	379
198.	9.8.2.1. Радно окружење алата „GForge Advanced Server“	380
199.	9.8.2.2. Радно окружење алата „JIRA“ – додавање активности и материјала	380
200.	9.8.2.3. Радно окружење алата „JIRA“ – дефинисање радног тока, примена Jquery језика и подршка за Kanban	381
201.	9.8.2.4. Радно окружење алата „easy Redmine“ – агилна табла	382
202.	9.8.2.5. Радно окружење алата „easy Redmine“ –временски план и граф прогреса	382
203.	9.8.2.6. Радно окружење „Microsoft Team Foundation Server/ VS Online“	383
204.	9.8.3.1. Радно окружење алата „Team Viewer“	384
205.	9.12.6.1. Пример бизнис процес модела на примеру „Алумни“	409

**СПИСАК ТАБЕЛА**

<b>РБ</b>	<b>Ознака и назив табеле</b>	<b>Страна</b>
1.	1.2.2.1. Анализа софтверских пројеката у уводном емпиријском истраживању	8
2.	1.3.1.1.1.1. Преглед функција информационог система	11
3.	1.3.1.1.2.1. Компоненте архитектуре информационог система	13
4.	1.3.1.1.3.1. Фазе и активности процеса развоја информационог система	14
5.	1.3.1.2.2.1. Активности животног циклуса софтвера према стандарду IEEE 1074	18
6.	1.3.1.3.4.1. Општи модел процеса управљања пројектом, према Јовановићу	32
7.	1.3.1.3.4.2. Приказ најчешће коришћених метода и техника у планирању пројекта, према Јовановићу	32
8.	1.3.3.1.2.1. Фазе и активности у управљању софтверским пројектима, према Berntsson-Svensson&Aurum	52
9.	2.1.1.1.1.1. Приказ ДСД проблема који су приказани у прегледним радовима	64
10.	2.1.1.1.1.2. Резултати примене SE-PM матрице за анализу покривености ДСД проблема у прегледним радовима	65
11.	2.1.1.1.1.3. Резултати примене SE-PM матрице користећи координате и референце прегледних радова	66
12.	2.1.1.1.1.4. Резултати примене SE-PM матрице сумарним приказом броја радова који разматрају одговарајућу групу ДСД проблема	66
13.	2.2.1.1. Општи оквир анализе истраживања у области метрички заснованог управљања софтверским пројектима	79
14.	2.2.2.1.1. Димензије успеха и одговарајуће мере, према Shenhar et al	80
15.	2.2.2.1.2. Индикатори мерења успеха пројекта на основу DeLone & McLean модела, према Crowston et al	82
16.	2.2.2.1.3. Димензије успеха и мерење за пројекте у области информационог система према Аткинсон	82
17.	2.2.2.2.1.1. Фактори успеха софтверских пројеката, према Berntsson-Svensson&Aurum	84
18.	2.2.2.2.1.2. Упоредни приказ фактора који доводе до успеха пројекта по годинама, на основу истраживања Standish group, према Eveleens& Verhoef	85
19.	2.2.2.2.2.1. Приказ фактора који доводе до статуса пројекта „под изазовом“ према истраживању Standish group, према Eveleens& Verhoef	90
20.	2.2.2.2.2.2. Приказ фактора који доводе до прекида пројекта према истраживању Standish group, према Eveleens& Verhoef	90
21.	2.2.2.2.2.3. Критични фактори успеха ИТ пројекта и поставке за метрике, према Hartman&Ashrafi	91
22.	2.2.2.4.2.1. Хипотезе Nguyen-а у вези зависности успеха пројекта и статус потврђености	93
23.	2.2.3.1.1. Нивои Capability Maturity Model за софтверске пројекте према Herbsleb et al	96
24.	2.2.3.4.2.1. Развојне фазе софтвера и индикатори мерења перформанси софтверског пројекта према Кан-у	103
25.	2.2.4.1.1. Функционалне карактеристике као критеријуми вредновања софтверског пакета, према Jadhav&Sonar	112
26.	2.2.4.1.2. Карактеристике квалитета софтвера као критеријуми вредновања софтверског пакета, према Jadhav&Sonar	112
27.	2.2.4.3.1. Елементи WOSP модела за мерење перформанси информационог система, према Whitworth&Zaic	114
28.	2.2.4.4.1. Van Belle-ов оквир за евалуацију модела у области информационог система	116
29.	2.3.2.1. Карактеристике агилних метода развоја софтвера, према Boose	124
30.	2.3.2.2. Компарација агилних и традиционалних метода развоја софтвера, према Javanmard&Alian	125
31.	2.3.3.1. Приказ карактеристика најчешће коришћених агилних метода,	126

	према Abrahamsson et al , Dyba& Dingsøy, Agile PrepCast	
32.	2.3.3.2. Компарација Агилних метода, према Agile PrepCast	130
33.	2.3.3.3. Компарација Агилних метода, према Javanmard&Alian	131
34.	2.3.3.4. Компарација агилних метода према Stojanovic et al	131
35.	2.3.4.1. Улога моделовања у примени агилних метода развоја софтвера, према Stojanovic et al	133
36.	2.3.5.1. Компарација принципа агилног приступа развоја софтвера и теорије комплексних адаптивних система према Meso&Jain	137
37.	2.4.2.1.1. Преглед истраживања у области самоадаптивног софтвера према Salehie&Tahvildari	143
38.	2.4.2.1.2. Типичне адаптационе (јаче и слабе) акције у оквиру самоадаптивног софтвера према Salehie&Tahvildari	143
39.	2.5.3.1. Компарација контроле квалитета у оквиру Waterfall и Агилних метода, према Huo et al	161
40.	2.5.4.1. Кључни индикатори перформанси стандардног пројекта у односу на Balanced Scorecard методологију, према Keyes	165
41.	2.5.4.2. Кључни индикатори перформанси колаборативног пројекта у односу на Balanced Scorecard методологију, према Keyes	166
42.	2.5.5.1. Анализа примене web сервиса у оквиру професионалних апликација подршке пројектном менаџменту	172
43.	3.1.3.1. Функционални теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката	174
44.	3.1.4.1. Основни резултати (производи) примене PRINCE2: 2009	175
45.	3.1.4.2. Записи и извештаји према методологији PRINCE2: 2009	175
46.	3.2.1.1. Теоријско-истраживачке основе за креирање предложених модела развоја система	178
47.	3.2.5.1. Општи модел табеле пресликавања примитивних пословних процеса у софтверске функције	183
48.	3.2.5.2. Пример примене табеле пресликавања пословних процеса у софтверске функције	184
49.	3.2.5.1. Табела пресликавања пословних процеса у софтверске функције	183
50.	3.2.6.1. Списак складишта података за нормализацију	188
51.	3.3.1.1. Теоријско-истраживачка основа предложених модела за евалуацију артефакта развоја софтвера	193
52.	3.3.2.1. Метрички модел за евалуацију модела пословних процеса	194
53.	3.3.3.1. Метрички модел за евалуацију модела софтверских функција	194
54.	3.2.4.1.1. Улога концептуалног моделовања података у оквиру интегралног процеса развоја софтвера информационог система	195
55.	3.2.4.1.2. Методологија концептуалног моделовања применом интеграције метода концептуалног моделовања у оквиру развоја софтвера информационог система	196
56.	3.2.4.2.1. Метрички модел за евалуацију концептуалног модела података са синтаксног, семантичког и прагматичког аспекта	196
57.	3.3.5.1. Метрички модел за евалуацију софтверске апликације као артефакта у процесу развоја софтвера информационог система	198
58.	3.4.1.1. Теоријско-истраживачке основе модела за процену и мониторинг успеха процеса развоја софтвера	200
59.	3.4.2.1. Општи модел за дефинисање критеријума за мониторинг успеха активности на реализацији пројекта који се одвија у оквиру организационог система	201
60.	3.4.2.2. Општи метрички модел и методолошка заснованост општег метричког модела за мониторинг успеха процеса софтверског пројекта	202
61.	3.4.2.3. Матрица повезаности стратешких, тактичких и оперативних компоненти управљања софтверским пројектом на основу општег метричког модела за мониторинг софтверског процеса	203
62.	3.4.2.4. Проширени метрички модел за мониторинг софтверског пројекта – категорија процеса	203
63.	3.4.2.5. Метрички модел за вредновање софтверског пројекта – категорије производа, финансија, учења и напредовања	204
64.	3.4.2.6. Метрички модел за оперативни ниво планирања и праћења успеха реализације софтверског пројекта у области развоја	205



	информационих система	
65.	3.4.3.1. Анализа постојећих приступа процени функционалне величине, трајања и трошкова реализације софтверског пројекта базирани на BPM, USE CASE, CDM и RM и предлози унапређења приступа	206
66.	3.4.3.1.1. Метрички модел за процену трајања пројекта будућег софтверског решења на основу модела пословних процеса	207
67.	3.4.3.2.1. Метрички модел за процену трајања пројекта будућег софтверског решења на основу модела дизајна и могућности имплементације	208
68.	4.2.2.1. Структура табела базе података реализованог прототипа	216
69.	5.2.3.1. Број месеци и број софтверских пројеката који је трајао одговарајући број месеци у оквиру узорка емпиријског истраживања	235
70.	5.2.3.2. Приказ примењених технологија и броја пројеката реализованих примењеном технологијом у оквиру студентских софтверских пројеката	236
71.	5.2.3.3. Приказ броја реализованих софтвера према категоризацији софтвера	236
72.	5.2.3.4. Приказ броја реализованих софтвера према категоризацији софтверских пројеката	237
73.	5.2.3.5. Статистика о улози реализованих софтверских пројеката у наставном окружењу	238
74.	5.2.3.6. Статистика о броју пројеката и броју учесника по пројекту	238
75.	5.2.3.7. Статистика о применљивости решења и нивоима сложености софтверских решења	238
76.	5.3.2.2.1. Преглед држава и броја учесника анкете	244
77.	5.3.2.2.2. Сумарни приказ основних демографских података испитаника у анкетирању	244
78.	5.3.2.2.3. Сумарни приказ стручних биографских података испитаника у анкетирању	245
79.	5.3.2.2.4. Сумарни приказ стручних биографских података испитаника у анкетирању који се односе на типове реализованих софтвера	245
80.	5.3.2.2.5. Сумарни приказ ускостручних биографских података испитаника – искуства у тимском тимском дистрибуираном раду у развоју софтвера и у коришћењу алата за подршку тимском раду и управљању пројектима	246
81.	5.3.2.5.1. Сумарни приказ броја одговора према групама фактора који утичу на успех софтверског пројекта	250
82.	5.3.2.5.2. Сумарни приказ броја одговора према групама фактора који утичу на неуспех софтверског пројекта	251
83.	5.3.2.5.3. Сумарни приказ броја одговора у рангирању фактора утицаја на успех пројекта	252
84.	5.3.2.6.1. Сумарни приказ броја одговора у рангирању карактеристика информационог система подршке управљању софтверским пројектима	253
85.	5.3.2.7.1. Сумарни приказ одговора о употребљивости развијеног решења	254
86.	5.3.2.7.2. Сумарни приказ одговора о предностима коришћења развијеног решења – утицај на успешност процеса управљања пројектима и функционалне карактеристике	255
87.	5.3.2.7.3. Сумарни приказ одговора о предностима коришћења развијеног решења – карактеристике прототипа и утицај на успех пројекта	256
88.	5.3.2.7.4. Сумарни приказ група карактеристика са недостацима коришћења прототипа	257
89.	5.4.1.2.1. Бенчмаркинг модел за вредновање алата софтверског пројектног менаџмента	262
90.	5.4.1.3.1. Списак алата који припадају обема категоријама (за подршку тимском раду у развоју софтвера и управљању софтверским пројектима), који су наведени у анкетирању	263
91.	5.4.2.1. Број поена за појединачне алате у бенчмаркинг анализи према категоријама PMBOK, SWEBOOK и BSC	264
92.	5.4.2.2. Процент заступљености одговарајућих функционалних карактеристика за појединачне алате у бенчмаркинг анализи	264

	према категоријама PMBOK, SWEBOOK и BSC	
93.	5.5.2.1. Компарација развојних циљева, имплементираних карактеристика и утврђених особина анкетирањем	265
94.	5.5.5.1. Статистика о броју записа и активности у оквиру студентских софтверских пројеката реализованих у шк. 2013/14	269
95.	5.5.6.1. Сумарни приказ броја појава фаза, сегмената и артефакта у оквиру реализације софтверских пројеката у школској 2013/14 години	270
96.	5.6.1.2.1. Табеларни сумарни приказ података о примени метричког модела за евалуацију модела пословних процеса – шк. 2013/14	273
97.	5.6.1.2.2. Табеларни сумарни приказ података о примени метричког модела за евалуацију модела пословних процеса – шк. 2014/15	274
98.	5.6.1.3.1. Табеларни сумарни приказ података о о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2013/14	275
99.	5.6.1.3.2. Табеларни сумарни приказ података о о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2014/15	275
100.	5.6.1.4.1. Табеларни сумарни приказ података о о примени метричког модела за евалуацију концептуалног модела података – шк. 2013/14	276
101.	5.6.1.4.2. Табеларни сумарни приказ података о о примени метричког модела за евалуацију концептуалног модела података – шк. 2014/15	277
102.	5.6.5.1. Сумарни приказ успешности студентских софтвера према областима	278
103.	5.6.5.2. Сумарни приказ успешности студентских софтвера према областима уз примену тежинских фактора	279
104.	5.7.1.1. Табеларни приказ тока процеса развоја софтвера „Алумни“	280
105.	5.7.1.2. Сумарни приказ активности у оквиру процеса развоја софтвера „Алумни“	281
106.	6.1.1.1. Статистички приказ степена реализације захтеваних софтверских функција почетним прототипом	283
107.	6.1.3.1. Приказ области технолошког унапређења прототипа и план имплементације	286
108.	9.1.1. Списак софтверских пројеката и софтвера у емпиријском истраживању	335
109.	9.1.2. Радне улоге кандидата (Мр Љубице Кази) у оквиру реализације софтверских пројеката из емпиријске анализе	336
110.	9.2.1. Приказ појединачних проблема у дистрибуираном развоју софтвера и одговарајућих референци радова који реализују истраживање наведеног проблема	337
111.	9.5.1. Списак реализованих софтверских пројеката у школској 2013/14 години	344
112.	9.5.2. Списак реализованих софтверских пројеката у школској 2014/15 години	344
113.	9.5.1.1. Основне карактеристике реализованих софтверских пројеката	345
114.	9.5.2.1. Класификација пројеката према типовима софтвера и софтверских пројеката	346
115.	9.5.3.1. Реализација наставних обавеза у оквиру реализације софтверских пројеката	347
116.	9.6.1.1. Листа записа у оквиру реализације пројекта „Алумни“	349
117.	9.6.2.1. Листа записа у оквиру реализације пројекта „On-line скриптарница“	351
118.	9.7.1. Категоризација радних места испитаника у анкетирању	352
119.	9.7.2. Списак алата за подршку тимском раду, на основу анкете	353
120.	9.7.3. Списак алата за подршку управљању пројектом, на основу анкете	354
121.	9.7.4. Резултати примене методе ELIM-F над подацима о учесницима анкете ради селекције узорка истраживања	355
122.	9.7.5. Аналитички приказ одговора о факторима успеха софтверског пројекта, класификованих према категоријама	359
123.	9.7.6. Аналитички приказ одговора о факторима неуспеха софтверског пројекта, класификованих према категоријама	361

124.	9.7.7. Аналитички приказ одговора о недостацима развијеног прототипа, класификовани према категоријама	366
125.	9.8.4.1. Анализа карактеристика издвојених алата путем анкете, применом бенчмаркинг модела	384
126.	9.9.1. Приказ укупног броја радних ставки за реализоване пројекте у шк. 2013/14	388
127.	9.9.2. Приказ броја записа са једном радном активношћу у оквиру пројеката реализованих у шк. 2013/14	388
128.	9.9.3. Приказ броја радних ставки у односу на вишеструки унос – правилан начин	389
129.	9.9.4. Приказ броја радних ставки у односу на композитан унос – неправилан начин	389
130.	9.10.1. Категоризација фактора успеха, неуспеха и ризика у оквиру истраживања софтверских пројеката, дистрибуираних и агилних софтверских пројеката према предлогу метричког модела процеса	390
131.	9.11.1.1. Оцене радова студената у области моделовања пословних процеса применом метричког модела – шк. 2013/14	392
132.	9.11.1.2. Оцене радова студената у области моделовања пословних процеса применом метричког модела – шк. 2014/15	393
133.	9.11.2.1. Оцене радова студената у области модела дизајна софтверских функција применом метричког модела – шк. 2013/14	394
134.	9.11.2.2. Оцене радова студената у области модела дизајна софтверских функција применом метричког модела – шк. 2014/15	395
135.	9.11.3.1. Оцене радова студената у области концептуалног модела података применом метричког модела – шк. 2013/14	396
136.	9.11.3.2. Оцене радова студената у области концептуалног модела података применом метричког модела – шк. 2014/15	398
137.	9.11.4.1. Примена метричког модела за оцену софтверске апликације у области развоја информационог система на примеру „Alumni“	398
138.	9.11.4.2. Сумарни приказ резултата примене метричког модела за оцену студентских софтверских апликација реализованих у шк. 2013/14 и 2014/15	400
139.	9.11.4.3. Сумарни приказ резултата примене метричког модела за оцену студентских софтверских апликација реализованих у шк. 2013/14 и 2014/15 - након примене тежинских фактора	401
140.	9.12.1. Приказ примене модела за оперативни ниво мониторинга успеха софтверског пројекта у области развоја информационог система заснованог на агилном приступу – кумулативан приказ	403
141.	9.12.2. Приказ примене модела за оперативни ниво мониторинга успеха софтверског пројекта у области развоја информационог система заснованог на агилном приступу – појединачни приказ	405
142.	9.12.6.1. Приказ примене метричког модела за процену трајања софтверског пројекта на основу модела пословних процеса на примеру „Алумни“	410

## СПИСАК ЛИСТИНГА

<b>Рб</b>	<b>Ознака и назив листинга</b>	<b>Страна</b>
1.	2.4.4.3.1. Пример XML записа са правилима која омогућавају самоадаптивност софтвера и еквивалент у објектно-орјентисаној програмској имплементацији, према Wang	152
2.	2.4.4.3.1. Пример описа процеса применом BPEL4WS, према Charfi&Mezini	153
3.	2.4.4.3.2. Пример описа два аспекта у адаптацији процеса применом AO4BPEL, према Charfi&Mezini	154
4.	2.4.4.4.1. Део FScript акција која се може користити за адаптацију компоненте у складу са раније дефинисаном стратегијом, према David&Ledoux	156
5.	2.4.4.4.2. Реактивно адаптационо правило и скуп правила дефинисан политиком, према David&Ledoux	156
6.	2.4.4.4.3. Примена политике као скупа адаптационих правила у оквиру интерфејс класе Адаптациони Контролер, према David&Ledoux	156
7.	2.5.5.1. Политика примене web сервиса уз примену онтологија, агената и политике квалитета приказом ограничења, према Maxmilien&Singh	170
8.	9.13.1.1. Листинг кључних делова кода почетног прототипа - табеларни приказ пројеката пријављеног корисника	411
9.	9.13.1.2. Листинг кључних делова кода почетног прототипа - Снимање записа у оквиру изабраног пројекта	412
10.	9.13.1.3. Листинг кључних делова кода почетног прототипа - Табеларни приказ записа (радних ставки) изабраног пројекта са подршком визуализацији статуса реализације фазе пројекта	415
11.	9.13.2.1. Листинг кода унапређења прототипа - ASPX Web сервис који користи XML и XQuery	418
12.	9.13.2.2. Листинг кода унапређења прототипа - XML који се користи у оквиру web сервиса	420
13.	9.13.2.3. Листинг кода унапређења прототипа - ASPX Web апликација која користи ASPX Web сервис	421
14.	9.13.2.4. Листинг кода унапређења прототипа - PHP апликација која користи ASPX Web сервис, уз динамичку промену структуре корисничког интерфејса у односу на вредности из ASPX Web сервиса	421
15.	9.13.2.5. Листинг кода унапређења прототипа - адаптивност слоја података - креирање структуре XML као носиоца података	423
16.	9.13.2.6. Листинг кода унапређења прототипа - ASPX web сервис за анализу концептуалног модела података	424
17.	9.13.2.7. Листинг кода унапређења прототипа - ASPX web сервис који позива и користи други ASPX web сервис	426

## 1. УВОДНА РАЗМАТРАЊА

Управљање пројектима представља динамичну и комплексну савремену област која се примењује у различитим делатностима. Kerzner [Kerzner, 1989] дефинише пројектни менаџмент као „планирање, организовање, усмеравање и контролисање ресурса компаније за релативно краткоричне задатке који су остварени како би реализовали специфичне циљеве и задатке." Истраживања из области управљања пројектима односе се како на фундаменталне принципе и елементе, где се појмовно и структурно одређење пројектног менаџмента ставља у контекст актуелности [Cicmil et al, 2006], тако и на поједине активности, аспекте квалитета и примене пројектног менаџмента у посебним областима.

Област информационих технологија (ИТ) представља једну од најбрже развијаних области [Hartman&Ashrafi, 2002], па се услед великих улагања [Abdel-Hamid&Madnick, 1990] настоји унапредити ефикасност и продуктивност, односно успешност примене информационих технологија, при чему управљање пројектима има значајну улогу [Hartman&Ashrafi, 2002]. „ИТ пројектни менаџмент је релативно нова дисциплина која комбинује традиционални пројектни менаџмент са софтверским инжењерством и управљањем информационим системима како би учинила ИТ пројекте успешнијима"[Marchewka, 2006]. Посебном врстом пројектата, у оквиру којих се одвијају активности развоја софтвера бави се област управљања софтверским пројектима [Oktaba et al, 2007].

Према [Devedzic, 2002], управљање софтверским пројектима се бави процесом развоја софтвера, софтверском архитектуром, инжењерством захтева корисника, тестирањем софтвера, обезбеђивањем квалитета софтвера, продуктивношћу, стандардима, организационим аспектима, менаџмент стратегијама и техникама, проценом ризика, софтверским метрикама, управљањем софтверском конфигурацијом и применом резултата најбоље праксе и друго. Управљање софтверским пројектима као област делатности представља јединство опште области управљања пројектима и области софтверског инжењерства. Током више десетина година истраживања и искуства у овој области преточена су у упутства која су дефинисана кроз стандарде водећих светских професионалних институција у форми интегралних књига упустава (SWEBOOK [SWEBOOK, 2014] – Software Engineering Body of Knowledge и PMBOK [PMBOK, 2013] – Project Management Body of Knowledge, као и у форми међународних стандарда, као што је ISO/IEC/IEEE 16326:2009 Systems and Software Engineering—Life Cycle Processes—Project Management. Такође, истраживања и професионална решења у области подршке софтверском инжењерству, а посебно управљању реализацијом софтверских пројектата представљају савремене трендове.

Савремена истраживања у области управљања софтверским пројектима баве се различитим аспектима развоја софтвера и управљања пројектима [Jurison, 1999], као и њиховом интеграцијом, односно применом у оквиру ширег контекста управљања пројектима развоја и унапређења информационих система и примене информационих технологија [Hartman&Ashrafi, 2002]. У циљу унапређења квалитета и успеха пројектата, посебно су изражена истраживања у области анализе фактора успеха пројектата, мерења успеха и квалитета процеса и резултата софтверског пројектата. О актуелности истраживања у области мерења перформанси софтверског пројектата говоре чињенице да су у претходном периоду реализовани значајни резултати у оквиру научно-истраживачког рада у оквиру докторских дисертација [Moser, 1996], међународних пројектата [U-QASAR, 2014] као и у оквиру индустријских окружења [Daskalantonakis, 1992].

У току еволуције софтверског инжењерства [Voeht, 2006] утврђене су парадигме које су условиле одговарајуће методе и алате, почевши од 1950-тих година (софтверско инжењерство се третирао као инжењерство хардвера), 1960-тих (креирање софтвера као занат), 1970-тих (формалност процеса и модел водопада), 1980-тих (продуктивност и скалабилност), 1990-тих (конкурентно наспрам секвенцијалних процеса, развој

отвореног кода и искористивост). 2000-те године су карактерисане развојем агилних метода, вредносно-базираним софтверским инжењерством, модел-базираним развојем и интеграцијом софтверског и системског инжењерства. 2010-те године карактерише глобализација и развој комплексних система („system of systems“). Флексибилност процеса развоја софтвера и глобални дистрибуирани развој су два главна тренда у развоју софтвера и информационих система, где је подршка флексибилности изражена је кроз примену различитих агилних метода [Lgerfalk et al, 2009]. С обзиром на трендове у софтверској индустрији [Boehm, 1979] [Boehm, 2006], веома су значајна истраживања дистрибуираног развоја софтвера и управљања софтверским пројектима који се реализују у дистрибуираном окружењу.

Према [Wahyudin& Тјоа, 2007], савремени трендови дистрибуираног развоја софтвера захтевају, поред човек-базираног, подршку алат-базираног приступа мониторингу пројекта, с обзиром да се управљање дистрибуираним развојем софтвера суочава са бројним изазовима међу којима је: велика количина података која се односи на догађаје и процесе развоја, недостатак људских ресурса који би радили мониторинг, дистрибуираност и слаба повезаност дистрибуираних учесника у тиму или заједници која учествује у реализацији глобалног-дистрибуираног пројекта. С обзиром да компјутеризација генерално доприноси [Brynjolfsson&Hitt, 2003] директном и дугорочном унапређењу пословања и повећању продуктивности, тако и коришћење софтверских алата унапређује квалитет мониторинга, односно мерења и контроле пројекта, а самим тим и квалитет пројекта.

У циљу подршке дистрибуираном развоју софтвера и управљању софтверским пројектима који се реализују у дистрибуираном окружењу, у овом раду су разматране могућности примене дистрибуираних адаптивних информационих система. „Дистрибуирани систем, који за циљ има да обезбеди кориснички-оријентисане, интелигентне и стабилне услуге, треба да има реализоване механизме који треба да одрже квалитет услуге на одговарајућем нивоу независно од промена у окружењу и такав систем може да се назове адаптивни дистрибуирани систем (АДС). Адаптација дистрибуираног система састоји се у модификацији структуре и функција у односу на промене како би се реализовале и одржале потребне услуге које треба да задовоље захтеван квалитет услуга на одговарајућој платформи до максималног могућег нивоа у дефинисаном временском интервалу.“ [Fujita et al, 1998]

Циљ овог рада је приказ постојећих истраживања, теоријски и практични допринос у области развоја дистрибуираног адаптивног информационог система који треба да омогући подршку управљању реализацијом софтверских пројеката. С обзиром да су промене једна од кључних карактеристика савременог развоја софтвера, технолошка реализација адаптивности на промене је посебно разматрана у овом раду. О значају истраживања у области адаптивних дистрибуираних информационих система говори и велик број радова објављених у овој области, истраживања у ИТ индустрији, као и значајан број пројеката.

У овом раду у оквиру првог поглавља дата су уводна разматрања, кроз методолошки оквир истраживања, анализу оправданости истраживања и опис основних концепата. Друго поглавље односи се на приказ постојећих истраживања из области управљања софтверским пројектима у дистрибуираном окружењу, метрички и адаптивно заснованог управљања софтверским пројектима, као и из области адаптивних дистрибуираних информационих система. Треће поглавље представља модел предложеног система, док четврто поглавље описује имплементацију наведеног система као почетног прототипа. У оквиру петог поглавља представљено је емпиријско истраживање. Шесто поглавље описује унапређење прототипа. Седмо поглавље описује закључна разматрања. У осмом одељку дат је списак коришћене литературе, а у деветом одељку дати су прилози.

## 1.1. МЕТОДОЛОШКИ ОКВИР РАДА

У овом поглављу биће представљен методолошки оквир који представља основ за спроведено истраживање у овом раду. Методолошки оквир је дефинисан као предлог истраживања, који је усвојен пре почетка израде овог рада.

### 1.1.1. Предмет и проблем истраживања

*Предмет истраживања* ове докторске дисертације представља процес, методологија и технологија развоја адаптивних дистрибуираних информационих система за подршку управљању софтверским пројектима.

Структуру *проблема овог истраживања* чини проблем мерења, праћења и управљања успешношћу реализације софтверског пројекта; просторна и временска дислоцираност учесника у реализацији софтверског пројекта; карактеристике софтвера и софтверских пројеката у мерењу успешности елемената реализације пројекта, односно резултата појединих фаза реализације софтверског пројекта, пре свега коришћењем софтверских метрика, карактеристика квалитета и нивоа ризика у реализацији софтверског пројекта; методологија и технологија за прикупљање података о успешности реализације софтверског пројекта; адаптивност информационог система за подршку управљању пројектима на промене које се односе на тип софтверског пројекта, тип и технологију софтвера; разноврсност методологија развоја софтвера, односно методологија реализације софтверских пројеката и одговарајућих података који омогућавају праћење реализације пројекта; интеграција података о резултатима мерења карактеристика резултата реализације софтверског пројекта; обрада података и представљање у погодној форми (визуализација података) за доношење одлука у вези управљања реализацијом софтверског пројекта.

### 1.1.2. Циљ и задаци

Основни циљеви истраживања су:

- Креирање модела адаптивног дистрибуираног информационог система за подршку управљању софтверским пројектима
- Практична имплементација оваквог система, чија би употребљивост, ефекти, позитивне и негативне карактеристике биле утврђене кроз експериментално истраживање у образовању студената инжењерског профила у области информатике.

Дистрибуирани информациони систем за подршку управљању пројектима треба да буде адаптиван на технолошка развојна окружења и њихове промене, односно типове софтвера и софтверских пројеката, што се постиже употребом система за управљање пословним правилима, чији механизми омогућавају формализацију и аутоматизовану примену софтверских метрика општег типа и метрика које се односе на посебне типове софтвера, односно софтверских пројеката. Системи за управљање пословним правилима у основи су системи аутоматског резоновања који користе формализована правила (нпр. користећи предикатски рачун 1. реда, прологолике клаузуле и слично) и податке, на основу којих се изводе закључци, односно реализују одговори на постављена питања.

Задаци истраживања су:

1. Дефинисање теоријског модела развоја адаптивног дистрибуираног информационог система за подршку управљању софтверским пројектима,
2. Емпиријско истраживање карактеристика реализованог софтвера и софтверских пројеката са становишта управљања пројектом, уз дефинисање проблема и недостатака. Узорак ће представљати реализовани софтверски пројекти и софтвери у области образовања, здравства, културе и јавне управе. Биће

- анализирани типови софтвера, а посебно ће бити анализирани софтвери који се односе на области информационих система и образовног софтвера.
3. Анализа резултата емпиријског истраживања и синтеза закључака,
  4. Анализа и пројектовање вишеслојне архитектуре адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима, у складу са савременим објектно-орјентисаним приступом моделовању и језиком UML (Unified Modelling Language).
  5. Имплементација адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима. Овај систем треба да садржи модул за управљање пословним правилима којима се формализују и аутоматски примењују софтверске метрике за процену реализације софтверског пројекта,
  6. Припрема имплементираних система за коришћење у образовном процесу у наставном раду у оквиру часова наставе и израде семинарских радова студената, који би омогућили праћење успешности реализације софтверских пројеката студената.
  7. Емпиријско истраживање ефикасности дистрибуираног информационог система за подршку управљању реализације софтверских пројеката, употребом експерименталне методе. Узорак ће представљати софтверски пројекти, модели софтверског решења и софтвер у оквиру наставног процеса у раду студената у оквиру часова наставе и учења пројектовања софтвера и програмирања високошколске установе.
  8. Анализа резултата емпиријског истраживања и синтеза закључака, који се односе на позитивне и негативне карактеристике, односно предности и недостатке имплементираних система,
  9. Разматрање перспектива и могућих праваца даљег развоја система, како би се користили у реалној пракси реализације софтверских пројеката и у оквиру наставног процеса.

### **1.1.3. Хипотеза и подхипотезе**

#### **ОСНОВНА ХИПОТЕЗА**

Могуће је креирати теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима, који би омогућио имплементацију наведеног система и на тај начин обезбедио подршку за управљање софтверским пројектима у реалној пракси развоја софтвера, као и у оквиру наставног процеса високошколске установе у области развоја софтвера и управљања софтверским пројектима.

#### **ПОДХИПОТЕЗЕ**

1. Коришћење адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима смањује време реализације софтверског пројекта,
2. Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим типовима софтвера и софтверских пројеката,
3. Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим технологијама имплементације софтвера и реализације софтверских пројеката,
4. Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се користи у оквиру наставног процеса у области образовања развоја софтвера и управљања софтверским пројектима, чиме повећава ефикасност наставе у области софтверског инжењерства и развоја информационих система, као и квалитет реализованих софтверских пројеката у наставном процесу.



#### **1.1.4. Методе и узорак**

Научне методе које су планиране да буду примењене:

1. Анализа – теоријска анализа, анализа постојећих решења у области, анализа карактеристика софтверских пројеката (у оквиру емпиријског истраживања)
2. Синтеза – синтеза модела теоријске основе за имплементацију система, синтеза закључака на основу емпиријских података (емпиријско-индуктивна метода)
3. Дескриптивна метода – опис карактеристика теоријског модела и имплементационог решења
4. Експеримент – коришћење имплементационог система и мерење кључних карактеристика перформанси система

Узорак истраживања одређен је областима емпиријског истраживања:

- ЕМПИРИЈСКО ИСТРАЖИВАЊЕ карактеристика реализованог софтвера и софтверских пројеката са становишта управљања пројектом, уз дефинисање проблема и недостатака
  - УЗОРАК: Реализовани софтверски пројекти и софтвери у области образовања, здравства, културе и јавне управе. Биће анализирани типови софтвера, а посебно ће бити анализирани софтвери који се односе на области информационог система и образовног софтвера.
- Емпиријско ЕКСПЕРИМЕНТАЛНО ИСТРАЖИВАЊЕ ефикасности дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката.
  - УЗОРАК: Софтверски пројекти, модели софтверског решења и софтвер у оквиру наставног процеса у раду студената у оквиру часова наставе и учења пројектовања и програмирања високошколске установе.

#### **1.1.5. Очекивани резултати**

Очекивани резултати истраживања су:

- Теоријски модел за развој адаптивниог дистрибуираног информационог система за подршку управљању софтверским пројектима, развијен на основу теоријске анализе проблемских области истраживања, анализе постојећих решења и емпиријског истраживања софтверских пројеката реализованих у реалној пракси у различитим доменама примене,
- Имплементирани информациони систем, заснован на претходно развијеном теоријском моделу,
- Резултати експерименталног истраживања карактеристика примене имплементационог система у пракси, односно у наставном процесу који се односи на развој софтвера,
- Приказ перспектива и могућих праваца даљег развоја система.

## 1.2. ОПРАВДАНOST И МОТИВАЦИЈА ИСТРАЖИВАЊА

Оправданост истраживања у наведеној области поткрепљена је емпиријским и теоријским истраживањима. На овај начин представљена је мотивација за истраживање, приказом емпиријских и теоријских истраживања, као и кроз практичне проблеме непосредног искуства у реализацији софтверских пројеката.

### 1.2.1. АНАЛИЗА ЗНАЧАЈА И ФАКТОРА УСПЕХА ПРИМЕНЕ ПРОЈЕКТНОГ МЕНАЏМЕНТА

Истраживање поводом утврђивања вредности и значаја пројектног менаџмента за реализацију пројекта и пословање организација спровела је организација Project Management Institute, а резултати су објављени 2008. године [Thomas, 2008]. Истраживање је спроведено у 65 предузећа различитих делатности, поводом реализације пројеката који су реализовани у овим организацијама. Циљ је био да се кроз мерљиве и немерљиве показатеље успешности вреднује значај пројектног менаџмента кроз анализу пројектне документације за 418 пројеката, 60 извештаја о студији случаја, 447 интервјуа са пројектним менаџерима итд. Анализирани су резултати пројеката, организациони контекст реализације пројеката и вредности, односно значај који има пројектни менаџмент. У истраживању су учествовале организације из целог света, посебно из Европе 29%, Северне Америке 22%, Кине 22%, Јужне Америке 8%, Аустралије 6%, Русије 8%.

Siegelaub [Siegelaub, 2004] описује СММ (Capability Maturity Model) примењен у пројектном менаџменту: *1. ниво (Иницијални) – хаотични, непредвидљиви процеси, носиоци пројекта су компетентни појединци – хероји, 2. ниво (поновљив, управљив) – дисциплиновани стабилни процеси, примена језга пројектног менаџмента, 3. ниво (стандардизација процеса) – интегрисана методологија управљања процесима, обука, управљање ризицима, 4. ниво (квантитативна управљивост) – управљање квалитетом базирано на мерењу перформанси процеса и квантитативним анализама, 5. Ниво (оптимизација, континуално унапређење процеса) – коришћење ранијих резултата („лекција из претходних искустава“), управљање променама, организационе иновације.* Према истраживању [Thomas, 2008], у области СММ нивоа зрелости у области пројектног менаџмента наведених организација, већина се налази на 1. и 2. нивоу. То значи да већина организација има и користи тек основне елементе пројектног менаџмента у организацији.

На основу истраживања [Thomas, 2008], елементи који утичу на значај и вредност пројектног менаџмента су:

1. Контекст: Економски статус, Људски фактор, Култура, Тип и контекст пројекта, Карактеристике организације, Стратешка усмерења
2. Контекст утиче на имплементацију кроз: Обука, Алати, Људи, Мотивација, Организација
3. Имплементација утиче на вредности: задовољство, усклађеност са циљевима, конзистентност примене, резултати процеса, пословни резултати, реализоване користи (предности).

На основу истраживања [Thomas, 2008], на елементе вредности утиче:

1. Креирање вредности – колективна перцепција усклађености, континуирано инвестирање, фокус и посвећеност, схватање значаја пројектног менаџмента
2. Уништавање вредности – промена персонала задуженог за спровођење и руковођење имплементацијом, недостатак фокуса, одржавања, воље, претерана бирократизација, неконзистентност усклађености са потребама (посебно између менаџмента организације и пројектног менаџмента)
3. Неодређеност вредности – конфликти, промене стратегија, контролисање проблема, примена пројектног менаџмента формално

У оквиру истраживања [Thomas, 2008] је показано да је у половини пројеката исказан повраћај инвестиција (ROI - return of investment), што представља мерљив показатељ значаја и вредности пројектног менаџмента. Међутим, из разлога што је тешко прикупљати овакве податке, у већини организација се значај примене пројектног менаџмента огледа у немерљивим показатељима као што су: подршка одлучивању, комуникације, радна култура, усклађеност рада, ефективност организације. Степен немерљивих вредности је у корелацији са зрелошћу организационе примене пројектног менаџмента. Зрелост организације примене пројектног менаџмента је у директној сразмери савредностима које доноси. Такође је утврђено да имплементација пројектног менаџмента и вредност која доноси су у великој зависности од културе – националне, организационе и културе примене пројектног менаџмента.

С обзиром да су алати у примени у оквиру пројектног менаџмента једна од компоненти имплементације процеса, који директно утиче и на резултате, јасан је значај развоја софтверских алата, тј. апликативног софтвера који би као алат допринео повећању квалитета резултата и нивоа вредности примене пројектног менаџмента.

## 1.2.2. УВОДНО ЕМПИРИЈСКО ИСТРАЖИВАЊЕ – АНАЛИЗА ПРОБЛЕМА РЕАЛИЗОВАНИХ СОФТВЕРСКИХ ПРОЈЕКТА

Емпиријско истраживање проблема реализованих софтверских пројеката извршено је на основу искуства кандидата у оквиру непосредног учешћа у реализацији више софтверских пројеката. Табела 9.1.1. (прилог) приказује списак софтверских пројеката и софтвера који су анализирани са становишта проблема реализације. Табела 9.1.2. (прилог) приказује списак радних улога кандидата у оквиру реализације софтверских пројеката из емпиријске анализе. Табела 1.2.2.1. представља резултате емпиријске анализе кључних карактеристика реализованих пројеката и проблема у оквиру реализације тих пројеката.

Табела 1.2.2.1. Анализа софтверских пројеката у уводном емпиријском истраживању

РБ ПР.	ОБЛАСТ	ТИП СОФТВЕР А/ УЛОГА КАНДИДАТА	КАРАКТЕРИСТИКЕ ПРОЈЕКТА	ПРОБЛЕМИ РЕАЛИЗАЦИЈЕ ПРОЈЕКТА
1	Здравство	Инф. систем	Научно-стручни пројекат, Учешће већег броја учесника са 2 факултета и малог броја учесника из дате институције, више софтверских модула	Координација рада учесника са 2 факултета, интеграција решења, промена захтева и структуре документације у току имплементације
2	Образовање	Образовни софтвер	Научно-стручни пројекат, Учешће већег броја учесника са једног факултета, нема екстерних учесника	Усклађивање предзнања усвојене технологије од стране свих учесника у имплементацији, Усклађивање резултата рада учесника
3	Култура	Инф. систем	Стручни пројекат, Модул информационог система, мали број учесника пројекта и малог броја учесника из дате институције	Одређивање и фокусирање на језгро дефинисаних потреба, у складу са временским ограничењем, могућност интеграције у оквиру информационог система организације и могућности даљих проширења
4	Јавна управа	Web portal	Стручни пројекат, Мањи број учесника који реализују пројекат са факултета и малог броја учесника из дате институције	Усклађивање расположивог познавања технологија од стране учесника, погодност за даље одржавање
5,6	Здравство	Инф. систем	Стручни пројекат, Мали број учесника пројекта са факултета и малог броја учесника из дате институције	Фокусирање на језгро дефинисаних потреба, у складу са временским ограничењем импл., непостојање информатичких кадрова у организацији ради даљег одржавања решења
7, 8	Здравство	Инф. систем	Стручни пројекат, Мали број учесника пројекта са факултета и малог броја учесника из дате институције	Фокусирање на језгро дефинисаних потреба, у складу са временским ограничењем импл., непостојање информатичких кадрова у организацији ради даљег одржавања решења
9	Образовање	Инф. систем	Стручни пројекат, Учешће већег броја	Интеграција реализованог решења са постојећом

			учесника са једног факултета, нема екстерних учесника	технологијом
10	Образовање	Web portal	Стручни пројекат, учешће малог броја учесника са факултета и једног учесника из екстерне институције	Прецизно дефинисани захтеви поводом графичког дизајна, недовољно претходног искуства у технологији израде, кратак рок израде
11	Образовање	Windows апликација	Стручни пројекат, учешће малог броја учесника са факултета	Непрецизно дефинисани захтеви у функционалном и дизајн делу, недовољно претходног искуства у технологији израде, кратак рок израде, дизајн система погодан за лако коришћење и одржавање

Претходно представљена Табела 2. даје преглед основних карактеристика софтверских пројеката и најважнијих проблема који су се јавили у току реализације софтверских пројеката. Највећи број пројеката је у области имплементације софтвера који реализују основне пословне функције користећи базу података. Такође, највећи број пројеката су стручни пројекти.

Проблеми који су се јавили у оквиру реализације наведених пројеката могу се груписати на:

#### ПРОБЛЕМИ КОМУНИКАЦИЈЕ И САРАДЊЕ

- Координацију рада учесника различитих институција
- Усклађивање и интеграција резултата рада учесника

#### ПРОБЛЕМИ СПЕЦИФИКАЦИЈЕ ЗАХТЕВА

- Непрецизност захтева у делу функција и дизајна
- Прецизност захтева у делу дизајна
- Промену захтева и структуре пословне документације у току имплементације
- Одређивање језгра специфицираних захтева корисника
- Фокусирање на језгро дефинисаних потреба

#### ПРОБЛЕМИ ТЕХНОЛОГИЈЕ

- Усклађивање предзнања усвојене технологије од стране свих учесника
- Интеграција реализованог решења са постојећом технологијом.

#### ПРОБЛЕМИ РЕСУРСА

- Временско ограничење реализације
- Недовољно претходног искуства у технологији израде
- Кратак рок израде

#### ПРОБЛЕМИ ДИЗАЈНА ПОГОДНОГ ЗА ИНТЕГРАЦИЈУ И ОДРЖАВАЊЕ

- Дизајн решења са могућностима интеграције са постојећим решењима и могућношћу проширења
- Дизајн решења погодан за даље одржавање

На основу наведених проблема, јавља се потреба за софтверским алатима који би омогућили подршку управљању реализацијом софтверских пројеката, посебно таквих пројеката који се одвијају у дистрибуираном окружењу. Такви алати треба да омогуће:

1. комуникацију и интеграцију резултата рада учесника пројекта
2. евидентирање и флексибилност у односу на промене спецификације захтева
3. да не зависе од технологије имплементације, односно да омогуће подршку различитим технологијама и методологијама израде
4. да повећају ефикасност рада на пројекту.

## 1.3. ТЕОРИЈСКЕ ОСНОВЕ

### 1.3.1. Основне области

#### 1.3.1.1. РАЗВОЈ ИНФОРМАЦИОНИХ СИСТЕМА

##### 1.3.1.1.1. Дефиниција и функције информационог система

Да бисмо дефинисали појам информационог система, потребно је представити дефиниције основних појмова од којих се састоји – систем и информација.

Систем (од грчке речи „συστημα“ – целина састављена из елемената) представља „јасно издвојен скуп, на одређен начин међусобно повезаних елемената, који по некој заједничкој одредници, чине складну целину.“ [Šoti, 1978] Као битне одлике динамичких система издвајају се: јасна издвојеност, степен и начин повезаности система са окружењем [Šoti, 1978], јасно одређење функција појединих елемената и њихова усклађеност и подређеност захтевима функционисања система као целине [Stanojević, 1986]. „Систем је организиран скуп елемената који дјелују ради испуњења постављеног циља“ [Radovan, 1989] . Организациони системи су сложени динамички системи [Šoti, 1978] који се састоје од подсистема: социолошки и техничко-технолошки, односно функционалних компоненти која носи информациона, енергетска и материјална обележја [Stanojević, 1986].

Појам информације се најчешће тумачи у контексту примене у организационим системима. У том контексту, под информацијом се подразумева [Stanojević, 1986] „порука која се користи за доношење одлука или извршавању активности на остварењу донетих одлука. Састоји се од података уређених тако да као целина представљају исказе са одређеним смислом. Информације се добијају обрадом података. Подаци изражавају постојање одређене стварности – чињеница, врста атрибута те стварности и вредности атрибута“ [Stanojević, 1986]. Настанак информација се везује за процесе тумачења (разумевање) и вредновања (додељивање вредности, значаја) подацима – „субјективни (свесни) модел објективне стварности“ [Dulović, 1991] . Подаци су регистроване чињенице о особинама неког процеса или објекта [Balaban et al, 1996] . „Податке прикупљамо и региструјемо, да би их могли чувати и по потреби користити. Подаци постају информације у моменту њиховог коришћења, када се користе за предузимање акција или доношење одлука“ [Jauković, 1992]. Обрадом података се подаци трансформишу у информације. [Jauković, 1992]

Општа дефиниција информационог система описује га као „систем који има улаз и излаз информацијског карактера“ [Dulović, 1991] . „Информациони систем укључује све ресурсе који су ангажовани у прикупљању, управљању, коришћењу и достављању информационих ресурса у организацији.“ [Elmasri& Navathe, 2007] »Полазећи од системског приступа, информациони систем можемо дефинисати као сређени скуп метода, процеса и операција за прикупљање, чување, обраду, преношење и дистрибуцију података у оквиру једне организације, укључујући и опрему која се у те сврхе користи и људе који се тим активностима баве.« [Jauković, 1992] Рачунарски подржани информациони системи називају се заједничким именом „Информациони системи засновани на рачунару (Computer-based information system) [Balaban et al, 1996] . Информациони систем подразумева скуп метода, поступака и ресурса обликованих тако да потпомогну постизање неког циља [Stankić, 2005].

„Информациони систем у организацији обезбеђује процесе и информације корисне члановима организације и његовим клијентима“ [Avison& Fitzgerald, 2003] . „Основни циљ информационог система је стварање предуслова за боље функционисање пословног система, а тиме и постозање бољих резултата пословања.“ [Krsmanović& Mandić, 1995] Информациони систем може бити управљани (где је информацијски улаз већи од излаза), неутрални или управљачки (информацијски улаз је мањи од излаза)

[Dulović, 1991] . „Управљачки информациони систем производи информацијску подлогу за управљање и одлучивање у организацијама“ [Dulović, 1991], [Balaban et al, 1996] . Свака организација поседује информациони систем (неаутоматизован или у неком степену аутоматизације) [Dulović, 1991] . Информациони систем представља подсистем организационог система [Hawryszkiewicz, 2001], [Radovan, 1989], [Elmasri& Navathe, 2007] односно настаје као модел реалног система [Stanojević, 1986]. Сматра се да му је “основна намена да, као део управљачког подсистема организације, прикупља податке и трансформише их у информације”, те се често користи под термином Управљачки информациони систем – Management Information System [Jauković, 1992], [Radovan, 1989] . С обзиром да се (пословни) организациони системи састоје од пословних јединица - сектора, информациони систем се састоји од више подсистема, где сваки представља заправо подсистем појединачних пословних јединица [Hawryszkiewicz, 2001].

Функције информационог система су представљене групама активности у оквиру Табеле 1.3.1.1.1.1.

Табела 1.3.1.1.1.1. Преглед функција информационог система

<b>ФУНКЦИЈА ИНФОРМАЦИОНОГ СИСТЕМА</b>	<b>РЕФЕРЕНЦА</b>
<b>ДОКУМЕНТОВАЊЕ ПОСЛОВАЊА</b>	[Stankić, 2005]
- Прикупљање/обухват података	[Stanojević, 1986] [Radovan, 1989] [Dulović, 1991]
- Верификација података	[Stanojević, 1986]
- Регистровање података	[Stanojević, 1986]
- Чување/Меморисање података	[Stanojević, 1986] [Dulović, 1991]
<b>РАЗМЕНА ПОДАТАКА</b>	
- Пренос од места производње до места коришћења	[Stanojević, 1986]
- Дистрибуција	[Radovan, 1989]
- Достављање података корисницима	[Dulović, 1991]
<b>ПОДРШКА ОДЛУЧИВАЊУ</b>	[Stankić, 2005]
- Уређивање података	[Stanojević, 1986]
- Обрада података	[Radovan, 1989] [Dulović, 1991]
- Производња информација	[Stanojević, 1986]
- Презентовање података	[Dulović, 1991]

### 1.3.1.1.2. Класификација и архитектура информационог система

Јауковић [Jauković, 1992] даје класификацију информационог система:

1. Према врсти пружених услуга: системи за рачунарске услуге опште намене, системи за чување и претраживање података, системи за комуникацију порука, системи за управљање физичким процесима, системи за контролу и упозорења, системи за обраду трансакција.
2. Према области примене.
3. Према степену аутоматизације [Jauković, 1992] (односно према историјском развоју [Stankić, 2005]):
  - НЕАУТОМАТИЗОВАНИ ИНФОРМАЦИОНИ СИСТЕМИ – користе механографска средства обраде, носиоци података су документи, обрада није јединствена и понекад се не обавља увек на исти начин, подаци нису форматизовани и строго структурирани, задаци који се решавају често нису до краја дефинисани, значајно су заступљени усмени информациони токови, који су променљиви и непостојани, многе одлуке се доносе на основу усмених информација и искуства или на основу непотпуних и често застарелих података, субјективни чиниоци могу знатно да утичу на ток обраде података и добијања резултата, долази до дуплирања рада на обради података, нарочито код израде извештаја, обрада је спора, праве се грешке итд.
  - СИСТЕМИ АУТОМАТСКЕ ОБРАДЕ ПОДАТАКА (АОП) – коришћење рачунара за прикупљање и регистровање података, класифицирање, уређивање и ажурирање података, разна рачунања и сумирања, штампање извештаја.

Односе се на послове који су у потпуности познати и добро описани, чија обрада је регулисана законима и прописима организације. Системи АОП састоје се од посебних апликација које обухватају поједине функције или организационе целине (или њихове делове). Предности: систематизација прикупљања и чувања података, уводи се општа номенклатура и јединствени систем обележавања, убрзавање прикупљања и обраде података, повећава се квалитет информација, систематизују се извештаји и начин извештавања на појединим организационим и управљачким нивоима. Недоности: занемарен системски приступ, своди се на пројектовање програма и организацију података, а не подршку одлучивању, засебне апликације – редуванса.

- УПРАВЉАЧКИ ИНФОРМАЦИОНИ СИСТЕМИ – тежиште не на информацијама и њиховом коришћењу за доношење одлука, дају информације брзо и у облику који је прилагођен потребама оних који одлучују, даје извештаје за доношење одлука код структурираних проблема одлучивања (средњи ниво одлучивања у организацији – проблеми за које се може унапред утврдити коначно решење, који су подаци потребни да се оно добије, као и алгоритам његовог решавања). Представљају међусобно повезане подсистеме који чине јединствену целину.
- ИЗВРШНИ ИНФОРМАЦИОНИ СИСТЕМИ – дају информације о текућем стању организације и пројекцији у будућности, а намењене су највишем нивоу управљања.
- СИСТЕМИ ЗА ПОДРШКУ ОДЛУЧИВАЊА – дају подршку одлучивању код неструктурираних и слабо структурираних проблема, зависе од екстерних података (који могу бити непоуздани), решавање проблема се заснива на знању. Подржавају све нивое одлучивања, али и вертикалне информационе толове и интеграцију информација које се користе на различитим организационим и управљачким нивоима (посебно су значајни за више нивое), олакшавају синтезу информација из појединих подсистема за стратешко одлучивање, могућност уоћавања узајамних релација између различитих функција у разним организационим целинама и разумевање његовог утицаја на организацију. Састоји се од базе података, модела одлучивања и специјалног софтвера који их интегрише и омогућава корисницима коришћење ових модела.
- ЕКСПЕРТНИ СИСТЕМИ – опонашају рад експерата, развијени су применом техника вештачке интелигенције. Описују знања и правила одлучивања експерата. Проблеми: Како прибавити знање експерата, како то знање представити на рачунару, како на основу прибављеног знања издвојити решења конкретних практичних проблема. Састоје се од: базе знања, механизма закључивања, специјализоване базе података (база чињеница), механизам објашњења, кориснички интерфејс.

»Архитектура се дефинише као основна организација система садржана у његовим компонентама, њихова међусобна веза и веза са окружењем и принципи који владају развојем и еволуцијом система.« [ANSI/IEEE 1471-2000] »Архитектура информационог система је део ширег скупа архитектура и модела који су релевантни за организацију. На различитим нивоима архитектуре можемо разликовати:

- Архитектуру организације/предузећа (Enterprise Architecture),
- Архитектуру информационог система (ISA – Information System Architecture),
- Софтверску архитектуру (SWA – Software Architecture).« [Vasconcelos et al, 2007]

„Информациони системи расту и развијају се у различитости технологије, домена примене, броја корисника, географској дистрибуираности, броја функционалних јединица, компоненти и веза између њих. Мноштво архитектурних шаблона (pattern-a) појавили су се у настојању увођења реда. Дизајн архитектуре има кључну улогу у развоју информационог система. То је први корак којим се спецификације захтева претварају у радни систем који се састоји од хардвера и софтвера.“ [Lockemann, 2003] Zachman-ov framework [Zachman, 1987] је први одвојио концепт SWA и ISA, односно показао да SWA није једини који чини ISA.



Према [Vasconcelos et al, 2007], архитектуру информационог система чине подсистеми, односно архитектуре:

- ИНФОРМАЦИОНА АРХИТЕКТУРА – архитектура података,
- АПЛИКАЦИОНА АРХИТЕКТУРА – функционална архитектура,
- ТЕХНОЛОШКА АРХИТЕКТУРА – инфраструктурно-платформска архитектура.

“Поред технологије „информациони систем се темељи на систематично успостављеном и промењиваном скупу организацијских правила у вези с носиоцима задатака информирања. Потребно је утврдити путеве информирања, права и обавезе информирања, те методе и начине обраде информација које служе носиоцима задатака као подлога за доношење пословних и других одлука” [Dulović, 1991] .

Елементи (компоненте) компјутерски заснованог информационог система су представљени Табелом 2.

Табела 1.3.1.1.2.1. Компоненте архитектуре информационог система

<b>КОМПОНЕНТА АРХИТЕКТУРЕ ИНФОРМАЦИОНОГ СИСТЕМА</b>	<b>РЕФЕРЕНЦА</b>
ИНФОРМАЦИОНА АРХИТЕКТУРА	[Vasconcelos et al, 2007]
- Подаци („Data“)	[Elmasri& Navathe, 2007][Krsmanović& Mandić, 1995]
- Модели података	[Lazarević et al, 1988]
- Структуре података и системи за управљање базама података	[Vasconcelos et al, 2007] [Elmasri& Navathe, 2007]
АПЛИКАЦИОНА АРХИТЕКТУРА	[Vasconcelos et al, 2007]
- Модели функција	[Lazarević et al, 1988]
- Софтверска архитектура, софтвер („software“), апликациони софтвер	[Dulović, 1991] [Stankić, 2005] [Vasconcelos et al, 2007] [Zachman, 1987] [Elmasri& Navathe, 2007] [Krsmanović& Mandić, 1995]
АРХИТЕКТУРА РЕСУРСА	[Lazarević et al, 1988]
• ТЕХНОЛОШКА АРХИТЕКТУРА	[Vasconcelos et al, 2007], [Dulović, 1991]
- Медији за чување података	[Elmasri& Navathe, 2007]
- Рачунарска опрема (“Hardware“)	[Stankić, 2005] [Elmasri& Navathe, 2007] [Krsmanović& Mandić, 1995]
- Мрежна опрема (“Netware“)	[Stankić, 2005][Krsmanović& Mandić, 1995]
• КАДРОВИ (“Lifeware“)	[Lazarević et al, 1988] [Elmasri& Navathe, 2007] [Krsmanović& Mandić, 1995]
- Персонал који користи и управља подацима	[Elmasri& Navathe, 2007]
- Програмери апликација	[Elmasri& Navathe, 2007]
• ОРГАНИЗАЦИОНА АРХИТЕКТУРА (“Orgware“)	[Lazarević et al, 1988] [Dulović, 1991] [Stankić, 2005] [Krsmanović& Mandić, 1995]
- Принципи и концепти	[Dulović, 1991]
- Организациона правила коришћења система	[Dulović, 1991]
- Токови информисања	[Dulović, 1991]
- Методе и поступци коришћења система	[Dulović, 1991]

### 1.3.1.1.3. Процес развоја информационог система

Информациони систем организације треба да буде усклађен са потребама пословног функционисања, али и да прати промене функционисања и омогући реализацију стратешких циљева и њихових промена [Hawryszkiewicz, 2001]. Зато планирање развоја информационог система [Lazarević et al, 1988] мора да прати планирање развоја самог организационог система. Пројектовању и развоју информационог система као подсистема организационог система придаје се велик значај због захтева који се постављају пред функционисање организационог система у савременим условима [Stanojević, 1986]: а) изналагање оптималних режима коришћења ресурса, б) висока

продуктивност рада и ефектима рада, ц) оптимални пословни резултати у условима бројних утицаја из окружења.

Процес настајања и деловања информационог система називамо животним циклусом информационог система, који даје приказ фаза настајања и мењања информационог система [Radovan, 1989], [Avison& Fitzgerald, 2003]. Према Станојевићу [Stanojević, 1986], разликују се три најзаступљеније концепције приступа развоју информационог система:

- Концепција ПАРЦИЈАЛНОГ ПРИСТУПА, по којој се новим информационим системима у појединим етапама обухватају одређени послови или групе послова, без претходне идентификације целокупног организационог система и без разраде интегралног модела новог информационог система,
- Концепција КОМПЛЕКСНОГ СИСТЕМСКОГ ПРИСТУПА, која обухвата идентификацију целокупног организационог система, разраду интегралног модела новог информационог система и имплементацију новог информационог система као интегралне целине,
- Концепција СИСТЕМСКО-МОДУЛАРНОГ ПРИСТУПА (према Станојевићу, најприхватљивији приступ) по којима се новим информационим системом, у појединим етапама развоја, обухватају информациони подсистеми. На нивоу целокупног организационог система дефинишу се подсистеми, а затим се sukcesivним пројектовањем и развојем информационих подсистема формира и гради нови информациони систем.

Независно од усвојених приступа активности у пројектовању, имплементацији и функционисању информационог система се могу поделити у неколико фаза развоја информационог система, као што је приказано на наредној табели.

Табела 1.3.1.1.3.1. Фазе и активности процеса развоја информационог система

РБ	ФАЗА	РЕФЕРЕНЦА
I	ПЛАНИРАЊЕ	[Krsmanović& Mandić, 1995]
1	Испитивање система, дефиниција проблема; анализа проблема и потребних промена постојећег информационог система	[Hawryszkiewicz, 2001]; [Lazarević et al, 1988] [Avison& Fitzgerald, 2003]
2	Дефиниција концепта; израда студије оправданости развоја новог информационог система	[Hawryszkiewicz, 2001]; [Stanojević, 1986]
3	Дефиниција концепта; Анализа изводљивости реализације	[Hawryszkiewicz, 2001]; [Elmasri& Navathe, 2007] [Avison& Fitzgerald, 2003]
4	Селекција пројекта	[Hawryszkiewicz, 2001]
5	Спецификација захтева корисника	[Elmasri& Navathe, 2007]; [Hawryszkiewicz, 2001] [Avison& Fitzgerald, 2003]
6	Израда програма развоја и дефинисање пројектних задатака, дефинисање приоритета	[Stanojević, 1986] [Krsmanović& Mandić, 1995]
7	Снимање информационих подсистема, моделирање и анализа	[Stanojević, 1986]; [Hawryszkiewicz, 2001] [Avison& Fitzgerald, 2003]
II	ДИЗАЈН	[Krsmanović& Mandić, 1995] [Avison& Fitzgerald, 2003]
8	Израда идејног пројекта информационих подсистема, обухваћених пројектним задатком	[Stanojević, 1986]
	Дизајн система – архитектуре	[Krsmanović& Mandić, 1995]
9	Избор система обраде – рачунарска и мрежна опрема, оперативни системи;	[Stanojević, 1986] [Krsmanović& Mandić,

	дефинисање оквира контроле и безбедности система	1995]
10	Израда детаљног пројекта информационог подсистема	[Stanojević, 1986]
11	Детаљни дизајн; Дизајн базе података и апликација; функционално моделирање (модел функција система), информационо моделирање (модел података); апликативно моделирање (дизајн корисничког интерфејса и апликативног програма)	[Hawryszkiewicz, 2001]; [Elmasri& Navathe, 2007]; [Lazarević et al, 1988] ; [Veljović, 2003]
III	РАЗВОЈ ПРОЈЕКТА / ИМПЛЕМЕНТАЦИЈА	[Krsmanović& Mandić, 1995] / [Avison& Fitzgerald, 2003]
12	Обезбеђивање техничких предуслова за развој и имплементацију новог информационог система	[Veljović, 2003]
13	Развој система - конструкција; Имплементација; Програмирање	[Hawryszkiewicz, 2001]; [Elmasri& Navathe, 2007]; [Stanojević, 1986]
IV	ИМПЛЕМЕНТАЦИЈА	[Krsmanović& Mandić, 1995]
14	Припреме за увођење пројектованих система – обука, документовање, припрема тест података	[Stanojević, 1986]
15	Валидација и тестирање (прихватљивости) решења	[Stanojević, 1986]; [Hawryszkiewicz, 2001]
V	УВОЂЕЊЕ	[Krsmanović& Mandić, 1995]
16	Инсталирање система, тестирање и пробни рад	[Stanojević, 1986]; [Elmasri& Navathe, 2007]; [Hawryszkiewicz, 2001]
VI	ФУНКЦИОНИСАЊЕ / ОЦЕНА И ОДРЖАВАЊЕ	[Krsmanović& Mandić, 1995] [Avison& Fitzgerald, 2003]
17	Мониторинг коришћења система, одржавање	[Stanojević, 1986]; [Elmasri& Navathe, 2007]
18	Пост-пројектна евалуација	[Hawryszkiewicz, 2001]
VII	ПОВЛАЧЕЊЕ СИСТЕМА ИЗ УПОТРЕБЕ	[Krsmanović& Mandić, 1995]

У зависности од методологије развоја информационог система, у пракси се јављају различити животни циклуси, односно приступи процесу развоја, као што су: линеаран, еволуцијски и сл [Radovan, 1989] .

- Линеарни (водопад – Waterfall) приступ састоји се од фаза [Radovan, 1989] које се извршавају једна за другом: Дефинисање проблема (циљеви, ограничења, ресурси), оцена изводљивости (оквирни предлог система, очекивани трошкови и користи), анализа система/захтева (физички и логички модел постојећег информационог система, Пројектовање – оквирни и детаљни пројекат (логички и физички модел новог информационог система), Реализација (израда софтверам иницијализација система), Постимплементација (оцена оперативних особина, Одржавање система). Проблем линеарног приступа састоји се у немогућности будућег корисника система да на почетку развоја дефинише све потребне захтеве [Radovan, 1989] .
- Еволуцијски приступи [Hawryszkiewicz, 2001], [Radovan, 1989] - RAD (Rapid application development) [Hawryszkiewicz, 2001], DSDM (Dynamic System Development Method) pristup [Hawryszkiewicz, 2001], Спирални модел. Сви наведени еволуцијски приступи заснивају се на итеративном развоју, базираном на прототипском развоју, где почетни протитип настаје избором мањег скупа функција које се могу брзо реализовати и уз пробни рад корисника детаљније специфицирати.

- Агилни приступ [Ghada&Fitzgerald, 2013], [Abrahamsson et al, 2009], [AgileManifesto, 2001]– Развој орјентисан на брзу испоруку резултата (првенствено софтвера), уз сталну интеракцију са корисником и без много губљења времена на моделовање и документовање. Брза промена решења уз стално и ефикасно прилагођавање променама пословања и захтева корисника, с циљем задовољства корисника брзом испоруком у складу са потребама и захтевима.

С обзиром на сложеност архитектуре информационог система и мноштво активности на његовом развоју, природна је орјентација на методе управљања пројектима, које треба да омогуће реализацију овако сложеног система у оквиру пројектом дефинисаних елемената квалитета и ограничења ресурса.

## **1.3.1.2. МЕТОДОЛОГИЈА РАЗВОЈА СОФТВЕРА**

### **1.3.1.2.1. Области знања софтверског инжењерства**

Према [Roper&Millar 1999], „софтвер представља компјутеризоване инструкције које управљају рачунаром, манипулишу подацима и извршавају одређене функције или задатке“. У ужем смислу, под појмом софтвер подразумевамо рачунарски програм. Рачунарски програм [Turban et al, 2013] представља низ инструкција које рачунарски систем треба да изврши. Према [w1], [w2] под појмом софтвер подразумевамо скуп рачунарских програма, података, пратећих процедура, правила и документације који су намењени примени у оквиру рачунарског система.

«Софтверско инжењерство представља примену алата, метода и дисциплина које треба да омогуће креирање и одржавање аутоматизованих решења проблема реалног света. Софтверско инжењерство представља практичну примену научног знања у дизајну и конструкцији компјутерских програма и пратеће документације потребне за њихов развој, функционисање и одржавање» [Blum, 1992]

Према IEEE Software Engineering Body of Knowledge ([SWEBOK, 2014]), следеће области припадају областима знања у софтверском инжењерству:

- Софтверски захтеви (Software Requirements)
- Дизајн софтвера (Software Design)
- Конструкција софтвера (Software Construction)
- Тестирање софтвера (Software Testing)
- Одржавање софтвера (Software Maintenance)
- Управљање софтверском конфигурацијом (Software Configuration Management)
- Управљање софтверским инжењерством (Software Engineering Management)
- Процес софтверског инжењерства (Software Engineering Process)
- Модели и методе софтверског инжењерства (Software Engineering Models and Methods)
- Квалитет софтвера (Software Quality)
- Професионална пракса софтверског инжењерства (Software Engineering Professional Practice)
- Економија софтверског инжењерства (Software Engineering Economics)
- Рачунарске основе (Computing Foundations)
- Математичке основе (Mathematical Foundations)
- Инжењерске основе (Engineering Foundations)

### **1.3.1.2.2. Процеси у развоју софтвера**

SWEBOK представља стандардни интегрисани приказ знања из области софтверског инжењерства и укључује ISO и IEEE стандарде. Према [SWEBOK, 2014], неки од најзначајнијих ISO стандарда примењених у области софтверског инжењерства је стандард *ISO/IEC 90003:2004 Software Engineering—Guidelines for the Application of ISO 9001:2000 to Computer Software*, а посебно у области процеса животног циклуса софтвера је значајан стандард *ISO/IEC 12207 Software Life Cycle Processes*, а с обзиром да се софтвер развија у оквиру ширег система, значајни су и стандарди: *ISO/IEC 26702:2007 Standard for Application and Management of the Systems Engineering Process* и *ISO/IEC 15288:2008 Standard for Systems and Software Engineering—System Life Cycle Processes*.

Према [Castro Souto, 2010], „основне активности у креирању и управљању софтверским производом су[Erdogmus, 2008 ] [Bourque, 2002]:

- Спецификација софтвера (Software specification)
  - Развој софтвера (Software development)
  - Валидација софтвера (Software validation)
  - Еволуција софтвера (Software evolution)
-

Конструкција било ког система или апликације почиње концептуализацијом и спецификацијом. Глобални циљеви софтвера треба да су одређени, као и његове особине и карактеристике (функционалне и нефункционалне), као и функционалности и сервиси које треба да презентује или омогући. Овај процес анализе захтева и дефиниције је критичан за успех развојног пројекта [Sommerville& Kotonya, 1998]. Након што је систем скициран, започиње развој, који представља кључну фазу [Hatley et al, 2000], [Blanchard&Fabrycky, 2003] и генерално може бити подељен у две фазе: дизајн предложеног система (укључујући архитектуру, окружење, структуру) и имплементацију на дефинисаној платформи и користећи одабрани програмски језик. Као део развоја апликације или као посебан скуп задатака, трећа група активности има за циљ да обезбеди повратну информацију о квалитету производа и нивоу усклађености са пројектним циљевима. Тестирање софтвера може бити извршено на различитим нивоима [Craig& Jaskiel, 2002], [Black, 2002] од најмањих компоненти софтвера (тестирање unit-а или модула), подсистема (тестирање интеграције) до провере функционисања целокупног система као целине (тестирање система). Овај процес валидације вреднује да ли је систем погодан за коришћење у складу са његовом сврхом.

Након што је систем спреман да буде коришћен у оквиру радног окружења, животни циклус софтверског производа се не завршава [Pigoski, 1996]. Након испоруке система, континуирано одржавање и подршка је највероватније потребна и захтевана, не само ради решавања мањих проблема који и даље могу да се појављују, већ и ради унапређења и еволуције апликације, у случају појављивања нових потреба...[Castro Souto, 2010]

Организација IEEE успешно прилагођава ISO стандарде за практично коришћење. Према [Bruegge& Dutoit, 2004], IEEE 1074 је стандард за развој процеса животног циклуса софтвера. Процес је дефинисан као скуп активности које се изводе у складу са одређеном сврхом. Активност је задатак или група подактивности које су задате тиму или учеснику пројекта ради остваривања спесифичне намене. Задаци користе ресурсе да би произвели радни производ. За сваки задатак одређени су термини почетка и завршетка и контролне тачке ради праћења тока извршавања. IEEE даје списак од 17 просеса који чине животни циклус софтвера, а групише их у посебне целине назване групе процеса. У наредној табели дат је списак група просеса, самих просеса и Припадајућих активности, у складу са IEEE 1074 стандардом.

Табела 1.3.1.2.2.1. Активности животног циклуса софтвера према стандарду IEEE 1074

ГРУПА ПРОЦЕСА	ПРОЦЕСИ	АКТИВНОСТИ
<b>Моделовање животног циклуса</b>	Избор модела животног сиклуса	одређивање активности у складу са конкретним Пројектом
	<b>Управљање пројектима</b>	Иницијација пројекта
алоцирање ресурса за пројекат		
остваривање ( <i>establish</i> ) окружења за развој пројекта		
Планирање управљање пројектом		
Мониторинг и контрола пројекта		анализа ризика
		извођење планирања ( <i>contingency</i> )
		управљање пројектом
		управљање записима ( <i>retain records</i> )
		имрлементасија модела за извештавање о проблемима

	Управљање квалитетом софтвера	Планирање управљања квалитетом софтвера	
		Дефинисање метрика	
		управљање квалитетом софтвера	
		идентификација потреба за унапређењем квалитета	
<b>Пред-развој (Pre-development)</b>	Истраживање концепта	идентификација идеја и потреба	
		формулисање потенцијалних приступа	
		спровођење студије изводљивости ( <i>feasibility study</i> )	
		Планирање транзиције ( <i>transition</i> ) система (ако је изводљиво)	
		Детаљизирање ( <i>refine</i> ) и финализација идеје или потребе	
	Алокација система	анализа функција	
		развој архитектуре система	
		Декомпозиција системских захтева ( <i>system requirements</i> )	
<b>Развој</b>	Захтеви ( <i>Requirements</i> )	Дефинисање и развој софтверских захтева ( <i>software requirements</i> )	
		Дефинисање захтева за интерфејсом	
		Одређивање приоритета и интеграција софтверских захтева	
	Дизајн	извођење дизајна архитектуре	
		Дизајн базе података (по потреби)	
		Дизајн интерфејса	
		Селекција или развој алгоритама	
		извођење детаљног дизајна	
	Имплементација	креирање тест Података	
		креирање изворног кода	
		генерисање објект кода	
		креирање радне документације	
		Планирање интеграције	
		извођење интеграције	
	<b>Пост-Развој (Post-development)</b>	Инсталација	Планирање инсталације
			Дистрибуција софтвера
			Инсталација софтвера
			Прихватање софтвера у радном окружењу
Коришћење ( <i>operation</i> ) и подршка		коришћење система	
		обезбеђивање техничке Подршке и помоћи ( <i>consulting</i> )	
		одржавање записа о захтевима за помоћ	

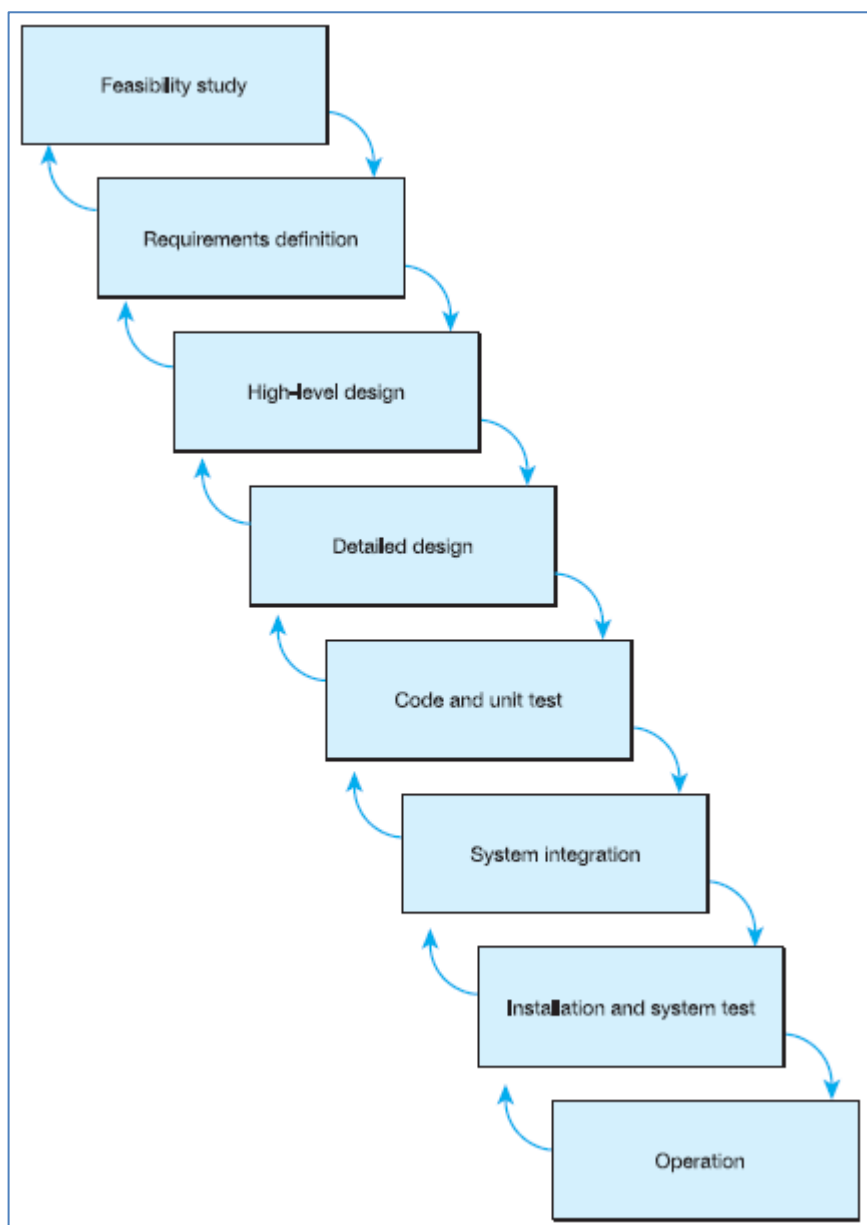
	Одржавање	Поновна примена животног циклуса софтвера
	Престанак коришћења ( <i>retirement</i> )	обавештавање корисника
		обезбеђивање и спровођење паралелног рада (по потреби)
		Престанак коришћења система
<b>Интеграција (Cross-development)</b>	Верификација и валидација	Планирање верификације и валидације
		извршавање задатака верификације и валидације
		Прикупљање и анализа метричких Података
		Планирање тестирања
		развој захтева тестирања ( <i>Develop test requirements</i> )
		извршавање тестирања
	Управљање софтверском конфигурацијом	Планирање управљања конфигурацијом
		развој идентификације конфигурације
		извођење контроле конфигурације
		извођење израчунавања статуса ( <i>status accounting</i> )
	Развој документације (documentation development)	Планирање документације
		имплементација документације
		Производња и дистрибуција документације
	Обука	Планирање програма обуке
		развој материјала за обуку
		валидација програма тренинга
		имплементација програма тренинга

### 1.3.1.2.3. Животни циклуси развоја софтвера

Према [Castro Souto, 2010], методологија развоја софтвера дефинише не само активности, већ и низ корака у развоју софтвера, који се морају применити одређеним редоследом ради добијања жењеног резултата, а понекад одређује и алате и технике које треба да буду у оквиру тих задатака примењени. Према [Castro Souto, 2010], методологије изградње софтверских производа се често називају животни циклуси софтвера. Неки од најпознатијих методологија развоја софтвера које су издвојене и описане у [Castro Souto, 2010] су:

1. Waterfall или класичан развој софтвера ([Royce, 1970] [Cadle&Yeates, 2008]) кроз низ фаза које се једна за другом извршавају: издвајање захтева у вези система (system requirements extraction), издвајање захтева који се односе на софтвер (software requirements extraction), анализа (analysis), дизајн програма (program design), кодирање (coding), тестирање (testing), коришћење (operation) и одржавање (maintenance).

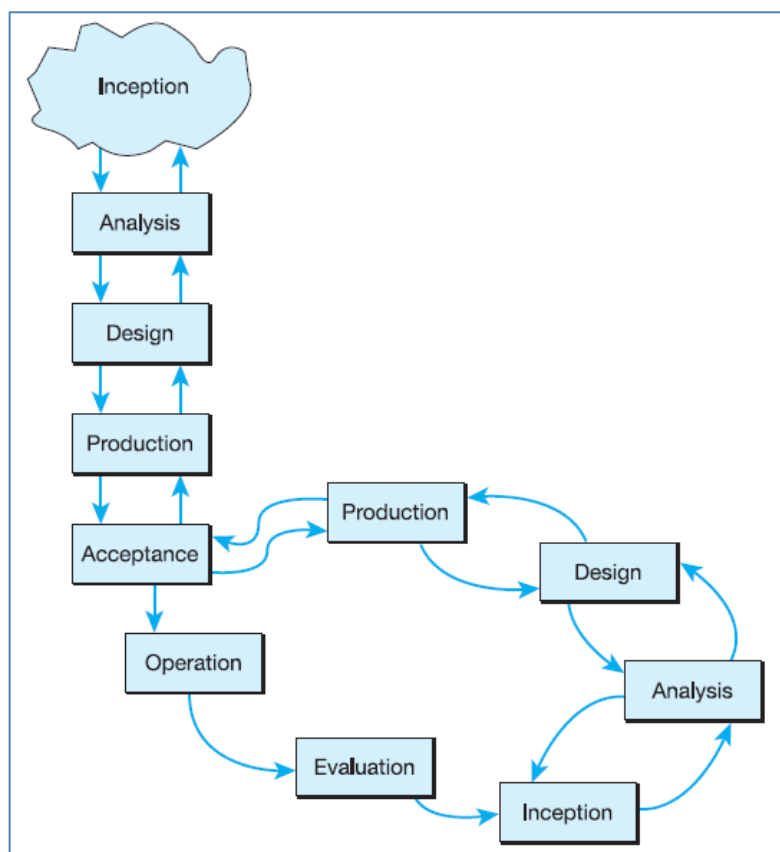




Слика 1.3.1.2.3.1. Waterfall модел животног циклуса софтвера према Cadle&Yeates [Cadle&Yeates, 2008]

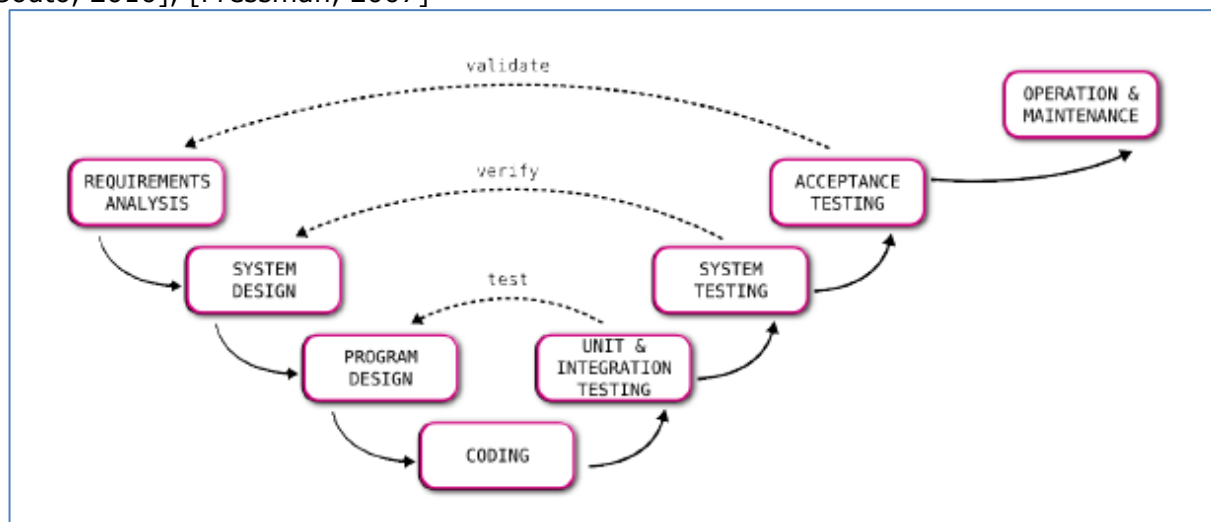
2. Waterfall са конструкцијом прототипа [Pfleeger& Atlee, 2003], који се користи у почетним фазама спецификације захтева, а затим одбацује

3. Модел "b" аутора N Birrell and M Ould, као још једна варијанта Waterfall модела, којим се коришћење и одржавање софтвера повезује са новим циклусом измена, где је укључено више од 70% укупних трошкова пројекта.



Слика 1.3.1.2.3.2. „b” модел животног циклуса софтвера према Cadle&Yeates [Cadle&Yeates, 2008]

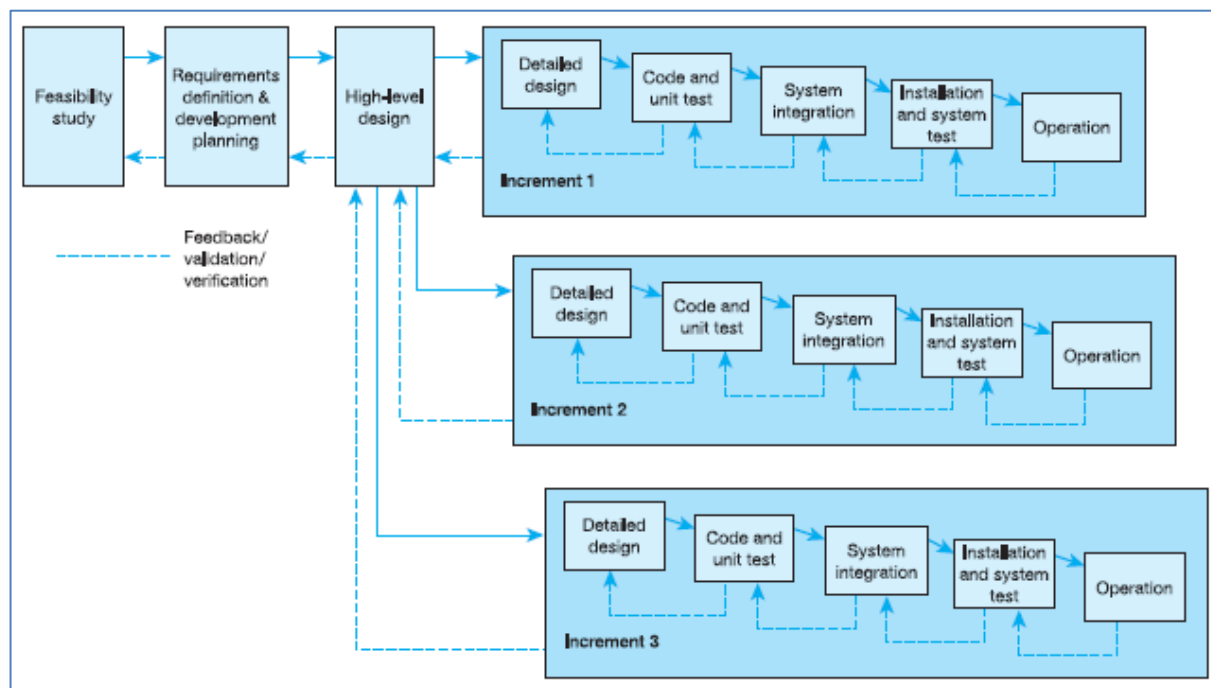
4. Модел “V” – који повезује поједине креативне фазе са фазама верификације [Castro Souto, 2010], [Pressman, 2007]



Слика 1.3.1.2.3.3. Model V животног циклуса софтвера према Castro Souto [Castro Souto, 2010]

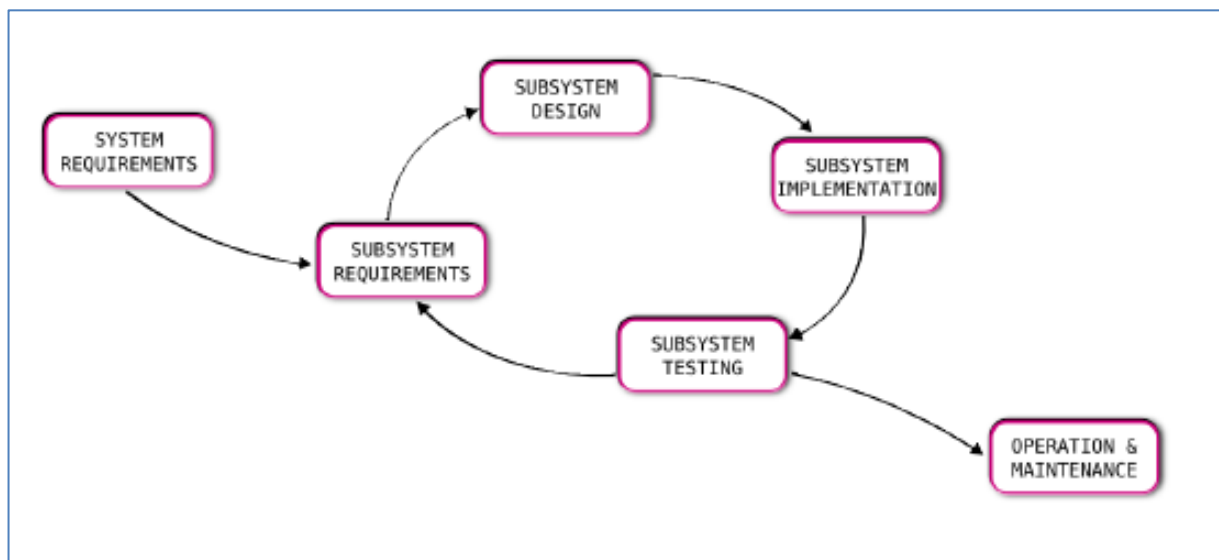
4. Прототипски развој [Brooks, 1975], где се прототипски дефинишу захтеви, дизајн и имплементира се прототип, али се затим кроз ревизије даље унапређује, уместо да се након почетне улоге у дефинисању захтева одбаца.

5. Инкрементални развој [Larman& Basili, 2003] са паралелним токовима развоја више различитих верзија истог софтвера, свака нова верзија добија нове функције у претходно дефинисаном скупу нових функција наведеног инкремента.



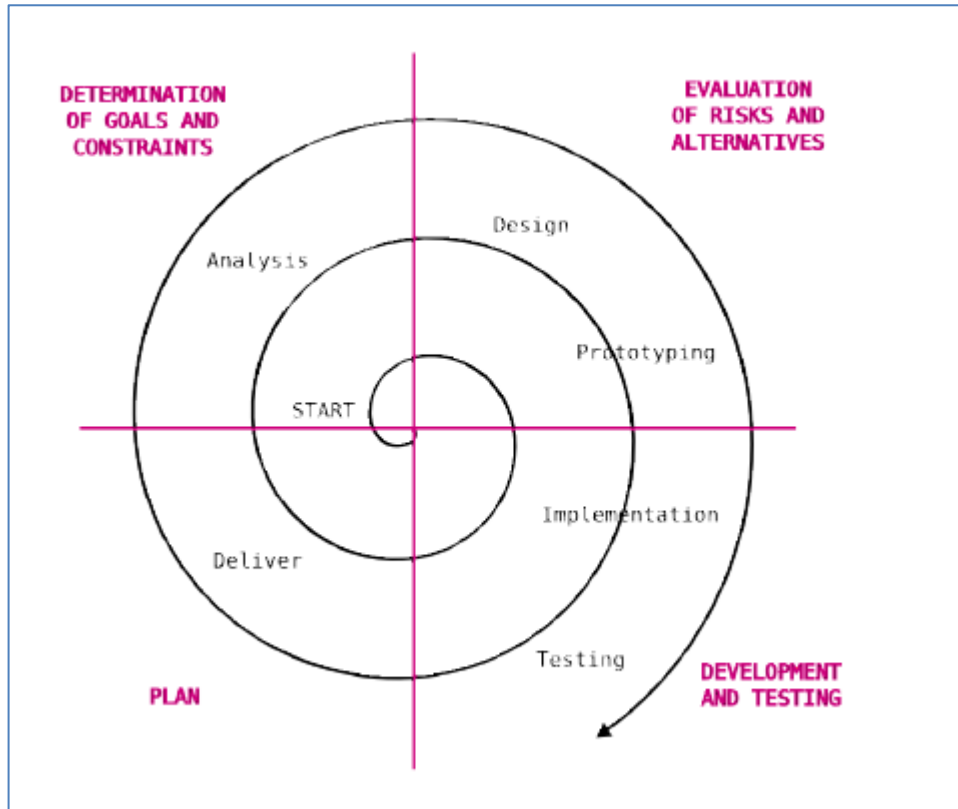
Слика 1.3.1.2.3.4. Инкрементални модел животног циклуса софтвера према Cadle&Yeates [Cadle&Yeates, 2008]

6. Итеративни развој [Larman& Basili, 2003] где се почетни систем испоручује са основним функцијама, а у низу итерација се сваки систем унапређује, кроз фазе:

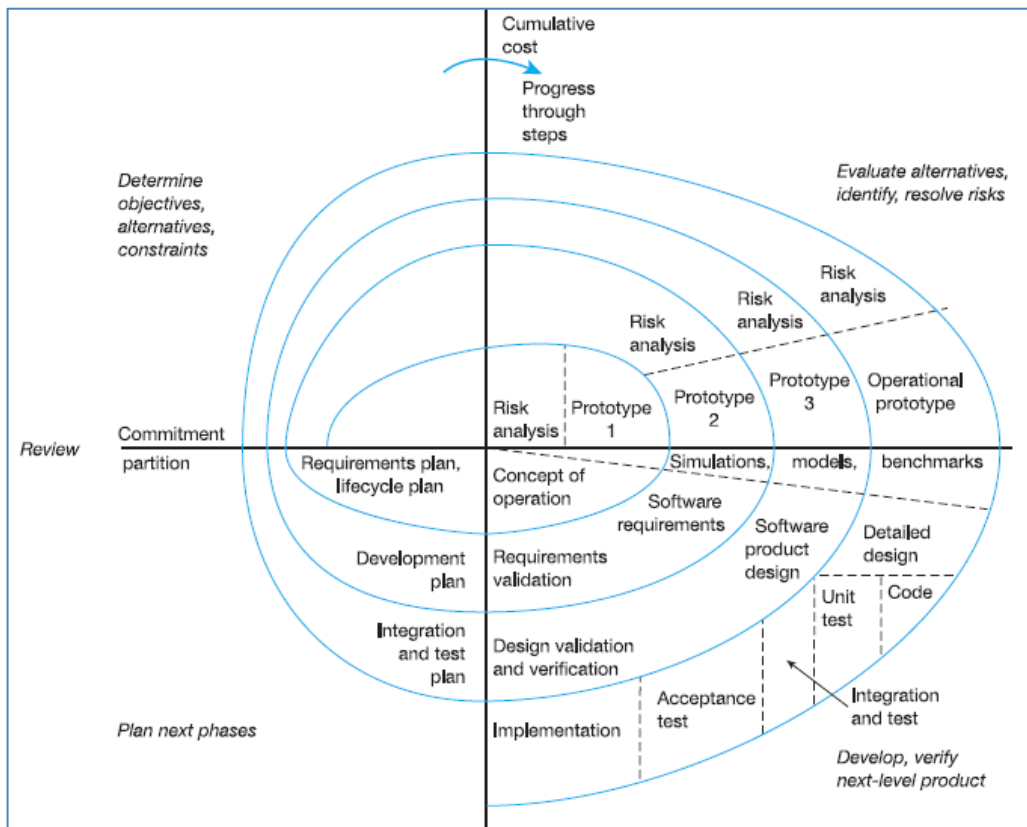


Слика 1.3.1.2.3.5. Модел итеративног развоја софтвера према Castro Souto [Castro Souto, 2010]

7. Спирални модел (Barry Boehm, 1986, [Boehm, 1986]), укључио је и елементе управљања пројектима по први пут у развој софтвера, као што је процена ризика у свим фазама.



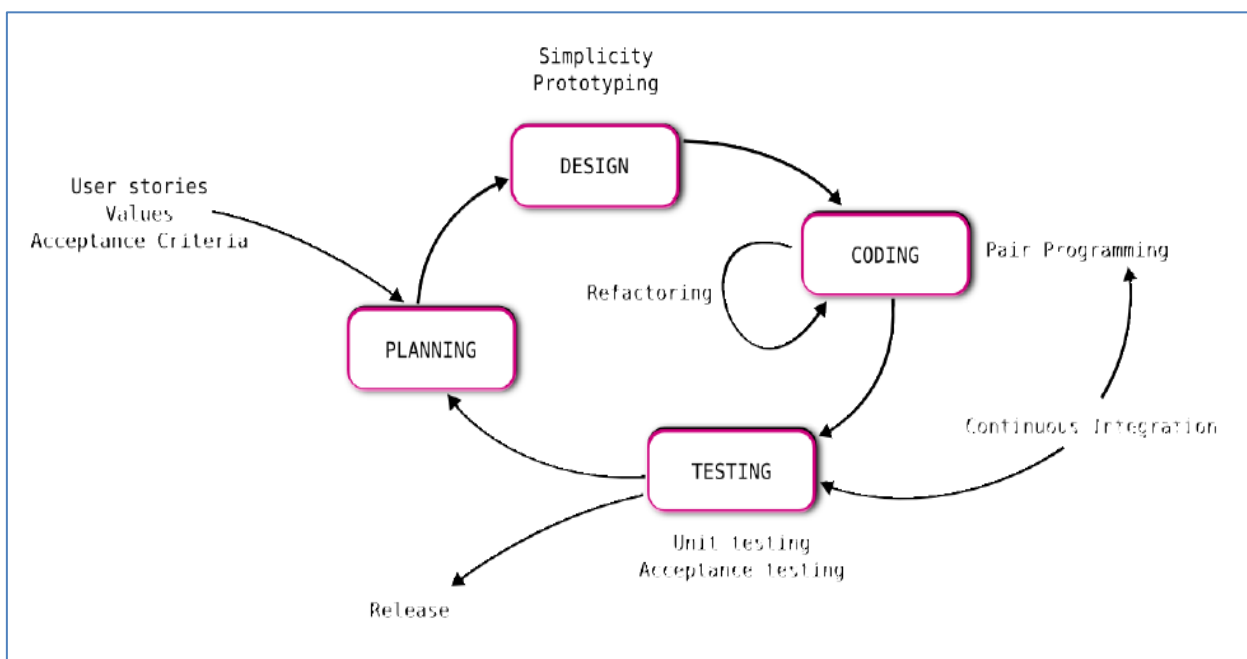
Слика 1.3.1.2.3.6. Спирални модел животног циклуса софтвера према Boehm-у [Castro Souto, 2010], [Boehm, 1986]



Слика 1.3.1.2.3.7. Спирални модел животног циклуса софтвера према Cadle & Yeates-у [Cadle & Yeates, 2008]

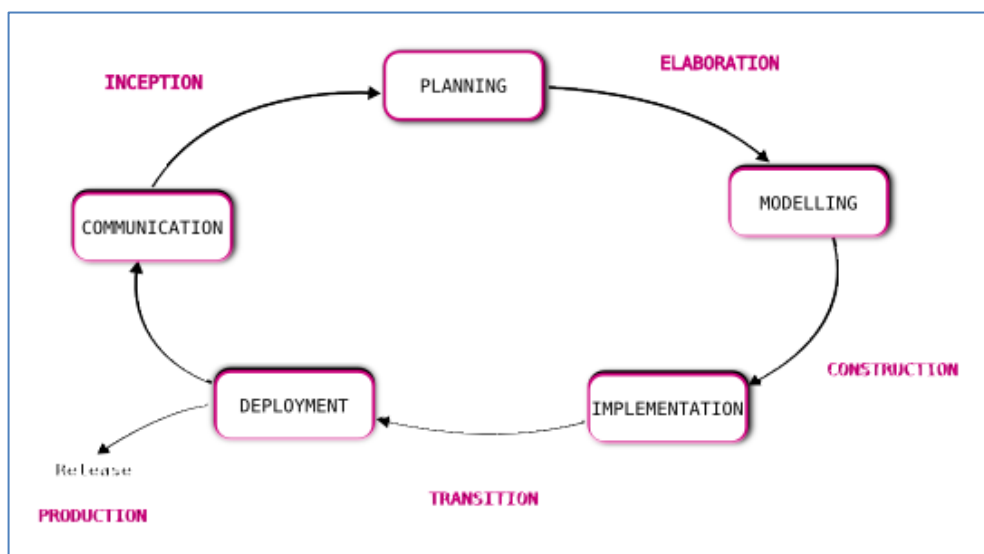
8. Rapid Application Development (RAD [Martin, 1991]), уводи јасно временско планирање рада и комуникације, као додатни фокус поред праћења ризика, како би унапредио брзину развоја и смањио време реализације.

9. Агилни приступ ([Cohen et al, 2004]) – поставља већи фокус на људе, њихове способности и непосредну комуникацију, пре него на менаџмент и комуникацију документацијом. Фокус ових метода је на ближој и фреквентнијој комуникацији са будућим корисницима софтвера, прихватању и реализацији промена у циљу повећања задовољства корисника и производњи доброг софтвера. Прилагодљивост на промене је главни основ адаптивности. Једна од најпопуларнијих метода агилног приступа је Extreme Programming [Pressman, 2007].



Слика 1.3.1.2.3.8. Животни циклус Extreme Programming према Castro Souto [Castro Souto, 2010]

10. Unified Process (Ivar Jacobson, Grady Booch, and James Rumbaugh, 1999, [Jacobson et al, 1999]) – укључују агилне методе и прилагођавају их практичној употреби



Слика 1.3.1.2.3.9. Модел Unified Process-а према Castro Souto [Castro Souto, 2010]

11. Неке од каснијих методологија укључују модификације раније већ приказаних методологија, као што су [Castro Souto, 2010]: Aspect-Oriented Software Development (AOD), Adaptive Software Development (ASD), Feature Driven Development (FDD), Test Driven Development итд.

#### **1.3.1.2.4. Приступу развоју софтвера у оквиру развоја информационог система**

Према [Cadle&Yeates, 2008], у току развоја методологија софтверског инжењерства, јављали су се различити приступи развоју информационог система, у оквиру којег се реализује развој софтвера:

1. Традиционални приступ - прати Waterfall модел развоја, минимизује учешће крајњих корисника, наглашава улогу потпуности спецификације захтева корисника на почетку пројекта и начина реализације, пре него да је фокус на томе шта ће бити реализовано и колико је то у складу са потребама пословања.

2. Структурни приступ - 1980-тих и 1990-тих година појављује се као унапређење у односу на традиционални приступ укључивањем корисника у евалуацију решења, одвајањем логичког и физичког аспекта, додавањем већег значаја подацима него процесирању, с обзиром да су стабилнији у односу на промене, дијаграмско документовање, јасно дефинисана структура система и организација реализације.

3. Агилни приступ – у циљу брже реализације резултата, јављају се Rapid Application Development (RAD) приступи, касније названи агилни приступи. Агилни приступи укључују коришћење прототипа, итеративно-инкременталног и спирално-еволутивног развоја, са нагласком на самосталност реализатора у доношењу одлука, флексибилност, квалитет у складу са потребама, јасно дефинисаним кратким временским роковима за испоруку. Агилни приступи су погоднији за мање пројекте. Најпознатије агилне методологије које прецизније дефинишу конкретни начин имплементације су су SCRUM и DSDM (Dynamic System Development Method).

4. Објектно орјентисане методе развоја – базиране на класама, наслеђивању, енкапсулацији и полиморфизму, омогућавају модуларни развој и олакшавање одржавања софтвера.

5. UML и Unified Process (UP) – базиран на Rational Unified Process-у фирме Rational, данас представља стандард у објектно орјентисаном развоју кроз визуални језик објектно орјентисаног развоја UML и развојни процес UP. Процес се састоји из четири фазе: концептуализација (inception), елаборација (elaboration), конструкција (construction) и пренос-испорука (transition). У оквиру сваке фазе, реализују се радни токови који се односе на: спецификацију захтева корисника (requirements), анализу (analysis), дизајн (design), имплементацију (implementation - building the software) и тестирање (test).

6. Компонент-базиран развој – у оквиру рада на неком софтверском пројекту, реализују се модули – компоненте које могу бити корисне и другим пројектима у будућности. Ти будући пројекти би тада повећали брзину и квалитет реализације, користећи готове библиотеке компоненти из претходних пројеката. Ипак, на почетку таква развој је спорији, јер захтева више времена у осмишљавању свеобухватнијих и општијих решења компоненти које би се могле користити и у другим системима.

7. Екстремно програмирање – намењено је мањим пројектима, са фреквентним изменама захтева или у оквиру рада на новим раније непознатим технологијама. Корисник мора стално бити у контакту са развојним тимом. Најчешће се примењује програмирање у пару (два програмера за истим рачунаром, раде заједно – програмирају и тестирају).

8. Пакет-базиран приступ – организације су наклоњене набавци и прилагођавању (ако је могуће) готових софтверских пакета или чак комплетних система, сматрајући их јефтинијим, бржим, поузданијим решењима. Узимајући то у обзир, софтверске фирме креирају пакет-базирана решења која могу да се прилагођавају различитим корисницима.

С обзиром да информациони систем развија за потребе и у окружењу пословних организационих система, на развој и улогу информационог система утичу промене организационог система применом [Cadle&Yeates, 2008]:

1. Методологија софт система (197-тих и 1980-тих, Peter Checkland са Lancaster University) – приступ да у природи и свим системима нису јасно дефинисани захтеви, већ се у низу интеракција прецизније дефинишу захтеви, разматрају различита решења у складу са ограничењима и изведивости.
2. Социо-технички приступ – унапређење задовољства запослених утиче на повећање квалитета рада и боље економске ефекте
3. Реинжењеринг пословних процеса – промена начина размишљања, основних структурних и функционалних елемената организације.

### 1.3.1.3. УПРАВЉАЊЕ ПРОЈЕКТИМА

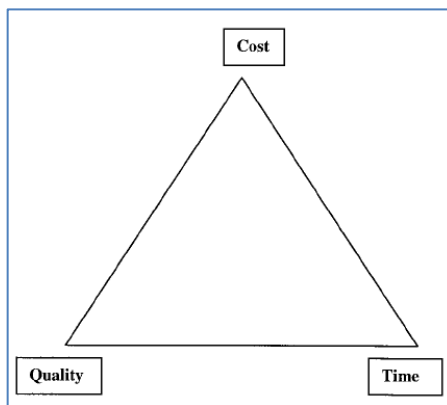
#### 1.3.1.3.1. Дефиниција

«Управљање пројектом представља научно заснован и у пракси проверен концепт којим се, уз помоћ одговарајућих метода организације, информатике, планирања, вођења и контроле, врши рационално усклађивање потребних ресурса и координација потребних активности да би се одређени пројекат реализовао на најбољи начин.» [Јовановић, 2006] «Управљање пројектом је умијеће како изводити пројекат сарадњом људи у договореном времену, одређеним средствима рада и трошковима, са жељеним учинком.» [Наус, 1975]

Према [Atkinson, 1999], британски стандард за пројектни менаџмент (BS60794) из 1996. године дефинише пројектни менаџмент као: „Планирање, мониторинг и контрола свих аспеката пројекта и мотивација свих укључених у пројекат како би реализовали пројектне задатке на време и у складу са специфицираним трошковима, квалитетом и перформансама“. Асоцијација Уједињеног Краљевства за пројектни менаџмент је у оквиру UKBOK Body of Knowledge дефинисала пројектни менаџмент као: „Планирање, организација, мониторинг и контролисање свих аспеката пројекта и мотивација свих укључених како би реализовали пројектне задатке безбедно и у оквиру договореног времена, трошкова и критеријума перформанси.“

Основни циљ управљања реализацијом сваког пројекта је да се реализацијом обезбеде захтеване техничке перформансе и квалитет пројекта уз најмање време и трошкове реализације.» [Јовановић, 2006]

Слика 1.3.1.3.1.1. представља основне параметре у управљању пројектима. Планирањем и изменама ових параметара утиче се на вероватноћу наступања ризичних догађаја који могу да утичу на успешност исхода пројекта. [Јовановић, 2006]

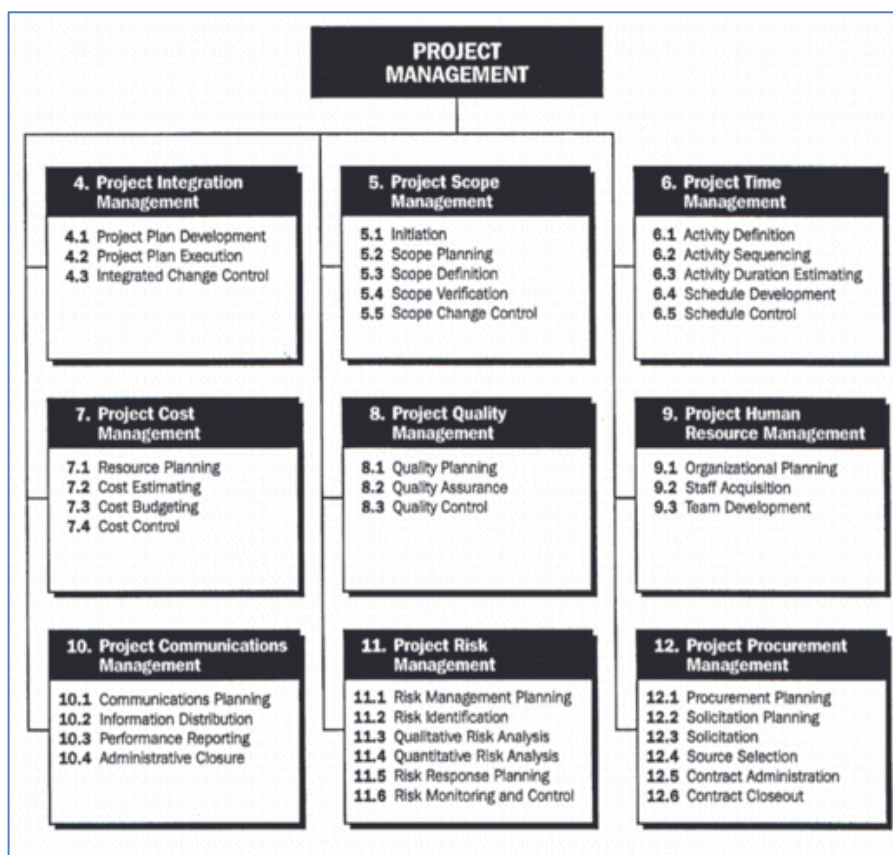


Слика 1.3.1.3.1.1. Основни параметри у управљању пројектима (Гвоздени троугао) према Atkinson-у [Atkinson, 1999]

#### 1.3.1.3.2. Функционалне области и стандарди

Функционалне области које представљају области знања и активности у области управљања пројектом дефинисане су дескриптивно у оквиру PMBOK („Project Management Book of Knowledge“) [PMBOK, 2004] методологије, која представља међународно признати стандардни методолошки оквир у управљању пројектима, развијен од стране Project Management Institute из Сједињених Америчких Држава.





Слика 1.3.1.3.2.1. Основне области управљања пројектима, према РМВОК [РМВОК, 2004]

Наведене функционалне области покривају најважније аспекте управљања пројектима и обухватају [РМВОК, 2004]: Управљање обимом пројекта, Управљање временом, Управљање трошковима, Управљање квалитетом, Управљање уговарањем, Управљање набавком, Управљање људским ресурсима, Управљање комуникацијама, Управљање конфликтима, Управљање променама, Управљање ризиком. Према слици 1.3.1.3.2.1. у оквиру одговарајућих области обухваћено је:

- Управљање обимом пројекта – дефинисање циља, резултата пројекта и његовог обима у квантитативном и квалитативном смислу, као и дефинисањем обима посла ( броја и врсте активности, трајања активности и утрошка ресурса) у процесу планирања и касније праћењем промена у пројекту.
- Управљање временом у пројекту – планирање времена кроз животни циклус пројекта и праћење реализације активности у смислу упоређивања планираног и реализованог времена, оптимизација времена трајања активности, процене трајања пројекта....
- Управљање трошковима у пројекту – планирање трошкова по активностима, праћење реализације трошкова, оптимизација трошкова...
- Управљање квалитетом – планирање квалитета кроз евиденције уговорних захтева, стандарда и прописа, прописивање карактеристика резултата и дефинисање начина одвијања процеса реализације пројекта, контролисање реализације елемената пројекта и усклађивање са потребним захтевима, корекције.
- Управљање уговарањем – односи се на планирање уговарања и реализацију уговарања и праћење испуњавања уговорних обавеза и одговорности учесника у пројекту, пре свега извођача реализације пројекта.
- Управљање набавком – процес набавке свих потребних ресурса, у складу са одвијањем активности и њиховим потребама.
- Управљање људским ресурсима – планирање, обука, распоређивање, руковођење (усмеравање, координација, информисање, мотивисање,

управљање конфликтима) људским ресурсима у току реализације пројекта, ради постизања циљева пројекта – успешан завршетак пројекта у оквиру предвиђеног времена, трошкова и квалитета.

- Управљање комуникацијама – односи се на двосмерну комуникацију руководиоца тима са свим осталим учесницима пројекта, као и међусобну комуникацију учесника пројекта. Реализује се кроз креирање и размену свих потребних информација у току целог животног циклуса пројекта. Важно је да се информације дистрибуирају у обиму, квалитету и времену које је адекватно улози учесника пројекта, статусу активности на коју се односи итд.
- Управљање конфликтима – дефинисање могућих извора конфликта и планирање реакција и реализације одговора, реаговање на настале конфликте, решавање конфликта.
- Управљање променама у пројекту – дефинисање могућих промена, планирање промена, увођење промена, праћење реализације промена у сврху ефикасније реализације пројекта.
- Управљање ризиком – идентификација могућих ризичних догађаја, анализа и процена њиховог утицаја на реализацију пројекта, планирање могућих одговора на ризичне догађаје и контрола спровођења одговора, односно реакција. Величина ризика се процењује на основу критеријума:
  - флексибилност пројекта (већа флексибилност крајњег исхода пројекта носи већи ризик),
  - величина пројекта (параметри: трајање пројекта, број људи итд – што је већа величина пројекта, већи је ризик),
  - технологија (висока технологија односи се на сложеније пројекте и потребна специфична знања и искуства, чиме се повећава ризик пројекта).

Мењањем параметара пројекта (обим, време, карактеристике квалитета) утиче се на смањење вероватноће наступања ризичних догађаја који могу да утичу на исход пројекта. [Јовановић, 2006]

Према РМВОК из 2013. године [РМВОК, 2013], следеће области припадају управљању пројектима:

- Управљање интеграцијом у пројекту (Project Integration Management)
- Управљање обухватом пројекта (Project Scope Management)
- Управљање временом у пројекту (Project Time Management)
- Управљање трошковима пројекта (Project Cost Management)
- Управљање квалитетом у пројекту (Project Quality Management)
- Управљање људским ресурсима пројекта (Project Human Resource Management)
- Управљање комуникацијама у пројекту (Project Communications Management)
- Управљање ризиком у пројекту (Project Risk Management)
- Управљање набавкама у пројекту (Project Procurement Management)
- Управљање заинтересованим странама пројекта (Project Stakeholder Management)

Према истом стандарду РМВОК [РМВОК, 2013], укупно 47 активности у пројекту подељене су у неколико група процеса:

- Група процеса иницијације (Initiating Process Group) Ови процеси се изводе како би се дефинисао нови пројекат или нова фаза постојећег пројекта остваривањем ауторизације како би се покренуо пројекат или фаза.
- Група процеса планирања (Planning Process Group) Ови процеси су потребни да остваре обухват пројекта, прецизније дефинишу задатке, дефинишу курс активности потребних за остваривање задатака и циљева које је пројекат преузео да оствари.
- Група процеса извршавања (Executing Process Group) Ови процеси се извршавају да би завршили посао који је дефинисан планом пројекта, како би задовољио спецификације пројекта.
- Група процеса мониторинга и контроле (Monitoring and Controlling Process Group) Ови процеси потребни су да прате, прегледају и регулишу напредак и перформансе

пројекта, идентификују области где су промене плана потребне и инцирају одговарајуће промене.

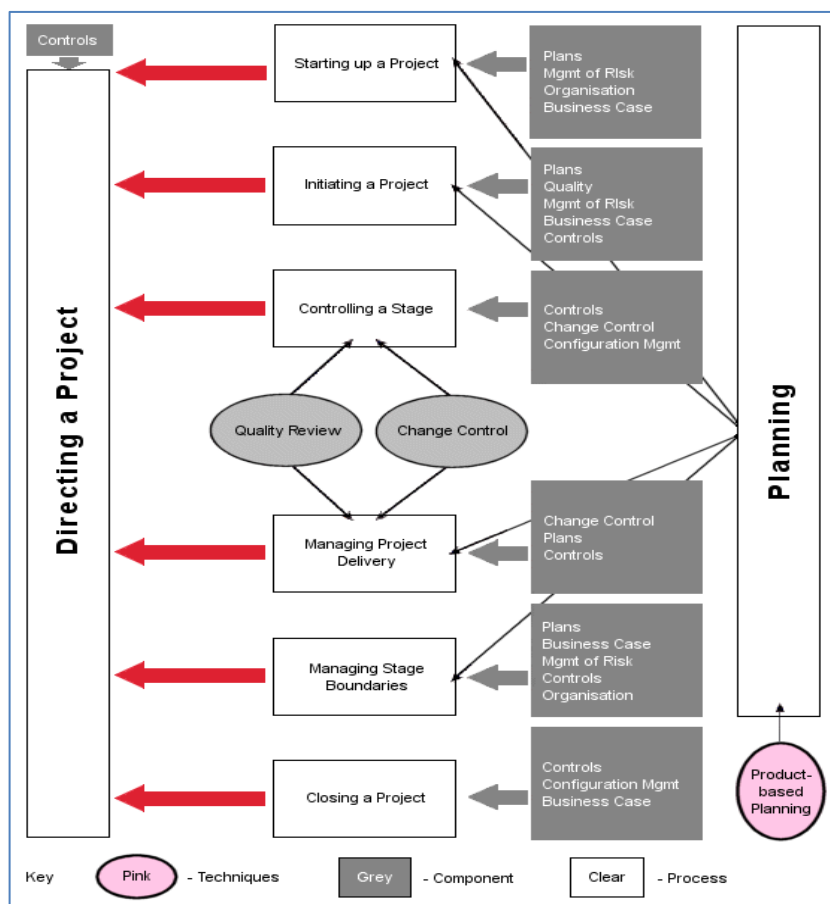
- Група процеса затварања (Closing Process Group) Наведени процеси се извршавају како би финализирали активности кроз све остале групе процеса како би формално затворили пројекат или фазу.

### 1.3.1.3.3. Компарација PMBOK и PRINCE 2 методологије

Методолошки оквир дат у PMBOK представља дескриптивни основ за развој прецизнијих методологија управљања пројектима у специфичним областима. Једна од најпотпунијих светски признатих и широко коришћених методологија која даје прескриптивни приступ у смислу конкретних упутстава, структуре документације и описа начина извођења појединих активности у процесу управљања пројектима је PRINCE2 (Projects in Controlled Environments) [CCTA, 1999]. Развој ове методологије започео ј 1975. године у Великој Британији, а последња верзија је развијена 2009. године. Ова методологија развијена је у складу са ISO 9000 и CMM методологијама за управљање квалитетом. PRINCE2 методологија се састоји од осам процеса, осам компоненти и три технике.

Siegelaub [Siegelaub, 2004] је приказао компарацију наведена два стандарда, односно методологије за управљање пројектима:

- PMBOK – свеобухватна, широко дескриптивна методологија. Прескриптивна (даје конкретна упутства) само на високом нивоу посматрања, орјентисана је на корисничке захтеве, изузетан значај спонзора и заинтересованих страна (stakeholders), USA/међународни стандард
- PRINCE2 – фокусира се на кључне области ризија, висок ниво прескриптивности (даје много конкретних упутстава), адаптабилна на било коју величину пројекта, орјентисан на пословни проблем, изузетан значај старијег менаџмента организације у одговорности у пројекту, UK стандард.



Слика 1.3.1.3.3.1. Структура PRINCE2 методологије, према CCTA [CCTA, 1999]

На основу претходно представљене анализе, јасно је да РМВОК представља широк опис знања (појмова, техника и поступака) у овој области, док PRINCE2 методологија обезбеђује детаљнија упутства о начину извршавања конкретних активности у оквиру животног циклуса пројекта, те је погодна као детаљнији оквир примене у реалним пројектима.

#### 1.3.1.3.4. Основне фазе, методе и технике у управљању пројектима

Основне фазе у реализацији пројекта представљене су следећом табелом:

Табела 1.3.1.3.4.1. Општи модел процеса управљања пројектом, према Јовановићу [Jovanovic, 2006]

<b>УПРАВЉАЊЕ ПРОЈЕКТОМ</b>	<b>Постављање циља</b>	Резултат
		Време
		Трошкови (ресурси)
	<b>Планирање</b>	Одређивање структуре резултата
		Израчунавање трајања
		Планирање ресурса
		Утврђивање буџета
	<b>Организација</b>	Функције
		Комплетирање персонала
		Неопходне инструкције
		Узајамна повезаност
	<b>Контрола</b>	Резултат
		Време
		Трошкови (ресурси)

У оквиру различитих фаза животног циклуса пројекта примењују се различите методе и технике [Jovanovic, 2006]. У наставку је дат табеларни приказ најчешће коришћених метода и техника у области управљања пројектима, са кратким објашњењем сврхе и начина примене.

Табела 1.3.1.3.4.2. Приказ најчешће коришћених метода и техника у планирању пројекта, према Јовановићу [Jovanovic, 2006]

<b>МЕТОДА-ТЕХНИКА</b>	<b>СВРХА</b>	<b>НАЧИН ПРИМЕНЕ</b>
PBS – product breakdown structure	Одређивање структуре производа	Подела на технолошки јасно дефинисане целине
WBS – work breakdown structure	Одређивање структуре послова и активности, добија се организациона шема, тј. Управљачка структура организације Хијерархијско уређење активности од делова пројекта до појединачних задатака	Одређивање структуре послова и активности, добија се организациона шема, тј. Управљачка структура организације Хијерархијско уређење активности од делова пројекта до појединачних задатака
OBS – organization breakdown structure	Одређивање структуре организације	Расподела послова из WBS према организационим јединицама и радним местима-појединцима који треба да буду одговорни за реализацију
RACI (Responsability, Accountability, Communication, Information) матрица	Одређивање одговорности појединаца за поједине активности	Прелапањем WBS и OBS структуре, прецизно одређивање одговорних појединаца и извршилаца.
Процена времена –Метод кључних	Кључни догађаји су најчешће почетак или завршетак неког дела пројекта.	Процена времена –Метод кључних догађаја Одређивање

догађаја Одређивање кључних догађаја (милестоне) у пројекту		кључних догађаја (милестоне) у пројекту
Процена времена – Гантограми	Графички приказ ради планирања одвијања задатака и послова у времену	Гантограм је матрица – вертикална оса – активност, хоризонтална оса – време. Дужина линије означава трајање активности. Користи се за једноставније пројекте.
Процена времена – Технике мрежног планирања Планирање целог тока реализације пројекта са становишта времена и трошкова, ресурса	Код веома сложених пројеката са много активности и учесника. Приказује поред трајања и повезаност активности у реализацији пројекта. Мрежни дијаграм – Кружићи су догађаји (стања система – немају временску димензију), стрелице су активности (дужина стрелице не означава трајање).	Процена времена – Технике мрежног планирања Планирање целог тока реализације пројекта са становишта времена и трошкова, ресурса
- CPM метод	Critical Path Method	Време активности познато, односно може се једнозначно одредити (проценити). Израчунава се најранији и најкаснији почетак и завршетак активности и издвајају активности које су критичне и критичан пут.
- PERT метод	Program Evolution and Review Technique	Време трајања активности непознато, процењује се са три вредности времена: оптимистичко, нормално и песимистичко
- PD метод - MDM метод	Precedence Diagramming Modelling Potential Methode	Мрежни дијаграм – у правоугаонцима активности, могућност да се једна активност надовезује (наставља) на другу, стрелице означавају ток и зависност активности.
Распоређивање радне снаге	Распоред радника по активностима	На основу профила радника, компетенције и броја расположивих радника, користећи гантограме, уз померање извршавања активности ако је потребно услед недостатка радне снаге
Процена трошкова	Процена трошкова активности и пројекта у целини	На основу трајања активности, учешћа ресурса: - искуствени метод - статистички метод (на основу реалних трошкова сличних пројеката) - нормативни метод (на основу норматива или стандарда)
Анализа трошкова – PERT COST метода	Проналажење најбољег односа времена и трошкова реализације	Однос времена и трошкова је обрнута сразмера. Врши се анализа и оптимизација.
Метода остварене вредности	Процена степена извршености пројекта и предвиђање исхода	Поређење планираног времена и трошкова реализације са остваренима по питању времена и трошкова. Користе се показатељи: планирани

		трошкови планираног рада, планирани трошкови извршеног рада, стварни трошкови извршеног рада.
--	--	--

## 1.3.2. ТЕХНОЛОШКА ОСНОВА

### 1.3.2.1. ДИСТРИБУИРАНИ ИНФОРМАЦИОНИ СИСТЕМИ

#### 1.3.2.1.1. Дефиниција дистрибуираног информационог система, технологије и примене

Главни разлог за изградњу дистрибуираног информационог система и дистрибуиране базе података је да се на одговарајући начин обезбеди подршка информационих захтева корисника реалног система, чије пословање обухвата шире географско подручје [Mogin et al, 2000].

Према [Mogin et al, 2000], под појмом дистрибуирани информациони систем подразумева се информациони систем који подржава дистрибуирану обраду података, над дистрибуираном базом података. У истраживањима и пракси термин дистрибуираног информационог система се изједначава са термином дистрибуирани систем. "Дистрибуирани систем је софтверски систем у којем су компоненте смештене на рачунарима повезаних рачунарском мрежом и у којем те компоненте комуницирају и координирају своје акције прослеђивањем порука. Компоненте реализују међусобну интеракцију како би оствариле заједнички циљ." [Coulouris et al, 2011]

Према [Goodchild et al, 2004], дистрибуирано рачунарство (distributed computing) у пуном смислу речи подразумева подршку:

- симултаном приступу већем броју сервера од стране једног клијента,
- подршку дистрибуираним базама података, тако што корисник види јединствену базу података, али се табеле или чак делови табеле налазе на различитим серверским локацијама,
- подршку дистрибуираном софтверу, тако што корисник види јединствено софтверско окружење, али модули се налазе на различитим серверским локацијама.

Према [Tsui&Jeng, 2009], дистрибуирани систем представља интеграцију сервиса система, који се извршавају на више рачунара са дистрибуираним ресурсима и контролом, док коришћењем корисник има утисак да се ради о јединственом рачунару. Дистрибуирани систем се састоји од конкурентних процеса који приступају дистрибуираним ресурсима кроз размену порука у мрежном окружењу.

Технологије дистрибуираних информационих система обухватају [Coulouris et al, 2011]:

1. Технологије организације сервера, сервиса и клијената – grid и cloud computing, mobile computing
2. Технологије дистрибуираних база података – дистрибуирани системи за управљање базама података, уз подршку репликацијама, фрагментацији и дистрибуираним трансакцијама
3. Технологије дистрибуираних софтверских компоненти – web servisi, агенти, дистрибуирани мултимедијални системи итд.

Неки од примера примене дистрибуираних система обухватају [Andrews, 2000], [Ghosh, 2007], [Lynch, 1996], [Peleg, 2000], [Elmasri& Navathe, 2000]:

- Телекомуникационе мреже: телефонске и мобилне мреже; рачунарске мреже, као што је интернет; Wireless сензорске мреже; Алгоритми рутирања
- Мрежне апликације: World Wide Web и Peer-to-peer мреже; Масивне вишекорисничке он-лине игре и заједнице (communities) он-лине виртуалне реалности; Дистрибуиране базе података и системи за управљање дистрибуираним базама података; Мрежни фајл системи; Дистрибуирани системи процесирања информација, као што су банкарски системи или системи резервација авионских летова
- Контрола процеса у реалном времену – контрола лета авиона, индустријски контролни системи

- Паралелно процесирање: кластер рачунарство, грид рачунарство, дистрибуирано процесирање (rendering) слика у компјутерској графички...

### **1.3.2.1.2. Дистрибуирана обрада података и клијент сервер модел**

„Дистрибуирана обрада података је обрада података над дистрибуираном базом података, при чему задаци логике обраде података и управљања обрадом података могу се расподелити на све сервере који партиципирају у дистрибуцији. На основу ове дефиниције произилази да обрада података, код које су делови базе података смештени на више различитих рачунара, или различитих меморијских уређаја, при чему је сама обрада података увек централизована, односно није могуће извршити дистрибуцију задатака логике обраде података и управљања обрадом података, не представља дистрибуирану обраду.“ [Mogin et al, 2000]

Клијент-сервер модел обраде података „представља такву врсту дистрибуиране обраде података, код које се функције једног корисничког програма расподељују на најмање два процеса, који међусобно комуницирају. Процеси који учествују у клијент-сервер обради се класификују на два типа: клијентски процеси и серверски процеси, при чему је логика комуникације клијентског и серверског процеса таква да клијентски процес шаље поруку серверском процесу, којом захтева „услугу“ серверског процеса, а серверски процес одговара на захтев клијентског процеса тако што, било успешно или неуспешно, извршава захтевани задатак и шаље поруку клијентском процесу. Та порука представља резултат траженог задатка.“ [Mogin et al, 2000] Рачунар намењен извршавању клијентских процеса назива се клијент, а рачунар намењен извршавању серверских процеса назива се сервер. [Mogin et al, 2000] Расподела задатака у клијент-сервер систему може се груписати на:

1. Управљање презентацијом: рад са корисничким интерфејсом - екранским формама, приказом и штампом извештаја, употреба менија, икона и других графичких елемената корисничког интерфејса
2. Логика презентације: трансформисање података са презентационог на концептуални ниво и обратно, покретање трансакција и обрада порука о резултатима рада трансакција, обрада података који се налазе у радној зони програма, обрада резултата дијалога корисника и програма.
3. Прослеђивање порука и података
4. Логика обраде података: трансформисање података с концептуалног на физички ниво и обратно, управљање операцијама читања и ажурирања података, спровођење контроле конзистентности података, обрада грешака које могу настати при реализацији операција читања и ажурирања или контроли конзистентности.
5. Управљање обрадом података: улазно-излазне активности – отварање и затварање датотека, физички и логички пренос података ка-из меморијског уређаја у радну меморију, детекција грешака при реализацији улазно-излазних операција.

У трослојном моделу клијент-сервер система, основни елементи архитектуре су [Mogin et al, 2000]:

1. Сервер базе података – задужен је за подршку управљања обрадом података и логике обраде података
2. Апликациони сервер – задужен је за подршку логике презентације
3. Апликациони клијент – задужен је за подршку управљања презентацијом.

### **1.3.2.1.3. Дистрибуирана база података**

„Дистрибуирана база података представља такву базу података у којој су подаци физички смештени на најмање два сервера базе података. Без обзира што су делови дистрибуиране базе података расподељени на више сервера, реч је о једној бази података која у логичком смислу представља јединствену целину. Предуслов за формирање и коришћење дистрибуиране базе података је да сви сервери базе



података који партиципирају у дистрибуцији, буду повезани комуникационом мрежом.” [Mogin et al, 2000].

Дистрибуирани системи за управљање базама података [Mogin et al, 2000]– системи за управљање базама података проширени механизмима и језиком за опис дистрибуције шеме базе података и репликације података по серверима, а такође и елементима који омогућавају симултано коришћење и ажурирање података, дистрибуираних на различите сервере базе података. Треба да омогућавају реализацију глобалних трансакција, закључавања, репликације, опоравка базе података.

Дистрибуирана база података може бити организована као хомогена (коришћен је само један систем за управљање базом података) или хетерогена (коришћено је најмање два система за управљање базом података). Једна од најважнијих особина дистрибуираних база података је дистрибуциона транспарентност односно независност – све апликације информационог система треба да буду независне од дистрибуције података по серверима, што значи да измене у начину дистрибуирања података по серверима не смеју утицати на рад апликација. Крајњи корисници дистрибуираног информационог система у принципу не треба да познају његову архитектуру. Ни програмер апликација информационог система не треба да познаје начин дистрибуције базе података по серверима, већ апликације треба да функционишу независно од тога како је извршена дистрибуција базе података.

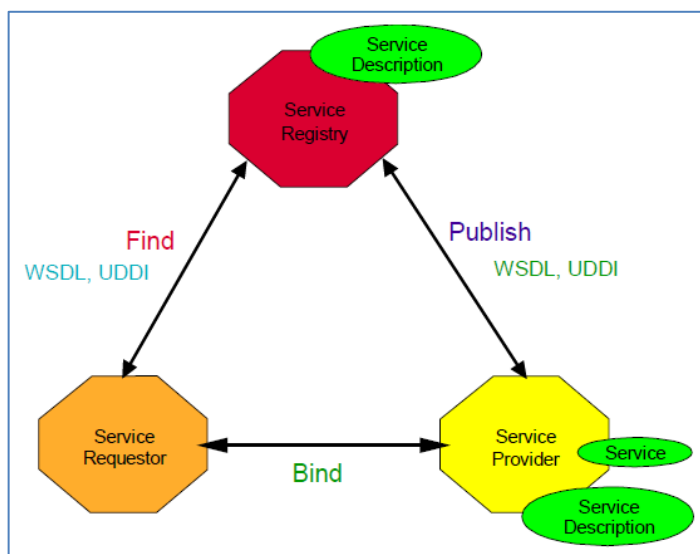
#### **1.3.2.1.4. Дистрибуиране софтверске компоненте – web сервиси**

Према [Alonso et al, 2004], коришћење web сервиса се заснива на претпоставци да је функционалност која је дата јавности на располагање обликована и изложена као сервис. Суштински, сервис представља поцедуру, метод или објекат са стабилним интерфејсом који може бити позван од стране клијента. Захтевање и извршавање сервиса подразумева да један рачунарски програм позива други (који реализује сервис). Према [Parazoglou&Georgakopoulos, 2003], сервис-орјентисано рачунарство представља примену сервиса као фундаментални елемент у развоју апликација. Сервисно-орјентисане архитектуре заснивају се на сервисима, њиховим описима и подршци основним операцијама (публикација сервиса, откривање сервиса, селекција и повезивање). Према [Parazoglou&Georgakopoulos, 2003], под сервисом подразумевамо самостално-описане отворене компоненте које подржавају брзу и јефтину композицију дистрибуираних апликација.

У [Alonso et al, 2004] описана је дефиниција web сервиса од стране UDDI конзорцијума као: „самосталне (self-contained), модулатне пословне апликације које имају отворене, интернет-орјентисане и на стандардима засноване интерфејсе.“, односно од стране World Wide Web конзорцијума: „Софтверска апликација идентификована коришћењем URI, чији интерфејси и везе имају способност да буду дефинисани, описани и откривени као XML артефакти. Web сервис подржава директне интеракције са другим софтверским агентима користећи XML-базиране поруке које се размењују путем Интернет-базираних протокола“. У оквиру [Kreger, 2011] описан је основни концепт примене web сервиса, који се састоји из Service Provider, Service Requestor, Service Registry (Слика 1.3.2.1.4.1.).

Према [Alonso et al, 2004] [Curbera et al, 2002], најчешће технике и протоколи у примени web сервиса укључују XML (Extensible Markup Language) за структурирање података, SOAP (Simple Object Access Protocol) за трансфер података, односно омогућава комуникацију између Web сервиса, WSDL (Web Service Description Language) за опис расположивих web сервиса, односно представља формалан опис web сервиса, а UDDI (Universal Description, Discovery and Integration) директоријум, односно регистар описа расположивих web сервиса, који се користи као листинг расположивих web сервиса. Према [Pautasso, 2008], савремени web сервиси су реализовани применом једног од два најчешћа технолошка приступа/стандарда: REST (RESTful – Representational State Transfer) и SOAP, чија је компарација извршена у овом раду.

Основни технолошки елементи RESTful стандарда су: RSS, XML, JSON, MIME, URI, HTTP, SSL. Основна сличност два наведена приступа је у удаљеном позиву процедура, а разлика је у томе што је код SOAP приступа изражена размена порука, где се комуникација одвија на основу протокола транспортног нивоа, а код REST се комуникација одвија на основу протокола апликационог нивоа.



Слика 1.3.2.1.4.1. Основни концепт примене Web сервиса према Kreger [Kreger, 2011]

У раду [Dustdar&Schreiner, 2005] разматрана је композиција web сервиса, где се од стране клијентске апликације и циљу реализације пословне логике клијентске апликације позива више web сервиса или се у циљу реализације логике једног web сервиса позивају други web сервиси и такви сложени web сервиси се називају композитни сервиси. Процес креирања композитних сервиса назива се сервисна композиција. За потребе реализације пословних процеса користећи web сервисе, креиран је језик BPEL4WS (Business Process Execution Language form Web Services). Web сервиси могу бити статички и динамички позивани у оквиру извршавања пословне логике. Редослед и услови под којима се web сервиси позивају описани су тзв. моделима оркестрације који користе језике моделовања процеса, као што су: дијаграм активности УМЛ-а, Петријеве мреже, графови стања, оркестрација заснована на правилима и други. Посебно је значајна повезаност семантике и web сервиса коришћењем онтологија у оквиру примене семантичког web-а.

#### 1.3.2.1.4. Дистрибуиране софтверске компоненте – софтверски агенти

Према [Badica et al, 2011], појам софтверског агента води порекло од више од 30 година уназад и дефинисан је у оквиру истраживања у области дистрибуиране вештачке интелигенције. Према [Green et al, 1997], под појмом софтверски агент (или краће агент) подразумева се софтверски систем или рачунарски ентитет који треба да помогне корисницима у реализацији задатака који се односе на примену рачунара, а задовољава следеће карактеристике: реализује активности у име неког другог ентитета у аутономном стилу, изводи акције са одређеним нивоом проактивности и/или реактивности, приказује одређени ниво особина учења, кооперације и мобилности. Према [Badica et al, 2011] кључне карактеристике софтверског агента: софтверски ентитети који реализују скуп операција у име корисника или другог програма са одређеним нивоом (или полу-) независности/аутономије (функционишу без директне интервенције човека или других система и исказују извесну контролу над својим акцијама и интерним стањем), примењује одређено знање, делује у складу са корисниковим циљевима или жељама, проактивни су (не делују само у одговору на окружење, већ имају и циљно-фокусирано понашање и предузимају иницијативу), адаптивни, интерагују са окружењем (перципирају окружење и реагују на промене) и другим агентима или људима примењујући одређени језик комуникације са агентима.

Строге дефиниције агената захтевају да агенти, у складу са одговарајућом формалном теоријом која је основ функционисања, исказују особине људских менталних карактеристика: веровања, планове, циљеве, намере, жеље, посвећеност...

Према [Badica et al, 2011] , агенти се могу класификовати према различитим карактеристикама, нпр. статични/мобилни агенти, велики/средњи/микроагенти, адаптивни (адаптирају се на непредвиђиве промене окружења) или неадаптивни агенти, агенти способни да уче (на основу datamining-a). У оквиру мултиагентног система, неки од језика за комуникацију између агената су: KQML, FIPA-ACL, ARCOL, KIF, COOL, а користе се и протоколи базирани на уговорима у повезивању агената. Програмски језици за програмирање агената укључују Java, C, C++, али и Pascal, C, Lisp, Prolog, Smalltalk. Посебно су развијени програмски језици за програмирање агената и могу се разликовати агент-орјентисани програмски језици (AGENTO, PLACA, Agent-K, MetateM, April, MAIL, Viva, Go!), веровање-жеља-намера (BDI) језици (AgentSpeak, AF-APL, ZAPL, JACK, JADEX), хибридни језици (Jason), као и језици засновани на декларативној парадигми (GOAL, GOLOG, FLUX, CLAIM). Неки од најпопуларнијих развојних окружења за креирање агената су: ZEUS, JADE, agenToo, RETSINA, JATLite, FIPA-OS, MADKIT, JAFMAS, Agent Building Shell, OAA, Cougar, AgentScape, Cybele.

### **1.3.2.1.5. Пројектовање дистрибуције информационог система**

Према [Tsui&Jeng, 2009], како би се превазишли потенцијални проблеми дистрибуираног система, реализује се дизајн система тако да се:

1. Дефинише модел система и одреде компоненте
2. Уреди се интеракција између компоненти
3. Обезбеди се комуникација компоненти
4. Изврши се заштита компоненти и реализују мере безбедности система.

Пројектовање дистрибуције информационог система састоји се у фазама [Mogin et al, 2000]: стратегија, анализа и пројектовање. Резултат пројектовања су документи:

1. Спецификација дистрибуција апликација по типовима локација – дистрибуциона шема апликација
2. Спецификација дистрибуције базе података по типовима локација – дистрибуциона шема базе података
3. Спецификација репликације података у дистрибуираној бази података – репликациона шема базе података.

Дистрибуциона шема базе података [Mogin et al, 2000] укључује тзв. „Чисту дистрибуцију“ шеме базе података, где поједини делови базе података на појединим локацијама немају редундансу (понављање података, односно структуре). Чиста дистрибуција се постиже фрагментацијом шеме базе података. Идеја увођења фрагментације шеме базе података је у томе да се изврши партиционисање шеме базе података, које ће представљати локалне базе података. Партиционисањем се добија фрагментациона шема базе података. Распоређивањем фрагмената шеме базе података на типове локација добија се дистрибуциона шема базе података. Постоје следеће врсте фрагментација:

1. Вертикална (одређен број колона табеле чува се у једној фрагментираној бази података, други скуп колона у другој фрагментираној бази).
2. Хоризонтална – записи се према неком критеријуму разврставања чувају у различитим базама података исте структуре табела.
3. Комбинована (хоризонтално-вертикална).

## **1.3.2.2. АДАПТИВНИ И АДАПТИБИЛНИ СИСТЕМИ И СОФТВЕР**

### **1.3.2.2.1. Управљачки системи**

Према [Jacic et al] разликујемо 3 врсте управљања, у односу на улогу коју човек има у извршавању управљања:

1. Мануелно (човек у потпуности преузима улогу управљачког система),
2. Полуаутоматско (управљачки систем је састављен делом од уређаја, а човек преузима извршење одређених задатака),
3. Систем аутоматског управљања - уређај реализује управљачки систем који сам генерише управљачка дејства.

Према [Jacic et al], систем је стабилан ако се, изведен из равнотежног стања и препуштем самом себи, после довољно дугог времена враћа у првобитно равнотежно стање. Нестабилан систем се након избацивања из равнотежног стања не враћа у првобитно стање, већ дивергира, удаљава се од првобитног равнотежног стања.

У односу на концепт управљања системи аутоматског управљања се деле на:

- Отворене системе аутоматског управљања
  - Отворени системи аутоматског управљања без компензације поремећаја – за формирање управљачког дејства се користи само информација о жељеном излазу система, без информације о поремећају
  - Отворени систем аутоматског управљања са компензацијом поремећаја – за формирање управљачког дејства се користи информација о жељеном излазу система и информација о поремећају
- Затворени системи аутоматског управљања (или систем аутоматског управљања са повратном спрегом или систем аутоматског регулисања) – за формирање управљачког дејства се користи информација о жељеном и стварном излазу објекта, без информације о поремећају. Управљање се формира на основу грешке излаза. То је индиректна компензација поремећаја. Грешка се може јавити и без поремећаја, од стране самог објекта, при свесним променама излаза објекта. Грешка увек настаје у објекту, јер ту долази до одступања стварног од жељеног понашања.
- Комбиновани системи аутоматског управљања – за формирање управљачког дејства користи се и информација о поремећају, као и информација о жељеном и стварном излазу, односно грешци. Адаптивни системи представљају посебну класу комбинованих система аутоматског управљања." [Jacic et al]

Према [Jacic et al], услови за увођење управљања неким објектом су:

- „Захтеви у погледу понашања објекта морају бити специфицирани
- Толеранције одступања понашања објекта од својих жељених вредности морају бити тачно назначене, а при могућим дозвољеним променама жељене величине и поремећаја
- Могућност настајања одступања из било ког разлога намеће потребу за управљањем објектом
- Све информације о објекту и поремећајима који делују морају бити доступне управљачком систему, како би се могло формирати одговарајуће управљање и све релевантне величине морају бити мерљиве
- Сваки објекат аутоматског управљања мора да поседује регулишући орган који је у стању да одговарајућу физичку величину мења у најширим границама, а све са циљем да се пренесу и таква управљачка дејства којима се могу компензовати и најкритичнија, дозвољена дејства поремећаја или екстремне промене жељене величине
- Усвојени концепт управљања објектом треба да обезбеди квалитетан рад система у целини, како у стационарном, тако и у прелазним режимима рада. Захтеви за остваривањем високе тачности и прецизности, безбедности, поузданости и рентабилности су категорије без којих савремено аутоматско управљање не може данас да се замисли.“

„Под компонентама управљачких система подразумевају се елементи који су неопходни при физичкој реализацији система аутоматског управљања или регулисања. Њиховим међусобним повезивањем остварује се одговарајућа задата законитост управљања објектом. Према својој намени, односно улози и функцији коју обављају, компоненте се могу поделити на следеће групе:

- Сензори (мерни елементи) – елементи који имају задатак да региструју постојање неке физичке величине или прате њено одступање у току времена и да ту информацију пошаљу управљачком систему, могу да мере интензитет сигнала и да врше претварање физичке величине. Захтеви: технички (висока осетљивост, брзина реаговања, линеарност, распон мерења, одсуство хистерезиса), функционални (поузданост, дуговечност, неосетљивост на спољне утицаје), технолошки (мали габарит и маса, једноставна конструкција, ниска цена).
- Елементи за постављање и меморисање величина
- Претварачи (трансмитери) – врше промену једне величине у другу, линеарно морају да буду усклађене, често у саставу сензора
- Појачавачи – појачавају интензитет сигнала, без промене природе, мора бити линеарна зависност излазне и почетне вредности
- Елементи за пренос и гранање сигнала
- Корекциони преносни органи – омогућава да управљачки орган постигне одговарајући карактер свог дејства,
- Извршни органи – имају задатак да остваре дејство управљачког система на објекат управљања, деловање може бити директно и индиректно„[Jacić et al]

Према [Jacić et al], за формирање управљачке величине у управљачки систем је неопходно увести информацију о жељеном понашању објекта, а такође и условима окружења и самом раду објекта над којим се спроводи управљање. „Начин достављања информација и сам скуп информација које унутар управљачког система конституишу одговарајућу управљачку величину, односно одговарајући закон управљања, назива се концепт управљања.

Према [Alleman 4.0], контрола представља усмеравање скупа променљивих према одређеном циљу. Контрола управљања („Management Control“) може бити предчињенично („before-the-fact“) или пост-чињенично („after-the-fact“). Постчињенична контрола се примењује као ефикаснија када су последице управљања лако могу мерити и надгледати.

#### **1.3.2.2.2. Адаптивни системи**

Управљањем се жељени циљ функционисања система постиже утицајем на функционисање појединих елемената. Саморегулишући системи поседују посебан вид управљања, којим се „интервенција“ усмерава на смањење одступања између задатог и стварног стања система. Свака регулација претпоставља затворену спрегу дејства – повратну спрегу. Уколико систем садржи елементе који управљају функционисањем осталих елемената, систем се назива саморегулишући [Šoti, 1978]. Повратна спрега је метод управљања понашањем неког система, уз коришћење резултата његовог ранијег деловања. Системи са негативном повратном спрегом – прилагођавају се штетном утицају или га избегавају, а системи са позитивном повратном спрегом под дејством штетног утицаја улазе у опасност и уништавају се. [Šoti, 1978] Према [Alleman 4.0], у теорији система контроле, историјски први су настали модели линеарне повратне спреге („linear feedback models“). С обзиром на широк спектар примене, савремени системи контроле укључују контролне системе са вишеструким улазима и излазима, адаптивне системе и системе оптимизационе контроле.

Код организованих система „међуодноси елемената у току функционисања бивају променљиви, зависно од тренутне ситуације“. [Šoti, 1978] „Код организационих система, задатак управљања је да формира одговарајућу организацију система и

осигура њено функционисање усмерено ка остваривању циљева система. Организациони системи под утицајем промена у окружењу, као и у њима самима, прилагођавају своје функционисање, а знатно ређе и своју структуру. Циљ адаптације организационих система је повећање ефикасности њиховог рада." [Stanojević, 1986]

„Системи који могу динамички да мењају своје унутрашње особине у зависности од измењених околности (окружења) називају се *адаптивни системи*. Овакви системи су способни да се прилагоде свом измењеном окружењу динамичким мењањем своје унутрашње структуре. У адаптивним системима се, поред интерне негативне повратне спреге, налази и адаптивна повратна спрега...“ [Šoti, 1978] која мења структуру и функционисање (алгоритам понашања) елемената система. [Šoti, 1978] Адаптивна повратна спрега интервенише на тај начин што се упоређују измењени улазни импулси са импулсима о стању регулисаног система и импулсима о његовим излазним величинама, те се на основу запажених одступања од задатих величина врши додатна – адаптивна регулација. Суштина адаптивне регулације је, дакле, у томе да се одређене особине система измене тако да се функционисање система може поново регулисати традиционалним методама, тј. интерном негативном повратном спрегом. Ова нова регулација се разликује од досадашње, јер су измењене раније спреге елемената и ранији алгоритам. [Šoti, 1978]

Према [Alleman 4.0], појам адаптивности је дефинисан као „билокоја промена структуре или функционисања организма како би се прилагодио у циљу преживљавања или мултиплицирања, односно промена у одговору на сензорске сигнале услед промене у условима окружења. У контексту адаптивности, користи се термин „индекс перформанси“ који представља квантитативну меру перформанси, односно одступања од идеалних перформанси. Адаптација указује на способност за само-подешавање или само-модификовање у складу са непредвидљивим променама у условима окружења или структуре. У адаптивном контролном систему, динамичке карактеристике морају бити идентификоване и праћене све време, тако да параметри контролера могу бити прилагођени како би се одржале оптималне перформансе.

Према [Alleman 4.0], систем адаптивне контроле је контролни систем у којем се реализују континуирана и аутоматска мерења динамичких карактеристика процеса, врши се компарација са жељеним динамичким карактеристикама и разлике се користе како би се подесили параметри система (најчешће карактеристике контролера) или генерисање побуђујућег сигнала тако да се одрже оптималне перформансе система, независно од промена у окружењу процеса. Да би се назвао адаптивним, нека форма самоорганизујућих особина мора да постоји. Адаптивни контролер састоји се од следећих три функција:

1. Идентификација динамичких карактеристика процеса
2. Одлучивање базирано на идентификацији процеса (компарацијом са оптималним карактеристикама или оптималним перформансама)
3. Модификација или активирање базирано на донетим одлукама:
  - А) модификација параметара контролера
  - Б) синтеза контролног сигнала базирано на функцији трансфера, индексу перформанси и жељеном пролазном одговору процеса.

Применом наведених функција континуирано, самоорганизација може да компензује непредвиђене промене у процесу.

Комплексни адаптивни систем [Pathak et al, 2007] представља мрежу међусобно повезаних елемената (или агената) који исказују адаптивно понашање у односу на промене у окружењу или у односу на карактеристике самог система. Перформансе система не зависе линеарно од перформанси појединачних елемената, већ динамички од паралелних активности ентитета. Ентитети међусобно комуницирају у складу са одговарајућим скупом правила. Елементи се могу мењати, еволуирати у току времена, учећи на основу интеракције са окружењем. CAS системи исказују особине самоорганизације, која настаје као последица интеракције између ентитета система. Само-

организација је дефинисана као процес у којем нове структуре, обрасци и особине „израњају“ (emerge) без да су екстерно изложени систему.

### 1.3.2.2.3. Адаптивни и адаптивбилни софтвер и информациони системи

Према [David&Ledoux, 2006], адаптација у општем смислу представља модификацију која је покренута од стране измењених околности, помоћу који систем почиње да буде боље усклађен са новим окружењем. У случају софтвера, адаптација је имплементирана када је програм одговоран за: 1) „посматрање“ окружења у којем се циљни софтвер извршава како би се детектовали нови услови, 2) одлучивање о одговарајућим модификацијама које треба да се примена на циљни софтвер, 3) примену тих модификација, прилагођавањем циљног софтвера новим условима. Савремени самоадаптивни софтвери треба да имају могућност да се адаптирају самостално.

Према [Zhang&Cheng, 2006] [McKinley et al, 2004] под појмом адаптивни софтвер подразумева се софтвер који динамички прилагођава понашање у извршном режиму („run-time“) као одговор на промене услова у окружењу рачунарско-комуникационе инфраструктуре и физичког окружења. У раду [Zhang&Cheng, 2006] представљена је анализа постојећих истраживања и решења у креирању и моделовању адаптивних софтвера и класификација адаптираних система у односу на структурне адаптације и адаптације у понашању.

У контексту адаптивности као промене понашања, (обичан) програм се може представити као машина стања (аутомат са коначним скупом стања) који приказује одређено понашање и функционише у одређеном домену (тј. улазном простору). Динамички адаптивни програм [Zhang&Cheng, 2006] [Allen et al, 1998] функционише у различитим доменима, мења своје понашање у току извршавања као одговор на промене домена. Простор стања адаптивног програма може бити подељен у одређени број међусобно независних региона (програма), где сваки приказује различито понашање стабилног стања и функционише у различитом домену. Стања и транзиције које повезују један регион са другим називају се адаптациони скупови. У раду [Zhang&Cheng, 2006] описан је приступ адаптивним програмима такав да:

1. Разликује неадаптивно од адаптивног понашања програма, омогућујући лакшу проверу од стране људи и аутоматизацију провере,
2. Дефинише глобалне карактеристике програма које треба да су непроменљиве иако су друге карактеристике променљиве у току адаптације
3. Адаптација програма се своди на примене стања програма, из почетног стања (изворни програм - „source program“) у циљно стање (циљни програм - „target program“, тј. Програм након адаптације).
4. Дефинише „стања мировања“ („quiescent states“) адаптивног програма. Стање мировања изворног програма је такво стање изворног програма које када се трансформише у стање циљног програма не нарушава вредности глобалних непроменљивих карактеристика.

Према [Zhang&Cheng, 2005] [Zhang&Cheng, 2006], најчешћи облици адаптивног понашања у програмима су: адаптација једне тачке („one-point adaptation“), вођена адаптација („guided adaptation“), преклапајућа адаптација („overlap adaptation“).

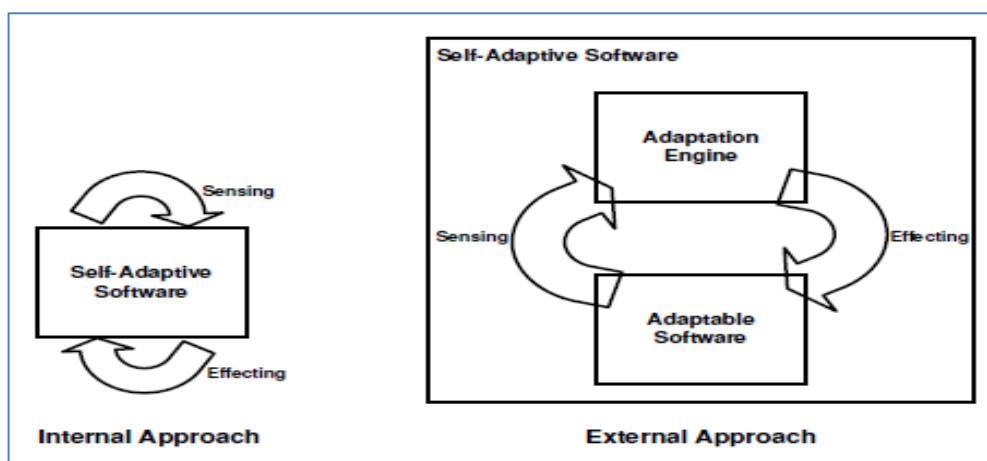
1. Адаптација једне тачке односи се на ситуацију када се понашање почетног програма (изворног програма) мора завршити, а тек потом циљно понашање (након адаптације) може да почне. Процес адаптације се завршава након једне транзиције. Главни задатак ове врсте адаптације је дефинисати стања која су погодна за адаптацију и дефинисати адаптивне транзиције из наведених стања.
2. Вођена адаптација односи се на ситуацију када изворни (почетни) програм добија захтев за адаптацијом, улази у ограничени („restricted“) режим рада када неке функционалности програма су најчешће блокиране. Тада програм улази у

„стање мировања“, из којег адаптација једне тачке преводи програм у простор стања циљног програма.

3. Преклапајућа адаптација се односи на то да се почетно (изворно) и циљно понашање преклапају. У току извршавања изворног стања почиње циљно стање и у једном преклапајућем интервају оба стања функционишу истовремено, односно програм показује понашање оба стања. Преклапајућа адаптација је типична за програме са вишеструким нитима („multi-threaded programs“), где свака нит за себе реализује адаптацију једне тачке или вођену адаптацију.

У раду [Salehie&Tahvildari, 2009] појам само-адаптивног софтвера дефинисан је као систем затворене повратне спреге где се повратна информација добија из самог система („self“ – целина софтвера, најчешће организована кроз слојеве архитектуре) и окружења („context“ – све што се налази у оперативном окружењу што утиче на особине или понашање система). Динамичке промене у току извршавања софтвера („run-time“) представљају основу адаптације у самоадаптивном софтверу. „Само-адаптивни софтвер врши евалуацију своје понашање и мења понашање када евалуација индицира да не реализује оно за што је намењен или када су боља функционалност или перформансе могуће.“ [Zhang&Cheng, 2006] [Laddaga, 1997] „Само-адаптивни софтвер модификује своје понашање као одговор на промене у оперативном окружењу, под чиме се подразумева било шта што је видљиво од стране софтверског система, као што је улаз корисникових података, екстерни рачунарски уређаји или сензори или програмска инструментација.“ [Oreizy et al, 1999] Према [Zhang&Cheng, 2006] само-адаптивни софтвер припада широј категорији аутономног рачунарства, где софтверски-интензивни системи обухватају апликације, средњи апликациони слој („middleware“), рачунарске мреже, оперативни систем и хардвер [McKinley et al, 2004] и у свим компонентама може бити примењени елементи аутономног рачунарства. Према [Zhang&Cheng, 2006], само-адаптивни софтвер припада средњем апликативном слоју. Основна премиса само-адаптивног софтвера је његов животни циклус не престаје инсталацијом и подешавањем за коришење, већ треба да је стално отворена могућност промена посебно у домену промене корисникових захтева, конфигурације, безбедности итд.

Слика 1.3.2.2.3.1. приказује две врсте само-адаптивних система – интерни приступ са интегралним активностима сензора и ефектора или екстерни приступ, где је језгро адаптације посебан подсистем у односу на софтвер који је *адаптибилан*. Адаптибилни (под)систем реализује основне функције, а њиме у смислу адаптивности управља посебни модул за адаптацију.

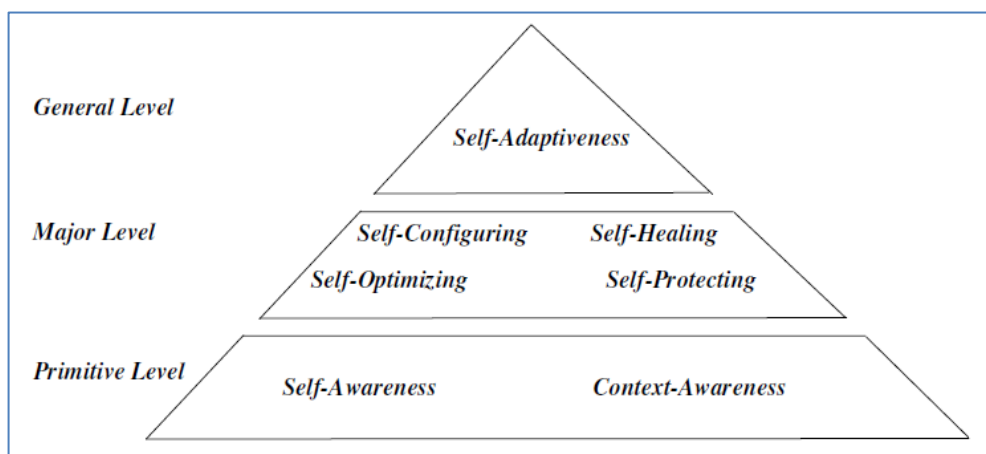


Слика 1.3.2.2.3.1. Два приступа архитектури само-адаптивних система, према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

Слика 1.3.2.2.3.2. Приказује различите аспекте интерних карактеристика самоадаптивности у оквиру самоадаптивних система. У раду [Salehie&Tahvildari, 2009]



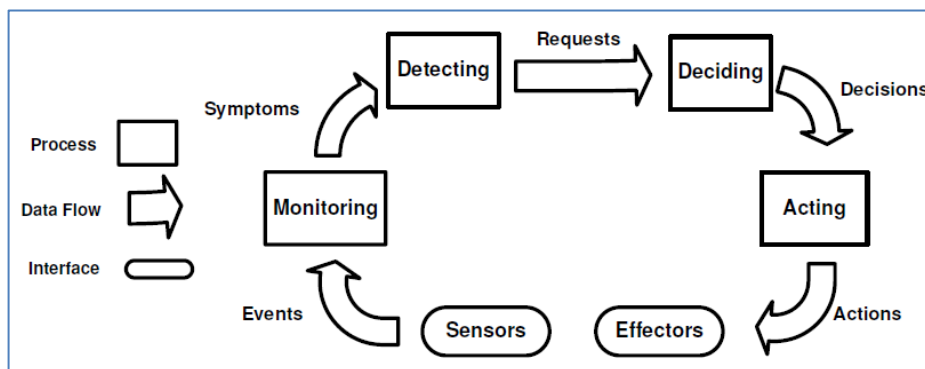
неке од карактеристика квалитета софтвера према стандарду ISO 9126 упоређене су у контексту подршке у самоадаптивним софтверским системима.



Слика 1.3.2.2.3.2. Хијерархијска уређеност интерних карактеристика само-адаптивног софтвера, према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

- "Самоконфигурација („Self-configuring“) је способност реконфигурације аутоматски и динамички као одговор на промене помоћу инсталирања, унапређења („updating“), интеграције и композиције/декомпозиције софтверских ентитета.
- Само-излечење („Self-healing“) повезано је са само-дијагнозом и само-поправком је способност проналажења, дијагностиковања и реаговања на прекиде. Такође може да предвиди потенцијалне проблеме и у складу са тиме предузме одговарајуће акције како би спречио неуспех. Само-дијагностика се односи на дијагностицирање грешака и недостатака, док се само-поправка фокусира на опоравак од грешака.
- Само-оптимизација – назива се и само-подешавање је способност управљања перформансама и алокацијом ресурса у циљу задовољења захтева различитих корисника. Праћење времена одговора система, пропусности, коришћења и оптерећења су важни аспекти.
- Само-заштита је способност детектовања безбедоносних претњи и опоравак од њихових ефеката кроз одбрану од малициозних напада и предвиђање проблема и предузимање акција како би се избегли могући проблеми.
- Само-свесност је особина система да је свестан својих стања и понашања. Ова особина је базирана на само-мониторингу.
- Свесност контекста значи да је систем свестан контекста, односно оперативног окружења.“ [Salehie&Tahvildari, 2009]

На слици 1.3.2.2.3.3. Приказан је процес самоадаптације софтвера кроз четири кључна подпроцеса.



Слика 1.3.2.2.3.3. Процес самоадаптације софтвера кроз четири кључна подпроцеса према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

- „Процес мониторинга је одговоран за прикупљање и корекцију података од сензора и конвертовање у шаблоне понашања и симптоме. Овај процес може бити реализован кроз корелацију догађаја, проверу прага, као и друге методе.
- Процес детектовања је одговоран за анализирање симптома обезбеђен мониторинг процесом и историјом система и циљу да детектује када је промена (одговор) потребна. Помаже у идентификовању где је извор транзиције у ново стање (девијација од жељених стања или циљева).
- Процес одлучивања одређује шта је потребно променити и како да промена постигне најбоље резултате. Заснива се на одређеним критеријумима који служе за компарацију различитих начина примене промене, односно различитих могућих путева за акцију.
- Процес деловања („acting process“) одговоран је за примену акција које су утврђене од стране процеса одлучивања. То укључује управљање не-примитивним акцијама кроз предефинисане радне токове или мапирање акција које су обезбеђене ефекторима и њиховим техникама динамичке адаптације..“ [Salehie&Tahvildari, 2009]

У контексту адаптивних web базираних информационих система [Houben, 2005], један од аспеката адаптивности је адаптивност у односу на корисника. У контексту моделовања корисника систем треба да се прилагоди профилу корисника (који се формира на основу знања, циљева, интересовања, историје ранијег коришћења итд). Разликујемо:

1. *Адаптибилност система* – прилагођавање статичком профилу (припадности групи) корисника или уређају, које се дешава на почетку коришћења
2. *Адаптивност система* – вишеструка прилагођавања „динамичком профилу корисника“, у току његовог коришћења система. Корисник у току коришћења система мења карактеристике и оне морају бити стално праћене како би се систем динамички прилагођавао променама корисника.

У раду [DAISY] описана је разлика између адаптивних и адаптибилних система у софтверском смислу. Адаптација у информационим системима односи се на људски и рачунарски, односно софтверски део. У рачунарско-софтверском делу, разликујемо:

- Аутоматску адаптацију – *адаптивни* софтвер
- Полуаутоматску адаптацију – *адаптибилни* софтвер
- Неаутоматску адаптацију – адаптивно одржавање софтвера од стране човека

### 1.3.2.3. Системи за управљање пословним правилима, аутоматско резонување и онтологије

Архитектура савремених пословних апликација орјентисана је на издвајање података, пословних правила и пословних процеса у посебне модуле, при чему тим слојевима управљају посебни системи:

- Подацима управљају системи за управљање базама података
- Пословним правилима управљају системи за управљање пословним правилима
- Пословним процесима управљају системи за управљање пословним процесима. [Naab, 2012]

Систем за управљање пословним правилима представља „софтверски систем који се користи за дефинисање, постављање, извршавање, мониторинг и одржавање разноврсну и комплексну логику доношења одлука које се користе од стране операционих система у оквиру организација или предузећа.“ [Thirumaran, 2010] „Пословно правило представља тврђење која дефинише или ограничава неки аспект пословања. Намена му је да утиче на структуру пословања или да контролише понашање пословања. Пословна правила су најчешће приказана као ограничења или у форми „ако услови онда акција. Услови се такође називају премисе правила. Приступ пословним правилима укључује скуп терма (дефиниција), чињеница (везе између терма) и правила (израчувања, ограничења и условна логика). Терми и чињенице су тврђења која садрже релевантне опсервације пословања, док правила представљају тврђења која се користе за откривање нових информација или да усмерава доношење одлука“. [Thirumaran, 2010]

У основи имплементације система за управљање пословним правилима најчешће су системи аутоматског резонувања. „Системи аутоматског резонувања су рачунарски програми који поседују одређене компоненте интелигентног понашања и који се могу примењивати као љуске експертних система, дијалогски системи, преводиоци природних језика, образовни рачунарски софтвери, информациони системи и др. Извођење закључка у оваквим системима најчешће се базира на резолуцијској методи оповргавања, тј. негирању тврђења које се доказује у систему за аутоматско доказивање теорема (АДТ) и извођењу закључка из одређеног скупа правила и чињеница. Првобитна област примене ових програма је била искључиво математика, али се временом увидело да се могу користити за решавање одређених проблема и у другим областима“. [Kazi, 2014]

У оквиру система за управљање пословним правилима често се у савременим истраживањима и техничким реализацијама, јавља потреба за представљањем знања о доменској области над којом се врши процесирање. „Онтологије се дефинишу на различите начине у зависности од научне области у којој се користе. У рачунарском инжењерству онтологија се односи на представљање знања. У области информационих система постоји проблем постојања различитих класа података које имају своје термине којима се представљају информације. Када се такве информације налазе на једном месту заједно, термилошке и концептуалне различитости се могу превазићи употребом одговарајућих онтологија. Онтологија се дефинише као скуп термина који се користе да би се описао домен, тј. област знања. Онтологије користе људи, базе података и софтверске апликације које деле информације из одређеног домена. Домен представља специфичну предметну област знања. Најпопуларнију дефиницију онтологије је дао Грубер [Gruber, 1995] као „формалну, експлицитну спецификацију дељене концептуализације. Појам формална се односи на чињеницу да онтологија треба да је разумљива, тј. читљива за машине. Експлицитна значи да су експлицитно дефинисани сви концепти и ограничења која се користе. Дељена указује на то да онтологија треба да представи знање прихваћено од стране заједнице у којој се уводи и користи. Концептуализација се односи на апстрактан модел феномена из стварног света који се добија идентификовањем релевантних концепата ових феномена [El-Ghalayini et al., 2005].“ [Kazi, 2014]

### 1.3.2.4. Визуализација података

„Визуализација је графичко представљање информација, са циљем обезбеђивања квалитативног разумевања садржаја информације од стране корисника. То је такође и процес трансформисања објеката, концепата и бројева у форму која је видљива човековом оку. Када кажемо „информација“, то се може односити на податке, процесе, релације или концепте. Овде се фокусирамо на податке. Визуализација података се односи на разумевање односа и релација између бројева. Не односи се на разумевање појединачних бројева, него образаца, трендова и релација које постоје међу групама бројева [Kamran&Chignell, 1993]. Визуализација података помера фокус са нумеричког резонувања на визуално резонување. Добијање информације са слика много више штеди време од посматрања текста и бројева. То је разлог зашто многи људи који доносе одлуке би радије да имају информације презентоване у графичкој форми, насупрот текстуалне форме.“ [Kaidi, 2000] Визуализација података се разликује од научне визуализације, која „користи анимацију, симулацију и софистицирану компјутерску графику да креира визуалне моделе структура и процеса који се на други начин не могу видети или се не могу видети у довољно детаљном приказу. Са друге стране, визуализација података је начин презентовања информација такав да подстиче одговарајућу интерпретацију, селекцију и асоцијацију. Он користи људске способности препознавања образаца и анализе трендова, користи способност људи да издвајају велике количине података у кратком временском периоду из визуалног представљања у стандардизованом формату.“ [Kaidi, 2000]

„У оквиру визуализације података, неки од коришћених графичких ентитета и атрибута могу бити [Kamran&Chignell, 1993]:

1. Ентитети: тачка, линија, закривљена линија, вишеструка линија (polyline), површина, раван (solid), слика, текст
2. Атрибути: боја, интензитет, локација, стил, величина, релативна позиција, покрет“ [Kaidi, 2000]

„Под подацима подразумевамо записе са различитим специјалним карактерима и често могу бити подељени у неколико група, као што су [Ward, 2009]:

- Нумерички, симболички или мешовити
- Скалари, вектори или комплексне структуре
- Различитих мерних јединица – метри, инчи
- Дискретни, континуирани, симболички
- Просторни, квантитативни, категоријски, временски, релациони, структурни
- Прецизни или приближни
- Уређени или неуређени
- Међусобно искључујући или преклапајући
- Бинарни, енумерације, вишенивовски
- Независни или зависни
- Вишедимензионални, итд.“ [Kaidi, 2000]

„Сматрамо да су подаци правилно визуализовани ако је визуализација [Ward, 2009]:

- Ефективна (Effective): корисник може лако да је интерпретира (разуме).
- Прецизна (Accurate): довољна за коректну квантитативну евалуацију.
- Ефикасна (Efficient): минимизира неискоришћени простор графикона, а максимизира коришћење података на графикону (minimize data-ink ratio and chart-junk, show data, maximize data-ink ratio), минимизује површине без података и редундантне податке( brase non-data- ink, brase redundant data-ink)
- Естетична (Aesthetics): не сме угрожавати-вређати корисничкова чула
- Прилагодљива: може се прилагодити да може да се користи различитим потребама“ [Kaidi, 2000]

„Неки од најчешћих начина представљања визуализације података су:

- Графикони (Charts): стубични или пите
- Графови (Graphs): добри за приказ структуре и релација

- Цртежи (Plots): 1- до n-димензионални
- Мапе (Maps): једне од најефективнијих
- Сlike (Images): коришћење боја/ интензитета уместо удаљености(површина)
- 3Д површине и равни (3-D surfaces and solids)
- Изоповршине (Isosurfaces/slices)" [Kaidi, 2000]

Уобичајени редослед поступака у процесу визуализације података подразумевају [Kamran&Chignell, 1993] низ корака, као што је приказано на слици 1.3.2.4.1.:



Слика 1.3.2.4.1. Редослед поступака визуализације података, према Kaidi [Kaidi, 2000]

#### “НУМЕРИЧКА ТРАНСФОРМАЦИЈА

Визуализација је облик трансформације нумеричких података. Бројеви су апстрактни концепти и да бисмо их представили као тачке и линије то захтева трансформацију. Трансформација укључује:

1. Промену дистрибуције: измена дистрибуције бројева како би били погоднији за анализу или визуално представљање. Неки фреквентни начини који се користе укључују: линеарну трансформацију, логаритамску трансформацију, трансформацију нормализације, Синусну (Arcsin) трансформацију, трансформацију квадратног корена, инверзну трансформацију.
2. Редифинисање значења: прилагођавање бројева како би имали више значења или више репрезентативни у односу на концепт за који је аналитичар података заинтересован.
3. Креирање нових агрегираних значења

#### АНАЛИЗА ПОДАТАКА

Анализа података је процес примене различитих метода над подацима како би помогле интерпретацији. Неке од методе анализе података које се користе су: статистичке методе, анализа кластера, мултидимензионално склалирање и факторска анализа. Анализа података може бити коришћена за трансформацију података или за сумирање самих података или њихових статистичких особина.

#### ГРАФИЧКА ИНТЕРПРЕТАЦИЈА

Графичка интерпретација се састоји од неколико кључних активности као што је просуђивање (judgement) о величини (или релативној величини), просуђивање о

пропорцији (или релативној пропорцији), просуђивање о тренду и груписању. Може користити начине као што су:

- Скалирање и offset како би се ускладио са опсегом
- Коришћење изведених вредности (residuals, logs) да би се нагласиле промене
- Коришћење пројекција, других комбинација, како би се извршила компресија информација и добила статистика
- Издвајање преклапања
- Коришћење вишеструких погледа да би се обрадиле скривене релације, више димензије
- Коришћење ефективних табела (grids), кључева и ознака да би се помогло разумевање

#### КОРИСНИЧКА ИНТЕРАКЦИЈА

Када су представљени резултати визуализације, корисник можда утврди да тај приказ не одговара његовом начину размишљања. Из тог разлога, корисници могу желети да ураде неке операције, које ће можда захтевати да се поново другачије реализује неки од претходних корака:

- Динамичко прилагођавање мапирања
- Померање података изменом погледа
- Означавање да би се добили оригинални подаци
- Брисање ради елиминисања clutter-а
- Чишћење (Brushing)/Наглашавање (Highlighting) како би се видела повезаност у вишеструким погледима
- Зумирање (Zooming) како би се фокусирао пажња
- "Panning" како би се истражило окружење." [Kaidi, 2000]

### 1.3.3. ЕЛЕМЕНТИ УПРАВЉАЊА СОФТВЕРСКИМ ПРОЈЕКТИМА

#### 1.3.3.1. Категоризација софтвера и софтверских пројеката

##### 1.3.3.1.1. Категоризација софтвера

Међународне организације, међу којима је и организација Уједињених Нација, реализују класификацију производа и услуга која би представљала међународни стандард и омогућила интеграцију у пословању различитих земаља. Тако је реализован у оквиру Уједињених Нација документ UNSPSC (United Nations Standard Products and Services Code) [UNSPSC], где се сви производи и услуге категоришу користећи 4 нивоа: „Segment, Family, Class and Commodity“. Софтвер према наведеној класификацији [UNSPSC-cc] [UNSPSC-sw] категорише се у односу на намену као:

- 11- 43230000 (Software):
  - 432315 - BUSINESS FUNCTION SPECIFIC SOFTWARE
  - 432316 - FINANCE ACCTING, ENTERPRISE RESOURCE PLANNING ERP SOFTWARE
  - 432320 -COMPUTER GAME OR ENTERTAINMENT SOFTWARE
  - 432321-CONTENT AUTHORIZING AND EDITING SOFTWARE
  - 432322-CONTENT MANAGEMENT SOFTWARE
  - 432323-DATA MANAGEMENT AND QUERY SOFTWARE
  - 432324-DEVELOPMENT SOFTWARE
  - 432325-EDUCATIONAL OR REFERENCE SOFTWARE
  - 432326-INDUSTRY SPECIFIC SOFTWARE
  - 432327-NETWORK APPLICATIONS SOFTWARE
  - 432328-NETWORK MANAGEMENT SOFTWARE
  - 432329-NETWORKING SOFTWARE
  - 432330-OPERATING ENVIRONMENT SOFTWARE
  - 432332-SECURITY AND PROTECTION SOFTWARE
  - 432334-UTILITY AND DEVICE DRIVER SOFTWARE
  - 432335-INFORMATION EXCHANGE SOFTWARE
  - 432336-ELECTRICAL EQUIPMENT SOFTWARE
  - 432337-SYSTEM MANAGEMENT SOFTWARE
- 11- 81160000 (Information Technology Service Delivery):
  - 811615 - ACCESS MANAGEMENT SERVICES
  - 811616 -ELECTRONIC MAIL AND MESSAGING SERVICES
  - 811617- TELECOMMUNICATION SERVICES

У истраживањима у области развоја софтвера, различити аутори дају класификације у складу са садржајем одговарајућих истраживања:

I) Рачунарски софтвер се може поделити према намени на [Turban et al, 2013]:

1. Системски софтвер (нпр. оперативни системи,услужни софтвер, драјвери уређаја, антивирусни програми итд.)

2. Апликативни софтвер различите намене:

- апликативни софтвер опште намене (софтвер за подршку канцеларијском пословању – word процесори, рад са презентацијама, рад са табелама...)
- апликативни софтвер специфичне намене (пословне апликације, индустријски софтвер итд.)

3. Софтвер за развој апликација (алати за програмирање, тестирање софтвера итд.)

II) Рачунарски софтвер се може поделити према корисницима на [Turban et al, 2013]:

- Рачунарски софтвер опште намене
- Специфичан рачунарски софтвер за одређену категорију корисника
- Специфичан рачунарски софтвер који се реализује за конкретног корисника

III) Рачунарски софтвер се може поделити према типу корисника на [Nguyen, 2006]:

- Комерцијални софтвер (за општу намену, за шири круг корисника)
- Софтвер по наруџби (за конкретног корисника)

##### 1.3.3.1.2. Софтверски пројекти

Управљање софтверским пројектима припада областима развоја софтвера и управљања пројектима [Berntsson-Svensson&Aurum, 2006]. Основне фазе и активности у управљању софтверским пројектима зависе од усвојене методологије развоја софтвера, као и методологије управљања пројектима, али према [Berntsson-Svensson&Aurum,

2006] основне фазе управљања софтверским пројектима се у суштини могу представити следећом табелом:

Табела 1.3.3.1.2.1. Фазе и активности у управљању софтверским пројектима, према Berntsson-Svensson&Aurum ([Berntsson-Svensson&Aurum, 2006])

Концепт пројекта	Планирање	Извршавање	Завршавање
Идентификација потреба	Припрема планова	Извршавање посла	Пренос одговорности
Успостављање циљева	Припрема плана буџета	Набавка материјала	Ослобађање ресурса
Утврђивање изводљивости	Припрема временског плана –распореда	Изградња и тест	Трансфер чланова тима
Припрема предлога	Окупљање пројектног тима	Верификација перформанси	Награђивање људи
Процена времена и ресурса (грубо)	Изградња и тестирање прототипова	Модификација по потреби	Спровођење ревизије
Идентификација кључних људи	Добијање сагласности за следећу фазу		
Добијање сагласности			

### 1.3.3.1.3. Категоризација софтверских пројеката

Према [Shenhar et al, 2000] [Shenhar et al, 2001], пројекти се могу класификовати према *нивоу значаја и нивоу управљања* на:

- Операционално управљане пројекте („operationally managed projects“) – фокусирани су на испуњење краткорочних циљева: извршавање посла и испоруку резултата у оквиру планираног времена и буџета
- Стратегијски управљане пројекте („strategically managed projects“) – фокусирани су на постизање пословних резултата, задовољавањем потреба корисника, компетитивне предности и будућег успеха у освајању тржишта, дугорочне циљеве за добробит пословања

Према [Chow&Дас-Вуу, 2008] софтверски пројекти се могу класификовати на „животно-критичне“ и „животно-не критичне“ у смислу утицаја примене софтвера на ризик за животну опасност човека који би био обухваћен системом који га користи.

Пројекти, па тако и софтверски пројекти, могу се категорисати и на основу *нивоа познавања технологије* у тренутку иницијације пројекта („level of technological uncertainty at the moment of project initiation.“), која представља основ реализације [Shenhar&Dvir, 1996] [Shenhar et al, 2001] [Dvir et al, 1998]:

- Пројекти ниског нивоа технологије („Low-Tech Projects“) базирају се на постојећој и добро познатој технологији, где се најчешће реализује понављање изградње истог производа
- Пројекти средњег нивоа технологије („Medium-Tech Projects“) ослањају се највише на постојеће базичне технологије, али укључују и нове технологије или особине. То су пројекти инкременталне иновације као и унапређења и модификације постојећих производа
- Пројекти високе технологије („High-Tech Projects“) су дефинисани као пројекти у којима је већина коришћене технологије нова, али постојећа, развијена непосредно пре покретања овог пројекта. То су пројекти развоја нових компјутерских система или одбрамбених система.
- Пројекти супер високе технологије („Super High-Tech Projects“) базирани су примарно на новим, до тада непостојећим технологијама, које морају бити развијене у току извршавања пројекта. Овакви пројекти су ретки и изводе се од стране неколико најчешће великих организација или владиних агенција.

Категоризација софтверских пројеката може се вршити у односу на више критеријума [Berntsson-Svensson&Aurum, 2006]:

1. Према *типу активности*:



- Софтверски пројекат развоја новог софтвера
  - Софтверски пројекат одржавања постојећег софтвера
  - Софтверски пројекат унапређења постојећег софтвера.
2. У односу на однос клијента и добављача софтвера могу категорисати као:
- Клијент орјентисани пројекти развоја софтвера („Bespoke software development“) – развија се специфични „custom-made“ производ или услуга за једног или неколико клијената
  - Тржишно-орјентисан развој софтвера – добављач развија производ за специфично тржиште без дефинисања клијента унапред
  - Комбиновани клијентски и тржишно орјентисани развој софтвера – добављач развија производ за специфично тржиште, али са намером да је могуће да се прода производ клијенту са специфичним подешавањима.
  - Интерни развој софтвера – добављач је истовремено и клијент. Систем се развија за своју компанију или организацију.

Према [Nassif, 2012] [Boehm, 1981], софтверски пројекти се могу категорисати у односу на *искуство учесника тима и променљивост захтева* као:

- Органски (Organic) – пројекти где мањи тим са добрим искуством ради на пројекту са флексибилним захтевима (non-strict requirements)
- Полу-зависни (Semi-detached) – пројекти где тим средње величине са мешовитим искуством ради на пројекту са помешаним строгим и флексибилним захтевима
- Угњеждени (Embedded) – пројекти са строгим ограничењима (tight constraints).

На основу [Reifer, 2000], софтверски пројекти се могу категорисати према *технологији реализације* на:

- Традиционалне софтверске пројекте
- Софтверске пројекте развоја Web апликација.

Наравно, савремена технолошка решења укључују и друге категорије софтверских пројеката, нпр. пројекти развоја мобилних апликација итд.

Развој софтвера често припада широј категорији ИТ (Информационе технологије) пројекта. Посебној категорији ИТ пројеката припадају пројекти у области развоја информационих система. Према [Cadle&Yeates, 2008], пројекти у области информационих система се могу поделити у 9 категорија:

- Развој софтвера (Software development) – развој новог софтверског производа, у складу са захтевима корисника
- Имплементирање готовог софтверског пакета (Package implementation) – инсталација и подешавање готових софтверских решења специфичним потребама конкретног клијента
- Унапређење постојећег система (System enhancement) – додавање нових функција, усклађивање са променама у окружењу, првенствено са променама прописа
- Консултантске услуге анализе пословног проблема и истраживању и предлогу бољег решења (Consultancy and business analysis assignments)
- Миграција система (Systems migration) – превођење на нови систем (који је исти али треба да функционише у оквиру другог технолошког окружења или се мења из наведеног разлога) уз минимални утицај на радне активности које користе систем
- Промена инфраструктуре (Infrastructure implementation) – увођење нових или замена постојећих хардверских и мрежних уређаја
- Outsourcing и-или in-sourcing пројекти – ангажовање других фирми у реализацији појединих ИТ послова
- Пројекти опоравка система (Disaster recovery) – уколико због временских непогода, вируса или других разлога дође до оштећења опреме или софтвера, реализују се пројекти опоравка система; наравно, било би боље превентивно реализовати активности како би се спречиле последице оваквих ситуација
- Мали пројекти информационог система (Smaller IS projects) – мања унапређења система ипак захтевају примену техника управљања пројектима, са документацијом која је због тога што је мањи пројекат, адекватно умањена.

### 1.3.3.2. Дистрибуирани развој софтвера

Под термином дистрибуираног развоја софтвера подразумева се развој софтвера који реализују тимови који су дистрибуирани, односно најчешће географски удаљени. "Дистрибуирани развој софтвера (ДСД) омогућава члановима тима да буду смештени на различитим удаљеним местима у току животног циклуса софтвера и на тај начин формирају мрежу виртуалних тимова који раде на истим пројектима. Ти тимови могу бити чланови исте организације или може бити потребна сарадња или outsourcing од стране других организација. Иако се овај феномен појавио у току 90-тих година, у току претходне деценије препозната је његова стратешка важност [Davison, 2007] и студије које се односе на овај феномен су сасвим недавне [Prikladnicki et al, 2008]." [Jiménez et al, 2010]

Као супротност дистрибуираном развоју јавља се колокацијски (collocated) развој, где се сви учесници у развоју налазе на истој локацији уз могућност непосредне личне комуникације [Kosaguneli et al, 2013]. Дистрибуирани развој најчешће подразумева коришћење електронских технологија за комуникацију, првенствено Internet-а.

Таксономија дистрибуције презентована је у [Gumm, 2006] кроз:

- Објекте дистрибуције (шта је дистрибуирано – људи, артефакти, задаци, итд.)
- Типови дистрибуције (како је дистрибуција организована, димензије дистрибуције):
  - Физичка или географска дистрибуција (названа и "глобални развој софтвера" (ГСД) [Herbsleb&Moitra, 2001])
  - Организациона дистрибуција,
  - Временска дистрибуција,
  - Дистрибуција између група заинтересованих страна.
- Изазови дистрибуције (проблеми који треба да буду решени, као што је контрола верзија, комуникација, културне разлике, „доживљај“ удаљености итд.)
- Решења изазова (како су проблеми решени, нпр. друштвена решења, техничка решења итд.).

Кључни елемент дистрибуираног развоја је сарадња тимова која може бити класификована као колаборација, кооперација итд. „Под термином координација, Herbsleb [Herbsleb, 2007] подразумева управљање зависностима између задатака“. Термини колаборација и кооперација су „најчешће коришћени као синоними за описивање дистрибуираног процесирања заједничких софтверских артефаката базирано на подели радне снаге. У оквиру тога, неколико аутора разликује више нивоа колаборације. Она се састоји из (а) информисања, (б) координације, (с) сарадње и (д) кооперације. Супротно од колаборације, кооперација у овој таксономији је спроведена од стране учесника са једнаким правима и појединачних циљева који су подређени једном колективном циљу. Слично колаборацији, кооперација такође указује на константну и интензивну интеракцију. Штавише, особине кооперације односе се на заједнички процес и заједничке излазе као што су заједнички креирани документи као и колективна евалуација тима и резултата.“ [Hildenbrand et al, 2008]

Неки типови организације у дистрибуираном развоју софтвера су:

- Виртуални тимови – "састоје се из географски дистрибуираних сарадника повезаних путем информационих технологија како би реализовале организациони задатак" [Zhang et al, 2008]. "Виртуални тим се може описати као суштинска јединица која изграђује виртуалну организацију. Традиционални тим је дефинисан као друштвена група појединача који се налазе на истој локацији и међусобно су повезани задацима, група узима и координира њихове активности како би се остварили заледнички циљеви и делила орговорност за резултате. Виртуални тимови имају исте циљеве и задатке као и традиционални тимови и интерагују кроз међусобно зависне задатке, али функционишу преко географских, временских и организационих граница. Често функционишу у мултикултуралном и мултијезичком

окружењу; комуникација између чланова виртуалних тимова је уобичајено електронска и често асинхрона." [Noll et al, 2010]

- *IT Outsourcing* – " је акт делегирања и преношења неких или свих права одлучивања, пословних процеса, интерних активности и услуга, која се односе на ИТ, екстерним обезбеђивачима, који развијају, управљају и администрирају своје активности у складу са договореним резултатима који треба да буду испоручени, у складу са стандардима перформанси и излаза, као што је дефинисано у уговору" [Dhar& Balakrishnan, 2006]. Велике ИТ компаније [Kommeren& Parviainen, 2007], као и мала и средња предузећа [Boden et al, 2007] укључују дистрибуирани развој софтвера ангажујући outsourcing тимове из других земаља.

- *Open Source развој софтвера (ОССД)* – са неформалношћу комуникације и кооперације, где појединци и групе доприносе унапређењу заледничког производа унапређивањем појединачних особина. Они који доприносе унапређењу решења су истовремено и корисници и развијају га; њихове сугестије за унапређење нису формално записане као захтеви. [Mockus& Herbsleb, 2002]

Када су у питању ДСД пројекти, "рани резултати [Spinellis, 2006] упозорили су да такви пројекти могу да пате од нижег квалитета због географске дисперзије" [Kosaguneli et al, 2013]. Истраживање [Kosaguneli et al, 2013] је извршило проверу тврђења да ДСД утиче на квалитет софтвера и доказало је да ДСД приступ је одговарајући за коришћење у индустрији.

Постоје многе "познате" предности ДСД, међу којима су [Ågerfalk et al, 2008]:

- Уштеда трошкова,
- Приступ великом броју расположиве радне снаге која је обучена и има одговарајућа знања и способности,
- Смањено време производње и бржи излаз производа на тржиште (користећи различите временске зоне и радно време [Herbsleb&Moitra, 2001],
- Близина тржишту и купцима, укључујући локална тржишта

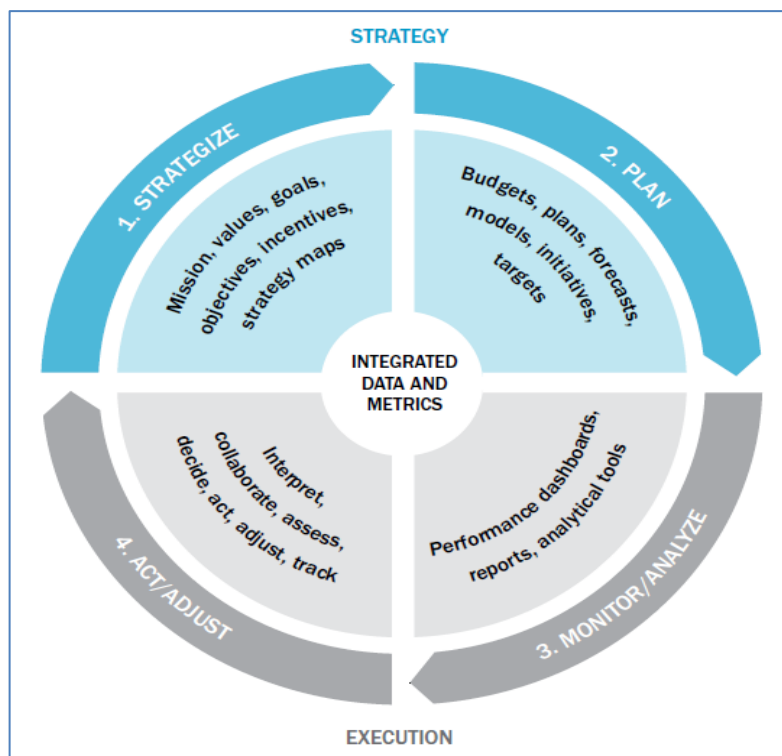
Неке „непознате“ предности ДСД, према [Ågerfalk et al, 2008] су:

- Организационе предности – а) иновација и дељење најбоље праксе, б) унапређена алокација ресурса,
- Предности на нивоу тима – а) унапређење модуларизације задатака, б) смањени трошкови координације, с) унапређење аутономије тима,
- Предности на нивоу процеса – а) формални запис о комуникацији, б) унапређена документација, с) јасно дефинисани процес.

### 1.3.3.3. Управљање перформансама пословних процеса

„Управљање перформансама је дисциплина која усклађује перформансе са стратегијом.“ [Eckerson, 2009] Под појмом управљања пословним перформансама (Business Performance Management - BPM) подразумевамо скуп пословних процеса и апликација дизајниране да оптимизују развој и извршавање пословне стратегије [Frolick&Ariyachandra, 2006]. Основни кораци дизајнирања, имплементирања и управљања BPM системом су [Frolick&Ariyachandra, 2006]:

1. Дефинисање стратегије (Strategize)
2. Планирање (Plan)
3. Надгледање и анализа (Monitor and analyze)
4. Предузимање корективних акција (Take corrective action) [BPM, 2004]



Слика 1.3.3.3.1. Животни циклус управљања перформансама пословних процеса, према Eckerson-у [Eckerson, 2009]

Управљање перформансама према [Isaev, 2011] [BPM, 2004] представља „методологију за оптимизацију извршавања пословне стратегије која се састоји из скупа интегрисаних затворених циклуса, аналитичких процеса, подржаних технологијом, који се односе на финансијске и оперативне податке. Управљање пословним перформансама - BPM (Business Performance Management) омогућава пословању да дефинише, мери и управља перформансама у односу на стратешке циљеве. Језгро BPM процеса укључује планирање, консолидацију и извештавање, анализу и примену повезаних кључних индикатора перформанси (linked key performance indicators - KPIs) кроз целу организацију“.

Неки од приступа управљању квалитетом процеса, односно управљању перформансама пословних процеса, у складу са развојном стратегијом организације су: TQM, Six Sigma, Balanced Scorecard.

Према [wGershon], TQM (Total Quality Management) представља једну од првих методологија које се односе на континуално унапређење пословних процеса. TQM се базира на Deming-ових 14 тачака унапређења пословних процеса, односно на 4 главне области:

1. Одговорност менаџмента за континуално унапређење
2. Фокус на радним процесима, како би се постигла унапређења
3. Коришћење статистике како би се мериле перформансе процеса
4. Укључивање и давање овлашћења („empowerment“) запосленима у процесу унапређења

Према [Magnusson et al, 2003], Six Sigma представља приступ остваривању корпоративних стратешких циљева. Представља метод заснован на подацима, чији је циљ отклањање дефеката у било ком процесу. Фокус Six Sigma методологије је у имплементацији стратегије базиране на мерењу, са циљем да се унапређење процеса реализује кроз примену две Six Sigma под-методологије: *DMAIC* (Define, Measure, Analyze, Improve, and Control – применљива код организација које имају формално дефинисане процесе) и *DMADV* (Define, Measure, Analyze, Design, and Verify – код пословних процеса без формалних процеса).

Према [Bianchi, 2001], Balanced Scorecard представља технику за превођење пословне визије и стратегије у скуп перспектива (финансијска перспектива, задовољство корисника, ефикасност пословног процеса, иновације и обука) које могу бити трансформисане у скуп метрика. Мерења сваке перспективе могу бити међусобно комбинована у скуп индикатора који обезбеђују вредне информације о унапређењу организације.

### **1.3.3.4. УСПЕХ СОФТВЕРСКОГ ПРОЈЕКТА, СОФТВЕРСКЕ МЕТРИКЕ, КОНТРОЛА И МОНИТОРИНГ СОФТВЕРСКОГ ПРОЈЕКТА**

Област истраживања мерења успеха софтверског пројекта, као и истраживање у области софтверских метрика представљају актуелне области истраживања и над над методама и приступима врше се истраживања и валидације [El Emam, 2000]. Ипак, основни појмови, методе и приступи су приказани у наставку.

#### **1.3.3.4.1. Дефиниције успеха пројекта и софтверских метрика**

У оквиру истраживања [Baccarini, 1999], дефинисана је разлика између успеха софтверског пројекта и софтверског производа. Према [Baccarini, 1999] [Berntsson-Svensson&Aurum, 2006], успех пројекта може бити дефинисан као:

Успех пројекта = успех пројектног менаџмента + успех производа

Успех пројекта је потребно представити мерним карактеристикама, које се могу квантитативно изразити мерењима у току и након завршетка пројекта. Мерење перформанси пројекта, према Аткинсон [Atkinson, 1999] треба да се односи на:

1. Процес развоја („извршавати посао како треба“) - односно ефикасност процеса реализације пројекта
2. Резултат реализације пројекта („добивање адекватног резултатућег система“) – карактеристике система, у постпродукционом периоду у току испоруке
3. Корисност система за организацију („користи система за организацију“) – колико развијени систем унапређује пословање организације, колико је организација имала користи од развијеног система.

Према [Frakes&Kang, 2005] [Kaner&Bond,2004] [Fenton& Pfleeger, 1997], „мерење је процес у којем су бројеви и симболи додељени атрибутима ентитета реалног света на начин да их описују у складу са јасно дефинисаним једнозначним правилима“. Према [Kaner&Bond,2004], „Мерење је емпиријско, објективно додељивање бројева, у складу са правилом изведеном из модела или теорије, атрибутима објеката или догађаја с намером да буду описани.“

Према IEEE Standard 1061 [IEEE QMetrics 1061-1998] атрибут представља „мерљиву физичку или апстрактну особину ентитета“. Фактор квалитета софтвера представља тип атрибута – „менаџмент-орјентисан атрибут софтвера који доприноси његовом квалитету“. Метрика представља функцију мерења, а метрика квалитета софтвера представља „функцију чији улази су софтверски подаци и чији излаз представља јединствену нумеричку вредност која може бити интерпретирана као ниво до којег софтвер поседује наведени атрибут који утиче на његов квалитет“. Према наведеном стандарду, разликујемо директне и индиректне метрике. Директна метрика представља „метрику која не зависи од мерења другог атрибута.“ Према [Kaner&Bond,2004], директна метрика представља функцију чији је домен само једна променљива, док индиректна метрика представља функцију чији домен представља n-торка.

Софтверска метрика представља „квантификацијски погодан мерење особине софтверског производа или процеса.“ [Frakes&Kang, 2005] Софтверске метрике се користе како би се добила мерења која су објективна и која је лако репродуковати, а која могу бити корисна за обезбеђење квалитета, мерење перформанси, откривање грешака, управљање и процену трошкова [Mair& Shepperd, 2011].

„Термин компонента односи се на јединицу посматрања над којом се примењује метрика. Може да се односи на процедуре, фајлове, пакете, класе, методе.“ [El Emam, 2000] У зависности од приступа, „јединица мерења може да се разликује. У процедуралном програмирању апликација, јединица може бити модул, фајл или

процедура. За објектно-орјентисани софтвер, метрике могу бити дефинисане на нивоу методе, класа или целог система." [El Emam, 2000]

Један од аспеката употребне вредности метрика је квантитативно представљање утврђених недостатака. „Користићемо појмове из IEEE стандардног речника [IEEE SE Glossary, 1990]. Грешка („mistake“) је људска активност коа производи некоректан резултат. Манифестација грешке је софтверска грешка („fault“), која може бити некоректни корак, процес или дефиниција податка. Софтверска грешка може резултовати неисправним резултатом рада софтвера („failure“). На пример, у току тестирања софтвер може да искаже неправилност рада, ако производи некоректне резултате у поређењу са спецификацијом..“ [El Emam, 2000]

У оквиру научних истраживања, метрике се могу користити као основ за креирање модела који представљају зависности варијабли у мерењима, а те зависности се могу испитати експериментално. [Frakes&Kang, 2005] У току више деценија истраживања у овој области, дефинисано је више модела квалитета софтвера, који се састоје од скупа метрика. Неки од модела дефинисани су у оквиру међународних и националних стандарда, као и процедура на нивоу организација.

#### 1.3.3.4.2. Типови метрика

Да бисмо дефинисали метрике квалитета софтвера, најпре треба да дефинишемо појам квалитета софтвера. Постоје многобројне дефиниције квалитета софтвера, које су прихваћене од међународних институција, међу којима је дефиниција из IEEE 729-1983 стандарда “Композитне карактеристике софтвера које одређују ниво до ког софтвер у употреби задовољава очекивања клијента” [IEEE SE Glossary 729-1983].

Према стандарду ISO/IEC 9126:1991 [ISO/IEC 9126] метрике које се односе на квалитет софтвера или “software quality metric” дефинисане су као „квантитативна скала или метод која може бити коришћен за утврђивање вредности особине коју има за конкретан софтверски производ”.

Постоје различита мерења које се односе на софтвер као производ, односно на софтверски процес који га ствара. Софтверске метрике се могу односити на софтверски систем, процес развоја или пратећу документацију. Дакле, метрике могу бити [IEEE SE Glossary, 1990]:

- Метрике производа: величине, комплексности, квалитета
- Метрике процеса
- Метрике ресурса
- Метрике пројекта

Према [Kaner&Bond,2004] типови софтверских метрика су:

1. Метрике захтева корисника (Requirements metrics): величина захтева, следљивост (Traceability), комплетност (Completeness), усклађеност и недвосмисленост (Volatility)

2. Метрике производа (Product Metrics):

а) метрике кода (линије кода –LOC, дизајн метрике – на основу захтева или докумената дизајна а пре имплементације система, објектно орјентисане метрике – омогућавају да се уочи како класе и објекти могу бити једноставнији и са мање грешака)

б) метрике тестирања – метрике које се извршавају у фази тестирања, а могу се односити на тестирање производа, али и софтвера ([Arvinder et al, 2007], [Farooq et al, 2011])

в) метрике комуникације – проучавањем арטיפакта из e-mail-а и састанака.

3. Метрике процеса – мере процес развоја софтвера, коришћене од стране менаџмента како би се проверавао буџет и ефикасност процедура, вреднују и прате аспекте процеса дизајна софтвера који се односе на: људске ресурсе, време, распоред рада, методологију

Типови мерења [El Emam, 2000]:

1. Директно мерење, нпр. број линија кода, при чему можемо разликовати [Yu et al, 2011]:

- Број линија кода – све линије кода, укључујући коментаре и празне редове
- Физички број линија кода – линије кода без коментара и празнина
- Логички број линија кода – број програмских наредби (statements) које могу бити написане и у више редова, без коментара и празнина.

2. Индиректно-изведено мерење, нпр. Густина дефекта = број defekata softverskog proizvoda / ukupna veličina proizvoda или комплексност софтвера [Yu et al, 2011] (нпр. коришћењем McCabe Cyclomatic complexity методе).

3. Предвиђање, нпр. Предвиђање труда (effort) потребног да се развоје софтвер, на основу мерења функционалности, нпр. Бројање функционалних поена

Према [El Esmat, 2000], метрике које се односе на производ могу бити:

- Статичке – подаци могу бити прикупљени из статичке анализе особина софтверских артефакта, нпр. Статичка анализа изворног кода.
- Динамичке метрике – захтевају извршавање софтверске апликације, у циљу прикупљања метричких вредности.

Вредновање у оквиру метрика могу бити [El Esmat, 2000]:

1. номинално – без одређивања редног броја, додељивањем назива
2. ординално – одређивањем редоследа, али квантитативним компарацијама (нпр. ниско, средње, високо)
3. интервално – одређивањем вредности у неком интервалу
4. пропорције, односи (ratio) – нпр. софтверски систем 1 је 2 пута већи од система 2
5. апсолутно – нпр. софтвер има 350,000 линија кода

У оквиру [Cavano&McCall, 1978] дате су дефиниције основних појмова који се користе у оквиру мерења квалитета софтвера: „Коректност – ниво до ког програм задовољава своју спецификацију и испуњава задатке корисникове мисије. Поузданост – ниво до ког се од програма може очекивати да извршава предвиђене функције захтеване прецизности. Ефикасност – обухват рачунарских ресурса и кода потребних од стране програма да би извршио функцију. Интегритет – Ниво до ког приступ софтверу или подацима од стране неауторизованих особа може бити контролисан. Искористивост („Usability“) – Труд потребан да би се научило коришћење, користио („operate“), припремио улаз и интерпретирао излаз програма. Могућност одржавања („Maintainability“) – Труд потребан да би се лоцирала и поправила грешка у програму који је у употреби („operational program“). Могућност тестирања („Testability“) – Труд потребан да се тестира програм како би се обезбедило да извршава предвиђене функције. Флексибилност – Труд потребан да би се модификовао програм у употреби. Портбилност – Труд потребан да би се извршио трансфер програма са једне хардверске конфигурације или софтверског системског окружења у друго. Поновна искористивост („Reusability“) – Ниво до ког програм мож бити коришћен у другим апликацијама, односи се на програмске пакете и обухват функција које програм извршава. Интероперабилност – труд потребан да се повеже („couple“) један систем са другим.“

#### **1.3.3.4.3. Контрола и мониторинг софтверског пројекта**

У области евалуације процеса реализације и примене информационих технологија, као и у области процеса мерења у софтверском инжењерству, развијени су међународни стандарди, међу којима су [ISO/IEC 15504], [SO/IEC 15939:2007].

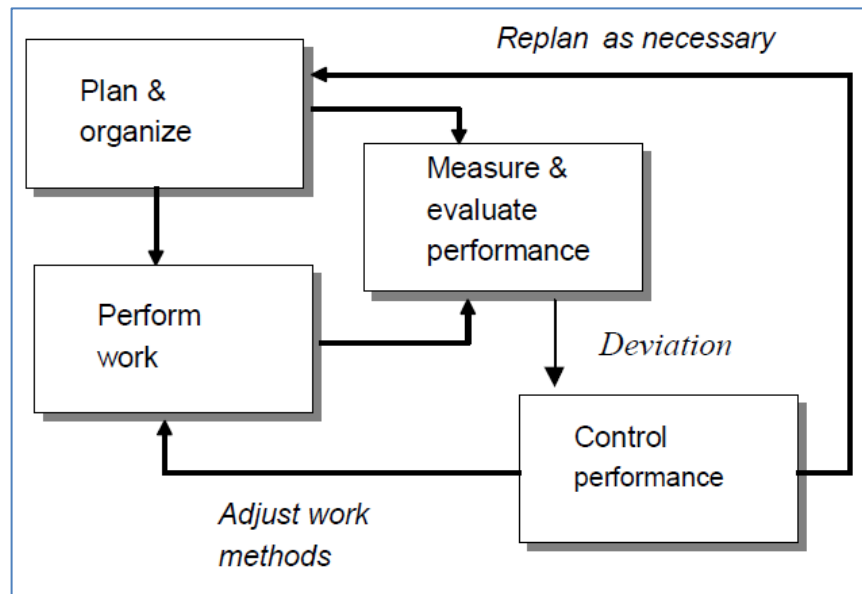
„Сврха контроле пројекта је да се:

- одржи пројекат на курсу“ (планираном путу) „и што ближе плану што је могуће,
- идентификује проблеме пре него што се десе и



- имплементира планове опоравка пре него што се непоправљива штета деси. Контрола укључује компарацију прогреса са планом и предузимање корективних акција када перформансе одступају значајно од плана.

За благовремену у ефективну контролу, пројектни менаџер мора имати потпуну видљивост напретка. Пројекат мора бити праћен и надгледан систематично. Ниво формализације и фреквенција мониторинга зависи од типа пројекта. Главне варијабле које се прате су трошкови, временски план (schedule) и техничке перформансе.“ [Jurison, 1999]



Слика 1.3.3.4.3.1. Активности у контроли пројекта, према Jurison-у [Jurison, 1999]

„Критични фактори успеха могу бити коришћени да се развију одговарајуће мере перформанси које могу бити моћни алати у управљању пројектима. Процес добијања тих мера одија се у следећим корацима:

- Идентификовати пројектне циљеве и задатке
- Дефинисати и одредити приоритете међу критичним факторима успеха
- Развити скуп одговарајућих мера
- Имплементирати систем континуалног мониторинга тих мера.“ [Jurison, 1999]

Према [Haban& Wybranietz, 1989], у оквиру мониторинга, најчешће разликујемо мониторинг систем и систем који се надгледа и мери. Под мониторингом подразумевамо „издвајање података о систему у току његовог рада“ [Haban& Wybranietz, 1989]. Када је у питању мониторинг, разликујемо:

- хардверске мониторе (уређаје који мере карактеристике других уређаја или системског софтвера)
- софтверске мониторе (софтверске апликације које мере карактеристике хардверских или софтверских компоненти). Софтверски монитори се најчешће налазе у истој радној зони са објектом свог надгледања, што може утицати на резултате мерења. Софтверски монитори су флексибилнији на динамику промена и омогућавају праћење и анализу историјских података.

Важне карактеристике мониторинг система су: минимизација утицаја надгледаног система на монитор, транспарентна интеграција надгледаног система и монитора, подршка континуираном мониторингу у току рада надгледаног система, графички интерфејс за представљање и анализу података, флексибилност и прилагодљивост променама софтвера и хардвера, локално процесирање мониторинг система на сваком чвору који се анализира (ради растеређења централног система), повратна спрега – достављање информација о мереним величинама надгледаном систему, како би могао да унапреди своје понашање. [Haban& Wybranietz, 1989]

Према [Marinescu et al, 1990], у оквиру мониторинга, разликујемо циљни систем (чији се процеси и догађаји надгледају) и мониторинг систем (који надгледа циљни систем, бележи податке, интерпретира итд.). Мониторинг систем може да утиче на рад циљног система (intrusive monitor) или да не утиче или утиче врло мало (non-intrusive monitor).

Према [Siles, 2004], мониторинг пројекта представља процес рутинског прикупљања, складиштења, анализе и извештавања о информацијама у вези пројекта које се користе за доношење одлука у оквиру пројектног менаџмента. Мониторинг обезбеђује пројектном менаџменту и заинтересованим странама информацију потребну за вредновање прогреса пројекта, идентификацију трендова, образаца или девијација, како би се мерио прогрес у складу са очекиваним циљевима реализовао пројекат у оквиру планираних ограничења.

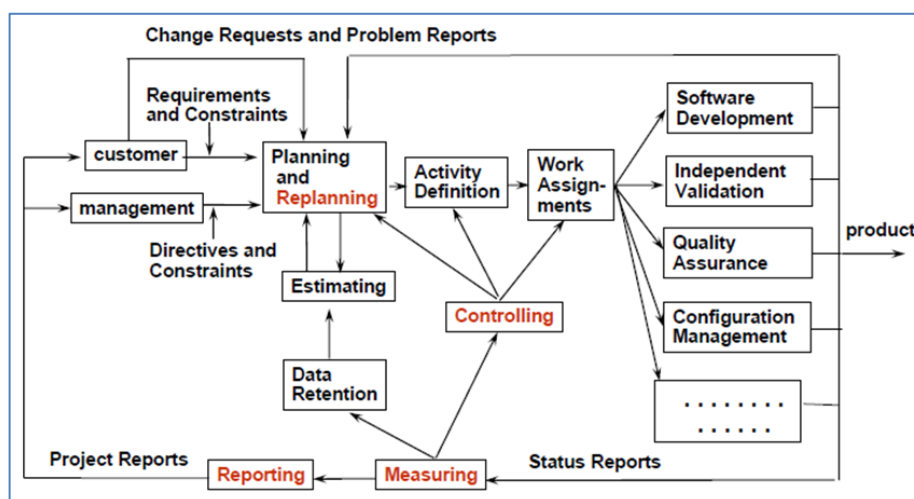
У оквиру [Fairley, 2009] описани су основни концепти мерења и контролисања софтверског пројекта. Разлози за мерење вредности атрибута активности и процеса развоја софтвера су: обезбедити фреквентне индикаторе прогреса, обезбедити рано упозорење о проблемима, омогућити анализу трендова пројекта, омогућити процену трошкова и трајања пројекта, изградити складиште података где би се евидентирао ток и историја пројекта у организацији. Области мерења софтверског пројекта су:

а) особине производа (које захтеване особине су имплементирание и демонстриране да функционишу), квалитативни атрибути производа (дефекти, поизданост, доступност, време одговора...)

б) особине процеса развоја: б1) ангажовање (effort): Количина рада утрошена на поједине радне активности; б2) распоред: постигнуће објективно мерљивих кључних тачака; б3) трошкови: утрошак различитих ресурса, укључујући људски рад; б4) напредак: производи рада који су основни, завршени, прихваћени

ц) особине ризика: статус фактора ризика и активности на смањењу ризика

Према [Fairley, 2009], мерење се односи на прикупљање, валидацију и анализу информација о статусу пројекта. Контролисање се односи на примену корективних акција када текући статус није у складу са планираним статусом: статусом радног производа, квантитета и квалитета који се односи на радни производ, статус развојног процеса, распоред, буџет, ресурси, напредак, статус фактора ризика и активности смањења ризика.



Слика 1.3.3.4.3.2. Модел радног тока у мерењу и контролосању софтверских пројеката, према Fairley-у [Fairley, 2009]

Корективне акције се могу односити на: проширење распореда рада, додавање ресурса, коришћење напреднијих ресурса, унапређење елемената развојног процеса, измене у захтеваним особинама. Ресурси који могу бити унапређени, додати или измењени односе се на: људе, софтверске компоненте, хардверске компоненте, софтверски алати итд.

## **2. ПРЕГЛЕД ПОСТОЈЕЋИХ ИСТРАЖИВАЊА, ТЕХНОЛОШКИХ РЕШЕЊА И ИСКУСТАВА ИЗ ПРАКСЕ**

### **2.1. УПРАВЉАЊЕ СОФТВЕРСКИМ ПРОЈЕКТИМА У ДИСТРИБУИРАНОМ ОКРУЖЕЊУ**

#### **2.1.1. Анализа истраживања проблема дистрибуираног развоја софтвера и примена SE-PM матрице**

У свету комуникација и глобализације, пројекти развоја софтвера су често имплементирани у сарадњи дистрибуираних тимова и софтверских компанија. Дистрибуирани развој софтвера (ДСД) је уобичајена пракса у софтверској индустрији [Jimenez et al, 2009], за фирме које настоје да редукују трошкове, повећају расположивост људских ресурса, повећају квалитет итд. [Herbsleb&Moitra, 2001]. ДСД је најчешће имплементиран у оквиру већих софтверских пројеката [Herbsleb, 2007]. Из наведених разлога, у последњих 10-20 година истраживања су фокусирана на ДСД проблеме.

Нека од приказаних истраживања у представљеним радовима који се баве ДСД проблемима су спроведена у оквиру великих ИТ компанија које раде outsourcing у оквиру других земаља, као што је Alcatel [Ebert & De Neve, 2001], IBM [Sengupta et al, 2006] и Philips [Kommeren&Parviainen, 2007]. Мањи је број ДСД радова који се односе на мала и средња предузећа - SME (Small and Medium Enterprises) и њихова искуства (као што је нпр. [Jimenez et al, 2010]).

Један од истраживачких и практичних подухвата у области ДСД организован је у оквиру међународног пројекта OPHELIA (Open Platform and meTHodologies for deVELOpment tools and IntegrAtion in a distributed environment) [OPHELIA], који је започео октобра 2001. године, са партнерима из 6 земаља Европске Уније, с циљем да дефинише платформу подршке софтверском инжењерству у дистрибуираном окружењу (у оквиру целог животног циклуса софтвера), кроз развој методологије и процеса оптимизације софтверског животног циклуса, као и интеграцију алата у оквиру овог окружења.

Истраживања у области проблема ДСД развоја рађена су и у оквиру научно-истраживачких активности у оквиру више докторских дисертација. Julia Kotlarsky [Kotlarsky, 2005] у оквиру свог доктората анализира рад великих компанија (SAP, BAAN, LeCroy, TCS) у управљању глобалним развојем софтвера интегрисаним са приступом компонент-базираног развоја, издваја факторе успеха које се односе на менаџмент таквим пројектима, где се посебно истичу друштвене везе између учесника пројекта. Marcelo Cataldo [Cataldo,2007] анализира зависности које постоје између софтверских модула и рада у развоју софтвера дистрибуираном окружењу и издваја две зависности: логичку зависност модула и зависност радних активности као два кључна фактора који утичу на успех пројекта, а посебно издваја и друштвене зависности учесника пројекта као још један важан фактор успеха.

Анализа истраживања у области ДСД је презентована у оквиру неколико прегледних радова, али и већег броја радова који се баве појединачним типовима ДСД проблема. У [Jimenez et al, 2009] систематским прегледом публикованих радова у области ДСД приказана је тенденција пораста ДСД истраживања. "Кроз систематску претрагу, може бити закључено да је предмет ДСД очигледно област која није широко истраживана све до пре пар година и тек недавно је већи број радова публикован." Према [Jimenez et al, 2009], већина ДСД истраживања укључују: студије случаја (47%), експерименте (27%), прегледне радове, симулације and упитнике. Стандарди који су представљали

основу у публикованим радовима односили су се на СММ, СММI, ISO 12207, ISO 9001, ISO 15504. Истраживања су најчешће извршена у оквиру организација – фирми (40%) и на универзитетима (16%). У последњих 20 година, проблеми ДСД истражени су у оквиру великог броја радова анализирајући појединачне аспекте (референце одабраних радова су приказани у табели 9.2.1. у секцији Прилог), као и мањег броја прегледних радова који сумарно приказују релевантне проблеме и одговарајућа истраживања. ДСД проблеми који су издвојени у неким од прегледних радова приказани су у оквиру табеле 2.1.1.1.1.1.

Табела 2.1.1.1.1.1. Приказ ДСД проблема који су приказани у прегледним радовима

РЕФЕРЕНЦА	ДСД ПРОБЛЕМИ
[Gorton&Motwani, 1996]	Организација виртуелног тима (кооперативни, делегацијски и консултативни модел) комуникације у тиму, процес развоја софтвера, додељивање задатака
[Herbsleb&Moitra, 2001]	Расположивост ресурса, различита инфраструктура, различита експертиза у различитим технологијама, стратешки проблеми (подела посла на локације, координација), отпор због страха (губитак посла, смањење контроле, потреба за пресељењем и честим путовањима), проблеми културе (однос према хијерархији, стил комуникације), неадекватна комуникација (формална – довољно прецизна и свеобухватна, неформална – да буде на располагању), управљање знањем, управљање информисањем (документовање, обавештавање, истицање рокова, проблема, потребног знања), недостатак синхронизације, пројектни и процес менаџмент, нестабилне спецификације и захтеви, недостатак добрих алата за комуникацију, контрола промена и верзија софтвера, технички проблеми (споре и непоуздане мреже, усклађеност верзија алата, компатибилност формата података)
[Carmel&Agarwal, 2001]	Удаљеност, колаборација, координација, контрола, културне удаљености, језичке баријере, временска удаљеност
[Ebert&De Neve, 2001]	Језик, културне баријере, кохерентност, колокација, алокације, радне улоге (језгро компетенције, инжењерство, услуге), обука, менторство, конкурентно инжењерство, развој орјентисан на особине, управљање променама, инкрементални развој, СММI нивои
[Dhar&Balakrishnan, 2006]	Ризици IT outsourcing - фактори: људи (различите способности, искуства), Знање (Функционално, технолошко, управљачко), Културни, Политички, Финансијски (Рачуноводствени стандарди, варијација курсне листе), Стандарди квалитета, Стандарди мерења перформанси, Обухват, трошкови, процена времена, Различите компаније из страних земаља имају различити начин управљања и суштинске компетенције, Правни уговори и стандарди интелектуалног власништва, Безбедност – заштита и контрола података, Опоравак од катастрофа, Управљање уговарањем (формулисање уговора, планирање распореда, планирање активности, слање и прихватање испорука, одјава).
[Sengupta et al, 2006]	Неадекватна комуникација, посебно неформална, културне разлике, стратегијски проблеми, процес, технички проблеми, управљање знањем, погрешно разумевање захтева, договор око корисничког интерфејса, географска дисперзија, проблеми временских зона, културне и организационе разлике, кључни индикатори перформанси, СММ нивои
[Herbsleb, 2007]	Мање фреквентна комуникација, мање ефективна комуникација, временске разлике, друштвено-језичке разлике, географска удаљеност, непознавање задужења и експертизе, некомпатибилност алата, знања, процеса, радних навика и културе, софтверска архитектура, дефинисање захтева, развојна окружења и алати, организација
[Ågerfalk et al, 2008]	Географска удаљеност: недостатак неформалне комуникације, погрешно разумевање, измене захтева; Временска удаљеност: временске зоне, кашњење одговора (feedback), различито радно време; Социо-културне разлике: Природа процеса развоја софтвера, језик, недостатак фамилијарности, недостатак поверења
[Jiménez et al,	Људски ресурси, организационо управљање, комуникација чланова

2009]	тима, инфраструктура, усклађеност са организацијом, управљање пројектом, усклађеност са стандардима, управљање конфигурацијом, тестирање, управљање знањем, контрола процеса, групна свесност, координација, колаборација, подршка процесу, квалитет, мерење
[Jiménez et al, 2010]	Комуникација, управљање конфигурацијом, управљање знањем, Управљање квалитетом, управљање ризиком, управљање пројектом, подршка процесу, координација, колаборација
[Noll et al, 2010]	Географска, временска, културна и језичка удаљеност, дељење знања, способности, организација, процес, управљање, страх и поверење, инфраструктура, архитектура производа
[Kocaguneli et al, 2013]	Квалитет софтверског производа, комуникација, координација, организација, стратегије развоја

У циљу систематизације претходно приказаних резултата, на основу стандарда за софтверско инжењерство [SWEBOOK, 2014] и за управљање пројектима [PMBOOK, 2013], креирана је тзв. SE-PM матрица која приказује унакрсну покривеност истраживања у односу на наведене 2 области. Вертикално (приказано са леве стране) су приказане области софтверског инжењерства, а хоризонтално (приказано као заглавља колона на горњој десној страни матрице) су приказане области пројектног менаџмента. Наведена матрица је предложена и иницијално је примењена за систематизацију резултата анализе прегледних радова (резултати анализе су приказани у табели 2.1.1.1.1.), а резултати примене су приказани у раду [Kazi et al, AIIT, 2014].

Табела 2.1.1.1.1.2. Резултати примене SE-PM матрице за анализу покривености ДСД проблема у прегледним радовима [Kazi et al, AIIT, 2014]

<b>SE – PM матрица</b>	<b>Интеграција</b>	<b>Обухват</b>	<b>Време</b>	<b>Квалитет</b>	<b>Људски ресурси</b>	<b>Комуникације</b>	<b>Ризици</b>	<b>Набавке</b>	<b>Заинтересоване стране</b>
<b>Захтеви</b>							√		
<b>Дизајн</b>							√		
<b>Конструкција</b>	√		√	√	√	√	√		
<b>Тестирање</b>							√		
<b>Одржавање</b>							√		
<b>Управљање конфигурацијом</b>				√			√		
<b>Инжењерски менаџмент</b>		√	√	√	√		√		
<b>Процес инжењерства</b>			√	√	√		√		
<b>Инжењерски модели и методе</b>		√			√		√		
<b>Квалитет</b>				√			√		
<b>Инжењерска економија</b>		√	√				√	√	

На овај начин се може приказати покривеност појединих области проблема ДСД у оквиру објављених радова. Визуалним приказом анализираних резултата у претходној табели, може се видети да су све области софтверског инжењерства обухваћене прегледним радовима који представљају проблеме ДСД. Такође, може се видети да је област квалитета присутна у обе стандардне методологије – софтверског инжењерства и управљања пројектима. Свакако, приказане су и области које нису покривене и могу представљати могуће даље области истраживања.

Детаљнија анализа прегледних радова коришћењем SE-PM матрице реализована је у оквиру рада [Kazi et al, COMENG, 2014]. Колонама су додељена слова, а редовима бројеви, а затим је извршено додељивање сваком издвојеном проблему из прегледног рада одговарајућа координата ради прецизнијег одређивања позиције наведеног

проблема у SE-PM матрици (нпр. software development process (G8), task allocation (G7) итд.). Резултати су унети у SE-PM матрицу као референце радова, као што је приказано на табели 2.1.1.1.1.3.

Табела 2.1.1.1.1.3. Резултати примене SE-PM матрице користећи координате и референце прегледних радова [Kazi et al, COMENG, 2014]

<b>SE – PM matrices</b>	<b>Column Id</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
<i>Row ID</i>		<b>Integration</b>	<b>Scope</b>	<b>Time</b>	<b>Quality</b>	<b>Human Resource</b>	<b>Communication</b>	<b>Risk</b>	<b>Procurement</b>	<b>Stakeholder</b>
1	Requirements							5, 6		
2	Design				10			5, 10		
3	Construction							9		
4	Testing						4	4, 7		
5	Maintainance						4	4		
6	Configuration Management	4	4				4	3, 4, 5, 7, 8	7, 9	
7	Engineering Management	2, 3, 5, 6, 7, 8, 9	3, 5, 6, 7, 8	6, 7, 8, 9	5, 7, 8	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	1, 2, 4, 5, 7, 8, 9, 10	7, 8	7, 8
8	Engineering Process		7	6			4	1, 5, 6, 7, 8, 9		
9	Engineering Models and Methods									
10	Quality		3, 4, 7		3, 4, 5, 7			3, 4, 5, 7		
11	Engineering Economics				4		4			

1- [Gorton and Motwani, 1996], 2- [Carmel and Agarwal, 2001], 3- [Ebert and De Neve, 2001], 4 - [Dhar and Balakrishnan, 2006], 5 - [Sengupta et al, 2006], 6 -[Ågerfalk et al, 2008], 7 - [Jiménez et al, 2009], 8 - [Jiménez et al, 2010], 9 - [Noll et al, 2010], 10 - [Kocaguneli et al. 2013]

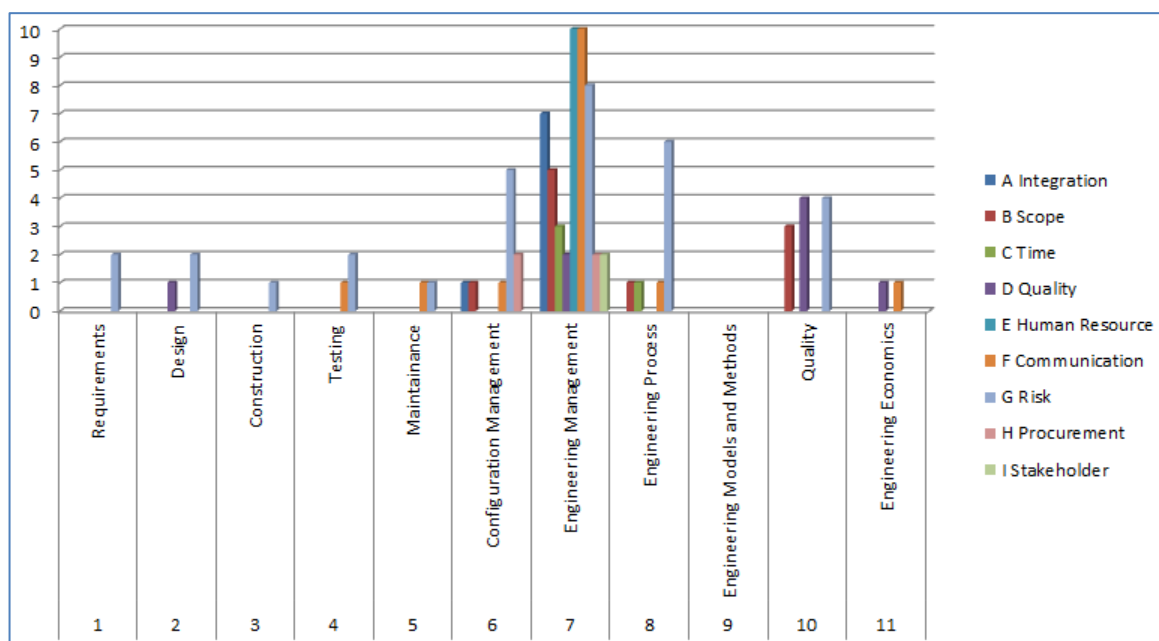
Укупан број радова у којима је одговарајући ДСД проблем, односно група проблема разматрана унет је у SE-PM сумарну матрицу, као што је приказано у табели 2.1.1.1.1.4.

Табела 2.1.1.1.1.4. Резултати примене SE-PM матрице сумарним приказом броја радова који разматрају одговарајућу групу ДСД проблема

<b>SE – PM matrices summary</b>	<b>Column Id</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
<i>Row ID</i>		<b>Integration</b>	<b>Scope</b>	<b>Time</b>	<b>Quality</b>	<b>Human Resource</b>	<b>Communication</b>	<b>Risk</b>	<b>Procurement</b>	<b>Stakeholder</b>
<b>1</b>	<b>Requirements</b>							2		
<b>2</b>	<b>Design</b>				1			2		
<b>3</b>	<b>Construction</b>							1		
<b>4</b>	<b>Testing</b>						1	2		
<b>5</b>	<b>Maintainance</b>						1	1		
<b>6</b>	<b>Configuration Management</b>	1	1				1	5	2	

<b>7</b>	<b>Engineering Management</b>	7	5	3	2	10	10	8	2	2
<b>8</b>	<b>Engineering Process</b>		1	1			1	6		
<b>9</b>	<b>Engineering Models and Methods</b>									
<b>10</b>	<b>Quality</b>		3		4			4		
<b>11</b>	<b>Engineering Economics</b>				1		1			

На основу резултата сумарног приказа, визуални приказ (слика 2.1.1.1.1.) у виду стубичног графика приказује сумарне резултате анализе ДСД проблема и могућност увида у покривене и непокривене области истраживања и разматрања ДСД проблема.



Слика 2.1.1.1.1.1. Графички приказ покривености ДСД проблема у анализи прегледних радова, на основу примене SE-PM матрице

На основу слике 2.1.1.1.1.1. можемо закључити да је највећи број ДСД проблема у прегледним радовима категорисан у оквиру области софтверског инжењерства – Engineering Management, а у оквиру области пројектног менаџмента у области Ризика и Комуникација.

## 2.1.2. Анализа истраживања у области управљања софтверским пројектима који се реализују у дистрибуираном окружењу

Иако су се практична решења у индустрији у области дистрибуираног развоја софтвера појавила већ средином деведесетих година [Nidiffer&Dolan, 2005], научна истраживања у овој области интензивније су се појавила у претходном периоду од 10-15 година, где је највише резултата забележено од 2006-2008. године, са тенденцијом раста [Jiménez et al, 2009]. У овом одељку биће описани радови који се односе на управљање софтверским пројектима у дистрибуираном окружењу.

Gumm [Gumm, 2006] разматра димензије дистрибуираности у пројектима развоја софтвера. Разликује објекте дистрибуције (људе, артефакте, задатке итд), типове дистрибуције (физичка/географска дистрибуција), организациона дистрибуција, временска дистрибуција, дистрибуција између група заинтересованих страна). Такође разматра и изазове/проблеме дистрибуције, као и постојећа решења. Историјат развоја дистрибуираног пројектног менаџмента приказан је у раду [Nidiffer&Dolan, 2005], кроз

приказ промена у прописима, захтевима корисника и технологији, почев од 80-тих година прошлог века до данас. Приказани су и најчешћи проблеми дистрибуираног развоја софтвера и управљања таквим софтверским пројектима.

Hande et al [Hande et al, 2012] описује покретаче развоја у области дистрибуираног развоја софтвера: оптимизација ресурса, сарадња са корисницима, глобални међународни развој, управљање трошковима, коришћење веће расположивости талената из ширег географског подручја, подршка сваког дана 24x7 одржавању критичних апликација, спајање организација. Најчешћи проблеми-изазови су: комуникација и колаборација, управљање развојем у оквиру промена и целокупног животног циклуса, управљање пројектима, видљивост резултата свим учесницима, прилагођавање екстерних учесника. У овом раду такође дат је детаљни приказ смерница које би водиле успешнијем управљању пројектима у дистрибуираном окружењу, а односе се на: а) принципе и правила управљања пројектима у дистрибуираном окружењу, б) посебно креиране софтверске алате који треба да пруже подршку комплетном животној циклусу, интегрисана решења која обједињују резултате различитих облика комуникације, подршку управљању пројектима, безбедност података, као и поуздане рачунарске мреже и хостинг системе који чувају податке и омогућавају њихову размену. Конкретна практична упутства која су настала на основу вишегодишњег искуства у пракси односе се на смернице у управљању пројектима у дистрибуираном окружењу:

- „Договорити се и комуницирати на почетку пројекта у вези циљева пројекта, у области квалитета, тачака контроле, садржаја и алокације ресурса.
- Постарати се да су обавезе описане у писаној форми која се може контролисати.
- Имати једног пројектног лидера који је у потпуности одговоран за остваривање пројектних циљева и доделити му пројект менаџмент тим, који представља главног носиоца културе у пројекту.
- У оквиру сваког пројекта, пратити континуирано 10 највећих ризика, који су мање технички, а више организациони.
- Дефинисати на почетку пројекта који тимови су укључени и шта треба да раде на свакој локацији.
- Поставити пројектну основну страницу („homepage“) која сумира садржај пројекта, метрике напретка, информације о планирању и специфичне информације о сваком тиму.
- Обезбедити интерактивни модел процеса, базиран на прихваћеној најбољој пракси, која омогућава извршавање процеса у складу са специфичним потребама пројекта или тима.
- Ригорозно се придржавати производних линија користећи договорене стандардне процесе који се односе на СММ ниво 3 организациони образац.
- Обезбедити довољно комуникационих средстава, као што је видеоконференсинг или дељени радни простори и глобалне софтверске библиотеке.
- Ригорозно подстицати СМ и градити управљачка правила (као што је гранање, спајање и фреквенција синхронизације) и обезбедити потребну подршку алатима.
- Ротирати менаџмент кроз разне локације и културе, како би се креирала потребна свесност у вези културне разноликости и како се ускладити са тиме.
- Организовати мешовите тимове из различитих земаља како би се интегрисали интелектуално културно наслеђе и усмерило ка организационом и пројектно оријентисаном духу.
- Учинити тимове у потпуности одговорнима за своје резултате.“ [Hande et al, 2012]

Zanoni & Audi [Zanoni&Audy, 2003] предлажу нови модел за управљање пројектима, који се односе на дистрибуирани развој софтвера. Поред стандарда PMBOK из 2000. Године, предлажу интеграцију са спиралним моделом развоја софтвера и објектно-оријентисаним приступом процесу развоја, дефинисаног у оквиру RUP (rational unified process) и UML (Unified Modeling Language). На овај начин, предлажу 6 фаза софтверског пројекта који се реализује у дистрибуираном окружењу: дефинисање



захтева, дефинисање пројекта, продукциони процеси, евалуација, транзиција и интеграција.

Raut [Raut, 2008] предлаже нови оквир за управљање пројектима у дистрибуираном окружењу, који се може применити у ИТ индустрији. Заправо, у оквиру постојеће методологије управљања пројектима, посебно истиче потребу за систематичним управљањем променама, управљањем кохезијом тима и управљање емоционалном интелигенцијом тима. Да би се од групе створио тим потребно је да менаџер пројекта:

- „Успостави захтеве за перформансама и усмерења
- Изабере учеснике на основу способности и потенцијала у способностима, а не на основу личности
- Посебно обрати пажњу на прве састанке и акционе ставке
- Постави јасна правила понашања
- Редовно позива чланове да му достављају „свеже“ чињенице и информације
- Истражи снагу позитивног одговора („feedback“), признања и награде“ [Raut, 2008].

Такође, да би се унапредила емоционална интелигенција тима [Raut, 2008], потребно је управљати групном емоционалном интелигенцијом која укључује: Поверење, Идентитет, ефикасност, што води ка учешћу, кооперацији и сарадњи, а коначно резултате: бољим одлукама, креативнијим решењима, већој продуктивности.

Управљање ИТ пројектима у односу на виртуалне дистрибуиране тимове који учествују у њиховој реализацији предмет је истраживања неколико радова. У раду [Beise, 2004] наглашено је да примена метода управљања пројектима није довољно практично истражена у области дистрибуираних софтверских пројеката. Посебан проблем у оквиру оваквих пројеката представља разноликост и разноврсност људских карактеристика у тиму. Хипотезе овог истраживања односе се на утицај пројектног менаџмента на смањивање разлика и конфликта услед разноликости учесника у тиму, односно унапредиће кохезију, перформансе и задовољство. Прелиминарно емпиријско истраживање са 22 студента са три америчка универзитета извршено је 2003. године.

Cassey & Richardson [Casey&Richardson, 2006] описују управљање пројектима и оквиру виртуалних тимова. Посебно детаљније разматра аспекте управљање софтверским пројектима са аспекта стратегије управљања виртуалним тимовима, управљање ризиком, инфраструктуром, процесе у виртуалном тиму, управљање конфликтима, структура и организација виртуалног тима.

### 2.1.3. АНАЛИЗА ЕМПИРИЈСКИХ ИСТРАЖИВАЊА ИЗ ИТ ИНДУСТРИЈЕ

Емпиријска истраживања у области дистрибуираног развоја софтвера и управљања пројектима у дистрибуираном окружењу у оквиру индустријских искустава односе се преваходно на „outsourcing“ фирми, које део продукције реализују у другим земљама. Поред „outsourcing“ индустријских искустава, емпиријска истраживања се такође односе и на „open source“ развој.

#### 2.1.3.1. Емпиријска истраживања „outsourcing“ предузећа

Истраживања која су се вршила у ИТ индустрији преваходно су се вршила у оквиру великих ИТ компанија која раде „outsourcing“ у оквиру других земаља, као што је фирма Alcatel [Ebert&De Neve, 2001], IBM [Sengupta et al, 2006], Siemens [Herbsleb et al, 2005] и Philips [Kommeren& Parviainen, 2007]. Мањи број истраживања односи се на емпиријска истраживања дистрибуираног развоја софтвера у оквиру малих и средњих предузећа [Jiménez et al, 2010], у појединачним земљама - у Финској [Paasivaara et al, 2009] [Komi-Sirvio&Tihinen, 2005] и у више других земаља [Herbsleb&Mockus, 2003] [Heeks et al, 2001].

У наставку ће бити детаљно приказана искуства из фирме Alcatel [Ebert&De Neve, 2001]. Иако је преваходно компанија која се бави телекомуникацијама, у укупном развојном буџету фирме Alcatel, чак 80-90% се односи на развој софтвера. Развој софтвера у овој фирми се одвија по тимовима у укупно 15 земаља. Ефикасност рада и квалитет комбинују се са трошковима локација на којима се одвија развој, тако да се део развоја одвија у источној Европи и Индији. Истовремено се реализује више пројеката на најчешће 2 или више локација истовремено. Ранија релативна независност појединачних развојних локација замењена је глобалним управљањем производним линијама. Нека од практичних искустава у овој фирми и савети како боље организовати посао у глобалном развоју софтвера су заснована на научним истраживањима у оквиру ове компаније. Резултати у виду закључака су дати у наставку:

- Креирати кохерентне колокацијске тимове који су потпуно радно алоцирани, а не виртуалне тимове. То значи да посао треба поделити у јасно одвојене модуле (кохезија) при чему сваки модул ради један тим. Тимови могу бити на различитим географским локацијама, али у оквиру истог тима људе организовати да на истом модулу раде тако што су физички смештени у истој згради (колокација) и да раде само посао на једном пројекту (радна алокација), а не на више пројеката истовремено.
- Одлуке о архитектури целог система, ревизије кључних тачака и тестирање ради се на једној локацији (централно).
- Радне улоге су: језгро компетенције (архитектура система, ревизија, главне одлуке), инжењерство (развојни део функционалности), услуге (документовање, одржавање).
- Јасно дефинисаним и договореним терминима почетка и завршетка посла, дефинисаним квалитетом и трошковима, садржај се развија итеративно, континуираном производњом резултата.
- Треба разликовати као посебне пројекте развој нових функција од корекција грешака у постојећим решењима.
- Јасна дефиниција одговорности сваког појединца за успех целог пројекта захтева детаљан пројектни план са јасно дефинисаним потребним знањима, трајањем активности, функцијама које се развијају, дефинисањем тимова и итерација развоја.
- Обука и менторство (coaching) над младим и неискусним кадровима унапређује квалитет будућег решења и потребно је да буде присутно и организовано.
- Развој орјентисан на карактеристике, а не на архитектуру (feature-oriented development) - већа интеграција унутар тима и орјентација на испуњавање функционалних целина пројекта, као и орјентација на циљеве пројекта и допринос

пројекту – дизајнер остаје укључен у пројекат и прати га до фазе тестирања, тестер учествује у дизајну, јер мора се и тестабилност решења уградити у сам дизајн... На овај начин су смањени време израде и трошкови.

- Укључивање јасно дефинисаних глобалних производних линија, односно фамилија производа који могу да деле дизајн и имплементацију – дефинишу се основни производи, који се касније прилагођавају појединачним тржиштима. Разликују се: 1) независни модули који не подлежу кастомизацији, 2) језгро производа које ако се мења, утиче на све остале производе, 3) специфични модули за појединачне локалне потребе корисника. Настојање да варијанти основног производа буде што мање, како би се лакше управљало променама.
- Потреба за алатима за представљање и синхронизацију промена у производним линијама. Реализовано је интранет решење, које даје свим учесницима увид у: фазе и успешност реализације пројекта, фазе развојног процеса, елементе и особине производа који се реализује.
- Инкрементални развој – повезати захтеве са функционалностима, повезати модуле у шири контекст интерфејса ка другим модулима и проценити њихову међузависност, дефинисати пројектни план, доделити један инкремент једном тиму (често реализује и више узастопних инкремента), посебне линије тестирања,
- Заједнички развојни процес, методологија и терминологија. Битна је размена свесности (информисаности о стању), комуникација и размена знања. Креиран је заједнички „on-line web“ базирани систем за представљање пројекта, документације и процеса.
- Организациона култура запослених - спремност на промене. Ниво организационог квалитета СММ ниво 2 је минимум који омогућује успешну реализацију пројекта, уз настојање повећања СММ нивоа.

Конкретна упутства и савети из искустава фирме *Alcatel* у организацији пројекта [Ebert&De Neve, 2001]:

- Договором дефинисати и комуницирати на почетку пројекта у вези циљева пројекта, као што су квалитет, тачке провере („milestones“), садржај или алокација ресурса.
- Постарати се да задужења појединаца и тимова постоје у писаној и контролисаној форми.
- Одредити вођу пројекта који је потпуно одговоран а остваривање циљева пројекта и одредити његов тим за управљање пројектом, који представља носиоца културе у оквиру пројекта.
- У оквиру сваког пројекта пратити континуирано 10 највећих ризика, који су најчешће у глобалним пројектима мање техничке, а више организационе природе.
- На почетку пројекта одредити који су тимови укључени и шта ће они радити на свакој локацији.
- Припремити web страницу пројекта која сумарно приказује садржај пројекта, метрике напретка, информације о планирању и информације специфичне за сваки тим.
- Обезбедити интерактивни модел процеса базиран на прихваћеној најбољој пракси, која омогућава извршавање процеса за специфичне потребе пројекта или тима.
- Строго подстицати реализацију производних линија, користећи усвојен стандардни процес развоја, који се односи на СММ ниво 3 организације.
- Обезбедити довољно комуникационих начина, као што је видеоконференсинг или дељена радна окружења и гловалне софтверске библиотеке.
- Строго подстицати управљање конфигурацијом и правила управљања изградњом решења (нпр. у гранању, спајању и синхронизацији, фреквенцији), обезбедити одговарајућу подршку алата.
- Ротирати чланове менаџмента кроз разне локације и културе како би се креирала потребна свесност културних разлика и како да се усклађује са њима.
- Креирати тимове људи из различитих држава како би се интегрисала индивидуална културна наслеђа и усмерило ка организационом и пројектно-орјентисаном духу.
- Учинити да су тимови у потпуности одговорни за своје резултате.

### 2.1.3.2. Емпиријска истраживања „open source“ развоја софтвера

Појам „open-source“ софтвера (OSS) може се дефинисати као „софтвер за који корисници имају приступа изворном коду. То га разликује од уобичајене праксе од стране комерцијалних издавача софтвера који само достављају бинарне извршиве верзије софтвера. Већина 'open source' софтвера се такође дистрибуира без икаквих трошкова са ограниченим рестрикцијама како може бити коришћен, тако да се под појмом слободан или бесплатан у односу на 'open source' односи на два значења: 1) ослобођен од трошкова б) слободан да се са тим софтвером може радити шта год се пожели (тј. оно што је најважније – слободно се може читати, тј. приступати изворном коду програма).“ [Madey et al, 2002].

„Пројекти развоја „open-source“ софтвера базирају се на Internet-заснованим заједницама програмера (софтвер девелопера) који волонтерски сарађују како би развили софтвер који је потребан њима или њиховим организацијама.“ [Hippel&Krogh, 2003] Најчешће су корисници „open-source“ софтвера истовремено и учесници у његовом развоју. Покретачи новог софтвера, мотивисани својим потребама, најчешће касније постају и руководиоци пројеката развоја тог софтвера [Hippel&Krogh, 2003] [Gasser&Ripoche, 2003].

Основне карактеристике „open-source“ развоја софтвера су [Mockus et al, 2000]:

- OSS системе развија потенцијално велик број (стотине, хиљаде) волонтера
- Радни задаци нису додељени, већ људи сами узимају да раде оно што изаберу
- Нема експлицитног дизајна система или чак детаљног дизајна
- Не постоји пројектни план, распоред рада или листа резултата који треба да буду испоручени,

Према [10] основне карактеристике „open-source“ развоја софтвера су:

- Главни доприноси долазе од учесника који нису плаћени
- Висок ниво дистрибуције учесника у развоју
- Слаба формализација у дизајну, планирању пројекта и менаџменту
- Отворен изворни код и контрола и ревизија пројекта од стране заједнице

Неки од најпознатијих „open-source“ софтвера, алата и језика су [Madey et al, 2002] [Gasser&Ripoche, 2003]:

- Web browser (Mozilla)
- Web server (Apache, iPlanet/Netscape)
- E-Mail server (Sendmail)
- Оперативни систем (Linux, BSDUnix).
- Алати за рад у канцеларији (OpenOffice - рад са текстуалним документима, унакрсним табелама, презентацијама)
- Програмских Језици (Perl, Java, Python, GCC, Tk/TCL).

У емпиријским истраживањима [Emanuel et al, 2011], успешним „open-source“ софтвером сматра се софтвер који има више од 100.000,00 преузимања („download“) од стране корисника.

Неке од предности „open-source“ развоја софтвера [Wahyudin &Tjoa, 2007] [Raymond, 1999]:

- Брзи развој и масивни провера/ревизија (massive peer review)
- Флексибилност у коришћењу и изменама изворног кода ради интереса корисника
- Развој ниског нивоа трошкова и трансфера технологије
- Наслеђивање развоја (Developer inheritance) и коришћење референци имплементације које помажу креирању стандарда.

Проблеми „open-source“ развоја софтвера представљени су кроз низ емпиријских истраживања која се односе на појединачне аспекте:

1. Аспект друштвеног умрежавања учесника у „open-source“ развоју, описан кроз теорију друштвених мрежа, базирану на теорији графова [Madey et al, 2002] [Crowston&Howison, 2003]
2. Аспект дистрибуираности у колективној пракси и одговарајућих приступа и метода за успешну реализацију [Gasser&Ripoche, 2003]
3. Иновација у интеграцији приватно и јавно мотивисаних активности учесника у развоју „open-source“ софтвера [Hippel&Krogh, 2003]
4. Мотивација учесника „open-source“ развоја софтвера и њихова ефикасност у оквиру иновативних активности и доприноса [Kogut&Metiu, 2001]
5. Фактори успеха и метрике успеха „open-source“ пројеката развоја софтвера, а посебно које се односе на метрике, које се односе на модуларност „open-source“ софтвера [Emanuel et al, 2011]
6. Координација активности у оквиру „open-source“ развоја софтвера и предности у односу на традиционални развој софтвера [Mockus& Herbsleb, 2002]
7. Размена знања у оквиру „open-source“ развоја софтвера [Sowe et al, 2009]

Посебно је детаљно описан конкретан „open-source“ развој Apache Servera [Mockus et al, 2000], уз илустрацију процеса, проблема и решења проблема у развоју. Емпиријско истраживање засновано је на хипотезама које су потврђене у складу са подацима који се односе на процес развоја Apache Servera. Развој Apache Servera почела је 1995. године, када је група од 8 програмера удружила снаге како би унапредила раније започет пројекат NCSA организације. Даље укључивање нових чланова реализовано је гласањем постојећих чланова. Такође, гласање је коришћено и приликом укључивања измена у постојећем програмском коду. Да би се неки нови члан прикључио тиму, морао је најпре дати допринос у пробном периоду од најмање 6 месеци. „Активности у развоју укључивале су проналажење проблема, утврђивање да ли ће волонтер радити на решавању проблема, идентификација решења, развијање и тестирање кода локално, презентовање промена кода језгру тима AG ради ревизије, слање кода и документације у репозиторијум. Овај процес се некад одвија у више итерација, али се настоји да промене које су потребне да би се решио појединачни проблем буду извршене у једној итерацији“ [Mockus et al, 2000]. У оквиру рада AG групе и волонтера који су кандидати за чланове групе, коришћени су разни алати за координацију рада: BUGDB – за евидентирање проблема, USENET newsgroup – за размену новости. Искуства из развоја Apache Servera указују на следеће потврде почетних хипотеза, а које говоре о условима за успешност OSS пројеката:

1. У развоју OSS софтвера, језгро од 10-15 људи развија око 80% процената софтвера.
2. Уколико је софтвер сложен и не може његов развој бити управљан и реализован од стране 10-15 људи, креирају се помоћни пројекти са одговарајућим језгром учесника.
3. Језгро тима развија нове функције софтвера, али нема довољно времена да се бави утврђивањем и поправком грешака. То треба да раде учесници из шире групе волонтера, који не припадају језгру.
4. У оквиру OSS развоја, програмери који учествују у развоју су најчешће и искусни корисници истих софтверских решења и доменски експерти, тако да лако могу да дефинишу захтеве и проблеме, а одмах и да их реше. Из тог разлога, број грешака (густина грешака) у OSS развоју софтвера је мања, у односу на комерцијалне софтвере. Такође, из тог разлога и поправке и нове верзије софтвера су много брже доступне корисницима.

### 2.1.3. АНАЛИЗА ПОСТОЈЕЋИХ ТЕХНОЛОШКИХ РЕШЕЊА

Развојна окружења за софтверско инжењерство предмет су истраживања и стандардизације (нпр. стандард ISO/IEC 15940:2006 Information Technology—Software Engineering Environment Services). У оквиру дистрибуираног развоја софтвера, посебно су унапређена постојећа и креирана нова развојна окружења која треба да омогуће тимски рад у развоју софтвера у дистрибуираном окружењу.

#### 2.1.3.1. Колаборативна развојна окружења и алати за подршку тимском раду у дистрибуираном развоју софтвера

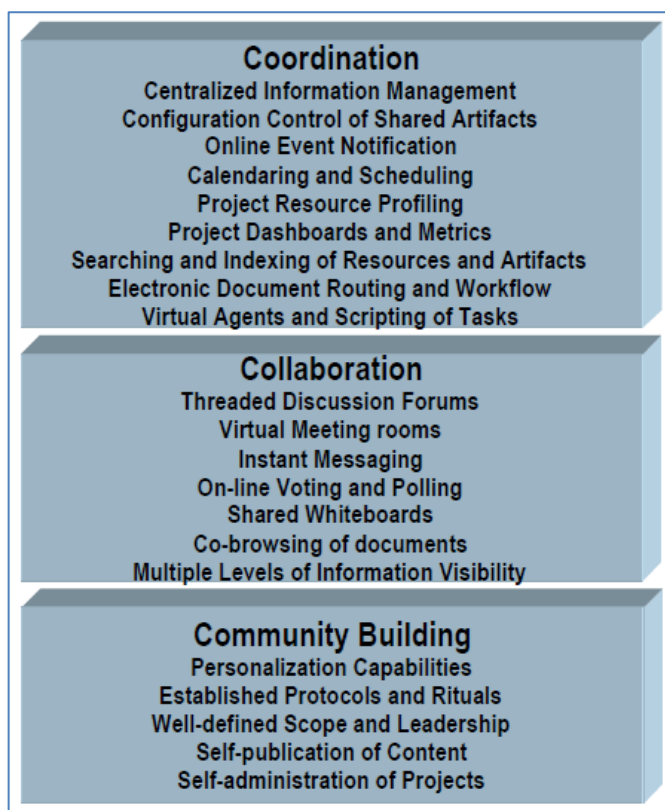
Проблеми дистрибуираности не морају да се односе само на географску дистрибуцију, већ и у оквиру исте локације могу наступити због тимског рада у условима смањених могућности непосредне комуникације. [Herbsleb&Moitra, 2001] Идеални услови за рад би омогућили „да локације максимално независно функционишу, уз једноставну, флексибилну и ефективну комуникацију између локација“ [Herbsleb&Moitra, 2001].

Решавање проблема дистрибуираног развоја софтвера од стране универзитетске заједнице и софтверских фирми резултовао је у креирању различитих софтверских алата који пружају подршку појединим аспектима у дистрибуираном развоју софтвера, а првенствено повећању квалитета комуникације у тимском раду. Такви алати припадају категорији Groupware и Task Manager софтвера. Специфичност алата за подршку колаборативном развоју софтвера огледа се у потребним карактеристикама. Ови специфични алати треба да омогуће [Sengupta et al, 2006]:

- Да учврсте колаборацију у дистрибуираном развоју,
- Управљање апликационим знањем, како би се олакшала миграција, очување и размена знања у вези апликације
- Омогућавање тестирање софтвера у дистрибуираном окружењу, нпр. тестирање јединица на удаљеним локацијама где тест подаци не могу бити директно доступни због приватности података и проблема са репликацијом
- Олакшавање интеграције модула који су развијени на различитим локацијама и редукација грешака интеграције
- Подршка проблемима процеса и метрика, која треба да омогући да се утврди које методологије развоја да се користе, који подаци и метрике да се прикупљају када је развој дистрибуиран,
- Подршка свим фазама развоја софтвера (спецификација захтева корисника, дизајн, кодирање, тестирање)

„Колаборативно развојно окружење („Collaborative development environment“-CDE систем) је виртуални простор где учесници пројекта, често раздвојени временом и простором, могу да се сусретну, размењују и деле, дискутују, преговарају, меморишу и генерално раде заједно како би извели до краја неки задатак, најчешће да креирају неки користан артефакт и његове пратеће објекте.“ [Booch&Brown, 2002] Према [Booch&Brown, 2002], разлика између колаборативног рада у другим областима и у развоју софтвера се разликује првенствено по тиме што учесници колаборативног развоја софтвера морају манипулисати семантички дубоким артефактима који су веома семантички дубоко повезани, а такође и по томе што су учесници софтверског развоја природно везани за Internet технологије, чиме се лакше прелази са физичке на виртуалну колокацију. Према [Booch&Brown, 2002] разлика између IDE (Integrated Development Environment) и CDE (Collaborative development environment“) је што је IDE примарно орјентисан на девелопера и њему потребну подршку, док је CDE есенцијално орјентисан на подршку потребама тима: интеракција, комуникација, координација...

Кључне карактеристике које CDE систем треба да има и да омогући су, према [Booch&Brown, 2002], приказане на слици 2.1.3.1.

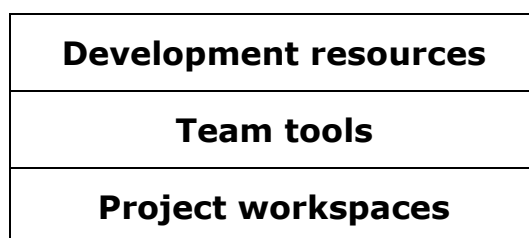


Слика 2.1.3.1. Карактеристике система за подршку колаборацији, према Booch&Brown [Booch&Brown, 2002]

Технолошка решења у области подршке колаборативном раду обухватају [Dustdar, 2004]:

- Groupware системе, тј. подршку тимском раду, а првенствено комуникацијама у тиму (instant messaging,... [Booch&Brown, 2002])
- Системе за подршку управљању знањем
- Документ менаџмент системе
- Системе за подршку пројектном менаџменту (пројектни web сајтови [Booch&Brown, 2002])
- Системе за подршку радним токовима (укључујући и тзв. Task manager, Event notification server и Issue Tracking тип софтверских алата [Booch&Brown, 2002]).

Специфичне потребе у тимском раду у оквиру дистрибуираног развоја софтвера захтевају подршку више различитих софтверских алата или њиховој интеграцији за подршку размени и интеграцији развојних ресурса, подршку тимском раду и подршку управљању пројектима. Концептуални модел таквог интегралног система представљен је на слици 2.1.3.2.



Слика 2.1.3.2. Кључни део концептуалног модела CDE (Collaborative development environment) система, према Booch&Brown [Booch&Brown, 2002]

У наставку ће бити дат приказ неких од софтверских алата који су креирани ради подршке тимском раду у дистрибуираном развоју софтвера, као и њихових карактеристика:

- Складишта и праћење промена изворног кода [Froehlich&Dourish, 2004], као што су нпр. системи:
  - CVS (Concurrent Versions System) [Cederqvist, 2006] - софтверски алат који поред складиштења изворног кода омогућава управљање конфигурацијом и подршку тимском раду, где је могуће бележити ко је мењао програмски код
  - Subversion [Pilato et al, 2004]
  - Microsoft SourceSafe и Team Foundation Server
- Software configuration management системи [Mahapatra& Das, 2012] [Booch&Brown, 2002]
- Дељење екрана [Cheng, 2004] – софтверски алат, као што је VNC (Virtual Network Computing) омогућава увид у радни екран другог члана тима
- Посебни правци развоја укључују проширења развојних окружења функцијама које омогућавају подршку тимском раду, као што је модул Jazz као проширење Eclipse IDE развојног окружења [Cheng, 2004] .
- Подршка размени порука (e-Mail и IM – instant messenger) такође може бити интегрисана у оквиру развојног окружења [Cheng, 2004] .
- Размена мултимедије у окружењу видео-конференција (аудио и видео синхронизација), систем CAIRO базиран на технологији агената [Pena-Mora et al, 2000]
- Процесно-орјентисани (process aware) систем CARAMBA [Dustdar, 2004] је објектно орјентисани систем који подржава workflow management са елементима управљања знањем, заснован на Јава технологији и релационим базама података, вишеслојне архитектуре.
- Алати за подршку континуираној координацији [Redmiles et al, 2007] – базирани су на мониторингу рада у развојном окружењу (нпр. систем Palantir – над Eclipse развојним окружењем), који користе Event Notification Server( нпр.систем YANCEES [Redmiles et al, 2007]) за размену података о утврђеним променама које је мониторинг софтвер утврдио. Такође, посебни алати визуално представљају и социо-технолошке везе између учесника у дистрибуираном развоју (нпр. систем Ariadne који се такође надовезује на Eclipse развојно окружење [Redmiles et al, 2007]).
- Web базирани системи за размену података у оквиру једног пројекта који омогућава праћење дневних промена и испорука, mailing листе, праћење грешака, огласне табле за обавештења и форуме, архивирање фајлова, бекап фајлова и администрацију, нпр: [Google Code], [Git Hub], [Launch Pad]
- Комбинована развојна окружења, где се локално на рачунару инсталира софтверски алат за контролу изворног кода, а путем web апликације реализује се регистрација корисника, пројеката и централна комуникациона локација пројеката. Један од таквих комбинованих система користи се за развој open-source "R (GNU-S) система" , а то је систем R-Forge [Theußl& Zeileis, 2009], базиран на Subversion клијентском систему.

Поред професионалних организација, поједина технолошка решења развијена су и у оквиру истраживања у оквиру пројеката или реализације докторских дисертација. У докторату Christian Heide Damm [Damm, 2002], представљен је нови алат Knight који омогућава објектно орјентисано моделовање у колаборативном развоју софтвера. У оквиру доктората Till Gaston Balz Bay [Bay, 2008] представљена је ORIGO платформа која је развијена за потребе хостинга дистрибуираних софтверских пројеката и омогућава интеграцију са развојним окружењима Visual Studio и Eclipse кроз посебно развијен ORIGO plug-in. У докторату Hani Ahmad Bani-Salameh [Bani-Salameh, 2011] је развијено радно окружење које интегрише алате друштвеног умрежавања и развојног окружења, који омогућавају колаборацију у реалном времену у оквиру 3D виртуалног света.



### 2.1.3.3. Алати за подршку управљању пројектима и софтверским пројектима у дистрибуираном окружењу

Постоји много комерцијално доступних софтверских пакета, намењених подршци управљању пројектима. Овакви системи омогућавају [Jurison, 1999]:

- Планирање и моделовање („Front-end planning and modeling“) – омогућавају креирање иницијалног плана и прецизирање евалуацијом алтернативних планова. Већина програма укључује методе PERT, анализу критичног пута и обезбеђује преглед и планирање коришћења ресурса
- Праћење трошкова и временског плана („Cost and schedule tracking“) – прикупљање података о трошковима и о статусу и праћење статуса у односу на план.
- Креирање извештаја – припрема извештаја о напретку у различитим формама, укључујући и извештаје и изузецима, гантограме, коришћење ресурса, податке о добијеној вредности (earned value) и трендовима. Већина алата омогућава корисницима да креирају темплејте и подешавају извештаје.
- Комуникација – додељивање задатака и добијање података о промени статуса, размена података о пројекту
- Управљање вишепројектним радом – одређивање међузависности између различитих пројеката, креирање вишепројектних извештаја и омогућавање да менаџери одреде утицај појединачних пројеката на пословање.

Према [Jurison, 1999], произвођачи популарних софтверских пакета у области управљања пројектима су: Microsoft (Project), Primavera (Project Planner, Sure Track), Artemis (Project View), Scitor (Project Scheduler), Computer Associates (Super Project). У оквиру [Fox & Spence, 1998] приказано је истраживање о најчешће коришћеним алатима у управљању пројектима. Истраживање је спроведено користећи упитнике који су послати за око 1000 учесника. Учесници су већином пројектни менаџери запослени у области софтверске индустрије, а мањи број у области индустрије која се односи на хардвер и трговину. Навели су преко 70 различитих алата, али најчешће коришћени су: Microsoft Project, Primavera Project Planner и ABT Project Workbench.

Према [Jurison, 1999], произвођачи константно укључују нове особине и додају нове функционалности раније развијеним софтверским пакетима за подршку пројектном менаџменту. Новија решења подржавају Web модуле, дозвољавајући пројектном менаџеру и члановима тима да пошаљу податке и извештаје на интранет систем компаније и да деле податке са клијентима у extranet окружењу.

Посебно за потребе подршке софтверским пројектима, у оквиру CASE (Computer Aided Software Engineering) алата, као и развојних окружења, подржано је [Jurison, 1999]:

- Контрола конфигурације (configuration control) – способност да се контролишу и прате промене, имплементирана је контрола верзија
- Процена софтвера (software estimating) – укључени су модели за процену величине софтвера, трошкова и процену временског плана (schedule estimating)

Неки од алата који омогућавају подршку развоју софтвера у дистрибуираном окружењу, али и управљању пројектима таквог развоја су:

- Систем „AUGUR“ [Froehlich & Dourish, 2004] – омогућава: 1) мониторинг активности у оквиру дистрибуираног софтверског пројекта; 2) претраживање података о активностима кроз простор и време, увидом у историју активности. Основне карактеристике система: 1) подршка приказу података о процесима и активностима који се извршавају у оквиру развоја процеса (конкурентност анализе података и развоја), али и ретроспективи података и припреми података за одлучивање ради унапређења процеса развоја; 2) подржава анализу програмског кода за време реализације (он-лине); 3) интероперабилност са другим сродним софтверским решењима, 4) подршка визуализацији података, која омогућава да корисник може да уочава обрасце, правила, насупрот системима аутоматског резоновања, који омогућавају директну анализу података и издвајање закључака.

- Систем „CODESAW“ [Gilbert& Karahalios, 2007] – визуализација као подршка дистрибуираном тиму у развоју софтвера кроз визуални приказ софтверског репозиторијума (изворног кода) и елемената управљања пројектима, тачније комуникације (из пројектног маил-а) која је претходила неком артефакту и која тај артефакт боље описује.
- У оквиру [Wahyudin & Tjoa, 2007] описан је концепт система који би био базиран на праћењу догађаја (event) које би се извршавало на основу мулти-агентског система прикупљања података у дистрибуираном окружењу.

## 2.2. МЕТРИЧКИ ЗАСНОВАНО УПРАВЉАЊЕ СОФТВЕРСКИМ ПРОЈЕКТИМА

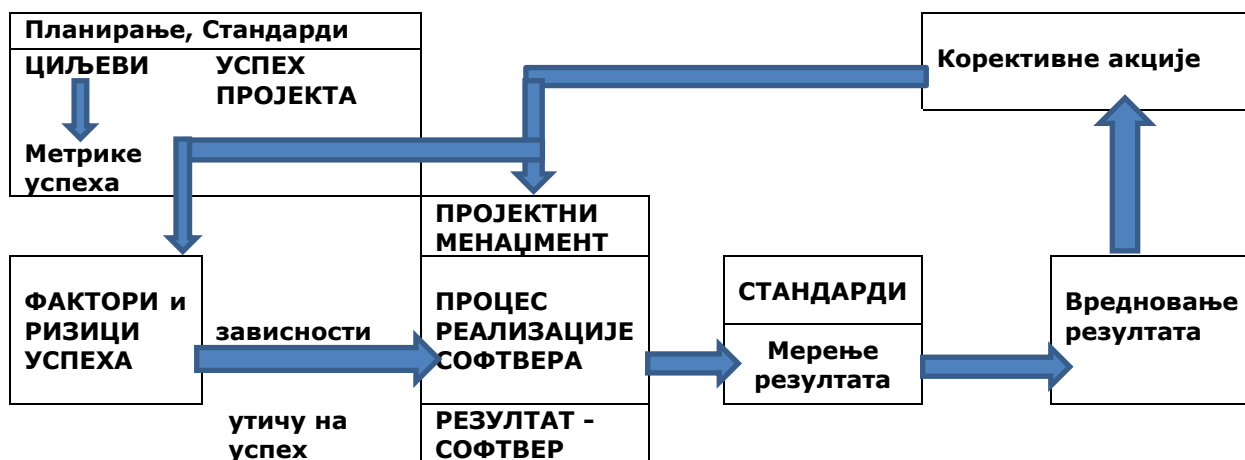
### 2.2.1. Модел анализе постојећих истраживања

У табели 1 приказан је општи оквир истраживања у области метрички заснованог управљања софтверским пројектима. Димензија планирања успеха-квалитета односи се првенствено на избор методологије развоја софтвера и управљања софтверским пројектом, као и усклађеност са стандардима у овој области. Мониторинг успеха врши се у току процеса развоја софтвера, користећи одређене индикаторе-метрике, а вредновање успеха и квалитета врши се након завршетка процеса развоја.

Табела 2.2.1.1. Општи оквир анализе истраживања у области метрички заснованог управљања софтверским пројектима

ПЛАНИРАЊЕ УСПЕХА/КВАЛИТЕТА	МОНИТОРИНГ УСПЕХА/КВАЛИТЕТА (мерење перформанси и подршка одлучивању)	ВРЕДНОВАЊЕ УСПЕХА/КВАЛИТЕТА
Процеса	Процеса	Процеса
Производа	Производа	Производа

На основу теоријских поставки које се односе на дефиницију успеха пројекта, контролу пројекта и софтверске метрике, општи модел анализе постојећих истраживања у области метрички заснованог управљања софтверским пројектима приказан је сликом 2.2.1.1.



Слика 2.2.1.1. Модел анализе постојећих истраживања у области метрички заснованог управљања софтверским пројектима

У наставку ће бити приказани поједини аспекти стандардизације, мерења и вредновања квалитета и перформанси процеса развоја софтвера и софтверског производа.

## 2.2.2. Димензије успеха софтверског пројекта

### 2.2.2.1. Успех пројекта у области информационих технологија

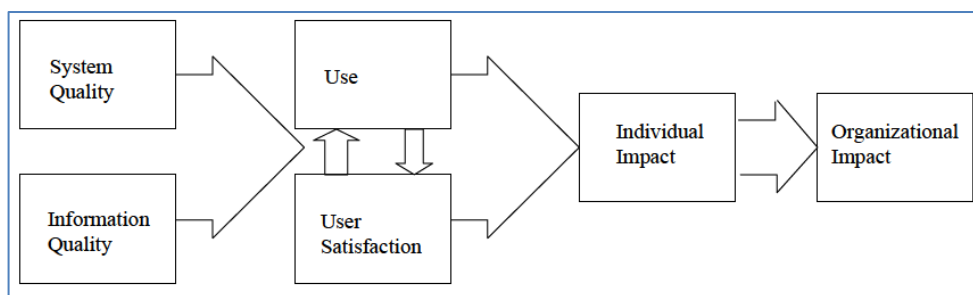
Према Project Management Institute [Vaccarini, 1999], успех пројекта не може бити једнозначно дефинисан, јер различити учесници у процесу могу имати различите перцепције [Jianga et al, 2002] [Pereira et al, 2008], па су критеријуми често базирани на субјективним проценама [Wohlin, 2000]. Vaccarini [Vaccarini, 1999] наглашава да је важно утврдити и договорити у оквиру тима критеријуме успеха пројекта, како би реализација пројекта била фокусирана. Vaccarini [Vaccarini, 1999] издваја као кључне елементе успеха пројекта: успех пројектног менаџмента, успех резултата пројекта, као и задовољство корисника, а предлаже логички оквир за дефинисање успеха пројекта у узрочно-последичној вези задатака кроз дефинисање: циља пројекта, сврхе пројекта, излаза (резултата) и улаза(ресурса).

Истраживање концепта успеха пројекта у смислу дефиниције и разматрања свих димензија, као и могућности метричке заснованости утврђивања нивоа успеха пројекта приказано је у [Shenhar et al, 2001]. У наведеном истраживању успех пројекта је дефинисан кроз четири димензије: ефикасност пројекта, утицај на корисника, успех пословања, припреме за будућност. Такође, дефинисане су и метрике којима се може утврдити ниво успеха пројекта у односу на сваку од наведене четири димензије.

Табела 2.2.2.1.1. Димензије успеха и одговарајуће мере, према Shenhar et al [Shenhar et al, 2001]

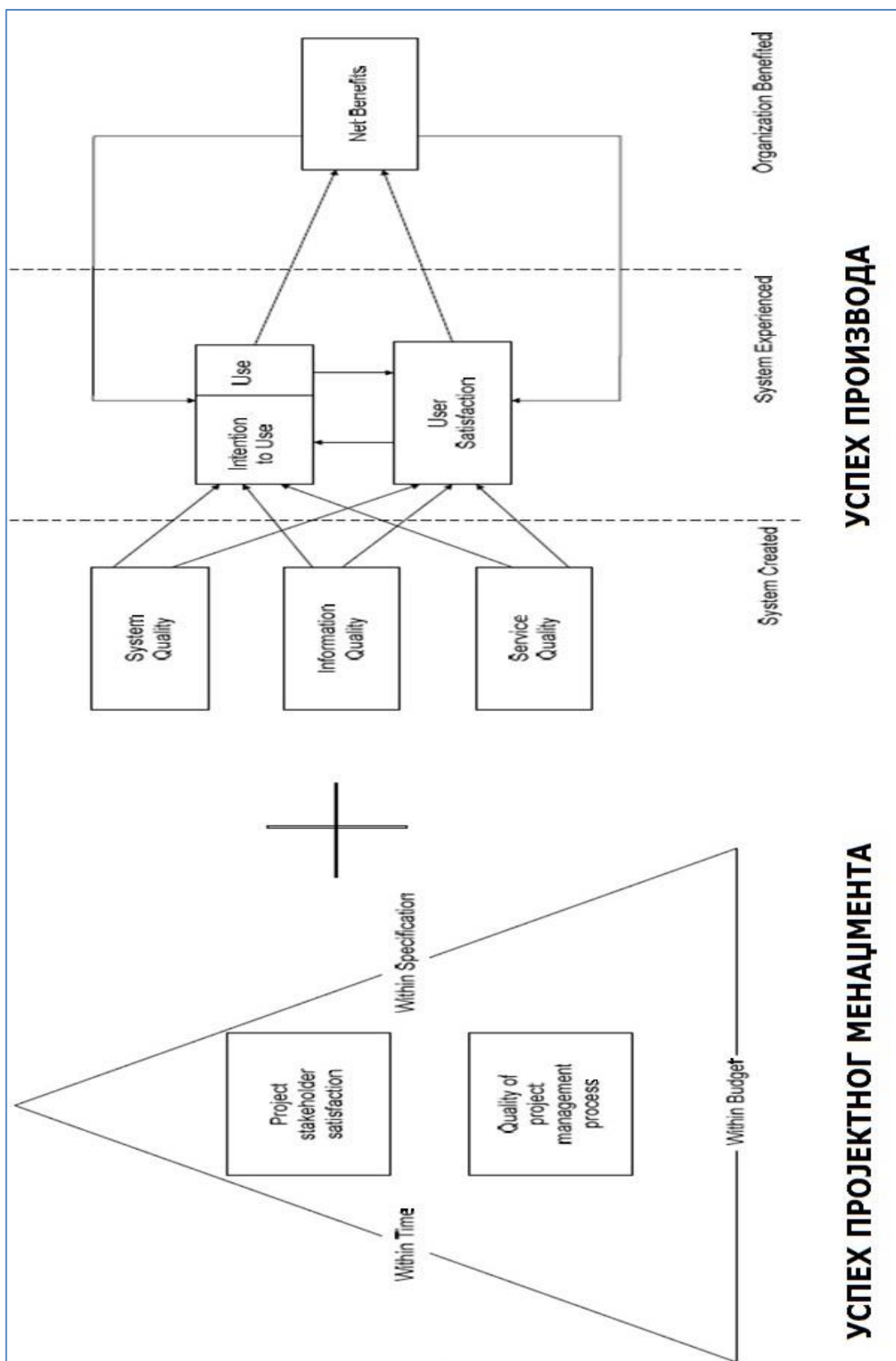
Димензија успеха	Мера
1. Ефикасност пројекта	Усклађеност са циљем који се односи на временски план – распоред
	Усклађеност са циљем који се односи на буџет
2. Утицај на клијента	Усклађеност са функционалним перформансама
	Усклађеност са техничким спецификацијама
	Задовољавање корисникових потреба
	Решавање проблема корисника
	Корисник користи производ
	Задовољство корисника
3. Пословни успех	Комерцијални успех
	Креирање великог удела тржишта
4. Припрема за будућност	Креирање новог тржишта
	Креирање нове производне линије
	Развој нове технологије

С обзиром да успех пројекта дефинишемо, према Vaccarini [Vaccarini, 1999] као успех пројектног менаџмента и производа, модел критеријума успеха реализованог информационог система према основном DeLone & McLean моделу [DeLone& McLean, 1992] може бити представљен сликом 2.2.2.1.1.:



Слика 2.2.2.1.1. Успех информационог система према DeLone & McLean моделу [DeLone& McLean, 1992]

Према проширеном DeLone & McLean моделу [Delone& McLean, 2003], мерење успеха пројекта у области информационих система може се представити сликом 2.2.2.1.2. [Westhuizen& Fitzgerald, 2005]:



Слика 2.2.2.1.2. Успех пројекта = успех пројектног менаџмента и производа, према Westhuizen& Fitzgerald [Westhuizen& Fitzgerald, 2005]

На основу анализе литературе која је урађена у оквиру рада [Crowston et al, 2003], критеријуми успеха у односу на претходно приказани DeLone & McLean модел могу се детаљније приказати кроз одговарајуће изворе података за мерење (индикаторе) нивоа успеха по појединим карактеристикама:

Табела 2.2.2.1.2. Индикатори мерења успеха пројекта на основу DeLone & McLean модела, према Crowston et al [Crowston et al, 2003]

МЕРА УСПЕХА	ИНДИКАТОРИ
Квалитет система и информације	<ul style="list-style-type: none"> <li>Квалитет кода (нпр. разумљивост, комплетност, концизност, портабилност, конзистентност, могућност одржавања, могућност тестирања, могућност коришћења, поузданост, структурираност, ефикасност)</li> <li>Квалитет документације</li> </ul>
Задовољство корисника	<ul style="list-style-type: none"> <li>Рангирање од стране корисника</li> <li>Мишљења прикупљена путем mailing листа</li> <li>Истраживања корисника</li> </ul>
Коришћење	<ul style="list-style-type: none"> <li>Коришћење (нпр. Debian-ово такмичење у популарности)</li> <li>Број корисника</li> <li>Број преузимања (downloads)</li> <li>Укљученост у дистрибуције</li> <li>Популарност или број прегледа на страници са информацијама</li> <li>Зависности пакета</li> <li>Поновна искористивост кода</li> </ul>
Индивидуални и организациони утицаји	Економске и друге импликације

За пројекте у области информационих технологија, односно информационих система, Аткинсон [Atkinson, 1999] предлаже „четвороугаоно усмерење за успех“ (Square Route) који се састоји из 4 области критеријума успеха:

Табела 2.2.2.1.3. Димензије успеха и мерење за пројекте у области информационих система према Аткинсону [Atkinson, 1999]

ОБЛАСТИ			
Гвоздени троугао	Информациони систем	Користи за организацију	Користи за заинтересоване стране (stakeholder)
(Процес развоја)	(Развијени систем)	(Корисност система)	(Корисност система)
КРИТЕРИЈУМИ			
Трошкови	Могућност одржавања	Унапређена ефикасност	Задовољни корисници
Квалитет	Поузданост	Унапређена ефективност	Организациони утицај и утицај на животну средину
Време	Валидност	Пораст профита	Лични развој
	Квалитет информација	Стратешки циљеви	Професионално учење
	Коришћење информација	Организационо учење	Профит, обезбеђивање капитала
		Смањени шкарт, губици, отпад	Економски утицај на друштво

У оквиру истраживања [Jugdev&Müller, 2005] дат је ретроспективни преглед радова који се баве дефинисањем концепта успеха пројекта. Извршена је категоризација радова у хронолошком смислу на:

1. период „имплементација пројекта“ (1960-1980) - кроз описне и метрички засноване једноставне мере које су се сводиле на гвоздени троугао – време, трошкови и спецификација захтева корисника која је задовољена.

2. период – листе „критичних фактора успеха“ (1980 – 1990) – дефинисане су листе фактора успеха, често субјективно и на основу једне или мањег броја студија случаја
3. период – оквири листа критичних фактора успеха (CSF frameworks) (1990-2000) – груписање фактора успеха и одређивање приоритета
4. период – стратегијски пројектни менаџмент (21 век) – реализација пројекта као део остваривања стратегијских циљева организације, дефинисање критеријума успеха, као и допринос успешности пројекта померен је на виши менаџмент и заинтересоване стране, спонзоре, с обзиром да пројекти доприносе организацији и окружењу, остварењу стратегијских циљева, а виши менаџмент обезбеђивањем ресурса и интересом за пројекат доприноси његовом успеху.

Сумарни приказ критеријума успеха производа приказан је у [Berntsson-Svensson&Aurum, 2006]:

- Задовољство корисника
- Производ који функционише
- Производ у складу са очекивањима корисника
- Производ који је у складу са дефинисаним захтевима корисника
- Производ који је коришћен од стране корисника

У оквиру истраживања [Crowston et al, 2003], описани су критеријуми успеха Open Source пројеката:

1. Корисник (задовољство, укљученост у процес развоја и одржавања)
2. Производ (у складу са захтевима, квалитет кода, преносивост, доступност)
3. Процес (активности, припадност процеса, решавање грешака, време, старост)
4. Девелопери (програмери) (укљученост, разноликост и измене, задовољство)
5. Коришћење (компетиција, број корисника, број преузимања - download)
6. Препознавање (реферисање, пажња и препознавање, „spin-off“)

Успех пројекта је у односу на ставове пројектних менаџера [Procaccino& Verner, 2006] истраживан са аспеката који се најчешће вреднују након завршетка пројекта:

- Захтеви корисника/клијента су задовољени завршеним системом
- Пројекат је завршен у оквиру предвиђеног буџета
- Пројекат је завршен на време
- Пројекат је завршен
- Систем се састоји од програмског кода који је конзистентан и добро тестиран, има минимум грешака, поуздан је и стабилан
- Систем је лак за коришћење

У оквиру овог истраживања, већина пројектних менаџера није повезала завршетак пројекта на време и у оквиру буџета са успехом пројекта, већ се успех пројекта односио на карактеристике квалитета софтверског решења:

- Стабилност система
- Безбедност система
- Код је погодан за одржавање
- Код је у складу са ИТ стандардима
- Да ли су корисници задовољни системом
- Да ли се систем користи
- Систем додаје вредност (за организацију)
- Корисници су задовољни и желе додатне опције и даљу сарадњу

## **2.2.2.2. Фактори успеха и ризици софтверских пројеката**

### **2.2.2.2.1. Фактори успеха софтверских пројеката**

Према [Jurison, 1999], критични фактори успеха (Critical Success Factors - CSF) представљају елементе који морају бити добри („things that must go right“), да би пројекат успео. Из тог разлога мора им се придавати посебна и континуална пажња од стране менаџмента како би се обезбедио успех. У мноштву истраживања критичних фактора успеха пројеката, релативност значаја издвојених фактора успеха јавља се у

односу на различите типове пројеката. Из тог разлога, генерички (заједнички, општи) фактори успеха могу представљати основ за дефинисање специфичних скупова фактора, који одговарају потребама специфичних типова пројеката. Предност дефинисања CSF је у томе што помажу пројектним менаџерима да се фокусирају на оно што је битно, дефинишу приоритете и омогућавају бољу комуникацију и разумевање пројектног тима.

Према [Jurison, 1999], критични фактори успеха софтверских пројеката су:

- Јасно дефинисани задаци (комплетна и јасна дефиниција пројектних задатака, обухвата и активности које треба да се ураде)
- Подршка вишег менаџмента (посвећеност пројекту од стране вишег менаџмента које се исказује обезбеђивањем потребних ресурса, давање довољног ауторитета пројектном менаџеру и подршка у време кризе)
- Адекватан буџет
- Реалистични временски план - превише оптимистичан план може дати нереална очекивања
- Учешће клијента/корисника – укључивање клијента-корисника посебно у фази планирања, даје боље и реалистичније дефиниције захтева у вези система и бољу посвећеност корисника пројекту
- Liderство пројекта – почиње избором пројектног менаџера и менаџмент тима; ефективно лидерство је потребно како би се тим одржао фокусираним и мотивисаним у току пројекта.
- Извештавање о пројекту (Project reviews) – редовно и фреквентно извештавање о пројекту обезбеђује видљивост напретка и проблема. Представља алат за дељење визије, мотивисања чланова тима и обезбеђивања комуникација.
- Контрола и управљање променама – процес контролисања и мониторинга променама је посебно значајно код комплексних пројеката
- Комуникације – добра комуникација између чланова тима и свих интересних страна је потребна.
- Решавање проблема – у сваком пројекту јављају се проблеми и успех пројекта зависи од ефективних менаџнизама за предвиђање и решавање проблема.

Према [Berntsson-Svensson&Aurum, 2006], истраживања у области успеха пројектног менаџмента су значајније заступљена у односу на истраживања успеха производа. Систематизацијом других истраживања која се односе на факторе који утичу на успех софтверских пројеката, као и у оквиру упитника који је слат организацијама и професионалцима софтверске индустрије у оквиру [Berntsson-Svensson&Aurum, 2006], утврђено је да су то следећи фактори:

Табела 2.2.2.2.1.1. Фактори успеха софтверских пројеката, према Berntsson-Svensson&Aurum [Berntsson-Svensson&Aurum, 2006]

КАТЕГОРИЈА	ФАКТОРИ УСПЕХА СОФТВЕРСКОГ ПРОЈЕКТА
Технички	Радни производ, дизајн производа, одговарајући ресурси пројекта
Психолошки	Технички изазови, Без притисака са временским ограничењима, Укључивање корисника и клијента, Искусност и способност пројектног менаџера, ауторитет пројектног менаџера, добар пројектни менаџер
Организациони	Високе перформансе пројектног тима, Укљученост и подршка врховног менаџмента клијентске организације, минимизиран обухват посла, пројектни тим, управљање променама, посвећеност и подршка спонзора, довољно времена за дефинисање и издвајање захтева корисника, добар распоред рада, добре процене трајања активности и распоређивања рада, подршка од стране пројектног менаџмента прековременом раду („working long hours“), способан тим, дефинисана методологија, промена пројектног менаџера, укључивање нових запослених како би се реализовао пројекат, идентификација ризика пре почетка пројекта
Информациони	Јасни пословни циљеви (пословни план и визија), јасни и чврсто одређени основни захтеви, ефективна комуникација, дефиниција пројекта, пројектни план, добро дефинисани (јасни и разумљиви) и комплетни захтеви корисника, повратне информације од пројектног менаџера, добра визија пројекта, добро дефинисан обухват пројекта, комуникационе способности



	пројектног менаџера, добро разумевање проблема клијента од стране пројектног менаџера
--	---

Организација Standish Group је спровела 1994. године истраживање о успеху софтверских пројеката и сачинила 1995. године тзв. CHAOS REPORT [CHAOS, 1995]. Истраживање је извршено са циљем да се утврде основне детерминанте неуспеха софтверских пројеката, фактори који доприносе неуспеху пројекта и елементи који могу да помогну у смањењу неуспеха пројекта. Основу методологије истраживања чине упитници који су послати ИТ извршним менаџерима (IT executive managers) у вези софтверских апликација. Популацију у истраживању чини 365 учесника који су учествовали у одговарању на питања у вези укупно 8,380 софтверских апликација. Резултати из 1994. године су:

1. Успешних пројеката - са захтеваним особинама, у оквиру предвиђеног времена и у оквиру предвиђених трошкова) – 16.2%
2. Пројекти са изазовом („challenging“) – пројекат је завршен и функционалан, али је премашен буџет, недостају неке особине софтвера или је „пробијен“ временски рок израде – 52,7%
3. Прекинути пројекти („cancelled“ или „failed“) – пројекти чије извршавање је прекинуто у неком тренутку – 31,1%

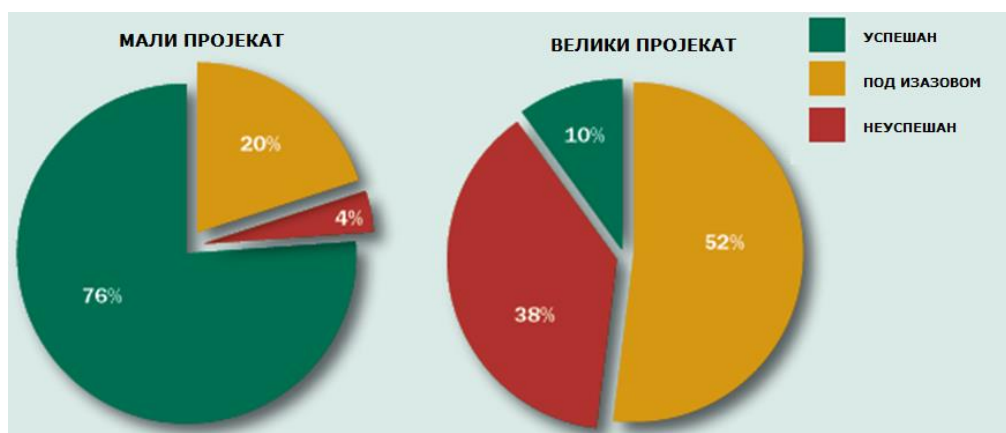
За наведена истраживања по годинама дат је списак фактора успеха софтверских пројеката (на основу [Eveleens& Verhoef, 2010]), табела 1.2.2.4.

Табела 2.2.2.2.1.2. Упоредни приказ фактора који доводе до успеха пројекта по годинама, на основу истраживања Standish group, према Eveleens& Verhoef [Eveleens& Verhoef, 2010]

<b>P6</b>	<b>1994</b>	<b>1998</b>	<b>2000</b>	<b>2004</b>	<b>2010</b>
1	Укљученост корисника	Укљученост корисника	Подршка оперативног менаџмента	Укљученост корисника	Подршка оперативног менаџмента
2	Подршка оперативног менаџмента	Подршка оперативног менаџмента	Укљученост корисника	Подршка оперативног менаџмента	Укљученост корисника
3	Јасна изјава о захтевима корисника	Мање пројектне тачке контроле	Компетентно особље	Мање пројектне тачке контроле	Јасни пословни задаци
4	Правилно планирање	Компетентно особље	Мање пројектне тачке контроле	Вредно, фокусирано особље	Емоционална зрелост
5	Реалистична очекивања	Власништво	Јасна визија и задаци	Јасна визија и задаци	Оптимизацион и оквир
6	Мање пројектне тачке контроле				Агилни процес
7	Компетентно особље				Експертиза пројектног менаџмента
8	Власништво				Способни ресурси (запослени)
9	Јасна визија и задаци				Извршавање, тј. контрола
10	Вредно, фокусирано				Алати и инфраструктур

	особље				a
--	--------	--	--	--	---

На успех пројекта утиче величина пројекта. Резултати из 2012. (према [CHAOS, 2013]) указују значајно већи успех мањих пројеката у односу на велике пројекте, као што је приказано на слици 2.2.2.2.1.1.



Слика 2.2.2.2.1.1. Однос успеха малих и великих софтверских пројеката, према истраживању 2012. године од стране Standish group, према CHAOS извештају из 2013. године [CHAOS, 2013]

Поједини научници и стручњаци [Eveleens & Verhoef, 2010] критички анализирају резултате организације Standish Group, с обзиром на методологију истраживања и доступност изворних података на основу којих су извршене статистичке анализе. Ипак, резултати истраживања организације Standish Group и даље представљају релеванан основ за даља истраживања у овој области.

Организација Standish group је 2013. године креирала документ „CHAOS Manifesto 2013“ [CHAOS, 2013], којим даје препоруке за унапређење процеса управљања пројектима у области информационих технологија, а посебно у области развоја софтвера, у оквиру тзв. малих пројеката. Према [CHAOS, 2013], фактори који утичу на успех малог софтверског пројекта су приказани редоследом важности:

1. *Подршка извршног менаџмента (Executive management support)*, односно извршног спонзора, који је лично заинтересован за реализацију пројекта, фокусиран на једноставну јасну визију, да има времена да буде посвећен пројекту, да брзо доноси одлуке, да је едукован у области управљања пројектима, да може да визуализује и мери напредак пројекта, да уме да преговара са свим учесницима у вези обухвата и осталих елемената пројекта, да уме да мотивише чланове тима, препозна и награди њихов труд.
2. *Укљученост корисника (User involvement)* – важно је укључити кориснике који познају одговарајућу област, остварити добру међуљудску сарадњу, добре системе комуникације, брзе одговоре на проблеме и честе испоруке
3. *Оптимизација (Optimization)* – с циљем да се уложи што мање рада и да се брзо испоручи резултат рада. Мали пројекти могу јасније да дефинишу мањи обухват посла, прецизније да процене трајање и трошкове. Најоптималнији је тим од максимално 6 људи који имају различита специјализована знања и задужења и имају позитиван став према свом ангажовању на пројекту. У оквиру мањих пројеката лакше се дефинишу и реализују очекивања, стиче се боље поверење корисника јер се успешно завршавају, честим испорукама олакшава се навикавање корисника на нова решења, процена ризика се лакше ради на нивоу сваке главне функције.
4. *Вешти, способни ресурси (Skilled resources)*, односно људи – прави људи који раде праве ствари у право време. Битне су компетенције и способности да се изврши посао, диверзитет квалификација у тиму. Избегавати „superstar“ синдром где појединци могу нарушити кооперацију у тиму. Важно је јасно одредити улоге учесника, управљати мотивацијом чланова тима, комуникацијом и кохезијом, као и обуком и дељењем искустава са једног за друге пројекте.

5. *Експертиза у области управљања пројектима (Project management expertise)*, односно управљања процесима, где је најважније управљање сарадњом заинтересованих страна и тима, добре процене и одлучивање. Основне активности управљања су планирање, праћење и контролисање. Дobar пројектни менаџер уме да препозна квалитете чланова тима, да их усмерава и фокусира на циљ, да размишља аналитички, да повезује људе и добро комуницира са свима, да разуме пословне проблеме клијента, јасно разграничава шта треба да се уради (дефинисано од стране пројектног спонзора) од тога како треба да се ради и одговоран је за процес реализације.
6. *Агилни процес (Agile process)* – итеративни, честе испоруке важних делова, мањи обухват посла, мањи тимови (5-6 људи) са мањим временским интервалима за реализацију пројекта (до 6 месеци), стална интеракција, конверзација и договори људи у тиму, прилагодљивост променама, чести повратни одговори (feedback), ретроспективе у оквиру претходне фазе (позитивни и негативни аспекти), издвајање функција које ће се заиста користити и које се могу брзо и лако имплементирати и фокусирање на њих,
7. *Јасни пословни циљеви и задаци (Clear business objectives)* – усклађеност циљева и задатака пројекта са циљевима и стратегијом организације у којој се примењује – јасноћа, концизност и фокус, јасна визија, дефиниција проблема и изјава о томе шта ће се радити, извештаји о статусу на дневном и недељном нивоу. Мали пројекти имају један заједнички циљ, али и улогу у ширем контексту програма развоја. Битно је фокусирати се на имплементацију функција које највише доприносе циљевима и функционисању организације за коју се софтвер реализује. Битно је често (једном недељно према Extreme Programming методологији) испоручивати реализоване модуле или завршавати задатке (један задатак не дужи од 80 часова).
8. *Емоционална зрелост (Emotional maturity)* – односи се на емоционално окружење у којем се реализује пројекат, односно на екосистем пројекта. Потребно је препознати и управљати проблемима као што су: преамбициозност, ароганција, изостајање, а унапредити етику, комуникацију, сарадњу, једноставност, формирати групе тимова у заједницу која се бави сличним пројектима, подстицати транспарентност, повећање свесности о статусу пројекта информисањем у одговарајуће време са одговарајућим нивоом детаља, објективност, промоција квалитета.
9. *Извршавање (Execution)*, односно вођење и управљање извршавањем пројекта – битно је дефинисати и придржавати се правила, јасно дефинисати пословни проблем који треба да реши систем који се развија, дефинисати и одржавати промене захтева корисника, мерење напретка пројекта и успеха тимова,
10. *Алати и инфраструктура (Tools and infrastructure)* – алати помажу у реализацији и управљању пројектом, али се не треба превише ослонити само на алате. Посебно у малим пројектима коришћење алата за пројектни менаџмент успорава процес и повећава трошкове. Свакако, потребно је ефикасно управљати ресурсима, захтевима, финансијама и елементима управљања пројектима.

У оквиру истраживања [Procaccino & Verner, 2006] приказани су резултати спроведене анкете међу пројектним менаџерима која се односи на факторе и карактеристике успеха пројекта. Према ранијим истраживањима презентованим у [Procaccino & Verner, 2006], мотивација учесника је други најважнији фактор који доприноси успеху софтверског пројекта. Неки од мотивационих покретача су: реализација радног изазова, реализација нечег новог, имати осећај да се испоручује нешто квалитетно и осећај добро урађеног посла, осећај постигнућа, имати довољно независности да би се посао обављао креативно. У истраживању [Sukhoo et al, 2005] приказане су карактеристике запослених потребних за рад у оквиру реализације софтверских пројеката, а посебно улога „hard“ и „soft“ карактеристика.

Када је у питању задовољство корисника у области развоја софтвера, у оквиру [Boehm & Ross, 1989] предложена су решења која омогућавају реализацију става да сви учесници софтверског пројекта треба да препознају своје улоге у успеху (теорија W – „Make everyone a winner“).

## 2.2.2.2. Проблеми и ризици ИТ и софтверских пројеката

Кључни учесници у пројекту свакако су пројектни менаџер и чланови развојног тима. У оквиру [Linberg, 1999] приказани су резултати истраживања перцепција чланова развојног тима у развоју софтвера у вези за факторима ризика и узроцима неуспеха софтверских пројеката.

Према истраживањима [Spinellis, 2006], [Kocaguneli et al, 2013], неуспех пројекта се изражава:

- прекорачењем времена израде за 30% или више (према овом истраживању највише је оваквих неуспешних пројеката - 87% и то 92% малих пројеката и 86% великих пројеката),
- 2 – прекорачењем буџета за 30% или више (56%), 3 – незадовољавањем планираних карактеристика и користи од реализованог решења (45%). Под малим пројектом подразумевао се пројекат чије је трајање планирано на 12 месеци или мање. Највећи број малих пројеката је прекорачио планирано време од 50-100%, док код великих пројеката највише је био оних који су прекорачили време израде за више од 100%. Према овом истраживању најчешћи узроци за неуспех пројекта су:
  1. Недовољно добро планирање пројекта, уз недовољну улогу управљања ризицима или је пројектни план био лош
  2. Пословни проблем пројекта није довољно добро дефинисан
  3. Недостатак укључености и подршке менаџмента

Пројекти у области информационих технологија, а посебно софтверски пројекти представљају пројекте са високим нивоом ризика [Royce, 1998] [Stellman& Green, 2005] и врло често резултују неуспехом.

Величина ризика у теорији управљања пројектима се најчешће процењује на основу следећих најважнијих критеријума [Јовановић, 2006]:

- флексибилност пројекта (већа флексибилност крајњег исхода пројекта носи већи ризик)
- величина пројекта (где су кључни параметри: трајање пројекта, број људи који је укључен, обим посла, сложеност архитектуре резултата итд. – што је већа величина пројекта, већи је и ризик)
- технологија (висока технологија односи се на сложеније пројекте и потребна су специфична знања и искуства, чиме се повећава ризик пројекта)

Воеhm [Voehm, 1991] је идентификовао 10 ставки које представљају ризике софтвера који се могу односити на пројекте развоја софтвера:

- Недовољно (shortfalls) ангажованих људи (Personnel shortfalls),
- нереални временски план и буџет (Unrealistic schedules and budgets),
- развијање погрешних функција и особина (Developing the wrong functions and properties),
- развој погрешног корисничког интерфејса (Developing the wrong user interface),
- Синдром Златног тањира (Gold plating) - додавање непотребних функција и особина
- Континуални ток промена захтева
- Непотпуност (shortfalls) екстерно креираних-подешених компоненти
- Непотпуност (shortfalls) екстерно реализованих задатака
- Непотпуност performansi realnog vremena (Real-time performance shortfalls)
- Настојање да се по сваку цену примене могућности компјутерске науке (Straining computer-science capabilities)

Ризици и узроци неуспешности пројекта, према [Whittaker, 1999] су:

1. Померање распореда рада (schedule)
2. Промене у обухвату технологије, функционалности или пословног проблема

3. Повећање трошкова преко предвиђене границе услед једне или више компоненти пројекта
4. Промене кључних појединаца – пројектни спонзор, пројектни менаџер или менаџер добављача
5. Некоректно процењена трајања активности
6. Некоректне претпоставке о расположивости ресурса
7. Неадекватна додела активности одговорнима
8. Недостатак или недовољна заступљеност активности ревизије, контроле и одобравања
9. Недостатак јасно дефинисаних елемената испоруке
10. Недостатак квантитативно изражених трошкова и добити-користи
11. Извршавање управљања пројектом
12. Пројектни тим
13. Одговорност у пројекту
14. Знање и способности пројектног менаџера
15. напредак пројекта није надгледан (monitored) и корективне акције нису на време спроведене
16. Искуство и ауторитет руководиоца пројекта нису у складу са обимом и ризицима пројекта

Истраживање [Jurison, 1999] приказује основне проблеме који се јављају у управљају софтверским пројектима, који се односе на део проблема развоја софтвера (нематеријалност – неопипљивост резултата, комплексност, променљивост захтева) као и проблеме управљања пројектима (недовољно добро дефинисани циљеви и спецификације, недостатак пројектног плана, нереално планирање буџета и рокова израде).

Аткинсон [Atkinson, 1997] дефинише два типа грешака који могу настати у оквиру реализације пројекта, а посебно ИТ пројекта:

1. тип грешке – када је нешто урађено погрешно, због лошег планирања, непрецизне процене, недостатка контроле
2. тип грешке – када је нешто изостављено, заборављено и није урађено, или је урађено непотпуно, као што је некомплетан скуп критеријума за успех.

У оквиру истраживања [Nelson, 2007], представљене су најчешће грешке које изазивају проблеме у реализацији ИТ пројекта:

1. Лоша процена трајања пројекта и креирање распореда рада
2. Неефикасно управљање заинтересованим странама (stakeholders)
3. Недовољно управљање ризицима
4. Недовољно планирање
5. Смањење обима посла на обезбеђивању квалитета
6. Недовољно јак персонал и проблеми са тимом
7. Недовољно спонзорство пројекта
8. Недовољно добро дефинисана спецификација захтева
9. Необраћање пажње на политику
10. Недостатак укључености корисника у развој
11. Нереална очекивања
12. Недефинисана мотивација
13. Неизвршавање посла од стране уговорне стране, тј. подизвођача (contractor)
14. Промене обухвата посла
15. Размишљање о жељама
16. Развој орјентисан на истраживање
17. Недовољно контроле од стране менаџмента
18. Неусклађеност између чланова тима у развоју и клијента
19. Губљење времена на почетку у нејасним преговорима и договорима обухвата посла
20. Програмирање хаотичног и лоше читљивог програмског кода („code-like-hell“)
21. Хероизам појединаца
22. Додавање нових чланова тима пројекту који касни

23. Синдром сребрног метка (коришћење нове технологије први пут у пројекту и разочарење када она не решава проблеме за које је рекламирано да треба)
24. Напуштање планирања под притиском
25. Неадекватан дизајн
26. Недовољни ресурси
27. Недостатак аутоматизоване контроле изворног кода
28. Прецењене могућности нових алата и метода
29. Планирање постизања активности касније („planning to catch up later“)
30. Превише функција планирано и обећано („requirements gold-plating“)
31. Преговори и натезања
32. Мењање алата у току развоја
33. Превише функција додато од стране чланова развојног тима због испробавања и фасцинираности новим технологијама („developer gold-plating“)
34. Прерана конвергенција
35. Бука и гужва у канцеларији
36. Проблематични запослени без контроле

Према мишљењу учесника у истраживању из 1995. године [CHAOS, 1995], односно одговорима на питања, фактори који доводе до статуса пројекта „под изазовом“ су представљени наредном табелом.

Табела 2.2.2.2.1. Приказ фактора који доводе до статуса пројекта „под изазовом“ према истраживању Standish group, према Eveleens& Verhoef [Eveleens& Verhoef, 2010]

Редни број	Фактор који доводи до статуса пројекта „под изазовом“
1	Недостатак података од стране корисника
2	Некомплетна спецификација захтева корисника
3	Мењање захтева и спецификација
4	Недостатак подршке оперативног менаџмента
5	Технолошка некомпетентност
6	Недостатак ресурса
7	Нереална очекивања
8	Нејасни задаци
9	Нереални временски рокови
10	Нове технологије
11	Друго

Фактори који доводе до прекидања пројеката су представљени наредном табелом.

Табела 2.2.2.2.2. Приказ фактора који доводе до прекида пројекта према истраживању Standish group, према Eveleens& Verhoef [Eveleens& Verhoef, 2010]

Редни број	Фактор који доводи до статуса пројекта „под изазовом“
1	Некомплетни захтеви корисника
2	Недостатак укључености корисника
3	Недостатак ресурса
4	Нереална очекивања
5	Недостатак подршке оперативног менаџмента
6	Мењање захтева и спецификација
7	Недостатак планирања
8	Пројекат више није потребан
9	Недостатак ИТ менаџмента
10	Технолошка писменост
11	Друго

У оквиру истраживања [Hartman&Ashrafi, 2002], користећи упитнике који су послати различитим фирмама у вези критеријума успеха ИТ пројеката, анализирани су фактори ризика и метрике којима се може пратити успешност реализације ИТ пројекта.

Табела 2.2.2.2.3. Критични фактори успеха ИТ пројекта и поставке за метрике, према Hartman&Ashrafi [Hartman&Ashrafi, 2002]

Редослед према рангу	Критични фактори успеха	Пројектне метрике
1	Власник је информисан о статусу пројекта и његово/њено одобрење је добијено у свакој фази	Пројекат је завршен на време или пре времена у складу са временским планом
2	Власник је консултован у свим фазама развоја и имплементације	Кључне контролне тачке су идентификоване и резултати су постигнути у проверама у тим тачкама
3	Одговарајући комуникациони канали су установљени на одговарајућем нивоу у пројектном тиму	Испоруке су идентификоване
4	Пројекат има јасно дефинисану мисију	Обухват пројекта је јасно дефинисан и квантификован
5	Највиши менаџмент је благонаклон (willing) да обезбеди потребне ресурсе (новац, експертизу, опрему)	Активности и логичке секвенце су утврђене и укључене у временски план (CPM)
6	Пројекат постиже пословну сврху која је раније планирана	Завршетак пројекта је прецизно дефинисан
7	Детаљни пројектни план (укључујући временски план и кључне контролне тачке) са детаљним планом буџета је дефинисан	Пројекат је завршен у оквиру предвиђеног буџета
8	Одговарајућа технологија и експертиза су на располагању	Захтеви за ресурсима су идентификовани и достављени у складу са потребама
9	Изменама у пројекту је управљано у оквиру формалног процеса	Одговорности су додељене
10	Пројекат је завршен уз минималне и узајамно договорене измене обухвата	Специфична нова технологија је прилагођена и прихваћена од стране крајњих корисника

## 2.2.2.4. Анализа зависности успеха софтверског пројекта

### 2.2.2.4.1. Истраживања зависности успеха пројекта

У раду [Munns&Vjeirmi, 1996] приказана је потреба за разликовањем пројекта и пројектног менаџмента и утицаја важности разликовања њихових елемената на успех пројекта.

У раду [Dvir et al, 2003] приказана је анализа корелације између нивоа напора који је уложен у три области планирања (функционална спецификација, техничка спецификација и процеси примене пројектног менаџмента) и успеха пројекта према мерама (Meeting planning goals; Enduser benefits; and Contractor benefits, Overall project success), који су проценили три главне врсте заинтересованих страна пројекта (крајњи корисник, пројектни менаџер и уговорена страна). Анализирани су подаци од 110 пројеката спроведених у Израелу, који су се првенствено односили на пројекте истраживања и развоја у области одбране. Истраживање је показало да су ставови испитаника о успеху пројекта такви да сматрају да примена процеса и процедура пројектног менаџмента не доприноси успеху пројекта, већ је потребно уложити више напора у дефинисање функционалних захтева и развој техничких спецификација. Исказана је висока корелација дефинисања функционалних захтева према дефинисању техничких спецификација, затим између добро дефинисаних функционалних и техничких спецификација према перцепцији о корисностима за крајњег корисника (а такође и високо, али не статистички значајно према корисностима за уговорену страну), а такође и висока корелација остваривања бенефита свих заинтересованих страна, успеха пројекта и планираних циљева.

#### **2.2.2.4.2. Истраживања зависности успеха ИТ пројекта**

У раду [Liang et al, 2007] приказано је истраживање ефекта диверзитета у тиму на успех софтверског пројекта, односно међусобним различитостима чланова тима у знању (knowledge diversity), различитост у припадности друштвеним категоријама (пол, године, националност) и различитостима у ставовима о вредностима, односно о томе шта је значајно, шта су стварни циљеви и мисија пројекта (value diversity). У овом раду је истраживан однос ових различитости у тиму и утицај на конфликте на нивоу задатака (task conflict) и на конфликте на нивоу међуљудских односа (relationship conflict). 185 упитника су послати професионалцима из ИТ области или менаџерима информационих система тајванских фирми. Резултати истраживања показују да различитост знања у тиму може допринети перформансама пројекта, док различитости у вредностима може смањити ефикасност тима и повећати штетне конфликте.

У раду [Mekonnen & Lamew, 2013] приказано је истраживање зависности избора методологије развоја софтвера од карактеристика личности чланова тима у развоју софтвера. Послато је 60 упитника софтверским фирмама или фирмама који имају ИТ одељење у Адис Абеби. Анализиран је утицај карактеристика личности (extraversion, neuroticism, agreeableness, conscientiousness, openness, cognitive) на успешност реализације софтверског пројекта користећи план-орјентисану или агилну методологију. Утврђен је статистички значајан утицај карактеристика личности на успех пројекта у оквиру развоја по план-орјентисаној методологији, док код агилних методологија није утврђен статистички значајан утицај типова личности.

У раду [Jiang et al, 2004] истражен је утицај зрелости процеса развоја софтвера на успех софтверског пројекта. Анализирани су одговори са 154 упитника који су послати члановима IEEE Computer Society поштом. Истраживање је показало да ниво зрелости управљања процесом развоја софтвера позитивно утиче на перформансе пројекта. Према овом истраживању, ефекти примене пројектног менаџмента тек долазе до изражаја у организацији која је достигла минимум ниво III (посебно су значајне активности процесног инжењерства и очувања квалитета процеса и производа), док активности нивоа II или нижег немају значајан утицај.

У раду [Milosevic & Patanakul, 2005] истражена је зависност успеха пројекта у области ИТ индустрије (индустрије која се брзо развија – high velocity industries) од стандардизованог пројектног менаџмента, који се може дефинисати као стандардизовани скуп практичних остварења пројектног менаџмента (standardized set of project management practices). У оквиру овог истраживања извршено је анкетање укупно 55 испитаника у циљу потврђивања хипотеза. Анализирана је улога следећих елемената стандардизоване праксе пројектног менаџмента: стандардизовани процес пројектног менаџмента, организација, информациони системи, алати, метрике, пројектна култура, лидерство. Истраживање је показало да елементи: PM tools, leadership, and process, према мишљењу испитаника, највише доприноси успеху пројекта и од већег су интереса за стандардизовану примену, док остали елементи су од мањег интереса за стандардизовану примену. Такође, истраживање је показало да су организације заинтересоване за стандардизацију само до одређеног нивоа.

У оквиру докторске дисертације [Grinter, 1996] истражене су, у оквиру приказа три студије случаја, зависности у оквиру развоја софтвера. Квалитет тих зависности утиче на општи квалитет софтвера и успех софтверског пројекта. Зависности су подељене по категоријама на: индивидуалне, групне, међуорганизационе, техничке, друштвене.

У оквиру докторске дисертације [Nguyen, 2006] истражена је критична улога планирања у реализацији софтверских пројеката, на примеру софтверске индустрије Вијетнама. У истраживању дефинисани су као фактори који утичу на квалитет резултата планирања: људски фактор, менаџмент и технички фактори. Успех планирања је оцењен кроз перформансе четири задатка: дефинисање захтева и



спецификација, процена трошкова и времена, одређивање распореда рада и анализа ризика. Резултати пројекта су вредновани уз помоћ пет критеријума: општи успех, квалитативне користи (нпр. инапређење способности пројектог тима, унапређење имиџа компаније), финансијске користи, време и трошкови. У наставку је дат преглед хипотеза наведеног истраживања, са становишта потврђености.

Истраживање [Nguyen, 2006] је вршено у односу на карактеристике пројекта (величину, тип и власништво) пројекта које представљају варијабле које описују однос између фактора планирања и перформанси планирања. Регресионом анализом је утврђено да ове пројектне карактеристике не утичу значајно на перформансе планирања пројекта. Истраживањем је утврђено да нема значајних разлика између пројекта у погледу величине, типа или власништва, односно величина пројекта, тип пројекта или власништво фирме нису значајно утицали на перформансе планирања пројекта.

Табела 2.2.2.4.2.1. Хипотезе Nguyen-а у вези зависности успеха пројекта и статус потврђености [Nguyen, 2006]

<b>ХИПОТЕЗА</b>	<b>СТАТУС ПОТВРЂЕНОСТИ ХИПОТЕЗЕ ПРИМЕНОМ ВИШЕСТРУКЕ РЕГРЕСИОНЕ АНАЛИЗЕ</b>
Позитиван ефекат напора пројектног менаџера на перформансе планирања	Подржана
Позитиван ефекат претходних искустава пројектног менаџера на перформансе планирања	Није подржана
Позитиван ефекат способности чланова тима на перформансе планирања	Подржана
Позитиван ефекат укључености клијента на перформансе планирања	Подржана
Ефекат примене метода пројектног менаџмента на перформансе планирања	Није подржана
Ефекат примене методе развоја система на перформансе планирања	Није подржана
Ефекат примене методе животног циклуса на перформансе планирања	Није подржана
Позитиван ефекат подршке менаџмента на перформансе планирања	Подржана
Ефекат пројектних задатака на перформансе планирања	Делимично подржана
Позитиван ефекат расположивости ресурса на перформансе планирања	Није подржан
Позитиван ефекат људи-орјентисаног стила вођства (leadership style) на перформансе планирања	Није подржан
Перформансе планирања позитивно утичу на успех пројекта	Подржан
Перформансе планирања позитивно утичу на квалитативне корисности (benefits) пројекта	Подржан
Перформансе планирања позитивно утичу на финансијске корисности (benefits) пројекта	Подржан
Перформансе планирања негативно утичу на време завршетка пројекта	Подржан
Перформансе планирања негативно утичу на трошкове завршетка пројекта	Подржан
Негативан утицај величине пројекта на перформансе планирања	Није подржан
Утицај власништва над пројектом на перформансе планирања	Није подржан

Према [Nguyen, 2006], ипак мале разлике постоје:

- Истраживањем је утврђено да су мале разлике у персоналу, техничким и менаџмент факторима који утичу на перформансе планирања у пројектима различитих величина. Разлике су биле значајне у резултатима пројекта. Мањи

пројекти су имали више позитивне квалитативне користи него велики пројекти. Мањи пројекти имају боље временско планирање, мања одступања у буџету и боље „неопипљиве“ користи, као што је унапређење способности пројектног тима, унапређење имиџа компаније итд, у односу на веће пројекте.

- Ако се узме у обзир власништво, пројекти страних компанија улажу више напора у пројектни менаџмент и унапређење способности тимског рада, као и имају бољу укљученост клијената у оквиру рада на софтверским пројектима него домаће (из Вијетнама) софтверске фирме. Стране фирме имају више иностраних клијената него локалне, што утиче на ниво укључености клијената и начин комуникације са клијентима (е-маил). Људски фактор је бољи у страним компанијама, посебно у делу управљања пројектима, знања и искуства чланова тима. Иако постоје значајне разлике у људском фактору и менаџменту, излазни резултати наведених пројеката се не разликују значајно.
- Незнатне су разлике између пројеката по типу, као што су комерцијални софтвери, софтвери по наруџби или outsourcing пројекти. Разлике се односе на корисника. Код комерцијалних пројеката веће је учешће корисника, боље дефинисање захтева за карактеристикама производа и боље дефинисане спецификације у односу на друге типове пројеката.

Истраживањем [Nguyen, 2006] није утврђена значајна повезаност између примене различитих техника и метода на перформансе пројекта. Према [Nguyen, 2006], у пракси примена техника у управљању софтверским пројектима није систематична и своди се најчешће на коришћење MS Project алата за временско планирање или коришћење spreadsheet-ова за евиденцију ресурса и праћење напретка. Коначно, у истраживању [Nguyen, 2006] утврђена је значајна повезаност између перформанси планирања и излаза пројекта, што потврђује важну улогу планирања. Такође, може се закључити да боља дефиниција захтева и спецификација повећава шансе за успех пројекта и боље квалитативне користи. Друго, боље временско планирање утиче на успех пројекта, повећава квалитативне користи и смањује време завршетка пројекта. Коначно, боље процењивање времена трајања пројекта и потребног напора резултује у смањењу времена реализације пројекта, смањује трошкове, што даје финансијске користи.

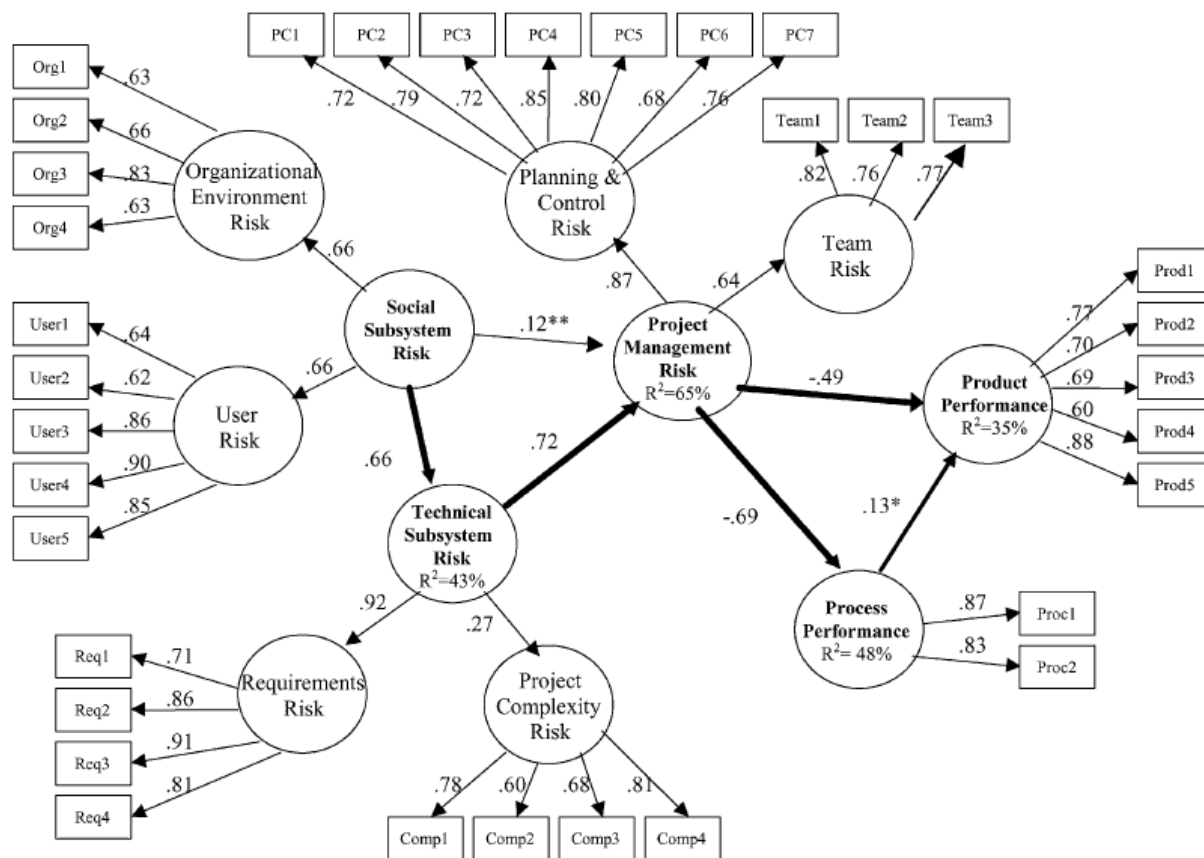
У оквиру [Sharif et al, 2012] истражене су зависности промена захтева корисника у току имплементације софтвера на трошкове реализације софтвера. Анализом 9 софтверских пројеката који су реализовани у различитим организацијама у Пакистану, издвојени су различити фактори, односно карактеристике које описују промене захтева и коришћењем SPSS софтвера утврђена је регресионом анализом зависност и коефицијенти корелације тако да је добијена формула којом би се могли израчунати трошкови захтева за изменом софтвера, а у којој трошкови представљају зависну величину која зависи од: фазе развоја софтвера, приоритета и типа захтева. Зависност је представљена линеарном једначином са коефицијентима који се односе на ниво значајности поједине променљиве:

$$\text{Cost} = - 45.967 + 8.605 \text{ SDLC Phase} \\ + 9.784 \text{ Priority} \\ + 1.844 \text{ Request Type}$$

Слика 2.2.2.4.2.1. Формула за изражавање трошкова измене софтвера, према Sharif et al [Sharif et al, 2012]

У оквиру [Wallace et al, 2004] истражен је утицај ризика софтверског пројекта на перформансе пројекта. На основу анализе литературе, издвојени су фактори ризика, затим на почетном малом узорку прелиминарно је урађено анкетање и истражен значај појединих фактора, а затим је коригована анкета (53 ставке анкете које се односе на 6 димензија ризика софтверског пројекта) коришћена у испитивању ставова 507 пројектних менаџера у вези утицаја појединих фактора ризика. Шест димензија ризика софтверског пројекта су подељене у групе: друштвени подсистем ризика (организационо окружење, корисник), технички подсистем ризика(захтеви,

комплексност пројекта), подсистем ризика управљања пројектом (ризик планирања и контроле, ризик који се односи на тим), а они утичу на перформансе производа и перформансе процеса.



Слика 2.2.2.4.2.2. Утицај подгрупа ризика на перформансе процеса и производа, према Wallace et al. [Wallace et al, 2004]

У оквиру [Wallace et al, 2004] утврђено је да друштвени подсистем ризика (нпр. нестабилна организација и корисници који не сарађују добро у току реализације пројекта) утиче на повећање подсистема техничких ризика (неодређеност захтева, повећање технолошке комплексности), што опет значајно утиче и на ризике пројектног менаџмента (планирање, контрола, управљање тимом) и коначно на перформансе процеса и производа.

## 2.2.3. Мерење перформанси софтверског пројекта

### 2.2.3.1. Стандарди у вредновању процеса развоја софтвера и управљања пројектима

У контексту вредновања процеса развоја, могу се користити стандардне процедуре и метрике, као што је стандард који дефинише вредновање процеса у области информационах технологија [ISO/IEC 15504].

Процес развоја софтвера у оквиру неке организације у дужем периоду може да се прати кроз стратешке циљеве унапређења наведене организације која се бави развојем софтвера у циљу унапређења пословања и успеха будућих пројеката. У том контексту дефинише се ниво зрелости организације у складу са СММ моделом (Capability Maturity Model) [Humphrey, 1988] [CMM-S, 1993], што даје основ за унапређење организације.

Табела 2.2.3.1.1. Нивои Capability Maturity Model за софтверске пројекте према Herbsleb et al [Herbsleb et al, 1997]

СММ ниво	ГЛАВНЕ КАРАКТЕРИСТИКЕ
1. Иницијални	Софтверски процес је карактерисан као ad hoc, и понекад чак хаотичан. Неки процеси су дефинисани и успех зависи од индивидуалних напора и хероја.
2. Поновљив	Основни процеси пројектног менаџмента су установљени како би се пратили трошкови, временски план и функционалност. Потребна дисциплина процеса је успостављена, како би се поновио успех ранијих пројеката са сличним апликацијама.
3. Дефинисан	Софтверски процес за менаџмент и инжењерске активности је документован, стандардизован и интегрисан у стандардни софтверски процес у организацији. Пројекти користе одобрене и прилагођене верзије организационих стандардних софтверских процеса за развој и одржавање процеса.
4. Управљан	Детаљна мерења софтверског процеса и квалитета производа се спроводе и прикупљају. Софтверски процес и производи су квантитативно представљени, разумљиви и контролисани.
5. Оптимизујући	Континуирано унапређење процеса је олакшано квантитативном повратном информацијом од стране процеса и од стране увођења иновативних идеја и технологија.

Интеграцијом процеса и области знања у управљању пројектима дефинисан је тзв. Project Management Process Maturity Model (PM)<sup>2</sup> који може бити примењен за утврђивање зрелости организације и дати правце даљег унапређења организације. [Kwak&Ibbs, 2002]

### 2.2.3.2. Метрике у управљању софтверским пројектима

„Фундаментални задатак пројектног менаџмента је учинити за посао буде урађен, односно да се реализују задаци у оквиру предвиђеног времена, трошкова и перформанси. Ове три варијабле су критичне димензије пројекта које захтевају континуалну пажњу пројектног менаџмента. У последње време, менаџери су додали и четврту димензију – задовољство клијената.“ [Jurison, 1999]

У контексту управљања софтверским пројектима, могу се користити софтверске метрике, ради мерења атрибута софтверског производа или процеса. [Frakes&Kang, 2005]

У области која се односи на примену софтверских метрика, следећи су правци истраживања [Vasilescu et al, 2011]:

1. Осмишљавање модела који дају описе и објашњења карактеристика софтверских система са статистички значајним утицајем
2. Развој модела за предикцију грешака (fault prediction models) [Nagappan&Ball, 2007]

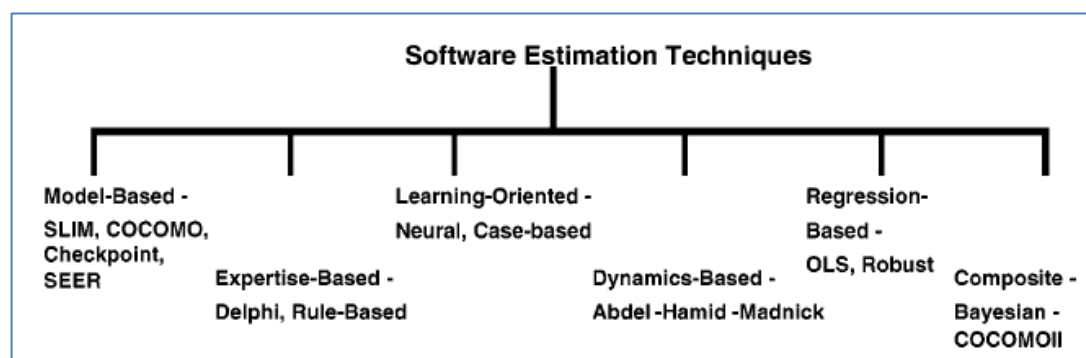
3. Проблеми људске процене у примени софтверских метрика [Mair& Shepperd, 2011]
4. Софтверске метрике за различите врсте грешака [Misirli et al, 2011]
5. Развој радних оквира (frameworks) који олакшавају примену и разумевање метрика
6. Проблем дистрибуције метрика и различитих нивоа сложености, доводи до потребе за агрегацијом метрика у хијерархијском уређењу [Herbsleb et al, 1997], односно примени метрика на нивоу софтверских пакета, у оквиру агилног приступа развоју [Martin, 2002]

Приликом избора метрика важно је узети у обзир [Mair& Shepperd, 2011] да превише поједностављена мерења могу да изазову више штете него користи. Поједностављивање се своди на избор мањег броја мерних величина које се приказују, јер је за њих лакше прикупити податке и лакше их је приказати. Поједностављивањем се често постиже ефекат да учесници пројекта добију погрешну слику о циљевима и напретку пројекта, често превише оптимистичну. Из наведених разлога, у управљању пројектима се јавља тенденција развоја или коришћења и прилагођавања готових framework-а (понекад формираних за друге примене), који дефинишу скупове метрика у складу са одређеним моделом.

### 2.2.3.3. Процена трајања, ангажовања и трошкова софтверског пројекта

Трајање, односно ангажовање (effort) на пројекту изражава се у јединици: човек-месец [Nassif, 2012]. У израчунавању вредности трајања пројекта, користе се различити приступи.

У раду [Jevtić&Letić, 2011] [Boehm et al, 2000] дат је преглед метода за процену трајања пројекта: методе мрежног планирања и технике процене трајања пројекта, процена експерата (базирана на аналогiji или симулацији), управљање стеченом вредности, Монтекарло симулација, примена Кларкових једначина [Jevtic, 2010].



Слика 2.2.3.3.1. Технике процене софтвера према Boehm-у [Boehm et al, 2000]

У раду [Khatibi&Jawawi, 2011] описане су суштинске разлике између других пројеката и софтверских пројеката, које се састоје у томе да софтверски пројекат резултује неопипљивим резултатима, чије се карактеристике мењају у току развоја у складу са променама захтева корисника. Наведене су методе за процену трајања пројекта [Khatibi&Jawawi, 2011]:

А) Алгоритамски приступи

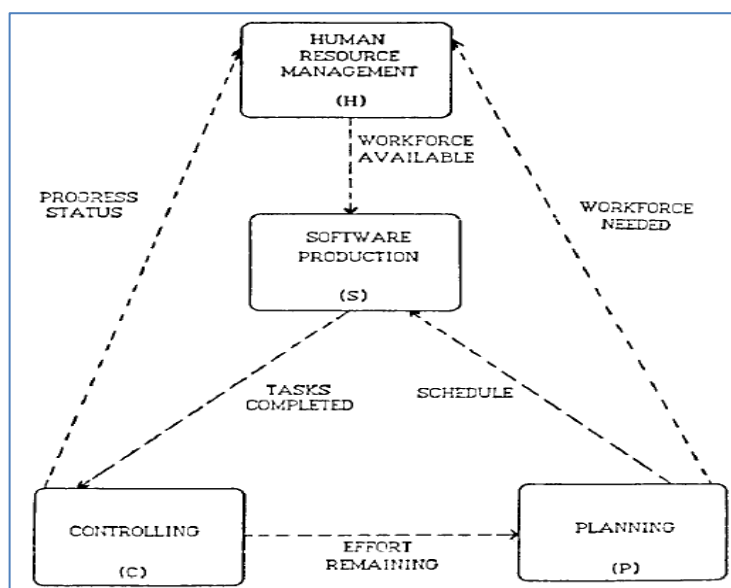
И) Улазни подаци – број линија кода

- Алгоритамски приступ мануелне процене, 1950-тих година
- Методе базиране на линеарним једначинама и регресионим техникама, 1965. године
- Први аутоматизовани алат (Interactive productivity and Quality (IPQ) [Khatibi et al, 2012], [Jones, 2007]) развијен у IBM 1973. године
- COCOMO метод (такође назван и COCOMO 81 [Nassif, 2012]), аутора Barry Boehm-а [Boehm, 1981]
- Software Lifecycle Management (SLIM) [Putnam, 1987] или Putnam модел, презентован 1978. године [Nassif, 2012]

- Software Evaluation and Estimation of Resources – Software Estimating Model (SEERSEM)
- II) Улазни подаци – Функционални поени – уводи Albrecht [Albrecht&Gaffney, 1983] 1979. године [Nassif, 2012]:
  - COCOMO II [Boehm et al, 1995], узима у обзир променљивост захтева [Nassif, 2012], користи логичке уместо физичких линија кода као мерило [Nassif, 2012] и подржава различите методологије развоја софтвера [Khatibi&Jawawi, 2011]
- Б) Неалгоритамски приступи:
  - Експертска процена, 1963. године [Dalkey& Helmer, 1963], са различитим методама:
    - Classification And Regression Tree (CART) 1984. године [Breiman et al, 1984]
    - Analogy Based Estimation (ABE) 1997. године [Shepperd& Schofield, 1997]
    - Примена неуронских мрежа и фази система.

У наведеном раду [Khatibi&Jawawi, 2011], за потребе процене процене ангажовања (трајања) у оквиру софтверског пројекта, предложен је приступ анализи карактеристика пројекта пре него што се приступи процени трајања пројекта, категоришући софтверски пројекат првенствено у односу на методологију развоја и тип софтвера који се развија.

У раду [Abdel-Hamid& Madnick, 1987] разматран је проблем прецизности израчунавања у процесу процене величине, ангажовања и трајања пројекта, као и проблем портабилности наведених израчунавања (примене истих модела процене у различитим организацијама). Као основ побољшања портабилности, предложен је модел који у оквиру процене узима и елементе и карактеристике из области управљања пројектима, а не само инжењерски засноване метрике.



Слика 2.2.3.3.2. Модел за подршку процени софтверског пројекта према Abdel-Hamid & Madnick-у [Abdel-Hamid& Madnick, 1987]

У наведеном истраживању, квантификована су наведена 4 фактора (људски фактор, процес развоја софтвера, контролисање, планирање) и емпиријски верификован њихов утицај на трошкове реализације софтвера.

У раду [Reifer, 2000] извршена је компарација традиционалног и развоја Web апликација у смислу елемената управљања пројектима и процене трајања пројекта. Предложен је модел са метрикама и једначинама за израчунавање процене ангажовања (effort) и трајања (duration) реализације пројекта развоја Web апликације. Израчунавање се заснива на броју елемената и процени њихове сложености.

### 2.2.3.3.1. Израчунавање трајања и трошкова софтверског пројекта у развоју информационог система

Савремени приступи процени величине („size“), ангажовања („effort“) и трајања („duration“) софтверског пројекта укључују методе за процену на основу основних модела који представљају основ за реализацију софтверског пројекта. У оквиру развоја софтвера за потребе информационог система, основни модели су: модел пословних процеса, модел софтверских функција и модел података. У наставку ће бити анализирани приступи процени величине софтверског пројекта на основу анализе модела који претходе његовој имплементацији.

У раду [Monsalve et al, 2011] приказан је приступ припреме модела пословних процеса за потребе процене функционалне величине софтвера на основу модела пословних процеса од стране COSMIC (Common Software Measurement International Consortium) методе, која представља међународно признати стандард [COSMIC]. У оквиру наведеног приступа, дефинисан је низ правила која омогућавају израду модела пословних процеса у одговарајућу форму погодну за процену. У раду [Vogelzang, 2006] описана је метода COSMIC-FFP примењена у оквиру развоја ERP система, где се јавља специфична ситуација низова повезаних активности за које се развија софтверска подршка. С обзиром да се ради о специфичној врсти софтверских пројеката, утврђена је потреба за додатном параметризацијом методе због специфичних услова и критеријума који могу да утичу на коначан резултат примене, што захтева даља истраживања. У раду [Vogelzang, 2006] описани су основни елементи примене методе COSMIC. Спецификација функционалних захтева корисника се дели на елементарне акције које софтвер треба да изврши: померања података (улаз, излаз, упис и читање података) и манипулацију подацима, где свака елементарна акција носи 1 Cfsu (COSMIC functional sizing unit). У оквиру примене COSMIC, може се израчунати процењено време испоруке, као и трошкови (који узимају у обзир директне трошкове девелопера и трошкове подршке – контроле квалитета, пројектног менаџмента итд.) На слици 3.4.3.1.1. приказана је формула за израчунавање временског трајања пројекта, с тим да се фактори 0.2 и 0.37 могу променити, али представљају вредности које су добијене на основу претходних емпиријских истраживања.

$$\text{Time}_{\text{Delivery}} = \frac{\text{Size} \cdot \text{Power}}{\text{PL}}$$

where:

Time	= Time to delivery of the process-chain in months
Size	= Functional size in Cfsu
Power	= 0,20 for a single production line and 0,37 for two production lines
PL	= Number of deployed production lines (1 or 2)

Слика 2.2.3.3.1.1. Формула за израчунавање времена испоруке на основу COSMIC методе према [Monsalve et al, 2011]

Иако су модели пословних процеса у наслову тематике претходно описаних радова, ипак нису јасно разграничени концепти пословног моделирања независног од софтверске аутоматизоване подршке, односно одговарајућих случајева коришћења. Наиме, у основи дефиниције случаја коришћења јесте спецификација захтева корисника у оквиру пословног контекста примене будућег софтверског решења (дакле блиска повезаност пословних процеса и софтверских функција) [Clemmons, 2006].

У раду [Issa&Rub, 2007] јасно се наглашава потреба да се процесима из модела пословних процеса (који је технолошки независан и представља организационе послове) изводе одговарајући случајеви коришћења који се односе на аутоматизовану подршку одговарајућим пословним процесима. На основу издвајања случајева коришћења добијамо кандидате за случајеве коришћења (који зависе од захтева

корисника која често могу бити и непрецизна). Укупно ангажовање (effort – изражен у човек-месецима) може се добити као:

$$Effort = Total\ Number\ of\ Candidate\ UseCases \times Average\ Effort\ per\ UseCase$$

Слика 2.2.3.3.1.2. Формула за израчунавање времена испоруке на основу броја кандидата за случајеве коришћења према [Issa&Rub, 2007]

Као просечно ангажовање по случају коришћења може рачунати 0.67 човек-месеци [Issa&Rub, 2007].

У докторској дисертацији [Nassif, 2012] предложен је приступ процени величине софтвера и потребног ангажовања у реализацији на основу UML USE-CASE дијаграма, применом регресије и модела soft computing. У раду [Clemmons, 2006] описан је приступ примени Use Case Points (UCP) методе, која омогућава процену ангажовања (човек-сати) које реализација софтверског пројекта захтева, на основу анализе случајева коришћења, профила корисника (actors) сценарија, али и техничких фактора (техничка комплексност) и фактора комплексности развојног окружења.

$$UCP = UUCP * TCF * ECF * PF$$

Слика 2.2.3.3.1.3. Формула за израчунавање Use Case Points према [Clemmons, 2006]

У наведеној формули:

- UUCP – Unadjusted Use Case Points – Израчунава се на основу UUCW (Unadjusted Use Case Weight) – укупан број активности или корака у оквиру сценарија случаја коришћења и UAW (Unadjusted Actor Weight) – комплексност свих учесника (actor) у свим случајевима коришћења (тип actora: једноставан – други софтверски систем којем се приступа користећи API, средње сложен – приступ другом софтверу користећи протоколе, комплексан – приступ корисника користећи графички кориснички интерфејс)
- TCF – технички фактор комплексности – тип софтвера (дистрибуирани или не), техничке карактеристике софтвера...
- ECF – „Environment Complexity factor“ – фактор развојног окружења, који се односи нпр. на претходна знања и искуства девелопера, способност, мотивацију, сложеност технологије израде, стабилност захтева
- PF – фактор продуктивности – рачуна се на основу ранијих искустава – однос броја сати са бројем поена случајева коришћења и добијемо фактор продуктивности у човек-сатима.

Коришћење анализе случајева коришћења омогућава прелиминарну процену величине и трајања пројекта у раној фази када није познат број табела у бази података или број ентитета или класа који учествују у реализацији сваког случаја коришћења. Тек познавањем структуре података и доделом одговарајућег броја табела базе података случајевима коришћења, могуће је прецизније одредити величину, трајање и трошкове реализације софтверског пројекта.

Величина и трајање пројекта може се израчунати и само на основу анализе елемената модела података. У раду [Tan et al, 2009] описан је став да величина изворног кода софтвера информационог система зависи од базе података, која зависи од одговарајућег концептуалног модела података. У наведеном раду одређена је формула за одређивање броја линија кода (величина софтвера) на основу броја ентитета, повезника и атрибута ER дијаграма (који представља концептуални модел података). За различита имплементациона окружења користе се сличне формуле, где су параметри исти (број елемената ER дијаграма), а фактори значаја су различити.



Basic-based systems is as follows:

$$Size = 6.788 - 0.062E + 2.169R - 0.007A.$$

The model that we built for Java-based systems is as follows:

$$Size = 4.678 + 1.218E + 0.028R + 0.023A$$

In both models, size was measured in thousand lines of code (KLOC).

Слика 2.2.3.3.1.4. Формула за израчунавање величине софтвера (изражено у хиљадама линија кода) према [Tan et al, 2009]

При чему су конкретне вредности фактора значаја изведене из статистике претходних искустава развоја софтвера одређене технологије.

У раду [Mishra et al, 2010] приказан је приступ процени величини (изражено у броју линија кода) и ангажовања у развоју софтвера базирано на величини и комплексности релационе базе података, односно броју табела, број атрибута у табелама, броја типова података и ограничења, рбој тригера, број улазних форми, број погледа.

#### 2.2.3.4. Мониторинг перформанси софтверског пројекта

Gartner Group [Hill, Thomson, 2001] класификује управљање ИТ процесима као „IT Management Process Maturity“ кроз следеће нивое зрелости:

- Хаотично (Chaotic – Нема конзистентне примене алата за мерење перформанси
- Реактивно (Reactive) – Организација користи конзоле за приказ догађаја (event consoles)
- Проактивно (Proactive) – Организација користи мониторинг перформанси и историјске алате (historical tools)
- Услужно (Service) – Организација користи планирање капацитета
- Вредносно (Value) - Организација користи метрике повезаности ИТ и пословања (IT/Business Metric Linkage)

У оквиру управљања софтверским пројектима, мониторинг софтверским пројектом је важан[Wahyudin &Tjoa, 2007] , јер пројектни менаџер треба у сваком тренутку у току животног циклуса софтверског пројекта да располаже и вреднује податке који описују статус пројекта, посебно у односу на ризике, како би могао да иницира одговарајуће активности. Према [Wahyudin &Tjoa, 2007], мониторинг се односи на податке које се односе на време – процесе/догађаје као и податке које се односе на артефакте креирања производа. „Генерално постоји два приступа мониторингу: алат-базирани и човек-базирани приступ. Алат базирани приступи су најчешће погодни за фреквентни мониторинг веће количине података који се односе на процесе развоја или када су људски ресурси за мониторинг недоступни или се тешко проналазе. С друге стране, човек-базирани мониторинг је најпогоднији за недељне или месечно извештавање о раду које презентују учесници.“, [Wahyudin &Tjoa, 2007] У оквиру традиционалног пројектног менаџмента, човек-базирани приступ је доминантан и он најчешће изоставља податке о процесима, већ је наглашено извештавање о статусу стања резултата пројекта и финансијском аспекту.

##### 2.2.3.4.1. Човек-базиран мониторинг перформанси софтверског пројекта

У погледу човек-базираног приступа, неке од уобичајених метода и техника у контроли софтверског пројекта су [Rasli&Mohd, 2008]:

1. Контрола трошкова и временског плана („Cost and Schedule Control“) – основни елементи планирања пројекта (WBS – Work Breakdown Structure, Gantt chart) као и извештаји о акумулацији трошкова („cost accumulation reports“) обезбеђују основне индикаторе о пројектном статусу. Користи се и „Earned Value“ метода као хибридна

мера која изражава вредност завршеног посла у облику буџета који је додељен том делу посла.

2. Састанци прегледа пројекта („Project Review Meetings”) - Састанци су најефикаснији када се одвијају у редовним интервалима (најчешће једном недељно) и када имају уобичајену структуру (статус високоризичних области, глобални напредак пројекта – основне контролне тачке, временски план и трошкови), напредак појединаних активности, статус ставки акција, планирање будућих активности.

3. Технички и квалитативни прегледи, чији је циљ детектовање и отклањање проблем техничке природе и у области квалитета, али се не односе на преглед напретка пројекта.

4. Извештаји о статусу („Status Reports”) – најчешће се шаљу у редовним интервалима учесницима или клијентима који су ван пројектне организације.

5. Контрола техничких перформанси – провера да ли су техничке карактеристике у складу са дизајном, одвија се у оквиру провере дизајна („design reviews”), који се одвија у оквиру главних контролних тачака (након дефиниције захтева, дизајна, програмирања), уз препоручљиво учешће корисника.

6. Обезбеђење квалитета – односи се на активности обезбеђивања да производ буде у складу са захтевима корисника у погледу функционалности и квалитета. Циљ је откривање и отклањање грешака што раније. Квалитет треба, слично као и захтеви техничких перформанси, бити дефинисан у квантификованом облику, како би биле разумљиве клијенту и пројектном тиму. Основне технике обезбеђења квалитета су технички прегледи „technical reviews” (пролази – „Walkthrough” и инспекције – „Inspections”) и тестирање софтвера. Метрике које се користе у области квалитета су густина дефекта (број дефекта у 1000 линија изворног кода) и ефикасност уклањања дефекта (процент дефеката који су отклоњени). Метрике се примењују за модуле и систем у целини, категорисане и примењене и према типу дефекта.

7. Контрола промена и конфигурације – промене у току реализације могу довести до значајних одступања у односу на почетне захтеве корисника. Технике решавања проблема у овој области се односе на аутоматизоване алате за контролу конфигурације. Промене започињу предлогом за инжењерску промену („engineering change proposal – ECP”), која садржи опис поребе за променом, карактер промене и утицај промене, Измена се одобрава од стране одговарајућег организационог тела („change control board”), а треба да се реализује тако да се врше само потребне измене, измене се одвијају контролисано и о изменама су информисани сви учесници на које измена може да утиче.

8. Евалуација пројекта – процес периодичног вредновања статуса пројекта у складу са циљевима. Разликује се од контроле пројекта у два аспекта:

- контрола пројекта је у одговорности пројектног менаџера, а евалуација пројекта је задатак вишег менаџмента организације
- Контрола пројекта је континуалан процес, док су евалуације пројекта периодичне. Одвијају се у главним контролним тачкама, где се проверавају измене и преиспитују циљеви и задаци, могућност реструктурирања плана пројекта или чак одустајање од пројекта.

#### **2.2.3.4.2. Индикатори у мониторингу перформанси софтверског пројекта**

Да бисмо мерили и вршили мониторинг перформанси пројекта, потребно је да буду дефинисани индикатори перформанси који треба да буду мерени. У раду [Garcia et al, 2006] описана је терминологија у области мерења софтвера кроз компарацију различитих стандарда и предложених метода у истраживању мерења софтвера. У наведеном раду дефинисана је онтологија којом су основни термини у области мерења софтвера дефинисани и уређени.

Најчешћи индикатори перформанси се односе на време, трошкове и квалитет [Rasli&Mohd, 2008]. Метрике и индикатори у мерењу перформанси софтверског пројекта могу бити дефинисане у односу на различите моделе (методе и приступе).

У [Kan, 2009] приказан је Effort-Outcome модел на основу ког је дефинисан скуп метрика које се могу спровести у разним фазама реализације софтверског пројекта, као што је приказано на табели 2.2.3.4.2.1.

У [Hartman&Ashrafi, 2002] утврђено је да се у току развоја софтвера разликују по значају фактори успеха пројекта у току различитих фаза и за различите учеснике/заинтересоване стране у пројекту. У односу на факторе успеха, дефинисане су и метрике успеха пројекта, за које је такође утврђено да се њихов значај разликује у односу на фазе и учеснике/заинтересоване стране. Такође, метрике се могу одредити и на основу стратешких циљева који се постижу пројектом за организацију која ради на развоју софтвера, као и за организацију која представља наручиоца пројекта. Метрике према фазама и радним улогама учесника у оквиру [Hartman&Ashrafi, 2002] су представљене дијаграмима, где су издвојене следеће метрике: Прихваћеност нове технологије, Додељеност одговорности, Обезбеђени ресурси, Реализација у оквиру буџета, Дефинисан критеријум завршетка, Дефинисан временски план активности, Дефинисан обухват, Дефинисани резултати испоруке, Реализовани резултати у оквиру контролних тачака, Реализован пројекат у оквиру временског оквира.

Табела 2.2.3.4.2.1. Развојне фазе софтвера и индикатори мерења перформанси софтверског пројекта према Кан-у [Kan, 2009]

Development Phase	Effort Indicators	Outcome Indicators
Requirements	- Coverage - req analysis and reviews - Coverage - Customer validation	- Requirements problem/issue rate
Design	- Design reviews coverage - Design reviews - inspector-hour per review - Design reviews progress metrics	- Design defect rate - Design rework indicators
Code	- Code inspection coverage - Code inspection progress metrics - static code analysis coverage	- Inspection defect rate - Rework indicators
Unit Test/Integration/Build	- Unit test plan coverage - test progress metrics - static code analysis coverage - Effectiveness/function coverage of build verification test	- Defect metrics
FVT (Functional Verification Test)	- Test plan function coverage/usage - Test Progress metrics	- Overall test defect rate - Test defect arrival pattern - Defect backlog metrics
SVT (System Verification Test)	- Same as FVT - CPU utilization - other indicators for system stress - Coverage of business/customer usage scenarios	- Same as FVT - Metrics for crashes and hangs - Mean time between outage
Beta	- Relative program progress - % of customers on production Vs testing - Customer usage of functions	- Customer problem rate - Cust. satisfaction

Мониторинг успеха софтверског пројекта може се вршити коришћењем Earned Value приступа, у комбинацији са USE CASE анализом [Liu, 2009]. Један од framework-а који се може користити за дефинисање метрика за праћење успешности софтверских пројеката је Balanced Scorecard [Kaplan&Norton, 1992]. Овај framework је првенствено осмишљен као оквир за примену метрика у праћењу успешности реализације стратешких циљева организације, односно као алат стратегијског менаџмента. С обзиром да пројекат може бити сматран организацијом са својим стратешким циљевима и задацима [Niebecker et al, 2008], могуће је прилагодити Balanced Scorecard примени у управљању пројектима, дефинисањем Project Scorecard, који мора бити стално унапређен кроз комуникацију и сарадњу учесника пројекта, чиме настаје Collaborative Project Scorecard [Niebecker et al, 2008]. Balanced Scorecard је нашао примену у области информационих технологија, прво кроз низ метрика које се могу применити у оцени

успешности усклађивања информационих технологија и организационих стратешких циљева [Keyes, 2005], у мерењу и мониторингу успеха реализације информационих система [Martinsons et al, 1999], ERP система [Rosemann & Wiese, 1999] и e-Government решења [Bertelsmann, 2002] [Dobrović et al, 2008].

За избор метрика за праћење успеха софтверског пројекта често се, као и у случају са Balanced Scorecard, примењују методолошки оквири који су првенствено намењени другим применама. Тако у раду [Mahnic & Zabkar, 2008], приказани су резултати истраживања у којем су примењени CoBIT индикатори за мерење перформанси процеса развоја софтвера применом SCRUM агилне методологије, усклађени са IT Balanced Scorecard приступом.

Поједина истраживања фокусирана су на специфичне врсте софтверских пројеката и дефинисање критеријума и метрика успеха. Тако у раду [Crowston et al, 2003] базирано на DeLone and McLean моделу за дефинисање критеријума квалитета информационог система, дефинисан је скуп метрика који се може применити на Open-source пројекте.

### 2.2.3.6. Мерење успеха софтверског пројекта

Мерење успеха софтверског пројекта може се вршити коришћењем различитих метода [Demir et al, 2009]:

1. Субјективна евалуација – субјективно рангирање успеха пројекта од стране учесника пројекта, без залажења у детаље, може се сматрати валидним у извесној мери, али не даје довољно података о узроцима и могућностима побољшања
2. Мерење засновано на упитницима – упитници се формулишу као низ питања која су у складу са одређеним моделом вредновања, најчешће у друштвеним наукама, али и у области управљања софтверским пројектима.
3. Примена софтверских метрика – врше се директна мерења одређених карактеристика софтвера или учесника и активности у софтверском пројекту, а затим се као индиректна мерења укључују као карактеристике из којих се изводе закључци о конкретним мерама. Вредности мерења се удружују у заједнички метрички сумарни израз који укључије метрике кроз метричке функције, у складу са одабраним метричким моделом.

$$SPMEM = Measurement\_function\left(\sum_{i=1}^n w_i m_i\right)$$

Слика 2.2.3.4.2.1. Формула за сумарно израчунавање резултата мерења софтверских метрика у софтверском пројекту према Demir-у [Demir et al, 2009]

Метрички модел у управљању квалитетом SPMEM одређује скуп мерних функција које меримо, а свака функција се састоји из збира резултата метрика. У оквиру метричког модела број метрика може бити 1..n. За сваку метрику  $m_i$  постоји одговарајући коефицијент  $w_i$  који одређује значај појединачне метрике у оквиру одговарајуће мерне функције.

4. Модел-базирано мерење – одређени начин вођења пројекта може се моделовати кроз коришћење одређених активности и ентитета као кључних елемената [Demir & Ostmundson, 2008]. На овај начин, мерење ефективности пројектног менаџмента може се вршити моделом где се квалитет вреднује мерном функцијом која укључује мерења за све активности и све ентитете, а њихове мерне вредности приказују сумарно.

$$SPMEM = Measurement\_function\left(\sum_{i=1}^m qa_i + \sum_{j=1}^n qe_j\right)$$

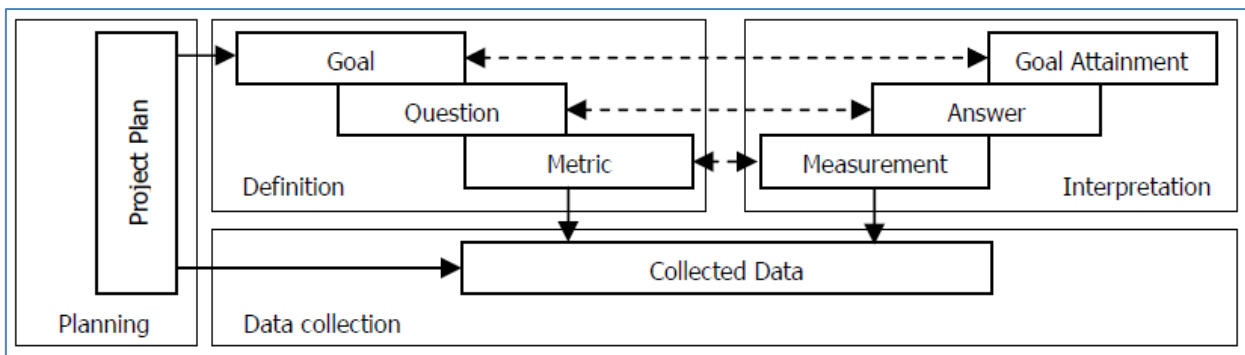
Слика 2.2.3.4.2.2. Формула за сумарно израчунавање резултата мерења квалитета активности и особина ентитета у софтверском пројекту према Demir-у [Demir et al, 2009]

Претходна формула изражава метрички модел SPEM који се састоји од метричких функција примењених над збиром резултата мерења квалитета активности  $qa_i$  и квалитета ентитета  $qe_i$ . Активности и ентитети су компоненте модела пројектног менаџмента. Број активности може бити  $m$ , а број ентитета може бити  $n$ .

Дефинисање кључних индикатора перформанси зависи од утврђене методологије и приступа. Један од приступа је Earned-Value приступ [Fleming&Koppelman, 1998] [Jurison, 1999], којим се у оквиру мониторинга успеха пројекта у сваком тренутку може израчунати вредност (финансијска, односно утрошена средства) урађеног посла у односу на расположиви финансијски фонд предвиђен за наведени пројекат, односно укупну количину посла за цео пројекат.

У раду [Joosten et al, 2011] је описан проблем мерења успеха пројекта у области информационих система. Извршено је полуструктурирано интервјуисање 9 експерата-учесника у развоју софтвера у односу на критеријуме успеха пројекта у области информационих система. У наведеном истраживању је утврђено да се у пракси као основни критеријуми успеха узимају стандардне карактеристике: време, буџет и испуњавање функционалних карактеристика. Разлози за ређе коришћење већег броја метрика је у томе што: 1) мерење по појединим метрикама се тешко реализује, 2) поједине карактеристике су изводиве из основних карактеристика успешности.

У области управљања перформансама софтверских пројеката користи се Goal-Question-Metric (GQM) Metod ([Basili& Weiss, 1984]). Према [Jurison, 1999] [Solingen& Berghout, 1999], [Lavazza, 2000] Goal-Question-Metric Metod представља систематску технику развоја програма мерења који се односи на софтверске процесе и производе. GQM процес је осмишљен на бази става да мерење треба да је оријентисано ка циљу, односно да прикупљање података треба да је базирано на експлицитно документованом образложењу. Дефинисане су четири фазе у примени методе GQM: планирањем дефинисање, прикупљање података и интерпретација.



Слика 2.2.3.4.2.3. Фазе у примени GQM методе, према Solingen& Berghout-у [Solingen& Berghout, 1999]

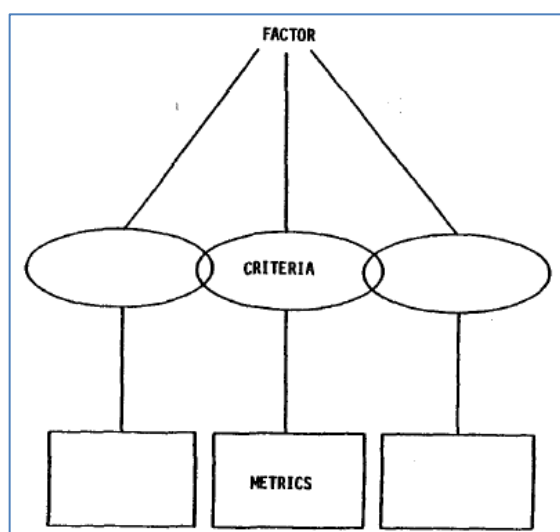
У фази планирања, врши се избор, дефиниција, карактеризација и планирање пројекта мерења, што резултује пројектним планом. У току фазе дефинисања, циљеви, питања, метрике и хипотезе су дефинисане и документоване уз помоћ интервјуа. У фази прикупљања података подаци се прикупљају у складу са претходним фазама. У последњој фази, подаци се обрађују у мерне резултате и обезбеђују одговоре на дефинисана питања, што води ка евалуацији постигнућа циљева.

## 2.2.4. Мерење карактеристика софтверског производа, артефакта процеса развоја софтвера и информационог система

### 2.2.4.1. Мерење успеха и квалитета софтверског производа

Према [Cavano&McCall, 1978], тестирање софтвера не може обезбедити квалитет софтвера, нити дати довољно информација о квалитету софтвера, већ може указивати само на грешке и верификовати функционалне карактеристике софтвера. Из наведеног разлога развијени су различити модели квалитета софтвера, који укључују различите карактеристике квалитета софтвера, а такође развијени су и алати за мерење и вредновање карактеристика на основу мерења.

У раду [Cavano&McCall, 1978] описан је оквир за квантитативан приступ квалитету софтвера, приказан на слици 2.2.4.1.1. који се састоји од хијерархије фактора, за који се дефинише скуп критеријума за вредновање, а за сваки критеријум се одређује скуп метрика.



Слика 2.2.4.1.1. Модел квантитативног приступа мерењу квалитета софтвера према Cavano&McCall [Cavano&McCall, 1978]

Према [Cavano&McCall, 1978], *фактор* представља менаџмент-орјентисан приступ квалитету производа, *критеријум* представља софтверски-орјентисане атрибуте који обезбеђују квалитет, а *метрике* представљају квантитативне мере наведених софтверски-орјентисаних атрибута. За сваки фактор квалитета може се креирати линеарна једначина следеће структуре:

$$r_f = c_1 m_1 + c_2 m_2 + c_3 m_3 + \dots$$

where:

$r_f$  is a rating of a quality factor,  $f$

$c_j$  are the regression coefficients

$m_i$  are the various measurements identified as relating to the quality factor,  $f$ .

Слика 2.2.4.1.2. Формула за израчунавање вредности фактора квалитета сумирањем вредности метрика према Cavano&McCall [Cavano&McCall, 1978]

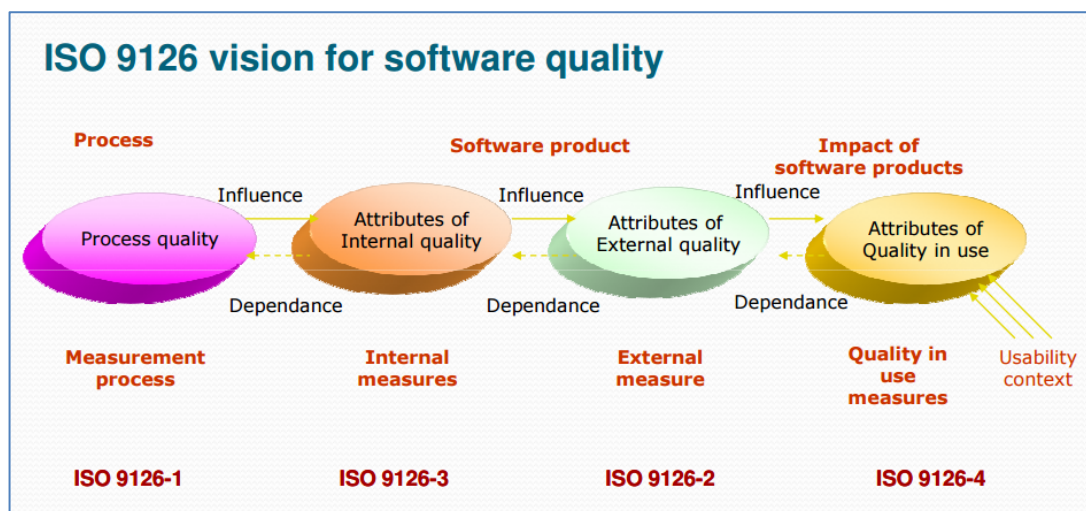
У области мерења квалитета софтвера, дефинисано је неколико стандарда, међу којима је ISO/IEC 9126: 1991 Software engineering - Product quality [Vollman, 1994], који је повучен, али је важећи стандард: ISO/IEC 25000:2005 (ревизија ISO/IEC 25000:2014)

Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Guide to SQuaRE. Дефинисани су и специфични стандарди за поједине типове софтвера, као што је: ANSI/IEEE 1209 стандард дефинише детаљније процес који би могао бити коришћен код евалуације и селектовања CASE алата.

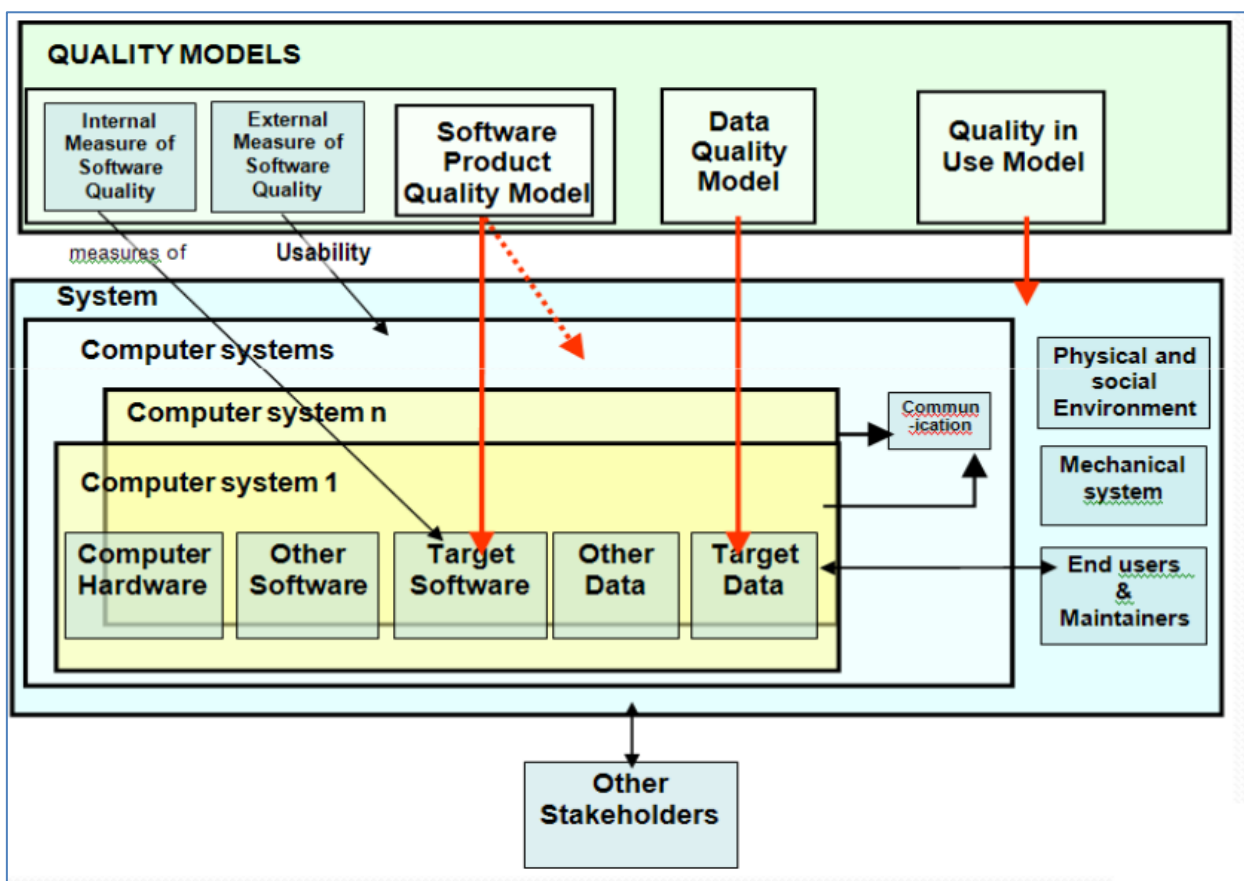
<b>Characteristics and subcharacteristics in ISO/IEC 9126</b>	
<b>Characteristic</b>	<b>Subcharacteristics</b>
Functionality	Suitability, accuracy, interoperability, security, functionality compliance*
Reliability	Maturity,* fault tolerance,* recoverability,* reliability compliance*
Usability	Understandability, learnability, operability, attractiveness, usability compliance*
Efficiency	Time behavior, resource utilization, efficiency compliance*
Maintainability	Analyzability, changeability, stability, testability, maintainability compliance*
Portability	Adaptability, installability, replaceability, coexistence, portability compliance*

Слика 2.2.4.1.3. Карактеристике и субкарактеристике квалитета софтвера према ISO/IEC 9126 [Jung et al, 2004]

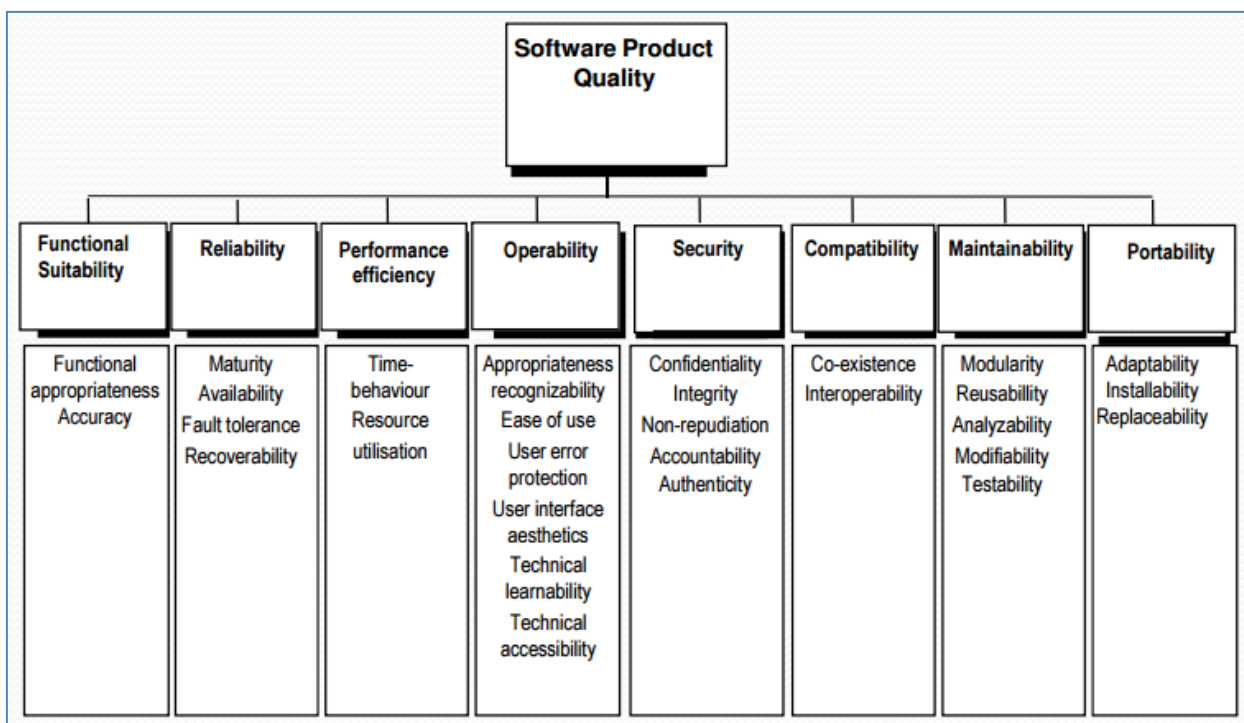
Стандард ISO/IEC 9126 може бити коришћен за вредновање готових софтверских производа и њихово рангирање, ради одлучивања у набавци. У раду [1] приказана је примена стандарда ISO/IEC 9126 и ANSI/IEEE 1209 ради евалуације DataLink алата. У раду [Jung et al, 2004] истражени су ставови различитих учесника у развоју софтвера према значају карактеристика квалитета софтвера дефинисани у стандарду ISO/IEC 9126.



Слика 2.2.4.1.4. Елементи ISO 9126 стандарда за квалитет софтвера [w Desharnais]

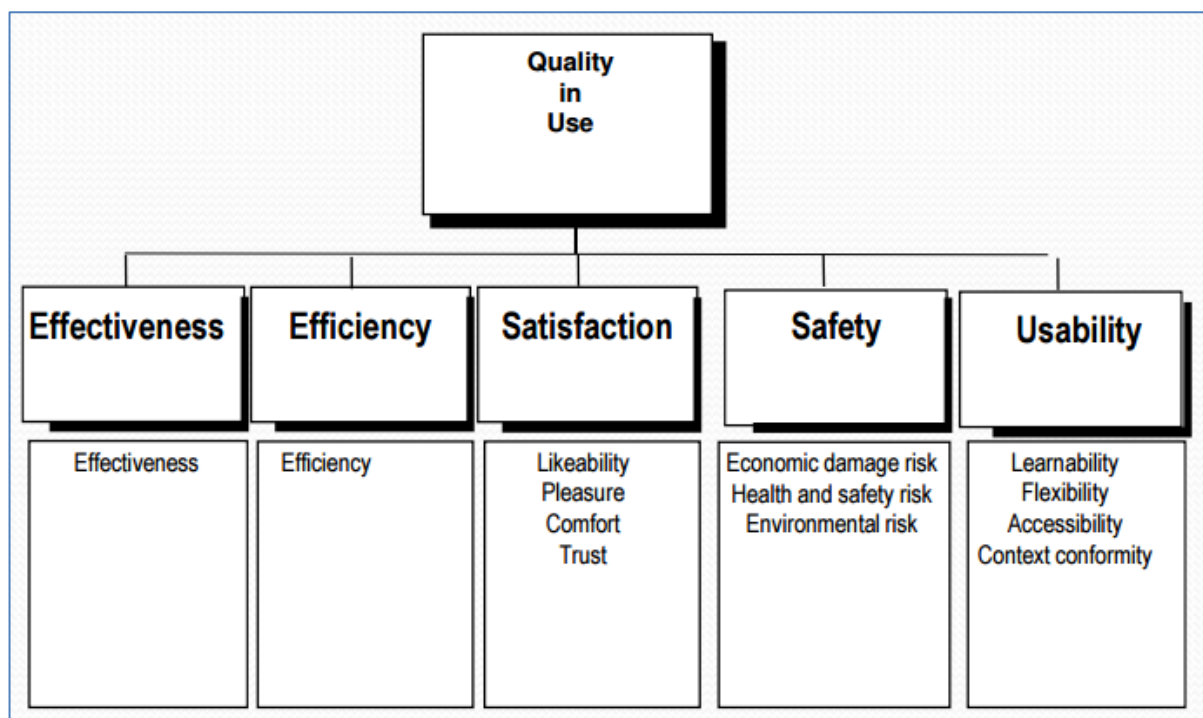


Слика 2.2.4.1.5. Основне компоненте квалитета софтвера, према ISO2010 [w Desharnais]



Слика 2.2.4.1.6. Основне компоненте квалитета софтвера као производа, према ISO2010 [w Desharnais]





Слика 2.2.4.1.7. Основне компоненте квалитета софтвера у употреби, према ISO2010 [w Desharnais]

<i>Characteristics</i>	<i>Inherent</i>	<i>System dependent</i>
Accuracy	X	
Completeness	X	
Consistency	X	
Credibility	X	
Currentness	X	
Accessibility	X	X
Compliance	X	X
Confidentiality	X	X
Efficiency	X	X
Precision	X	X
Traceability	X	X
Understandability	X	X
Availability		X
Portability		X
Recoverability		X

Слика 2.2.4.1.8. Основне компоненте квалитета података, према ISO2010 [w Desharnais]

У раду [Hendriks et al, 2000] анализирани су критеријуми квалитета софтвера према ISO9126 стандарду и утврђено је да нису сви критеријуми једнако важни и примењиви за све типове софтвера. Анкетирано је 7 корисника софтвера који се користи у области

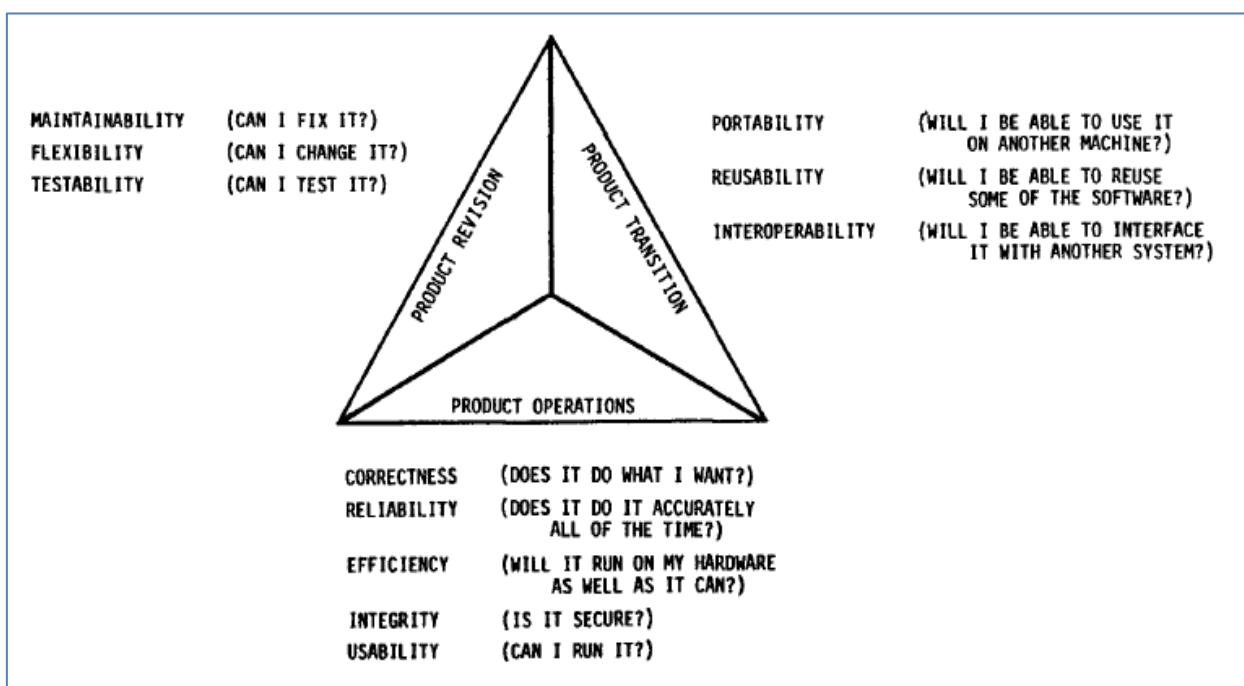
стоног издаваштва и утврђено је да су функционалност, поузданост и могућност одржавања најважније карактеристике.

Квалитет софтвера може да се мери статички, вредновањем елемената у тренутку завршетка реализације или динамички, у току коришћења. Истраживање о могућностима да се вредновање квалитета софтвера може вршити на основу резултата тестирања софтвера, односно на основу извештаја о грешкама у софтверу, описано је у раду [Yu et al, 2011]. У наведеном истраживању постављене су 2 хипотезе о: 1) линеарној корелацији броја измена у претходној верзији софтвера према броју грешака утврђених у наредној верзији софтвера; 2) линеарној корелацији броја грешака у претходној верзији софтвера и времену испоруке наредне верзије софтвера. Утврђено је да измена у величини софтвера нове верзије није у линеарној корелацији са бројем грешака утврђене у претходној верзији софтвера, као и да број грешака претходне верзије софтвера утиче на време испоруке наредне верзије софтвера.

У раду [Parkinson et al, 2008] описан је приступ који, на основу Goal-Question-Metric парадигме [Basili&Rombach, 1988], интегрише Top-Down приступ са Bottom-up приступом у дефинисању квалитета софтвера, који је назван Practitioner-Based Model (PBM), који дефинише низ метрика квалитета софтвера.

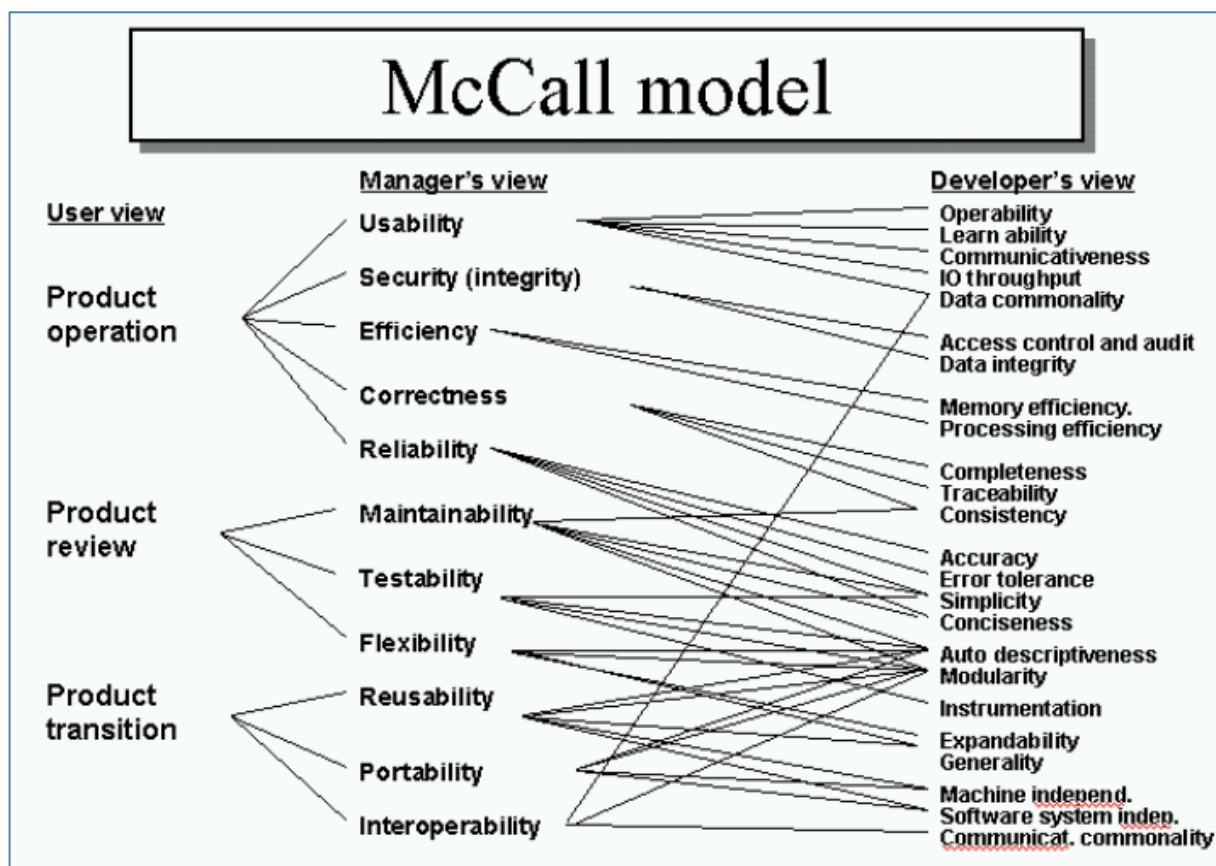
У раду [Graves et al, 2000] описане су методе за предвиђање погрешног рада софтвера, на основу историје измена, где се мерења врше над производом и процесом. У раду [Fenton& Neil, 1999] описан је критички приступ методама за предвиђање софтверских дефеката.

У оквиру [Cavano&McCall, 1978] дефинисан је преилиминарни модел квалитета софтвера, у односу на факторе квалитета (односно менаџер-базиране критеријуме квалитета софтвера).



Слика 2.2.4.1.9. Фактори квалитета софтвера, према Cavano&McCall [Cavano&McCall, 1978]

У раду [Sanz et al, 2005] описан је McCall модел квалитета софтвера, којим се декомпонују и прецизније дефинишу критеријуми и метрике квалитета које су иницијално дате стандардом ISO9126.



Слика 2.2.4.1.10. McCall-ов модел са три нивоа погледа на карактеристике квалитета софтвера, према Sanz et al [Sanz et al, 2005]

У наведеном раду [Sanz et al, 2005], метрике су подељене према учесницима и заинтересованим странама – менаџерски поглед (општије метрике) и поглед учесника у развоју (детаљније и многобројније, прецизније метрике). На овај начин омогућено је прецизније дефинисање мерних величина за које је изведива реализација мерења.

У раду [Porter&Selby, 1990] разматрана је могућност примене софтверских метрика за класификацију софтверских модула у односу на вероватноћу наступања одређених ризика, коришћењем класификационих стабала (classification trees). Описан је приступ, реализован алат за аутоматизацију примене, а примена је илустрована примером.

У раду [Banker et al, 1991] описан је критички приступ према метрикама које се односе на процену трошкова који се заснива на пребројавању функција и функционалним поенима и предлаже приступ који би се заснивао на пребројавању објеката и коришћењу тзв. функционалних поена у оквиру примене CASE (Computer Aided Software Engineering) развојног окружења. У раду [Banker et al, 1994] разматране су могућности аутоматизације примене софтверских метрика у циљу мерења продуктивности у развоју софтвера, у оквиру система репозиторијума резултата CASE алата.

У раду [Jadhav&Sonar, 2009] представљена је систематска анализа литературе у области евалуације софтвера. Представљени су сумарни резултати постојећих истраживања: сви анализирани радови односе на предлоге критеријума, техника и методологија евалуација конкретних типова софтвера; преко 70% радова предлажу критеријуме за евалуацију софтвера, преко 50% радова предлажу евалуационе технике, преко 40% радова предлажу методологије за селекцију софтвера и 10% радова описује развијене системе и алате који се користе у сврху евалуације других софтвера. Критеријуми за евалуацију софтвера систематизовани су у следеће категорије: критеријуми који се односе на функционалне карактеристике софтверског

пакета, карактеристике које се односе на излаз из система, критеријуме у односу на карактеристике квалитета софтвера (усклађеност са стандардима), карактеристике које се односе на добављача и пратеће материјале (корисничко упутство, туторијал, демо, услуге обуке, услуге одржавања, консултантске услуге, референце добављача, време одговора на захтев за помоћи или сервисирањем), критеријуме који се односе на трошкове и корисности система) итд. У наставку ће бити дат приказ детаља прве две групе критеријума:

Табела 2.2.4.1.1. Функционалне карактеристике као критеријуми вредновања софтверског пакета [Jadhav&Sonar, 2009]

Criteria related to functional characteristics of the software package		
Criteria	Criteria group	Criteria meaning
Included functionality	Functional	Areas or functions of the company that the software has to serve. It describes how software package covers each function
Main target	Functional	Functional area(s) for which software is specially oriented or strong
Completeness	Functional	It is defined as degree to which software satisfies functional requirements
Adaptability	Functional	Possible level of customization in general and for the specific company
Openness	Functional	Level of openness to additional development (internal and external) and to other existing applications
Interoperability	Functional	Capability to integrate with other tools and applications
Security levels	Functional	Breadth of security policies supported by the software package (user identification, auditing, data encryption)
Number of simultaneous users	Functional	Number of simultaneous users that can be linked and served by the system

Табела 2.2.4.1.2. Карактеристике квалитета софтвера као критеријуми вредновања софтверског пакета [Jadhav&Sonar, 2009]

Criteria related to software quality characteristics		
Criteria	Criteria group	Criteria meaning
Vertical solutions	Personalizability	Number of customized versions of a package accommodating the typical requirements of a specific industry
Customizable fields	Personalizability	Ability to personalize the layout of package interface
Customizable reports	Personalizability	Ability to personalize the layout of reports produced by package
Interface type	Personalizability	Interface type of the package
Programming languages	Personalizability	Ability to personalize modules by programming languages
Middleware standard	Portability	Breadth of middleware standard that are supported by software package (CORBA, DCOM, RMI, ODBC, JDBC, OLE-DB)
DBMS standards	Portability	Breadth of database management systems that can be accessed by software package (SQL server, oracle, DB2, Sybase, Informics)
Communication standards	Portability	Inter-organizational data exchange standards that are supported by software package (EDI, XML)
Platform variety	Portability	Capability of the software package to run on wide variety of computer platforms
Number of modules	Maintainability	Average size of independent code units
Number of independently installable modules	Maintainability	Level of independence among modules
Number of workstations	Maintainability	Maximum number of users that can be supported
Maximum number of distribution tiers	Maintainability	Ability to split package into separate application that can be distributed onto different servers
Number of modules that can be installed on separate servers	Maintainability	Ability to distribute modules on different servers
Scalability	Maintainability	Ability of the software package to handle increasing number of users and higher load of transaction
User interface	Usability	Ease with which user can use interface of the software package
User types	Usability	Ability of the software package to support beginners, intermediate, and advanced users or a combination of user types
Data visualization	Usability	Capability of the software package to present data effectively
Error reporting	Usability	Error reporting and messaging ability of the software package
Domain variety	Usability	Capability of the software package to be used in different industries to solve different kinds of business problems
Ease of use	Usability	Ease with which user can learn and operate the software package
Robustness	Reliability	Capability of the software package to run consistently without crashing
Backup and recovery	Reliability	Capability of the software package to support backup and recovery feature
Time behavior	Efficiency	Ability of the software package to produce results in reasonable amount of time relative to data size

### 2.2.4.2. Мерење успеха и квалитета софтверског производа посебних технологија

Мерење квалитета софтвера, тј. избор метрика зависи од одабране технологије којом је реализован софтвер [Pandey et al, 2014] [Cavano&McCall, 1978]. У оквиру истраживања [Pandey et al, 2014], приказани су различити могући сценарији избора метрика за вредновање у процесу миграција софтверске апликације из старих у нове технологије.

Истраживање зависности квалитета open-source софтвера (OSS) [Lee et al, 2009] извршено је у оквиру 145 упитника са питањима, усклађене са DeLone и McLean моделом вредновања квалитета информационог система, која су се односила на елементе потребне за утврђивање статуса хипотеза: 1) Квалитет OSS позитивно утиче на његово коришћење. 2) Квалитет услуге у вези OSS софтвера позитивно утиче на његово коришћење. 3) Квалитет софтвера позитивно утиче на задовољство корисника.

4) Квалитет услуге у вези OSS софтвера позитивно утиче на задовољство корисника. 5) Задовољство корисника позитивно утиче на коришћење OSS софтвера. 6) Задовољство корисника OSS софтвером позитивно утиче на индивидуалне користи у оквиру радне делатности. 7) Коришћење OSS софтвера позитивно утиче на индивидуалне користи у оквиру радне делатности. Утврђено је да је коришћење OSS софтвера одређено задовољством корисника и квалитетом софтвера, али квалитет услуге у вези OSS софтвера нема значајан ефекат на коришћење OSS софтвера.

С обзиром на савремене трендове коришћења компонент-базираног развоја софтвера, у истраживању [Gao et al, 2004] представљено је мерење перформанси софтверских компоненти. Описани су разлози за извршавање мерења перформанси (утврђивање нивоа усаглашености са захтевима, утврђивање капацитета у складу са потребама корисника, утврђивање проблема), дефинисане су метрике (време процесирања, доступност, поузданост, скалабилност), које су проверене у оквиру студије случаја.

У оквиру докторске дисертације [Ressmann, 2003] дефинисане су метрике којима се врши мониторинг успешности извршавања дистрибуиране софтверске апликације. У раду [Marinescu et al, 1990] описани су системи и технологије које служе за мониторинг извршавања дистрибуираних софтверских система.

У оквиру рада [Lilburne et al, 2004] описан је метрички систем који се односи на мерење квалитета и метрике квалитета Web апликација. Квалитет се може посматрати са два аспекта: са аспекта програмера (учесника у развоју Web апликације) и аспекта корисника Web апликације. У наведеном раду примењен је хијерархијски приступ: циљ квалитета -> критеријуми квалитета у односу на циљ -> метрике. На овај начин је дефинисан QCF (Quality Compliance Framework) метрички framework који представља флексибилни оквир за мерење квалитета у разним фазама развоја софтвера. Суштина метрика које су коришћене у овом раду односи се на метрике из ISO9126 стандарда.

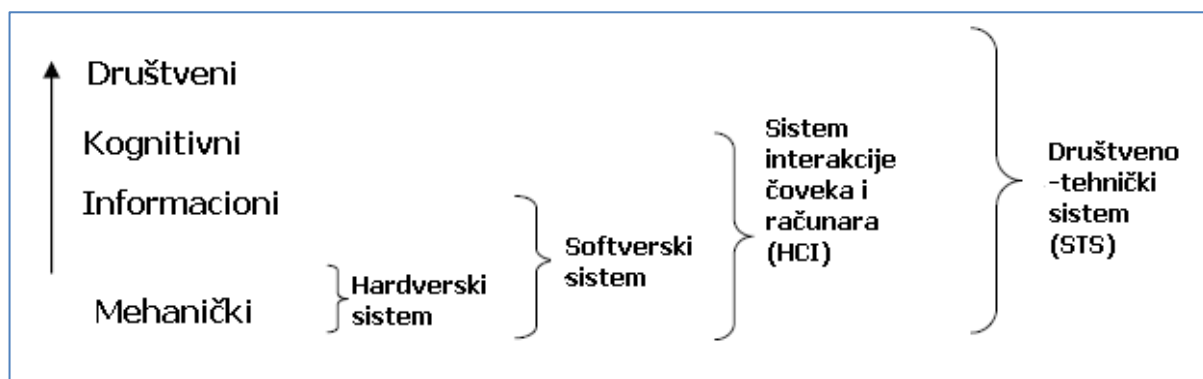
#### **2.2.4.3. Мерење квалитета информационог система**

У претходној декади, актуелна су истраживања у области евалуације информационих система кроз дефинисање критеријума евалуације и емпиријску евалуацију предложених метричких модела.

Евалуација информационих система у раду [Beunon-Davies et al, 2000] повезана је са животним циклусом развоја информационог система и дефинисана је кроз четири нивоа:

1. Стратешка евалуација – евалуација концепта система пре почетка имплементације, провером усклађености са стратешким циљевима
2. Формативна евалуација – упоређивање система са захтевима и циљевима у току реализације, са могућношћу да неусаглашености утичу на измене система
3. Сумативна евалуација – евалуација система након завршетка реализације, упоређивањем са циљевима, захтевима итд.
4. Пост-мортем анализа – анализа пројеката који су напуштени, односно неуспешних пројеката

Whitworth&Zaic [Whitworth&Zaic, 2003] дефинишу четири елемента система који утичу на перформансе система: граница система, интерна структура (подршка и координација активности система), рецептори (примају информације из окружења) и ефектори (извршавају акције над окружењем). У функцији евалуације информационог система дат је приступ информационом систему кроз 4 нивоа архитектуре информационог система.



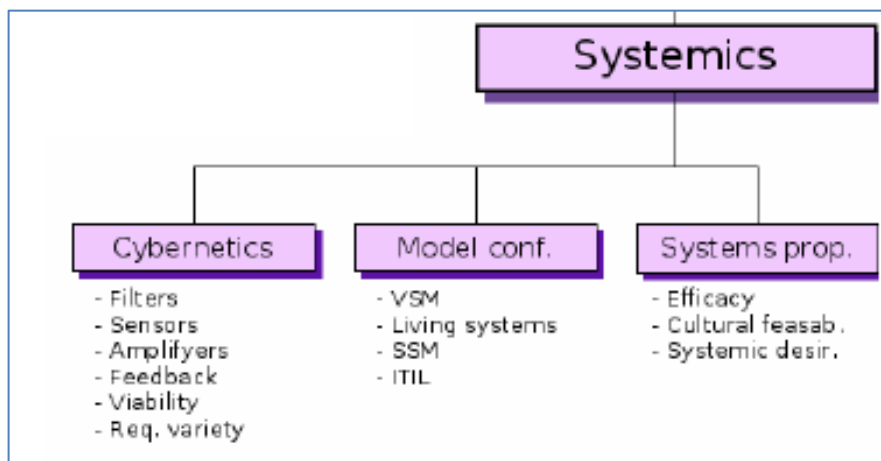
Слика 2.2.4.3.1. Нивои архитектуре информационог система према Whitworth&Zaic [Whitworth&Zaic, 2003]

Да би се минимизовао ризик од недостатка и грешака информационог система, Whitworth&Zaic [Whitworth&Zaic, 2003] дефинишу 8 особина дизајна информационог система који чине WOSP модел перформанси система. Активне особине чине систем успешним, а недостатак пасивних особина могу изазвати грешке.

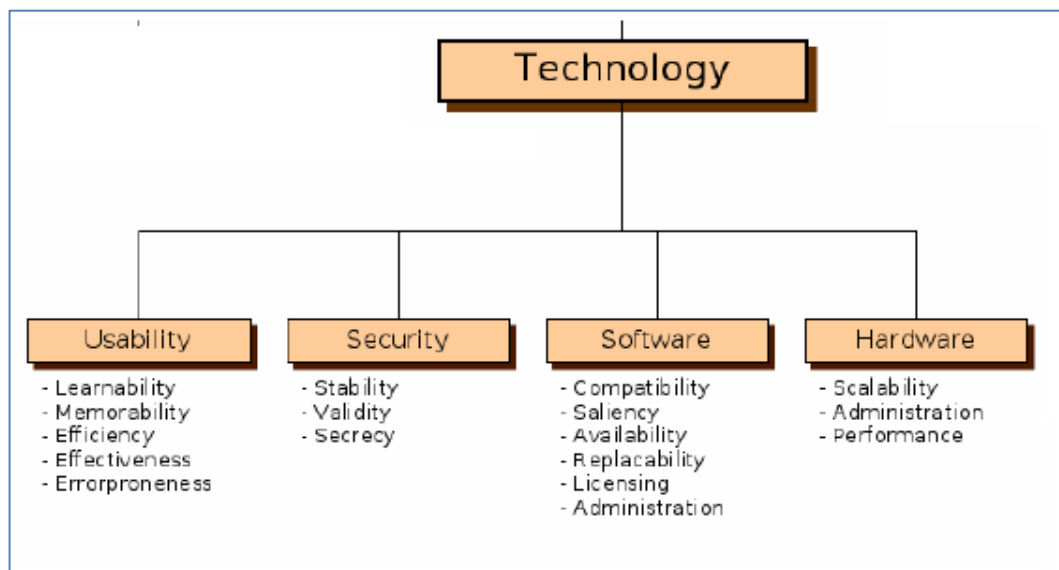
Табела 2.2.4.3.1. Елементи WOSP модела за мерење перформанси информационог система, према Whitworth&Zaic [Whitworth&Zaic, 2003]

ЕЛЕМЕНТ	ОСОБИНА	ТИП ОСОБИНЕ
Граница система	Безбедност (security)	Пасивна
	Проширивост (extendibility)	Активна
Структура система	Поузданост (reliability)	Пасивна
	Флексибилност (flexibility)	Активна
Рецептор	Повезивост (connectivity)	Активна
	Поверење (confidentiality)	Пасивна
Ефектор	Функционалност (functionality)	Активна
	Применљивост (usability)	Пасивна

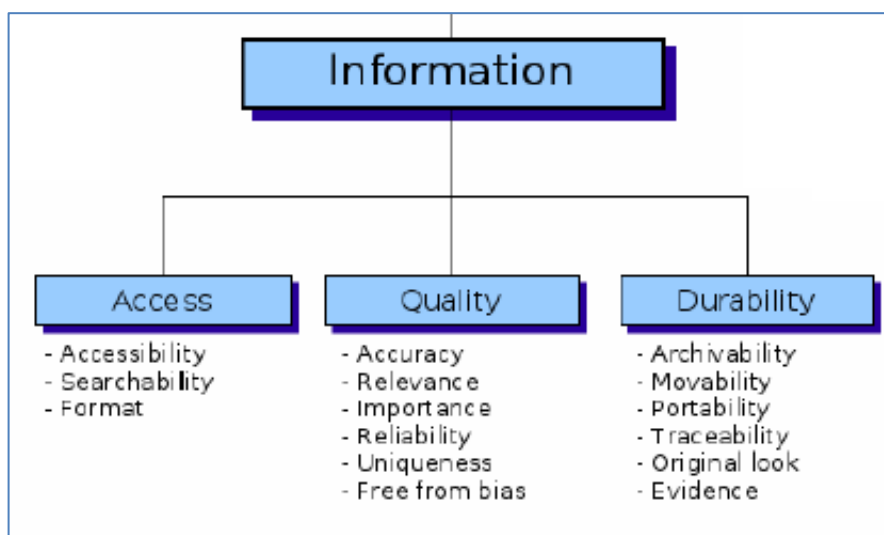
У раду [Palmius, 2007] приказани су критеријуми за евалуацију појединачних информационог система и њихову компарацију путем мерљивих особина. У наведеном раду издвојено је да се најчешћи критеријуми, из претходно анализираних литературе, односе на групе критеријума које изражавају потребу за економском корисношћу, употребљивости са становишта корисника и мерења задовољства корисника реализованим решењем. У раду [Irani, 2002] описан је приступ евалуацији информационог система са становишта стратегијског менаџмента, оперативних потреба пословних процеса и финансијског аспекта кроз критеријуме којима се дефинише допринос информационог система одговарајућим стратешким, оперативним и финансијским циљевима.



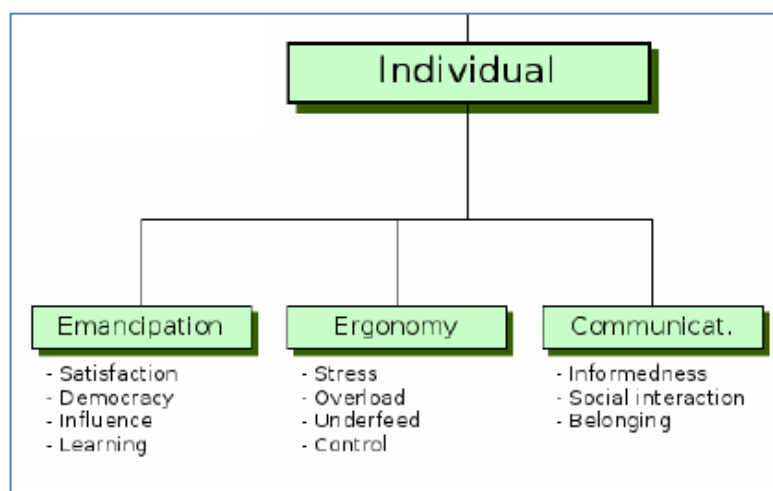
Слика 2.2.4.3.2. Системска група критеријума у евалуацији информационог система, према [Palmius, 2007]



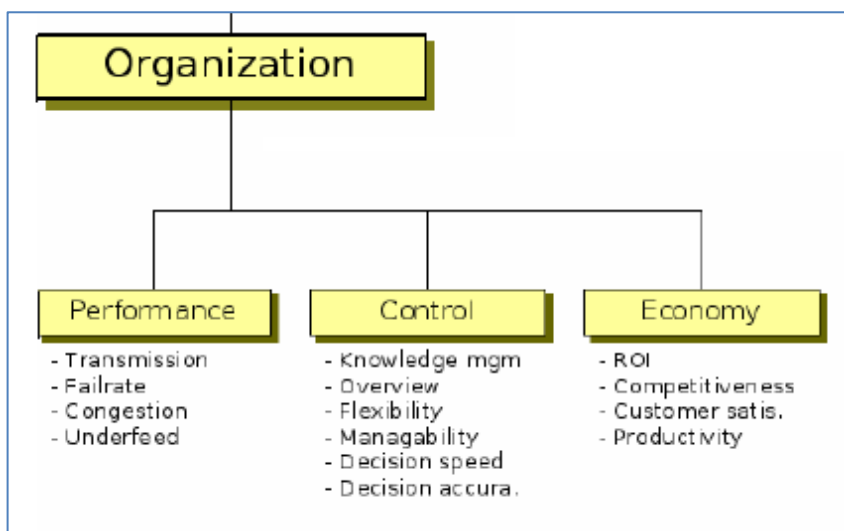
Слика 2.2.4.3.3. Технолошка група критеријума у евалуацији информационих система, према [Palmius, 2007]



Слика 2.2.4.3.4. Информациона група критеријума у евалуацији информационих система, према [Palmius, 2007]



Слика 2.2.4.3.5. Група критеријума која се односи на појединце (особе, људски фактор) у евалуацији информационих система, према [Palmius, 2007]



Слика 2.2.4.3.6. Организациона група критеријума у евалуацији информационих система, према [Palmius, 2007]

#### 2.2.4.4. Мерење квалитета осталих артефакта у развоју софтвера

Поред програма (апликације) који у ужем смислу представља софтвер, у ширем смислу постоји низ других артефакта који припада заједничкој категорији софтвера, односно артефакта који настају у процесу развоја софтверске апликације – програма. Мерење квалитета тих артефакта је такође значајно. У оквиру модел-базираног приступа развоју софтвера, модели су заправо примарни артефакти, на основу којих се реализује програмски код, документација и тестирање [Monperrus et al, 2008].

У раду [Monperrus et al, 2008] описана је генеричка метрика  $\sigma$  као општа метрика која се у конкретној имплементацији специјализира и конкретизује. У наведеном раду модел општег типа је дефинисан као усмерени граф. Примена наведене генеричке метрике извршена је над мета-моделима, као и конкретним моделима - трансформисан је јава код у моделе (дијаграме) различитог типа и реализована су модел-базирана мерења.

У оквиру рада [Kazi et al, ICDQM, 2011] извршена је анализа приступа евалуацији модела у развоју информационих система. Тип модела одређује приступе евалуацији [Belle J-P, 2006], тако да разликујемо:

1. Статичке и динамичке моделе
2. Мета моделе, узорке (pattern) и конкретне моделе.

Приступи евалуацији модела могу се класификовати [Belle J-P, 2006]:

1. Интерна (један модел) и екстерна евалуација (компарација више различитих модела)
2. Апсолутно мерење (теоријски засновано, објективно и технички изводљиво) и релативно мерење (примењени, субјективно и пословно орјентисани фактори).
3. Van Belle [Belle J-P, 2006] даје оквир (кроз аспекте, критеријуме и метрике) за оцену модела са синтаксног, семантичког и прагматичког аспекта. Синтаксни аспект се односи на форму и структуру, семантички на значење и усклађеност са доменом реалног света на који се односи, а прагматички на употребу модела у контексту окружења.

Tabela 2.2.4.4.1. Van Belle-ов оквир за евалуацију модела у области информационих система [Belle J-P, 2006]

АСПЕКТ	КРИТЕРИЈУМ (метрике)
Синтаксни аспект	Величина (број концепата)
	Коректност, непостојање грешака, интегритет, конзистентност, усклађеност са стандардима
	Модуларност (број група и нивоа дијаграма)
	Структура, хијерархија, степен поновне искористивости

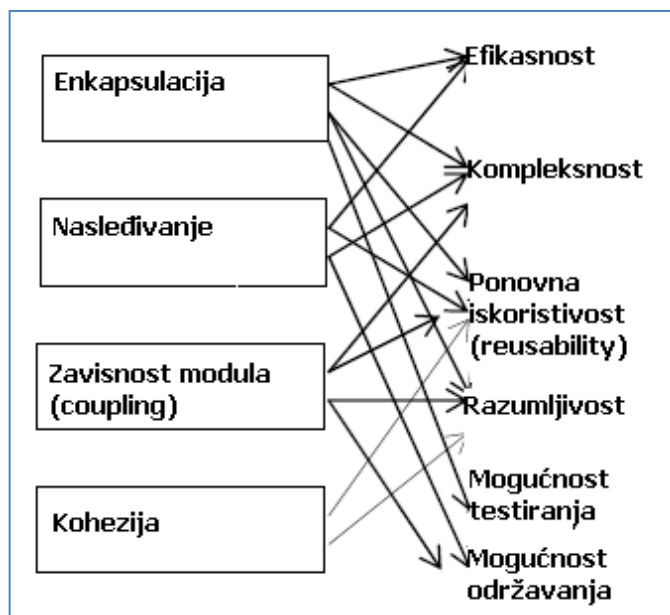


	Комплексност, густина
	Архитектурни стил
<b>Семантички аспект</b>	Генеричност (усклађеност са доменом)
	Покривеност (покривеност домена и основних концепата)
	Комплетност (степен покривености речника -exicon)
	Ефикасност, концизност
	Експресивност
	Сличност и преклапање са другим сличним моделима
	Разумљивост, читљивост
<b>Прагматички аспект</b>	Документација (комплетност, проширивост, читљивост)
	Валидност, прихваћеност од стране клијента и корисника
	Флексибилност, проширивост, адаптивност
	Зрелост, усклађеност са садашњим стањем (currency)
	Усклађеност са сврхом, циљем, релевантношћу, да одговара систему
	Доступност (availability)

Vanderfeesten et al [Vanderfeesten et al, 2008] путем компарације елемената софтверских програма и пословних процеса користи метрике из домена софтверског инжењерства у евалуацији модела процеса и предлаже 5 кључних принципа правилног дизајна:

- Величина – треба тежити што мањим модулима
- Модуларност – треба тежити што већој модуларности
- Комплексност – треба тежити што једноставнијем дизајну
- Интерна повезаност (Cohesion) – повезаност елемената у модулу треба да је што већа
- Екстерна повезаност (Coupling) – повезаност између различитих модула треба да је што мања.

У области евалуације модела података, поред теоријски заснованих методолошких приступа, дефинисана су и хеуристичка упутства правилном дизајну [Riel, 1996]. У контексту објектно-орјентисаног дизајна, Khan et al [Khan et al, 2007] дефинише приступ евалуацији појединачних класа приступом WCC (Weighted Class Complexity), заснован на кључним особинама, као што је приказано на следећој слици:



Слика 2.2.4.4.1. Кључне квалитативне особине у евалуацији модела класа према Khan et al [Khan et al, 2007]

С обзиром да је једна од кључних активности у развоју софтвера информационог система заправо концептуално моделовање, највећи број истраживања у области дизајна модела података односе се на концептуалне моделе података [Piatini et al,

2000]. Квантитативне метрике [Piattini et al, 2000] заснивају се на броју ентитета, атрибута и релација, при чему се разликују специфичне врсте елемената – функционално зависни атрибути, изведени атрибути, редунданса, композитни атрибути, релације различитих кардиналитета итд. Квалитативне метрике се одређују субјективно у односу на квалитет, односно кључне факторе који утичу на квалитет модела. Moody [Moody, 1998] наводи факторе: комплетност, интегритет, флексибилност, разумљивост, коректност, једноставност, интеграција, могућност имплементације. Pirani [Pirani, 2008] наводи факторе: тачност, комплетност, конзистентност, прецизност, поузданост, временска усклађеност, правовременост, јединственост, валидност. Kesh [Kesh, 1995] дефинише метрике квалитета модела у односу на онтолошку и бихејвиоралну компоненту. Онтолошке компоненте се односе на структуру (погодност, стабилност, конзистентност) и садржај (комплетност, кохезивност, валидност). Бихејвиорална компонента се односи на: употребљивост са корисничког и дизајнерског аспекта, могућност одржавања, тачност и перформансе.

## 2.2.5. Анализа постојећих технолошких решења

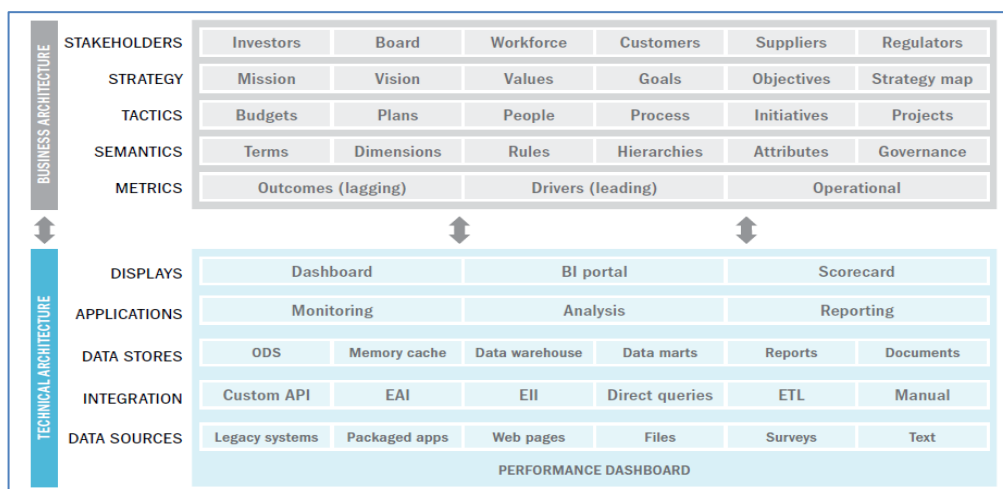
### 2.2.5.1. Системи за управљање перформансама организације и подршку одлучивању

Систем за управљање перформансама (Performance Management System – PMS) може бити дефинисан као „Скуп метода, процеса, информационих система и способности запослених, фокусираних на задатке прикупљања, усаглашавања, чувања, аналитичког процесирања и презентовања информација, које су критичне за информациону транспарентност организације и стратешко доношење одлука које се извршава од стране екстерних и интерних заинтересованих страна“ [Isaev, 2010], односно „еволуирајући формални и неформални механизми, процеси, системи и мреже коришћене од стране организације ради додељивања кључних задатака и циљева запосленима од стране менаџмента, за подршку стратешким процесима и текућем управљању кроз анализу, планирање, мерење, контролу, награђивање и шире управљање перформансама као и за подршку организационом учењу и променама“ [Ferreira&Otle, 2009]. Подршка информационих технологија одлучивању у организацијама кроз историју развијала се почев од система за подршку одлучивању, data warehouse и business intelligence системе којачно до система подршке управљању пословним перформансама (слика 2.2.5.1.1.).

Decision support system (DSS)	A computer-based support for management decision makers who are dealing with semistructured problems (Keen and Scott Morton, 1978)
Executive information system (EIS)	A computer-based system that serves the information needs of top executives (Turban, Aronson, and Liang, 2005)
Data warehouse (DW)	A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of decision making (Inmon, 1992)
Business intelligence (BI)	A broad category of applications and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make better business decisions (Turban, Aronson, and Liang, 2005)
Business Performance Management (BPM)	A series of business processes and applications designed to optimize both the development and the execution of business strategy

Слика 2.2.5.1.1. Историјски развој софтверске подршке одлучивању, према Frolick&Ariyachandra [Frolick&Ariyachandra, 2006]

Систем за управљање перформансама подржава животни циклус процеса управљања перформансама. Састоји се од повезане пословне и техничке архитектуре [Eckerson, 2009], као што је приказано на слици 2.2.5.1.2.



Слика 2.2.5.1.2. Структура система за подршку управљању перформансама према Eckerson-у [Eckerson, 2009]

### 2.2.5.2. Мерење перформанси дистрибуираних ИТ система и софтвера – приказ неких решења

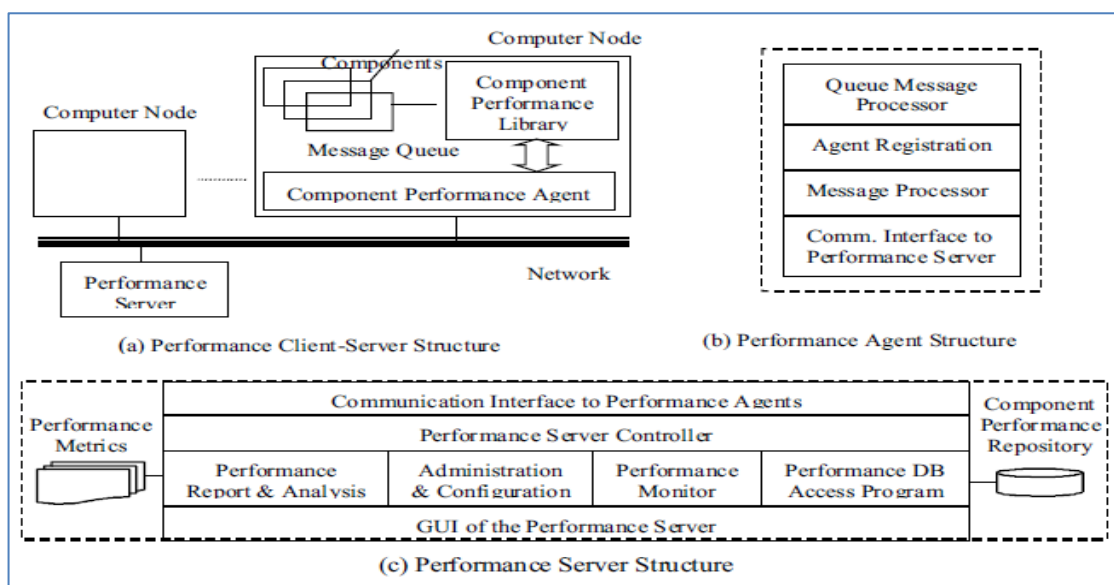
На основу дефинисаних нивоа зрелости управљања ИТ процесима (Gartner Group [Hill& Thomson, 2001] - „IT Management Process Maturity“) креирани су адекватни системи који омогућавају мониторинг ИТ процеса, а који се разликују према подршци различитим нивоима зрелости управљања:

- Event Consoles – упозоравају систем администраторе на проблем или потенцијални проблем.
- Performance Monitors – приказују детаљне информације о томе шта се дешава
- Historical Analysis Tools – приказују историјске податке, садашње трендове и предвиђају будуће вероватне догађаје.
- Capacity Planning Tools – предвиђају ефекте будућих пословних оптерећења и мењају (аутоматски) конфигурацију система.

У оквиру истраживања [Haban& Wybranietz, 1989], представљена су решења за мониторинг дистрибуираних система, који омогућује мерење перформанси и надгледање понашања дистрибуираног система у току извршавања. Наведени систем омогућава прикупљање, анализу и презентовање података који су прикупљени о раду 9 чворова. Дистрибуирани систем се састоји из дистрибуираног хардвера, оперативног система, програмских језика за развој дистрибуираних апликација (у наведеном раду коришћен је језик објектно оријентисаног програмирања LADY) и софтверских апликација. Представљено решење је хибридно (комбинација хардверске и софтверске компоненте мониторинга).

У раду [Delgado et al, 2004] представљена је анализа истраживања у области алата који служе за мониторинг за утврђивање погрешног рада софтвера у току извршавања (runtime software-fault monitoring). Дата је таксономија појмова из наведене области и представљени су алати из ове категорије, који су приказани у оквиру истраживања.

У раду [Gao et al, 2004] представљен је систем за евалуацију перформанси дистрибуираних софтверских компоненти. Систем се састоји из сервера перформанси, који прикупља податке од агената мониторинга перформанси, складишти податке, анализира и врши калкулације у складу са метрикама, омогућава администрацију система и извештавање. На сваком чвору (рачунарској радној станици) налази се агент мониторинга перформанси, који доставља податке о перформансама компоненте серверу путем низа порука.



Слика 2.2.5.2.1. Архитектура система за евалуацију перформанси дистрибуираних софтверских компоненти, према Gao et al [Gao et al, 2004]

### **2.2.5.3. Системи за мониторинг развоја софтвера**

У раду [Wu et al, 2012] приказан је систем Caribou који омогућава мониторинг перформанси процеса развоја софтвера који се одвија у дистрибуираном окружењу. Наведени систем базиран је на Web подршци развоју и агентима, који омогућавају мониторинг параметара реализације пројекта. У наведеном раду дефинисан је развојни оквир примене система Caribou који садржи дефиницију метрика, формула за израчунавање изведених вредности, као и визуализацију напретка пројекта.

У раду [German&Mockus, 2003] представљен је систем SoftChange који је намењен аутоматизованим мерењима у оквиру Open Source пројеката. Користи податке из mailing lista, као и податке о промени фајлова, из CVS log фајлова, ChangeLog фајлова, односно податке из система Bugzilla. Наведени систем се састоји из низа скрипт фајлова који врше екстракцију и преузимање изворних података, сумирање и валидацију. Наведени систем има могућност да утврди где су извршене измене у изворном коду. Примена наведеног система илустрована је на примеру развоја и измена софтвера Evolution.

Поједине фирме развијају алате које нуде као сопствени производ тржишту, а односе се на алате за подршку мониторингу и вредновању квалитета софтвера и софтверских пројеката. Фирма Agitar Technologies описује [Agitar] систем AgitarOne Management Dashboard као систем који обезбеђује скупове метрика који омогућавају учесницима пројекта (менаџменту и запосленима у развоју) да разумеју и пратек статус квалитета Java софтверских пројеката. Систем реализује мерење и процену ризика софтвера (односно поузданост програмског кода), базирану на оцени комплексности кода, као и покривености тестирањем (пребројавањем линија кода или делова кода који су тестирани и изражавањем у процентима). На овај начин визуално представља делове кода на које треба посебно обратити пажњу и унапредити ниво поузданости додатним тестирањем или смањивањем сложености.

У раду [Suri et al, 2009] представљен је симулатор (тачније, математичка основа симулатора, који се заснива на Марковом процесу), који омогућава процену ризика у оквиру мерења перформанси софтверског пројекта.

## **2.3. Адаптивност у реализацији и управљању агилним софтверским пројектима**

### **2.3.1. Адаптивност у одржавању софтвера и закони еволуције софтвера**

Појам адаптивности јавља се и у оквиру активности одржавања софтвера, тј. у активностима након испоруке. Е.В. Swanson је у оквиру рада [Swanson, 1976] иницијално дефинисао три категорије одржавања: корективно, адаптивно и перфективно. У оквиру стандарда ISO/IEC 14764 дефинисано је 4 категорије одржавања:

- Корективно одржавање: реактивне модификације софтверског производа које се реализују након испоруке како би се кориговали утврђени проблеми.
- Адаптивно одржавање: Модификација софтверског производа извршена након испоруке како би се одржао софтверски производ употребљивим у измењеном окружењу или окружењу које се стално мења.
- Перфективно одржавање: Модификације софтверског производа након испоруке како би се унапредиле перформансе или могућност одржавања.
- Превентивно одржавање: Модификација софтверског производа након испоруке да бисе детектовале и кориговале латентне грешке у софтверском производу пре него што постану ефективне грешке.

Према [Lehman, 1996], формулисано је 8 закона еволуције софтвера, који су раније дефинисани у [Lehman, 1978a] [Lehman, 1978b] [Lehman, 1980a] [Lehman, 1980b] [Lehman, 1991] на основу анализе података које су првобитно прикупљене у току студије IBM процеса програмирања [Lehman, 1969]. Закони еволуције софтвера односе се на програме е-Типа, односно „софтверске системе који решавају проблеме примене рачунара у реалном свету“ [Lehman, 1996].

1. закон- Рачунарски програми е-Типа који се користе морају се континуирано адаптирати, иначе постају прогресивно мање задовољавајући. Овај закон говори о потреби за констаном повратном информацијом о успешности употребе система и контролисаним одржавањем, како би се одржало задовољство корисника система на одговарајућем нивоу.

2. закон - Како програм еволуира, расте његова комплексност, осим уколико се не реализују активности одржавања и смањења комплексности

3. закон - Процес еволуције програма је саморегулишући, са дистрибуцијом мерења која су блиска нормалној дистрибуцији вредности мерења атрибута процеса и производа.

4. закон – Просечна ефективна глобална стопа активности које се односе на еволуирајући систем је непроменљива у току животног циклуса производа.

5. закон – У току активног живота еволуирајућег програма, садржај сукцесивних испорука је статистички непроменљив.

6. закон – Функционални садржај програма мора стално да расте како би се одржавало задовољство корисника у току животног циклуса програма.

7. закон – Програми е-Типа ће бити „доживљени од стране корисника“ као програми чији квалитет опада уколико се ригорозно не одржавају и адаптирају на променљиво операционо окружење.

8. закон – Процес програмирања у креирању програма е-Типа састоји се од више циклуса и вишенивовске повратне спреге и мора се тако третирати како би успешно били модификовани или унапређени.

### **2.3.2. Карактеристике агилног приступа развоју софтвера, адаптивност и истраживања мерења агилности**

Термин „агилност“ дефинисана је у [Wendler, 2013] на основу дефиниција из [Ganguly et al, 2009] и [Dove, 1999] као „ефективна интеграција способности одговора и управљање знањем у циљу да се брзо, ефикасно и прецизно прилагоди (у проактивном

и реактивном смислу) било којој неочекиваној (или непредвиђеној) промени у пословању/потребама или могућностима клијента без компромиса који би се односио на трошкове или квалитет производа/процеса." Агилност је блиско повезана са флексибилношћу и сужавањем („leanness“), међутим треба ове термине сматрати концептима у оквиру агилности као ширег појма. У оквиру [Conboy&Fitzgerald, 2004] термин агилност разматран је са историјског аспекта, где је приказано да је овај термин уведен прво у области менаџмента, односно истраживања у области пословних система као термин агилне производње, а тек касније је уведен у софтверску индустрију. У оквиру [Conboy&Fitzgerald, 2004] термин агилности је упоређен са термином флексибилност („способност за прилагођавање на промене“), где разликује проактивну и реактивну флексибилност. Агилност као термин укључује флексибилност и временску димензију (брзину одговора на промене), а такође и одбацивање сувишног, „урадити више са мање ресурса“, економичност и једноставност – „leanness“. Коначно, дефиниција агилности према [Conboy&Fitzgerald, 2004] је: „континуална спремност ентитета да брзо и инхерентно, проактивно или реактивно, прихвати промену кроз високо квалитетне, једноставне, економичне компоненте и релације са окружењем“.

У оквиру истраживања [Wendler, 2013] извршена је систематизација 28 фрејмворка и модела који описују концепте који одређују појам агилности и концепте који припадају том ширем појму као елементе којима би се могла мерити агилност. Општи појам агилности је декомпозицијом на 33 концепата категорисан кроз четири домена: Агилну производњу, Агилни развој софтвера, Агилну организацију и Агилну радну снагу, односно кроз пет домена: организациона култура, технологија, радна снага, клијент, организационе способности.

Примена агилног приступа развоју софтвера формално је започела 2001. године, дефинисањем принципа агилног развоја од стране Agile Software Development Alliance [Agile SD Alliance] кроз тзв. Agile Manifesto [AgileManifesto, 2001]. Међународне организације које се баве стандардима препознале су агилни приступ као приступ који води ка успеху софтверског пројекта и дефинисале низ стандарда који укључује агилни приступ, као што је ISO/IEC/IEEE 26515:2012 Systems and Software Engineering – Developing User Documentation in an Agile Environment.

Агилни приступ развоју софтвера подразумева следеће четири основне вредности [AgileManifesto, 2001]:

- Појединци и интеракције су важније од процеса и алата
- Софтвер који функционише је важнији од детаљне документације
- Сарадња са клијентом је важнија од преговора у односу на уговор
- Одговарати на промене је важније него бити доследан плану
- 

У складу са наведеним основним вредностима агилног приступа, дефинисано је 12 принципа агилног развоја софтвера [AgileManifesto, 2001]:

1. Наш највиши приоритет је задовољити клијента кроз ране и континуиране испоруке корисног софтвера..
2. Радо прихватити („поздравити“) измене захтева, чак и касно у развоју. Агилни процеси користе промене као клијентову конкурентску предност.
3. Испоручити радни софтвер често, у фреквенцијама од неколико недеља до неколико месеци, са настојањем да интервали буду што краћи.
4. Људи из пословног окружења и људи у развоју морају радити заједно сваког дана у току пројекта.
5. Градити пројекте око мотивисаних појединаца. Дати им радно окружење и подршку која им је потребна и веровати им да ће завршити посао.
6. Најефикаснији и ефектнији метод прослеђивања информација у развојном тиму је конверзација „лицем у лице“.
7. Софтвер који функционише је примарна мера прогреса.
8. Агилни процеси промовишу одрживи развој.

9. Спонзори, девелопери и корисници би требали да одржавају константну комуникацију бесконачно.
10. Континуално обраћање пажње на техничку изврсност и добар дизајн унапређује агилност.
11. Једноставност, тј. уметност максимизирања посла који није урађен, је кључна.
12. Најбоље архитектуре, захтеви и дизајн долази од самоорганизујућих тимова.
13. У редовним интервалима, тим размишља о тиме како да постане ефективнији, затим подешава и усклађује понашање у складу са тиме.

Према [Alleman, 2002], агилне методе у развоју софтвера имају следеће карактеристике:

1. Инкременталне, еволутивне и адаптирају се на промене у екстерним и интерним догађајима
2. Модуларне и сажете („lean“) омогућавајући да се компоненте процеса (или решења) постављају или уклањају у зависности од специфичних потреба учесника и заинтересованих страна
3. Базиране на времену – укључују конкурентне и итеративне радне циклусе.
4. Самоорганизујуће - у нормативном смислу, не нуде много упутства у смислу структуре и контроле. Агилне методе се заснивају на хеуристикама и партиципативним процесима, пре него на нормативним и рационалним методама и упутствима.

У раду [Bose, 2008] представљене су основне карактеристике и предности агилних метода развоја софтвера, као што је приказано следећом табелом.

Табела 2.3.2.1. Карактеристике агилних метода развоја софтвера, према Boose [Bose, 2008]

Особина	Предности
Континуално прикупљање захтева	Клијенти одлажу одлуке о кључним елементима, софтвер остаје флексибилан
Фреквентна „лицем у лице“ интеракција	Превазилазе се неспоразуми, гради се поверење између чланова тима
Програмирање у пару	Лакши тимски рад, боље власништво над кодом
Рефакторисање	Постепено унапређење кода без креирања шок таласа
Континуирана испорука и интеграција	Детекција и поправка грешака раније у пројекту, већи квалитет софтвера
Рани одговор од стране клијентског експерта	Избегавање скупих измена на крају, мањи трошкови развоја
Минимизације документације	Мање време развоја, нижи трошкови документовања

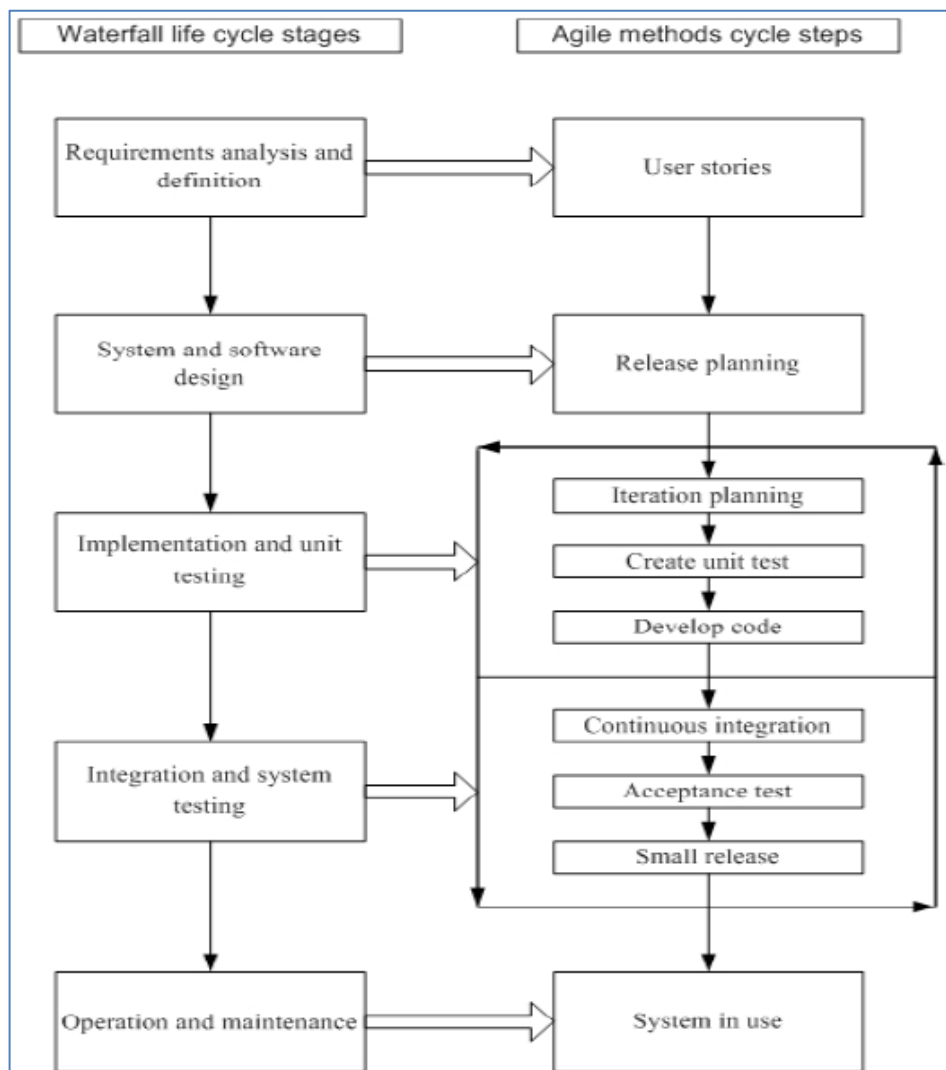
У почетном периоду примена агилних метода (пре дефинисања Agile Manifesto) је понекад сматрана ризикантна, јер је интерпретирана као супротна од модела који се заснивају на планирању и превиђању и зато је сматрано да примена агилних метода може довести до хаоса [Boehm, 2002].

У оквиру [Boehm, 2002] извршено је упоређивање карактеристика плански-заснованих и агилних метода. У раду [Nerur et al, 2005] [Leau et al, 2012] [Javanmard&Alian, 2015] дата је компарација традиционалних (waterfall) и агилних метода развоја софтвера. У раду [Boehm, 2002] утврђено је да свака од наведене две групе метода (традиционална/waterfall и агилна група метода) имају област одговарајуће примене и не може се тврдити да је нека група метода погоднија у општем смислу. Такође, разматрана је и могућности хибридног приступа у скалирању између наведена два екстрема [Ambler&Lines, 2013].



Табела 2.3.2.2. Компарација агилних и традиционалних метода развоја софтвера, према Javanmard&Alian [Javanmard&Alian, 2015]

	Agile Methods	Heavy Methods
Approach	Adaptive	Predictive
Success Measurement	Business Value	Conformation to plan
Project size	Small	Large
Management Style	Decentralized	Autocratic
Perspective to Change	Change Adaptability	Change Sustainability
Culture	Leadership-Collaboration	Command-Control
Documentation	Low	Heavy
Emphasis	People-Oriented	Process-Oriented
Cycles	Numerous	Limited
Domain	Unpredictable/Exploratory	Predictable
Upfront Planning	Minimal	Comprehensive
Return on Investment	Early in Project	End of Project
Team Size	Small/Creative	Large



Слика 2.3.2.1. Компарација фаза животног циклуса Waterfall приступа и корака у оквиру једног циклуса примене агилне методе у развоју софтвера [Huo et al, 2004]

Истраживања примена агилних метода извршена су као емпиријске студије анализе постојећих истраживања [Dyba& Dingsøyr, 2008] [Dingsøyr et al, 2012], као и

истраживања стратешких проблема прилагођавања агилног приступа примени [Lal, 2011] и примене у оквиру индустријске праксе [Petersen&Wohlin, 2009] [Laanti et al, 2011], а посебно су истраживане могућности мерења успеха примене агилних метода и могућности унапређења њихове примене [Qumer&Henderson-Sellers, 2008], као и мерења степена агилности. У оквиру [Conboy&Fitzgerald, 2004] описан је оквир за вредновање примене агилних метода у развоју софтвера који се састоји у мерењу креативних, проактивних и реактивних активности и смислу мерења нивоа агилности. Мерење се своди на компарацији броја промена које су идентификоване и реализоване у активностима у односу на трошкове спровођења наведених активности. Што је већи број промена по новчаној јединици, то је активност више агилна. Такође предложени фрејмворк разматра могућности мерења робустности (систем је више робустан уколико је потребно мање реактивних активности на промене) и мерење учења. У наведеном раду разматрани су извори и типови промена (промене захтева клијента, притисак конкуренције у односу на трошкове, промене методологије и технологије имплементације, промене радне снаге и њихових очекивања, итд.), могуће проактивне и реактивне активности. У раду [Qumer&Henderson-Sellers, 2006] описан је аналитички алат 4-DAT који служи за анализу карактеристика агилних метода развоја софтвера са становишта 4 димензије-перспективе уз одговарајући скуп карактеристика који се може мерити на основу одговора на одговарајућа питања карактеристике за сваку агилну методу (у наведеном раду предложени алат је примењен на примеру Extreme Programming методе):

- карактеристике обухвата методе: величина пројекта (да ли метода дефинише за коју величину пројекта је намењена?), величина тима, стил развоја, технолошко развојно окружење, физичко развојно окружење (колокацијски или дистрибуирано), пословна култура, механизми апстракције у развоју софтвера (објектно-орјентисани, агент-базиран)
- карактеристике агилности: флексибилност, брзина, једноставност ("leanness"), учење, способност одговора („responsiveness“)
- агилне вредности у складу са агилним манифесто: појединци и интеракција изнад процеса и алата, софтвер који ради изнад детаљне документације, сарадња са корисником изнад преговора о уговору, одговор на промене изнад придржавања плана, одржавање процеса агилним, одржавање процеса ефективним у односу на трошкове
- карактеристике у складу са софтверским процесом: развојни процес, процеси пројектног менаџмента, Подршка контроли конфигурације, Процес управљања

### 2.3.3. Компарација агилних метода развоја софтвера

У раду [Dyba& Dingsøyr, 2008] [Abrahamsson et al, 2003] дат је преглед најчешће коришћених агилних метода развоја софтвера и описане су најважније карактеристике, као што је приказано у табели 2.3.3.1.

Табела 2.3.3.1. Приказ карактеристика најчешће коришћених агилних метода, према Abrahamsson et al , Dyba& Dingsøyr, Agile PrepCast [Abrahamsson et al, 2003] [Dyba& Dingsøyr, 2008]

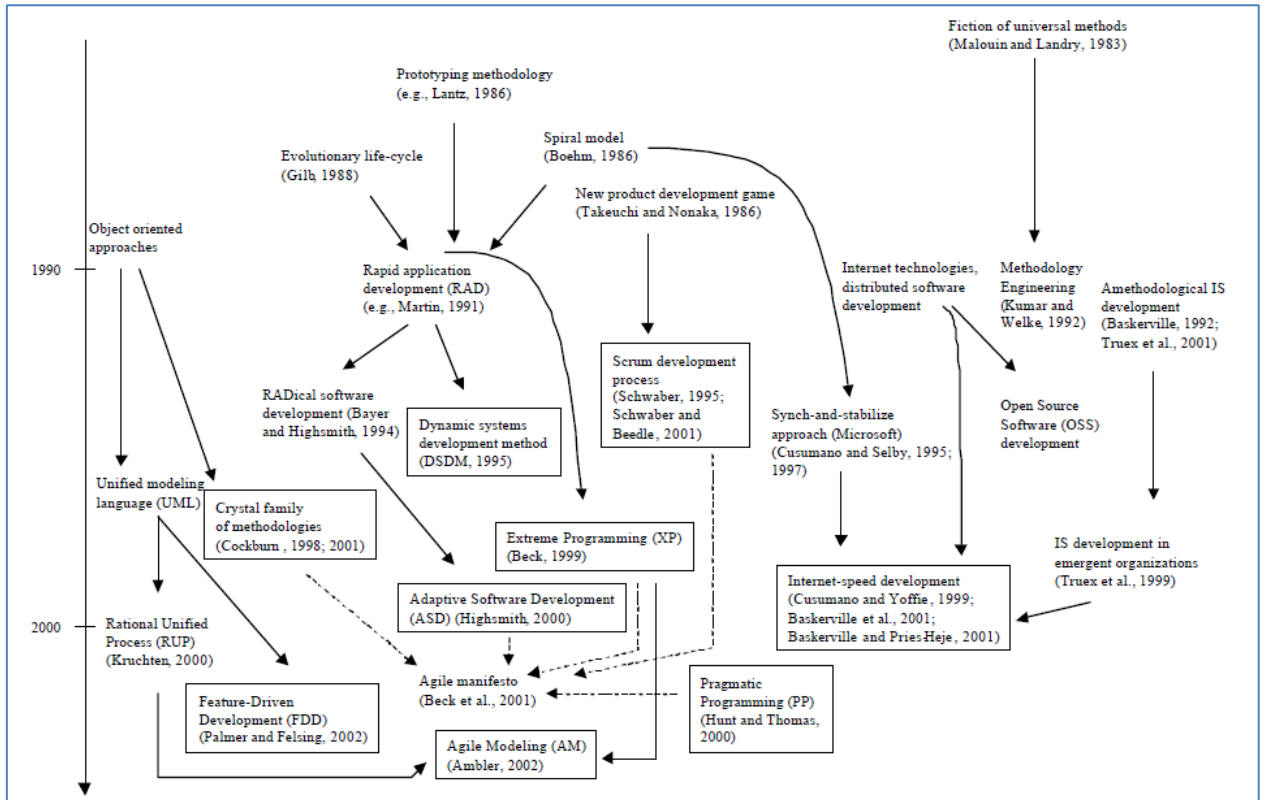
АГИЛНА МЕТОДА	ОПИС
„Crystal methodologie s“ Кристалне методологије	Фамилија метода за колокацијске тимове различитих величина и критичности: Clear, Yellow, Orange, Red, Blue. Једна од најагилнијих метода „Crystal Clear“ фокусирана је на комуникацију у малим тимовима у развоју софтвера који није животно-критичан, а карактеристике су: фреквентна испорука, рефлексивно унапређење, осмотска комуникација, лична безбедност, фокус, лак приступ експертским корисницима и захтеви за техничким окружењем. [Cockburn, 2004] Фамилија Crystal метода омогућава избор одговарајуће методе у складу са величином и критичности, а „боја“ методе одређује тежину методе. Већи пројекти захтевају више координације и теже методе. Crystal приступ омогућава прилагођавање метода како би одговарале различитим околностима и конкретним пројектима. [Abrahamsson et al, 2003] Crystal Methods – колекција приступа агилном развоју софтвера, који се фокусира примарно на људе и интеракцију међу њима док раде на пројектима

	<p>развоја софтвера. Такође, фокус је на критичности и приоритетима елемената система који се развија, у односу на значај и утицај на пословање. Људи и процеси, као и њихове интеракције представљају језгро развојног процеса. [Agile PrepCast, 2013]</p>
<p>„Dynamic software development method (DSDM)“ Метода динамичког развоја софтвера</p>	<p>Дели пројекте на 3 фазе: пред-пројекат, пројектни животни циклус, пост-пројекат. Девет основних принципа: укључивање корисника, оснаживање пројектног тима, фреквентна испорука, фокусирање на тренутне пословне потребе, итеративни и инкрементални развој, дозвољавање реверзних промена, дефинисање обухвата високог нивоа пре почетка пројекта, тестирање у току животног циклуса, ефикасна и ефективна комуникација. [Stapleton, 2003] Основна идеја DSDM је прилагођавање времена и ресурса и тек након тога усаглашавање функционалности предлога решења у складу са расположивим временом и ресурсима. [Abrahamsson et al, 2003] DSDM првенствено представља развојни оквир за агилну испоруку у оквиру билоког пројекта, али првенствено коришћен као метод развоја софтвера. Настао 1994 у циљу увођења дисциплине у RAD (Rapid Application Development), од 2007 постаје генерички приступ пројектном менаџменту и испоруци решења. Основни принципи су итеративни и инкрементални приступ уз континуално укључивање корисника/клијента. У оквиру DSDM фиксирају се трошкови, квалитет и време према спољашњем договору, а интерно користи MoSCoW класификацију према приоритетима особина софтвера који треба да буду реализоване – “must”, “should”, “could”, “won’t have to”. DSDM се може користити и за не-ИТ решења. [Agile PrepCast, 2013]</p>
<p>„Feature-driven development“ (FDD) Развој заснован на особинама софтвера</p>	<p>Представља метод процес-орјентисаног развоја софтвера за развој пословно критичних система. Фокусира се на дизајн и фазе изградње. Наглашава аспекте квалитета кроз процес развоја и укључује фреквентне и опипљиве испоруке, кроз прецизан мониторинг прогреса на пројекту. [Abrahamsson et al, 2003] Комбинује модел-базиран и агилни развој са нагласком на иницијалном објектном моделу, подели рада на особине и итеративном дизајну сваке особине софтвера. Тврди се да је овај приступ погодан за развој критичних система. Итерација особине састоји се од две фазе: дизајн и развој. [Palmer&amp; Felsing, 2002] Итеративни и инкрементални процес развоја софтвера са нагласком на перспективу клијента и његовог вредновања потребне функционалности (особина софтвера). Главни циљ је доставити опипљиве резултате, софтвер који ради и то испоручити у предвиђеном времену.</p>
<p>“Lean” развој софтвера</p>	<p>Адаптација принципа из „lean“ производње, а посебно из Toyota производног система за развој софтвера. Састоји се од 7 принципа: елиминација вишка (непотребнога), повећати учење, доносити одлуке што касније, испоручивати што пре, оснажити тим, градити интегритет и имати увид у целину. [Porpendieck&amp;Porpendieck, 2003] Lean software development – покретн намењен редукацији грешака и губљења времена, уз максимизирање едукације и ефикасности. Његови принципи су оригинално коришћени у производњи и ИТ, а након тога прилагођени програмирању. Основни принцип је примарна повећеност вредности која се ствара за крајњег корисника, уз интелигентно чување ресурса. [Agile PrepCast, 2013]</p>
<p>SCRUM</p>	<p>Фокусира се на пројектни менаџмент у ситуацијама где је тешко планирати унапред, уз механизме за „емпиријску процесну контролу“ уз вишеструке повратне спреге. Софтвер се развија од стране само-организујућих тимова у инкрементима (који се зову „спринтови“), почев од планирања до завршетка који укључује ревизију. Особине које треба да буду имплементирани у систему се региструју у оквиру „backlog“. Након регистрације, власник производа одлучује која ставка из backlog-а треба да буду развијене у следећем спринту. Чланови тима координирају свој рад у свакодневним састанцима. Један члан тима („scrum master“) је задужен за решавање проблема који су учинили да тим буде заустављен у ефективном раду. [Schwaber&amp; Beedle, 2001] Самоорганизујући тимови се сматрају јединицом која треба да достигне заједнички циљ, подстиче колокацију чланова тим, вербалну комуникацију између чланова тима. Главни принцип SCRUM је омогућавање да клијент може да промени мишљење о томе шта жели или шта му је потребно (назива се „requirements churn“) и прихватање да проблем који се решава не може бити у потпуности разјашњен и дефинисан, али се зато фокусира на максимизирање способности тима да испоручује брзо и одговара на актуелне захтеве. [Agile PrepCast, 2013]</p>
<p>Extreme Programming</p>	<p>Представља колекцију добро познатих практичних упутстава и искустава из софтверског инжењерства. [Abrahamsson et al, 2003] Састоји се од 12 принципа</p>

<p>(XP, XP2) Екстремно програмирање</p>	<p>практике: игра планирања, мале испоруке, метафора, једноставан дизајн, тестирање, рефакторисање, програмирање у пару, колективно власништво над артефактима, континуирана интеграција, 40-часовна радна недеља, стална комуникација са корисницима (on-site), стандарди кодирања. Ревизија путем XP2 предлаже принципе праксе: седети заједно, целина тима, информативни радни простор, енергизирани рад, програмирање у пару, корисникове приче („user stories“) као спецификација захтева, итерације на недељном нивоу и на 3-месечном нивоу, затишја (slack), изградња од 10 минута, континуирана интеграција, програмирање базирано на тестирању („test-first programming“), инкрементални дизајн. [Beck, 2004] Промовише фреквентне испоруке у малим развојним циклусима, у циљу унапређења продуктивности и увођења тачке контроле где нови захтеви корисника могу бити прилагођени. Остале особине: програмирање у пару или екстензивна ревизија кода, тестирање целог кода једног модула, избегавање програмирања особине софтвера све док заиста није потребна, равна структура менаџмента, једноставност и јасност кода, очекивање промена у захтевима корисника како време пролази и проблем је боље схваћен, фреквентна комуникација са клијентима и између програмера. [Agile PrepCast, 2013]</p>
<p>Адаптивни развој софтвера (ASD)</p>	<p>Настао на основама Rapid Application Development. Укључује принципе континуалне адаптације процеса развоја, као понављајуће серије шпекулација, сарадње и учења. У овом динамичком циклусу обезбеђује константно учење и адаптацију актуелном стању пројекта. Карактеристике ASD животног циклуса: фокусиран на мисију, базиран на карактеристикама софтвера које је потребно реализовати, итеративно, временски дефинисан(timeboxed), оријентисан на управљање ризицима и толерантан на промене.[Agile PrepCast, 2013]</p>
<p>Kanban</p>	<p>Представља систем временског планирања (scheduling) за lean и JIT (Just in time) производњу. Представља систем контроле логистичког ланца са производне тачке гледишта. Развијен је у Toyota, као систем за унапређење и одржавање високог нивоа производње. Kanban представља метод кроз који се постиже JIT и представља ефективни алат подршке извршавања производног система у целини, уз подршку промоцији унапређења. [Agile PrepCast, 2013]</p>

Рад [Qumer&Henderson-Sellers, 2006] послужио је као основ за развој методе за компарацију карактеристика различитих метода агилног приступа развоју софтвера применом реализованог аналитичког алата 4-DAT, а примена у компарацији XP и SCRUM метода је представљена у раду [Fendandes&Almeida, 2010].

У оквиру рада [Abrahamsson et al, 2003] дат је преглед агилних метода и поред претходно наведених у табели 2.3.3.1., описане су и метода адаптивног развоја софтвера и агилног моделовања, Internet-speed развојна метода и метода прагматичног програмирања. Анализа и компарација свих наведених метода извршена је у односу на тип животног циклуса софтвера који метода подржава, да ли подржава активности пројектног менаџмента, подршка апстрактним принципима развоја, универзалност или ситуациона усклађеност, емпиријска искуства и докази примене. Историјски преглед агилних метода приказан у овом раду дат је на слици 2.3.3.1.:



Слика 2.3.3.1. Историјски преглед метода агилног приступа развоју софтвера према Abrahamsson et al [Abrahamsson et al, 2003]

У оквиру [Agile PrepCast, 2013] дата је детаљна компарација (представљена у Табели 2.3.3.2.) агилних метода у односу на различите критеријуме:

- Скалабилност – ниво до ког агилна метода може бити коришћена за различите врсте развоја производа, различите типове пројеката и различите примене
- Величина тима – идеалан број чланова развојног тима за максималну ефективност примене агилне методе
- Дужина итерације – саветована дужина времена (периода итерације) за испоруку производног инкремента клијенту, према наведеног агилној методи
- Улоге и одговорности – Ниво до ког су дефинисане специфичне улоге и одговорности у примени конкретне агилне методе
- Процес центричне или човек-центричне – да ли је агилна метода орјентисана на процес или на људски фактор
- Подршка виртуалним тимовима – ниво до ког агилна метода подржава комуникацију и координацију рада виртуалних тимова
- Ниво смањења ризика – ниво до ког се ризик који је инхерентан пројекту може смањити применом наведеног агилног метода
- Интеракција са клијентом – ниво интеракције са клијентом потребан за максималну ефективност примене наведене агилне методе
- Предности – главни аспекти предности примене наведене методе
- Недостаци – главни недостаци наведене методе, због којих наведена метода не треба да се користи

У раду [Stojanovic et al, 2003], [Agile PrepCast, 2013] и [Javanmard&Alian, 2015] дата је компарација агилних метода у односу на различите критеријуме, а посебно у оквиру [Stojanovic et al, 2003] извршена је анализа и компарација приступа моделовању у оквиру развоја софтвера применом различитих агилних метода.

Табела 2.3.3.2. Компарација Агилних метода, према Agile PrepCast [Agile PrepCast, 2013]

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Scalability</b>	A wide range of software projects of various sizes; can also be adapted to non-software projects such as product/application development products.	Software development projects driven by need for quick delivery; also for large, complex projects where plan driven methods are ineffective.	Any technical or business environment where a system needs to be quickly developed and deployed; preferably any project that is a good fit with the DSDM Atern Method as outlined in the Project Approach Questionnaire.	Designed and first used for large, complex banking projects; good for any large, complex project with large teams.	Designed for project environments that are high-speed, high-change, and high-uncertainty; also good for organizations that are highly adaptive.	Based on Lean Manufacturing; used when project requires the use of a specific set of engineering practices or when cost and ROI are the main measurements of success.	Although Kanban is not considered an Agile development method, it can be used with any existing processes and projects to help reduce costs and increase efficiency.	Designed to be adaptable to projects ranging from small with low criticality to large with high criticality.
<b>Team size</b>	Small Teams (Should consist of at least 5 but no more than 9 team members)	Small teams (Generally 2-10 team members; emphasis is on highly-skilled and experienced team members)	Small team: (Generally 2-10 team members in a structured and disciplined environment)	Small to Large Teams (Generally 4-20 team members depending on complexity and length of project)	Small to Large Teams (Depends on project scope)	Not specified;	Not specified; (Normally used with existing team size)	Supports Any Team Size (Emphasis is on highly-skilled and experienced team members)
<b>Iteration length</b>	Medium (30-day iterations recommended as maximum length with preference for a shorter iterations)	Short (2-week iterations; never to exceed 3 weeks)	Not specified	Short (2-week iterations recommended as best practice)	Determined by project schedule and degree of uncertainty (4-week iterations for small projects and 8-week iterations for medium to large projects)	Not specified	Short (1-week iterations)	Project Specific (Up to 4-month iterations for large, complex projects)

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Roles &amp; responsibilities</b>	Specifically defined	Specifically defined	Specifically defined	Specifically defined	Not defined	Not defined	Not specified; (normally uses existing organization and project roles)	Not defined
<b>Process centric vs. people centric</b>	People Centric (Process is important but secondary)	People centric	Process centric	Process centric	People centric	Process centric	Process centric	People Centric (Process is important but secondary)
<b>Virtual team support</b>	Somewhat Supported (not specifically designed for distributed teams but multiple teams could easily be effectively distributed)	Not Supported (Team must be co-located distributed teams.)	Not Supported	Somewhat Supported (Designed for multiple teams; should be adaptable to distributed teams.)	Loosely Supported (Can be applied to distributed teams but becomes more challenging during collaboration phase.)	Not Supported	Somewhat Supported (Can be adapted to distributed teams through virtual Kanban boards and other Agile tools.)	Fully Supported (Designed to support distributed teams.)
<b>Risk mitigation level</b>	High Risk Mitigation (Risk is identified and mitigated daily.)	Medium Risk Mitigation (Despite the fact that risks are decreased by Collocation & Pair Programming, risks are also increased by the slower delivery of features)	High Risk Mitigation (Risk mitigation is inherent in most of the principles and the roles and responsibilities that are basic to this method.)	Medium Risk Mitigation (Follows most basic Agile principles except for daily collaboration with customer.)	High Risk Mitigation (Risk-Driven is one of the six basic characteristics and each Component Cycle Plan is initiated by analyzing risks.)	Medium Risk Mitigation (Follows most basic Agile principles except for making decisions as late as possible, which increases negative risk.)	Medium Risk Mitigation (Follows most basic Agile principles except often considers estimation as waste, which may increase negative risk.)	High Risk Mitigation (Risk mitigation is inherent in just about all of the principles and techniques that are recommended when using this method.)
<b>Customer interaction</b>	Medium Interaction (monthly input)	High Interaction (daily input)	High Interaction (daily to weekly input)	Low interaction (input not specified)	Low interaction (input not specified)	Low Interaction (input not specified)	Low Interaction (input not specified)	High interaction (daily to weekly input)

Criteria	Scrum	XP	DSDM Atern	FDD	ASD	LSD	Kanban	Crystal Family
<b>Pros</b>	Maximizes personal communication and informal knowledge sharing; breaks project into manageable pieces; progress is made even if requirements are not stable	Simple; iterative; values communication; based on best practices; puts emphasis on design	Highly dependable; provides well-defined guidelines for the business system, concept development, and requirements	Strong modeling features; provides detailed guidelines for multi-team projects	Very strong in non-technical aspects of software development	Seeks to change companies from the top down; guidelines for business enterprise very defined	Allows teams to visualize their work and eliminate bottlenecks; can lead to exponential improvements in operational efficiency and quality	Strong on communication; well-defined guidelines for project teams; well-defined technical practices
<b>Cons</b>	Weak on measurement practices; weak on business system, technical, and concept development practices	Lacks design documentation; highly prescriptive; lacks measurement processes; does not address deployment	Details and white papers available only to DSDM Consortium members; may tend towards bureaucracy	Only addresses design and implementation; requires highly-experienced experts for modeling	Does not provide guidelines for individual development projects; does not address technical aspects; weak on metrics	Allows little change in requirements; literature for applying in a software environment is limited; poorly-defined technical and measurement practices	Using JIT delivery will inevitably lead to delays at some point when JIT turns in to "Not In Time"; doesn't capture or show dependent tasks very well; hard to see overall project status	Largely theoretical; does not define guidelines for the business enterprise

Табела 2.3.3.3. Компарација Агилних метода, према Javanmard&Alian [Javanmard&Alian, 2015]

Method Name	Key Points	Special features	Identified weakness
<b>ASD</b>	Adaptive culture, collaboration, mission-driven component based iterative development	Organizations are seen as adaptive systems. Creating an emergent order out of a web of interconnected individuals	ASD is more about concepts and culture than the software practice
<b>DSDM</b>	Application of controls to RAD, use of timeboxing and empowered DSDM teams.	First truly agile software development method, use of prototyping, several user roles : "ambassador", "visionary" and "advisor"	While the method is available, only consortium members have access to white papers dealing with the actual use of the method
<b>XP</b>	Customer driven development, small teams, daily builds	Refactoring - the ongoing redesign of the system to improve its performance and responsiveness too change	While individual practices are suitable for many situations, overall view & management practices are given less attention
<b>SCRUM</b>	Independent, small, self-organizing development teams, 30-day release cycles.	Enforce a paradigm shift from the "defined and repeatable" to the "new product development view of Scrum"	While Scrum details in specific how to manage the 30-day release cycle, the integration and acceptance tests are not detailed
<b>FDD</b>	Five-step process, object-oriented component (i.e. feature) based development.	Method simplicity, design and implement the system by features, object modeling	FDD focuses only on design and implementation. Needs other supporting approaches.

Табела 2.3.3.4. Компарација агилних метода према Stojanovic et al [Stojanovic et al, 2003]

	Key characteristics	Special features	Shortcomings
<b>Scrum</b>	Small teams, Iterations (sprints) 7-30 days cycles	Daily meetings (scrums), integration with XP	Little about development practice
<b>ASD</b>	Adaptive organizational culture, collaborative teamwork	Non-linear overlapping lifecycle phases, component-based	Lack of software development practice
<b>Crystal family</b>	Family of methods, adaptable to different project size and complexity	Features and values common to the whole family, small teams, 1-3 months cycle	Not complete
<b>DSDM</b>	Controlled RAD variant, supported by consortium	Use of prototyping, several small teams (2-6 people)	Limited access to supporting materials
<b>XP</b>	Customer-driven, frequent release, pair programming	Refactoring, test-first development, constant testing	Little about management practice
<b>FDD</b>	Five basic process steps, short iterations, feature-centered	Combining features and object modeling, scalable	Needs management practice support
<b>Agile Modeling</b>	Applying agile principles and practices to modeling	Can fit well into different processes (XP or RUP)	Not complete process, need other development methods
<b>Extreme Modeling</b>	Integrating model-based and XP principles	Tool support needed, models testable and executable	The method and the tool support under development

Табела 2.3.3.4. Компарација агилних метода према Stojanovic et al [Stojanovic et al, 2003] (наставак)

	<b>Maturity</b>	<b>Documented</b>	<b>Adoption and Experience</b>
<b>Scrum</b>	Active	Book, website	Applied in practice
<b>Adaptive SD</b>	Active	Book, papers, website	Applied to some extent
<b>Crystal family</b>	Still under development	Website, web documents	Applied to some extent
<b>DSDM</b>	Active	Book, web site, papers (limited to DSDM members)	Applied in practice
<b>XP</b>	Active	Books, papers, websites	Applied in practice
<b>FDD</b>	Active	Books, papers, websites	Applied in practice
<b>Agile Modeling</b>	Still under development	Book, papers, website	No evidence of applying
<b>Extreme Modeling</b>	Still under development	Website, papers	No evidence of applying

	<b>Process support</b>	<b>Iterative and Incremental</b>	<b>Scalability – size of teams</b>	<b>Defined member roles</b>	<b>Modeling Activities</b>
<b>Scrum</b>	Management process	Yes	Up to 10 people	Yes	Not applicable
<b>ASD</b>	Management process	Yes	No restrictions	Not detailed	Not applicable
<b>Crystal family</b>	Partly Development and Management process	Yes	Up to 40	Yes	Not applicable
<b>DSDM</b>	Management process	Yes	Possible many small teams (from 2 to 6)	Yes	Not applicable
<b>XP</b>	Development practice	Yes	Up to 12	Yes	Minimized
<b>FDD</b>	Development and partly Management process	Design and build phases	From 50 up to 250	Yes	Minimized
<b>Agile Modeling</b>	Modeling	Yes	No restrictions	Not detailed	Full
<b>Extreme Modeling</b>	Modeling	N/A	No restrictions	N/A	Models are executable and testable

#### 2.3.4. Истраживање улоге инжењерства захтева, моделовања, дизајна и документације у агилном развоју софтвера

У оквиру рада [Сао&Balasubramaniam, 2008] описана су претходна истраживања, као и практична искуства анализом индустријске праксе (16 компанија) у области инжењерства захтева (који се изједначава као термин са анализом захтева) применом агилног приступа. Основни резултати су: 1) преферирање личне комуникације у односу на комуникацијом путем документације; 2) итеративно инжењерство захтева, где захтеви нису унапред дефинисани, већ еволуирају у току развоја; 3) захтеви се рангирају константно у току развоја у складу са приоритетима, који се дефинишу у складу са доприносом пословним вредностима. Листе захтеваних особина софтвера се константно формирају и рангирају, на почетку сваког развојног циклуса; 4) управљање изменама захтева кроз константно планирање кроз 2 типа примена: додавање или брисање особина софтвера као и измене већ раније реализованих особина; 5) реализација прототипа, који се најчешће касније унапређује и поставља као решење, уместо да буде третирано као привремено решење које ће бити замењено правим решењем; 6) тест-базирани развој где се дефинишу тестови пре писања кода као форма прецизне спецификације шта и како апликација треба да функционише; 7) састанци ревизије.

У раду [Paetsch et al, 2003] описане су технике инжењерства захтева у контексту агилног развоја софтвера. Технике издвајања захтева корисника служе за разумевање апликационог домена, пословних потреба, ограничења система, заинтересованих страна и самог пословног проблема. Најчешће технике обухватају: интервју, креирање случајева коришћења и сценарија, посматрање и социјална анализа, дискусије фокусних група, Brainstorming, реализација прототипа. Након прикупљања захтева, следи њихова анализа уз технике: удружени развој апликације (Joint Application



Development), одређивање приоритета захтева за развој (од стране клијента – издвајање захтеваних карактеристика софтвера које омогућавају највеће користи у пословању али и чланова развојног тима – разматрање могућности техничке реализације – ризика, трошкова, потешкоћа), моделовање система (најчешћи модели: модели тока података, модели семантике података, објектно-орјентисани модели). Након приоритизације, реализује се документовање, валидација и управљање захтевима. У овом раду су у свим сегментима инжењерства захтева описана искуства добре праксе у примени у оквиру агилних приступа.

У оквиру рада [Stojanovic et al, 2003] описано је истраживање улоге моделовања у агилном приступу развоја софтвера. Резултати истраживања представљени су у табели 2.3.4.1.

Табела 2.3.4.1. Улога моделовања у примени агилних метода развоја софтвера, према Stojanovic et al [Stojanovic et al, 2003]

	<b>XP</b>	<b>FDD</b>	<b>Agile Modeling</b>	<b>Extreme Modeling</b>
<b>Scope</b>	Development process	Development process	Modeling and design	Development process (not detailed)
<b>Amount of modeling</b>	Low	Low	Middle	Middle
<b>Tool support</b>	No	Not specified (possible UML-supportive)	The simplest possible e.g. whiteboard	Extensive support (transfer model to code)
<b>Notation</b>	User stories, CRC cards, Sketches of classes	Features, Objects/Classes, Sequence diagrams	UML + User stories, UI screens, Data modeling, etc.	Standard UML + Petri nets
<b>Architecture design</b>	Metaphor, Architecture Spikes, First Release	Based on object model	Domain packages	Based on object orientation
<b>Requirements elicitation</b>	User stories	Features	Use cases + user stories	Use cases
<b>Using components</b>	No	No	To some extent	No
<b>Business modeling</b>	No	No	Yes	No
<b>Model repositories</b>	No	N/A	No	Yes
<b>Designing large systems</b>	No	To some extent	Yes	Yes
<b>Incremental</b>	Yes	Yes	Yes	Yes
<b>Iterative</b>	Yes (very small increments)	Yes, only two phases	Yes, sequential and iterative	Model and code constantly in sync
<b>Complexity</b>	Low	Low	Middle	Middle – advanced tool support
<b>Model reusability</b>	No	No	No	Yes

У оквиру [Fox et al, 2008] разматран је аспект кориснички-орјентисаног дизајна у контексту примене агилних метода, кроз преглед постојећих истраживања и емпиријско истраживање реализације конкретних софтверских пројеката. Првенствени фокус истраживања овог рада је у итеративном дизајну корисничког интерфејса који се одвија паралелно са техничком имплементацијом решења.

У раду [Rubin&Rubin, 2010] описано је истраживање у области улоге документације у агилном развоју софтвера. Као један од главних доприноса агилног приступа у овом раду се сматра смањење бирократије традиционалних методологија развоја софтвера. Ипак, документација има важну улогу у размени знања. Суштинска идеја наведеног рада је у предлогу адаптивне документације која ће бити усклађена са агилним принципима развоја. Адаптивна документација се у овом раду уводи кроз имплементацију приступа активног документационо софтверског дизајна (Active Documentation Software Design) као активна документација која је блиско повезана са изорним кодом, где се измене и документацији рефлектују на измене изворног кода и обрнуто – измене изворног кода омогућавају измене релевантне документације.

### 2.3.4. Дисциплиновани агилни приступ развоју софтвера

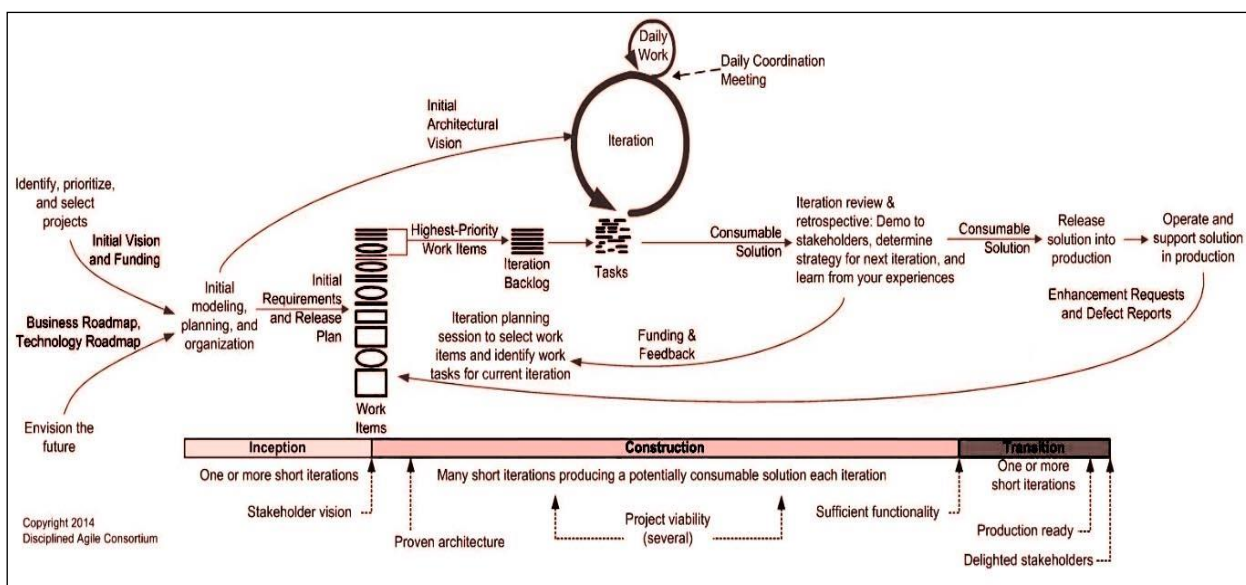
У току процеса прилагођавања агилних метода реалној употреби, различита терминологија из разних метода и некомплетност појединих метода довела је до потребе да се наведене методе анализирају у смислу издвајања заједничких карактеристика и општег модела агилног приступа развоју софтвера [Janus, 2012], као и интегришу [Ambler, 2013]. Из наведених разлога, од 2006-2012. године [Ambler&Lines, 2012] у оквиру фирме IBM развијен је тзв. Disciplined Agile Delivery (DAD) фрејмворк, као процесни оквир орјентисан на испоруку решења применом интеграције различитих агилних метода. Основне карактеристике DAD су [Ambler&Lines, 2013]:

- орјентисан на људе („people-first“),
- хибридан у примени стратегија из више агилних метода: Scrum, Extreme Programming (XP), Agile Modeling (AM), Unified Process (UP), Kanban, Outside in Development (OID), and Agile Data (AD)
- орјентисан на учење,
- орјентисан на решења, тј. решавање проблема у ширем контексту– „од једноставног креирања софтвера до обезбеђивања употребљивих решења која обезбеђују стварну пословну вредност заинтересованим странама у оквиру економских, културних и технолошких ограничења. Софтвер јесте важан, али у циљу обезбеђивања подршке потребама заинтересованих страна, често се поред софтвера јавља потреба за унапређењем хардвера и променама у пословним и оперативним процесима, чак и организационим структурама у организацијама где раде заинтересоване стране.“ [Ambler&Lines, 2013]
- орјентисан испоруку – орјентација на континуирану испоруку, кроз животни циклус
- орјентисан на реализацију циљева („goal driven“) – тим мора да се адаптира на промене у захтевима и развојним приоритетима, мора да разуме контекст у ком раде...
- вредновање ризика („risk- value delivery“),
- усклађеност са организационим потребама („enterprise-aware“) – „дисциплиновани агилни тимови препознају да представљају део већег организационог екосистема и у складу са тиме реализују своје активности, сарађујући са другим тимовима у оквиру организације.“ [Ambler&Lines, 2013]
- скалабилност.

На слици 2.3.2.1. приказан је животни циклус дисциплинованог агилног приступа испоруци ИТ решења:

1. DAD животни циклус започиње идентификацијом, приоритизацијом и селекцијом пројеката, где је иницијална визија и буџет (финансирање) дефинисано.
2. Фаза заснивања („inception phase“) пројекта пролази кроз једну или више итерација. Састоји се из иницијалног моделовања, планирања и организације, где су иницијални захтеви дефинисани и испоручен иницијални план испоруке и обезбеђена је сагласност заинтересоване стране.
3. Фаза конструкције се реализује кроз кратке итерације и свака итерација производи потенцијално употребљиво решење. Радне ставке („work items“) се дефинишу и радне ставке највишег приоритета су изабране да буду укључене у план следеће итерације („iteration backlog“). Задаци су дефинисани у оквиру плана итерације. Задаци се реализују и оквиру итерације у оквиру активности сваког дана и сваког дана се одржавају састанци координације. Након што је итерација завршена, потенцијално употребљиво решење се испоручује на преглед и ретроспективу итерације, креира се демо верзија за заинтересоване стране и она се њима презентује, дефинише се стратегија за следећу итерацију. У оквиру састанка планирања итерације („iteration planning session“) селекују се радне ставке и идентификују радни задаци за следећу итерацију. Фаза конструкције је завршена када је довољно функционалности реализовано.
4. Фаза транзиције почиње када се реализовано решење поставља у реалну радну средину где ће бити коришћено („released in production“). Ова фаза се такође састоји из неколико кратких итерација.

5. Коначно, решење се користи у реалном радном окружењу, уз подршку која треба да прихвати и реализује захтеве за унапређењем или извештаје о грешкама.



Слика 2.3.4.1. Животни циклус у оквиру дисциплиноване агилне испоруке (DAD) према Ambler&Lines [Ambler&Lines, 2013]

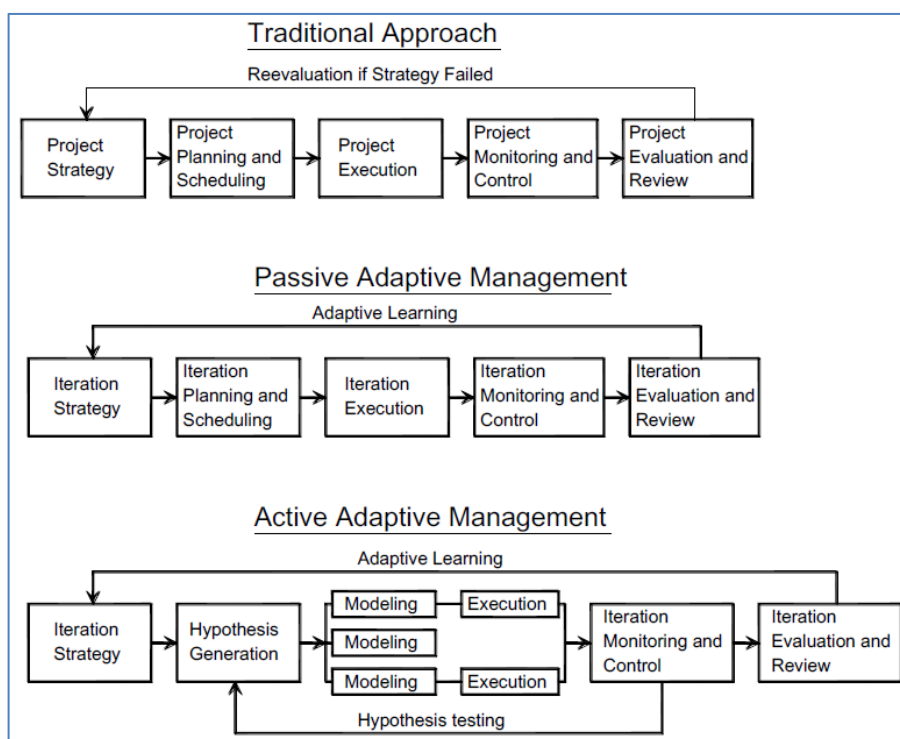
У складу са агилним приступом, адаптивност у смислу спремности и прилагођавању на промене представља једну од четири најважније вредности. У оквиру основних принципа агилног развоја, промене се односе на промене захтева корисника као и на омогућавање самоорганизованости тимова који интерно унапређују своју организацију и понашање ради унапређења ефикасности и ефективности рада.

### 2.3.5. Адаптивно управљање перформансама у агилном пројектном менаџменту

У циљу креирања услова за агилни пројектни менаџмент и агилне иновативне организације, потребно је унапредити систем управљања перформансама и у оквиру [Highsmith, 2006] предлаже се адаптивни систем управљања перформансама у организационим системима. Према [Highsmith, 2006], адаптивни систем управљања перформансама има два кључна атрибута, односно потребно је да омогући: 1) фокусирање тимова у организацији на циљне стратешке или тактичке излазне резултате, 2) подстицање тимова да реализују своје активности у складу са унапред дефинисаним мерним карактеристикама излазних резултата. Како би се успешно креирала адаптивна организација, важно је узети у обзир [Highsmith, 2006]: 1) систем мерења перформанси утиче на агилност, 2) мора се заменити „опсесија“ временом реализације „опсесијом“ креирања вредности за клијента, 3) треба разликовати систем мерења перформанси пројекта од система мерења перформанси тима који ради на пројекту. Адаптивни систем мерења перформанси се састоји из два дела: 1) систем мерења перформанси пројекта, који је фокусиран на резултате који дају вредност клијенту и 2) систем мерења перформанси тима, који се фокусира на информационе метрике које тим може да користи како би унапредио своје функционисање. У оквиру адаптивног система мерења перформанси, фокус се помера са временске и ресурсне компоненте (поштовања плана и постизања буџета) на квалитативну компоненту (добијања вредности за клијента). Овакав метрички систем треба да буде изграђен тако да се: 1) изгради систем мерења заснован на поверењу, поштењу и намерама да се повећа вредност организације, 2) постави највећи нагласак на мерењу излаза, не улаза, чак и када метрике није лако утврдити или нису прецизне, 3) креирати метрике излазних резултата тако да су ограничења што шира (због толеранције на варијације) у циљу подстицања адаптације, 4) креирати информационе метрике које подржавају

урођену људску интерну мотивацију и обезбеђује им агрегатне мере општих, радије него личних (појединачних) перформанси.

У оквиру [Intaver APM] разматране су карактеристике општег адаптивног пројектног менаџмента. Адаптивни менаџмент је дефинисан као структурирани и систематски процес континуалног унапређења одлука, политика управљања и праксе учећи на основу резултата који су настали на основу претходних одлука. Заснива се на принципу „учење радом – learning by doing“. Неки од принципа агилног развоја софтвера, садржаних у Agile Manifesto из 2001. године одговарају и принципима адаптивног менаџмента: редовна адаптација променљивим околностима, које укључују и промене захтева корисника, итеративни процес развоја, константна комуникација развојног тима и клијента. Близак агилном приступу је и флексибилни развој производа. Адаптивни пројектни менаџмент заснива се на итеративном одлучивању и стратешком ставу да се не доносе неповратне одлуке. Основне поставке адаптивног пројектног менаџмента: пројектима се управља базирано на учењу из стварних перформанси пројекта и то учење се заснива на резултатима примене и анализе квантитативних метода, као што су: мулти-модел анализа и тестирање хипотеза, мерење перформанси пројекта, мерење перформанси текућег пројекта, квантитативно базирана анализа ризика у односу на време и трошкове. Иако се адаптивни пројектни менаџмент показао као ефикаснији, ипак је традиционални пројектни менаџмент доминантан. Узроци отпора широј примени адаптивног менаџмента, према [Intaver APM], леже првенствено у психолошким елементима. Адаптивни пројектни менаџмент може бити пасиван и активан. Традиционални пројектни менаџмент податке о евалуацији целог пројекта користи за реевалуацију и повратну спрегу у односу на стратегију пројекта. Пасивна варијанта адаптивног пројектног менаџмента уводи адаптивно учење на нивоу итерације, док активно адаптивно учење уводи експериментацију, тј. више алтернатива и хипотеза и тестирање хипотеза, најчешће симулацијом (нпр. Монте Карло симулацијом), како би се добило најбоље решење на нивоу итерације.



Слика 2.3.5.1. Компарација традиционалног и адаптивног пројектног менаџмента, према Intaver Institute [Intaver APM]

У оквиру [Alleman, 2002] пројектни менаџмент је разматран са становишта теорије управљања, а посебно са становишта адаптивног управљања. Alleman [Alleman, 2002]

разликује општи пројектни менаџмент од софтверског пројектног менаџмента. Управљање у околностима промена и управљање променама су области у којима се разликује традиционални пројектни менаџмент и адаптивни пројектни менаџмент. Према [Alleman, 2002], под традиционалним пројектним менаџментом се сматра оквир дефинисан кроз РМВОК [РМВОК, 2013]. Према [Alleman, 2002], традиционални пројектни менаџмент се заснива на планирању и контролисању, као и на претпоставци да се планирање и извршавање одвија у стању потпуне рационалности. У традиционалном пројектном менаџменту („нормативан приступ“) активности су организоване линеарно, као секвенцијалан низ активности – планирање, извршавање, контрола, који су засновани на примени правила која омогућавају унапређење корисности за учеснике. Према [Alleman, 2002], теорија комплексних адаптивних система је блиска „агилном“ приступу управљању софтверским пројектима. Развој софтвера се разликује по томе што се не може сматрати да се одвија у стабилним и потпуно рационално заснованим околностима због промена захтева корисника, промена технологије и стања на тржишту. Коначан излазни резултат у пословном доприносу и технолошко-имплементационом смислу је тешко планирати. У развоју софтвера присутне су нелинеарне повратне спреге, где корисници након испоруке дефинишу често значајно различите захтеве у односу на почетне. У пракси софтверског пројекта, активности се реализују у нелинеарном и адаптивном облику као скуп међусобно зависних активности.

### 2.3.6. Адаптивност у управљању агилним софтверским пројектима

У раду [Meso&Jain, 2006] извршена је детаљна компарација управљања агилним софтверским пројектима, односно процеса агилног развоја софтвера и елемената-функција комплексних адаптивних система (CAS – Complex Adaptive Systems).

Табела 2.3.5.1. Компарација принципа агилног приступа развоја софтвера и теорије комплексних адаптивних система према Meso&Jain [Meso&Jain, 2006]

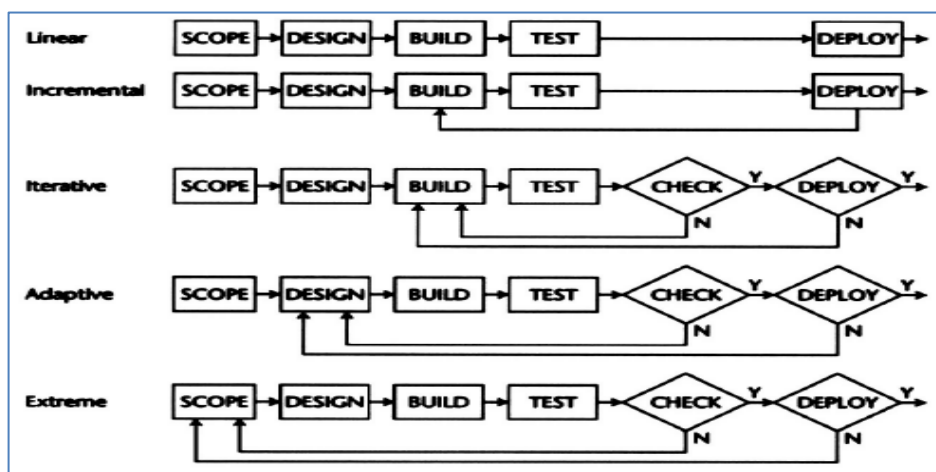
<b>ПРИНЦИПИ АГИЛНОГ ПРИСТУПА РАЗВОЈА СОФТВЕРА</b>	<b>ПРИНЦИПИ ТЕОРИЈЕ КОМПЛЕКСНИХ АДАПТИВНИХ СИСТЕМА (CAS)</b>
Фреквентне испоруке и континуирана интеграције	Принцип раста и еволуције
Потреба за фреквентном повратном информацијом	Принцип трансформативних повратних веза
Проактивно управљање променама које се односе на захтеве пројекта	Принцип хитности реда („emergent order“)
Лагано („Loosely“) контролисање развојног окружења	Принцип дистрибуиране контроле
Планирање одржавати на минимуму	Принцип раста и еволуције
	Принцип хитности реда
Подстицање континуираног учења и унапређивања	Принцип раста и еволуције
	Принцип интеракција и релација
Нагласак на софтверском производу који ради	Принцип путање са најмањим напором („path of least effort“)

У раду [Augustine et al, 2005] разматра пројекте који се односе на примену агилних методологија као комплексне адаптивне системе (CAS) и представљају у овом раду развој CAS-базираног развојног оквира Агилног пројектног менаџмента. Према [Augustine et al, 2005], CAS системи су нелинеарни (функција целине не може да се добије простим сабирањем делова), отворени (интерагују са окружењем, примају улазе и обезбеђују излазе) и динамички (мењају се и еволуирају своје понашање као одговор на улазе у систем) системи. Полу-аутономни агенти представљају један од примера CAS система, наведених у овом раду.

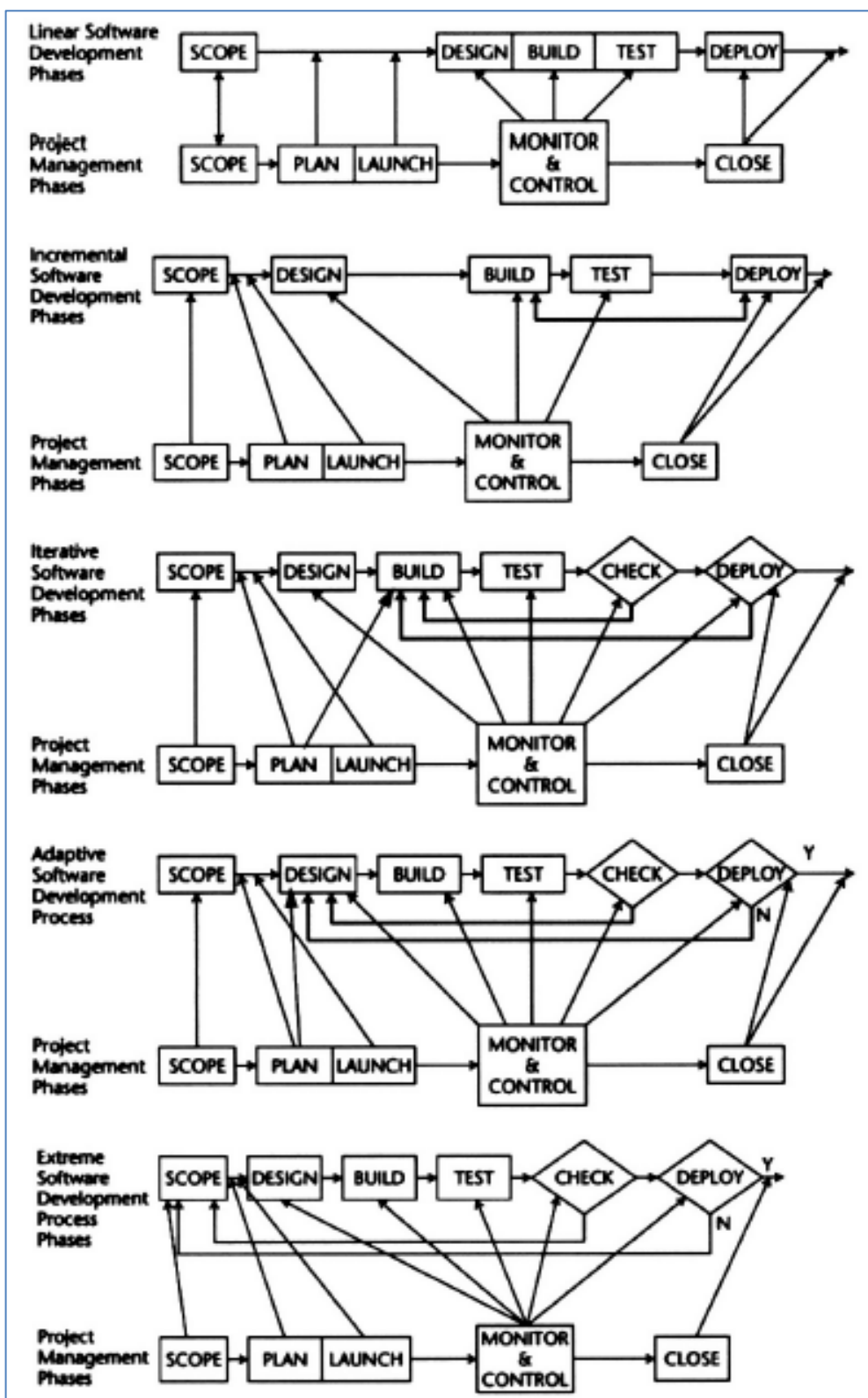
У раду [Coram&Bohner, 2005] разматран је утицај примене агилних метода развоја софтвера на елементе управљања софтверским пројектима и даје критички осврт на могућности и оправданост примене агилног приступа у оквиру управљања софтверским пројектима. Елементи који су разматрани у контексту утицаја су:

- Људски фактор - Примена агилног приступа има утицај на високе захтеве за квалитетним (способним да се прилагоде променама, имају добре способности комуникације...) кадровима у развојном делу, улога тестера је редукована уз увођење аутоматизације тестирања, пројектни менаџери су више ангажовани посебно у свакодневним састанцима са развојним тимом с једне стране, а с друге стране у фреквентнијим састанцима са клијентима. Агилни приступ подразумева константну сарадњу са клијентом, а у реалним пројектима је тешко успоставити такву сарадњу са клијентом који би обезбедио такву врсту расположивости свог представника, при чему представник треба да је посвећен, колаборативан, репрезентативан и да има довољно знања и утицаја у организацији да би могао да буде користан за сарадњу. Такође, кључни фактор успеха агилног приступа је кохезија тима, тј. способност да раде заједно, размењују често неструктурирана и недокументована знања и резултате.
- Процес – Планирање у агилном приступу је континуирано, за разлику од waterfall приступа где се одвија на почетку уз касније минималне измене. Планирање се своди на свакодневно одлучивање о приоритетима развоја, решавање проблема и додељивање задатака и алокацију ресурса. Документација је минимална, а процес развоја је фокусиран на рефакторисање, минимални развој, ревизије кода и континуалну интеграцију.
- Пројекат – у овом раду је наглашено да примена агилних метода не одговара свим типовима пројеката и не може се применити као универзални приступ развоју свих пројеката. Примена агилних метода је применљива код пројеката где спецификација није јасно дефинисана, где се примењују најновије технологије и где промене имају значајан утицај. Посебан утицај примене агилних метода јавља се на пословање, посебно код дефинисања уговора и потребе за дефинисањем захтева у оквиру уговора, код захтева вођења документације посебно код већих и државних пројеката, посебно код организација које имају уведен систем ISO9000 систем квалитета.

У раду [Fernandez&Fernandez, 2008] дефинисан је појам агилног пројектног менаџмента и описани су приступи у оквиру агилног пројектног менаџмента. Агилни пројектни менаџмент обухвата основне принципе добре праксе: претпоставке једноставности, прихватати измене, омогућити и фокусирати се на наредне напоре, инкрементално мењати, максимизирати вредност, управљати сврхом, испитивати акције, пројектни менаџмент треба да управља пројектом и границама процеса, брзи одговор од свих заинтересованих страна, испорука квалитета, креирање документације базирано на вредностима. У оквиру агилног пројектног менаџмента могу да се примене и обрасци (pattern), као и стратегије агилног пројектног менаџмента. Стратегије традиционалног пројектног менаџмента су: линеарна и инкрементална, док итеративна, адаптивна и екстремна стратегија припадају агилним стратегијама.



Слика 2.3.6.1. Стратегије пројектног менаџмента на основу комплексности и неизвесности у процесу развоја софтвера, према Wysocki [Wysocki, 2006] [Fernandez&Fernandez, 2008]



Слика 2.3.6.2. Стратегије пројектног менаџмента на основу комплексности и неизвесности у процесу развоја софтвера, према Fernandez&Fernandez [Fernandez&Fernandez, 2008]

У оквиру [Fernandez&Fernandez, 2008] описани су приступи стратешким обрасцима који су предложени у оквиру [Wysocki, 2006] (слика 2.3.6.1.), а проширење у контексту фаза пројектног менаџмента које се могу применити на одговарајуће фазе развоја софтвера предложено је у раду [Fernandez&Fernandez, 2008], што је приказано на слици 2.3.6.2.

## **2.4. АДАПТИВНОСТ И АДАПТИБИЛНОСТ ДИСТРИБУИРАНИХ ИНФОРМАЦИОНИХ СИСТЕМА**

### **2.4.1. Истраживања у области дистрибуираних информационих система**

#### **2.4.1.1. Предности и проблеми дистрибуираних информационих система**

Према [Krzyzanowski, 2000], предности коришћења дистрибуираних система огледају се у односу цена/перформансе, дистрибуција рачунара или мобилних уређаја су потреба свакодневног и пословног функционисања, умрежавање рачунара подржава кооперативни рад географски удаљених учесника, повећање поузданости, инкрементални раст, приступ удаљеним услугама, мобилност омогућава смештање/коришћење података са удаљених сервера (коришћењем мобилних уређаја). Према [DeBloch] развој дистрибуираног система мотивисан је економским разлозима (смањење трошкова хардвера), организационим разлозима (локална подршка организационих структура, интеграција постојећих система, локална аутономија), техничким разлозима (повећање перформанси – локалношћу процесирања и искоришћавањем паралелизма, висока расположивост и поузданост (због репликације), скалабилност).

Према [Tsui&Jeng, 2009], неки од проблема креирања дистрибуираних информационих система су: а) композиција ресурса различитих технологија који треба да омогуће безбедан трансфер података; б) услед коришћења мрежних ресурса размена података може бити непоуздана, в) саме компоненте софтвера које се налазе на различитим рачунарима могу бити непоуздане. Према [Krzyzanowski, 2000], неки проблеми креирања дистрибуираних система су: а) дизајн, имплементација и коришћење дистрибуираног софтвера може бити тешко, уз проблеме везане за оперативне системе и програмске системе који подржавају дистрибуиране системе, б) рачунарска мрежа може да „изгуби“ поруке или да буде преоптерећена. в) хардверско „појачавање“ мреже поновним умрежавањем може бити скупо и тешко изводљиво, г) Безбедност постаје највећи аспект проблема. Лак и погодан приступ подацима билогде из света креира проблеме безбедности. Према [Krzyzanowski, 2000], проблеми дизајна дистрибуираних информационих система односе се на постизање транспарентности (односно „скривања“ детаља имплементације), флексибилност и подела задатака који ће се извршавати на серверима, расположивост и поузданост, перформансе и скалабилност.

#### **2.4.1.1. Дистрибуиране базе података**

У раду [Milošević, 2010] описане су предности дистрибуираних система база података: повећање поузданости података репликацијама, боље перформансе система због могућности близине извора података кориснику, једноставан системски развој и проширивост система коришћењем каталога евиденције чворова. Основна карактеристика дистрибуираних база података је транспарентност, односно „скривање“ детаља о физичкој имплементацији фрагментације, репликације, трансакција итд.

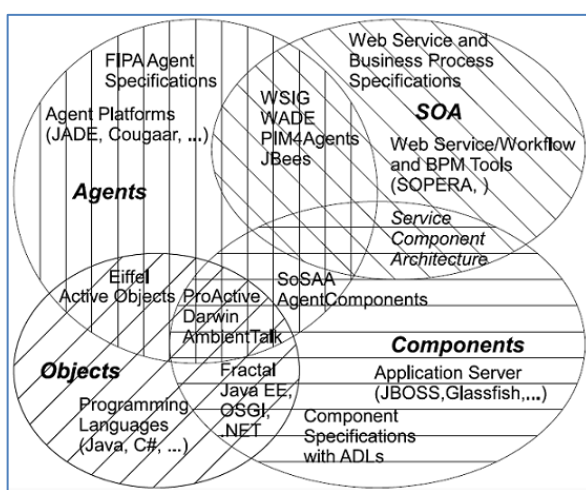
У раду [Khan&Ahmad, 2006] описан је проблем репликације података у дистрибуираном систему, које се може сматрати проширењем класичног проблема алокације фајлова. У наведеном раду приказано је истраживање унапређења репликације података користећи тзв. „себичне“ агенте. Проблем репликације података је обрађен користећи теорију игара. На сваком чвору „себични“ агенти се боре за могућност да реализују репликацију података у некооперативном окружењу. Борба агената се своди на борбу око меморијског простора који би био коришћен за реплику података, а заснива се на аукцијском систему.



У истом раду [Khan&Ahmad, 2006] описани су претходни резултати истраживања у области репликације података дистрибуираних система који се односе на: проблем алокација више фајлова у мултипроцесорском систему [Chu, 1969] [Mahmoud&Riordon, 1976], затим проблем алокације података у дистрибуираним базама података који се односи на стратегије за извршавање упита и одлуке у вези алокације [Apers, 1988], проблеме оптимизације и формализације проблема, коришћењем функција вишекритеријумског одлучивања [Dowdy& Foster, 1982], уз практичну примену у оквиру Интернет сервис провајдера [Rabinovich, 1998].

#### 2.4.1.2. Дистрибуиране софтверске компоненте

У оквиру истраживања [Braubach& Pokahr, 2011] дат је преглед софтверских концепата/елемената који се користе у оквиру програмирања дистрибуираних система, односно софтверских компоненти, објеката, агената и концепата сервисно орјентисане архитектуре (слика 2.4.1.2.1.). У наведеном истраживању предлаже се концепт активних компоненти.



Слика 2.4.1.2.1. Софтверски елементи дистрибуираних система према Braubach& Pokahr [Braubach& Pokahr, 2011]

У раду [Dzitas&Barbat, 2009] описане су основне карактеристике пост-индустријског друштва: пораст брзине промена, пораст комплексности система (архитектонске, когнитивне, структурне) и глобализација. У наведеном контексту, савремена ИТ решења карактерише: хетерогеност (технологија, приступа, методологија...), динамичност (брзина интерних и екстерних промена је велика), несигурност (информације и правила процесирања су неодређена, фази, променљива). Из наведених разлога, софтверске апликације морају имати особине: интелигентне и дистрибуиране. У наведеном раду описане су карактеристике софтверских агената који омогућавају јединство наведених особина.

У раду [Fortino&Russo, 2012] описана је методологија ELDAMeth која омогућава симулацију у области развоја дистрибуираних агентних система. Наведена методологија заснива се на агентном моделу ELDA (Event-driven Lightweight Distilled Statecharts Agents). Методологија је подржана одговарајућим CASE алатом за визуално креирање агента. Наведени приступ је илустрован студијом случаја у примени мобилних агената који служе за прикупљање података са различитих извора у мобилној мрежи.

У раду [Zeng et al, 2004] разматрају се аспекти квалитета web сервиса у оквиру композиције web сервиса. Уколико постоји понуда више web сервиса који обезбеђују исту функционалност, корисници могу изабрати web сервис са већим квалитетом, у складу са критеријумима који се односе на извршавање web сервиса: цена, трајање, репутација (поузданост), расположивост итд.

## 2.4.2. Истраживања у области адаптивних и адаптивних информационих система

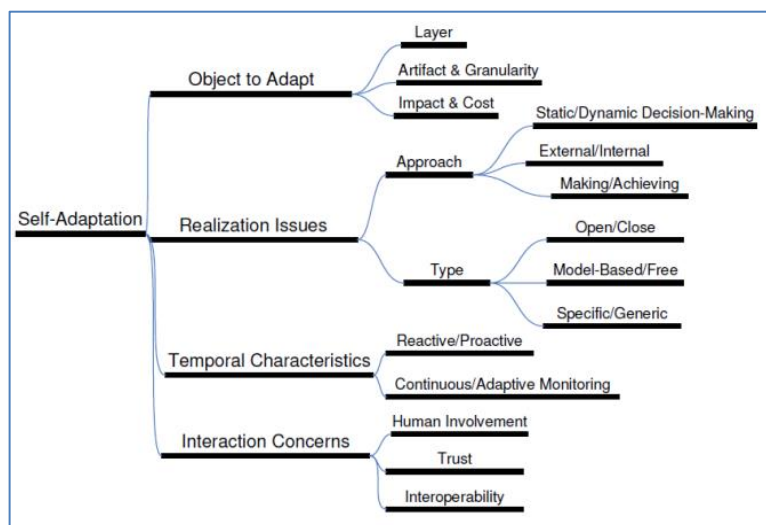
### 2.4.2.1. Адаптивни и само-адаптивни софтвер

У раду [Berg et al, 1995] приказана је имплементација структурно-базиране адаптације, заснована на објектно-орјентисаном програмирању, графу класа (који може бити променљив), графу пропагације (подграф графа класа) и спецификацији преношења (traversal specification). Према [Berg et al, 1995] под појмом адаптивни програм подразумева се скуп пропагационих образаца („collection of propagation patterns“), помоћу којих се граф објеката трансформише на основу спецификације преношења и мапе „умотавања“ (wrapper map) помоћу које се апстрактни објекти трансформишу у програмски код конкретног објектно-орјентисаног програмског језика. У наведеном раду дата су и општа правила трансформација. У транслацији адаптивног програма у конкретан циљни програмски језик, важно је очувати компатибилност, конзистентност и непроменљивост подкласа („subclass invariance“). Као конкретни програмски језик имплементације коришћен је C++.

У раду [Zhang&Cheng, 2006] представљен је модел-базирани развој динамички-адаптивног софтвера, где је адаптивност првенствено орјентисана на аспект понашања. Спецификација адаптације представљена је коришћењем графичких елемената представљања Петријевих мрежа („Petri nets“). У наведеном раду три врсте адаптације понашања су демонстриране на примеру софтвера за „GSM-oriented audio streaming protocol“ имплементацијом програма коришћењем Java програмског језика. Могући начини имплементације адаптивности су замена постојећег објекта новим или директан утицај на стања објеката. У наведеном раду примењена је замена објеката као начин имплементације.

У раду [Lemos& Fiadeiro, 2002] разматран је проблем обраде грешака код самоадаптивног софтвера и предложена је архитектура таквог софтвера код ког треба одвојити модул који се односи на процесирање основних задатака и услуга програма („computational“), модул који реализује координацију, односно контролише основно понашање програма и модул који реализује конфигурацију, односно садржи правила и сервисе који контролишу еволуцију система.

Таксономија истраживања у области само-адаптивности софтвера приказано је сликом 2.4.2.1.1. [Salehie&Tahvildari, 2009]:



Слика 2.4.2.1.1. Таксономија истраживања у области само-адаптивности софтвера, према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

У наставку је дат приказ резултата пројеката и истраживања у области самоадаптивних софтвера. [Salehie&Tahvildari, 2009]

Табела 2.4.2.1.1. Преглед истраживања у области самоадаптивног софтвера према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

Entity	Technique	Example
Sensors	Logging	GLA (Generic Log Adapter), LTA (Log Trace Analyzer) [IBM 2005]
	Monitoring & events information models	CIM (Common Information Model) [CIM ], CBE (Common Base Events) [IBM 2005]
	Management protocols and standards	Simple Network Management Protocol[SNMP ], Web-Based Enterprise Management[WBEM ], Application Response Measurement[ARM ], Siena [Carzaniga et al. 2001]
	Profiling	JVMTI (JVM Tool Interface) [JVMTI ]
	Management frameworks	JMX (Java Management eXtension) [JMX ]
	Aspect-oriented programming	BtM (Build to Manage) [IBM BtM ], JRat (Java Run-time Analysis Toolkit) [ShiftOne JRat ]
	Signal monitoring	Heartbeat and pulse monitoring [Hinchey and Sterritt 2006]
Effectors	Design patterns	Wrapper (Adapter), Proxy, Strategy Pattern [Gamma et al. 1995]
	Architectural patterns	Microkernel pattern, Reflection pattern, Interception pattern [Buschmann et al. 1996; Alur et al. 2001]
	Autonomic patterns	Goal-driven self-assembly, self-healing clusters and utility-function driven resource allocation [Kephart 2005]
	Middleware-based effectors	Integrated middleware, Middleware interception [Popovici et al. 2002], Virtual component pattern [Schmidt and Cleland 1999]
	Metaobject protocol	TRAP/J [Sadjadi et al. 2004]
	Dynamic aspect weaving	JAC [Pawlak et al. 2001], TRAP/J [Sadjadi et al. 2004]
	Function pointers	Callback in CASA [Mukhija and Glinz 2005]

У оквиру наредне табеле приказане су типичне адаптационе (јаке и слабе) акције у оквиру самоадаптивног софтвера.

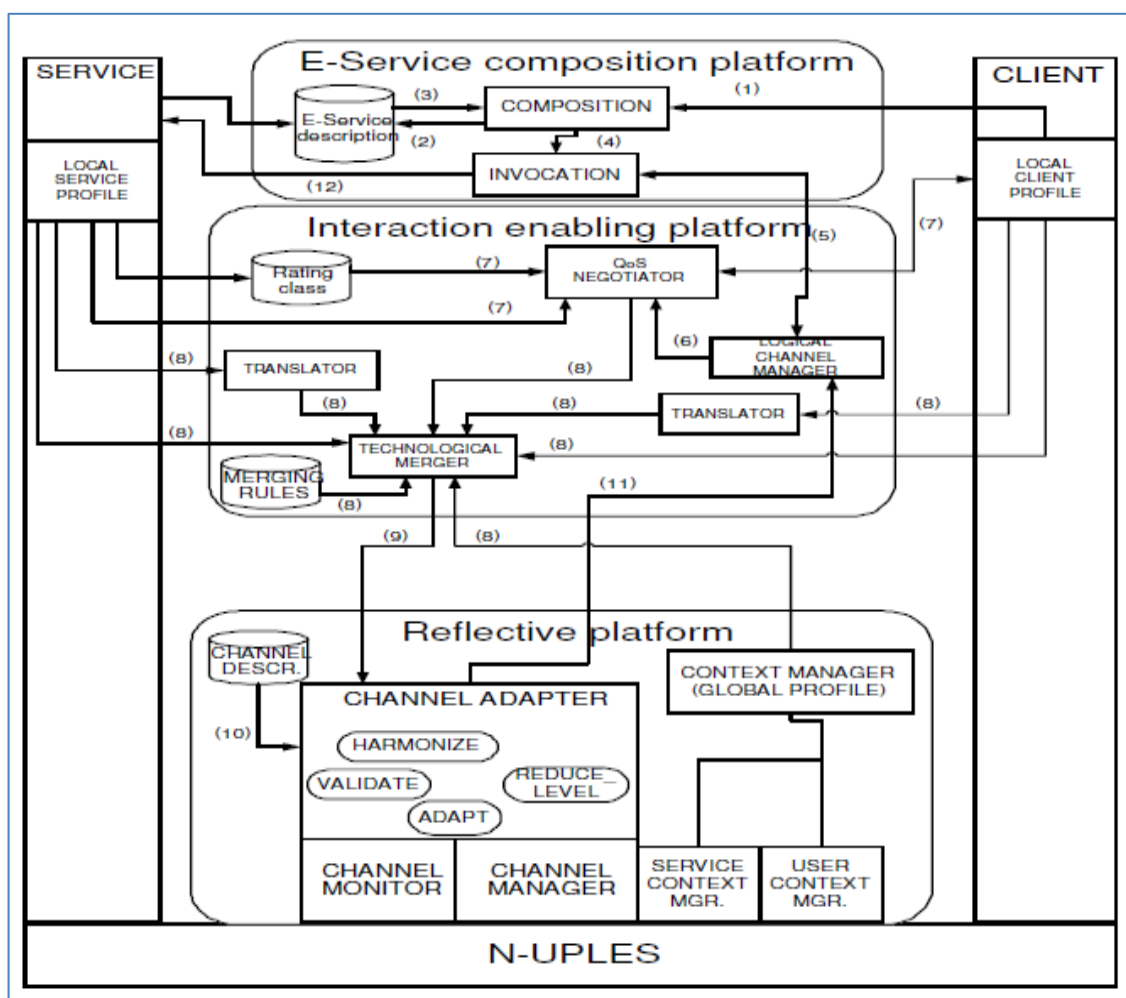
Табела 2.4.2.1.2. Типичне адаптационе (јаке и слабе) акције у оквиру самоадаптивног софтвера према Salehie&Tahvildari [Salehie&Tahvildari, 2009]

Type	Action	Description
Weak	Caching [Oreizy et al. 1999; Dowling and Cahill 2004]	Caching data, states, connections, objects or components in order to lower the response time, load of servers, or help decentralized management
	Changing data quality [Mukhija and Glinz 2005]	Changing data quality (i.e., lower resolution) to save bandwidth and increase speed
	Changing type of data [Cheng et al. 2006; Mukhija and Glinz 2005]	For instance, switching from video to image and even to text to save bandwidth and increase speed
	Compressing data [Laddaga et al. 2001]	Saving bandwidth by transceiving compressed data
	Tuning (parameter adjusting) [Karsai et al. 2001]	Adjusting parameters to meet some adaptation goals (i.e., buffer size and delay time)
	Load balancing [Willebeek-LeMair et al. 1993; Cardellini et al. 1999]	Fair division of load between system elements to achieve maximum utilization, throughput or minimum response time
	Changing aspects [Pinto et al. 2002; Suvéé et al. 2003]	Changing aspect of a component or object with another one with different quality
Strong	Changing algorithm/method [Oreizy et al. 1999; Robertson and Williams 2006]	Changing the algorithm/ method to meet self-* properties and run-time constraints
	Replacement, addition & removal [McKinley et al. 2004]	Replacing an entity (e.g., a component) by another one with the same interface but different quality (non-functional)
	Restructuring /changing architecture [Kramer and Magee 1990; Magee and Kramer 1996; Oreizy et al. 1998]	Changing organization/ architecture of the system (it may change the architectural style or design patterns of the system)
	Resource provisioning [Appleby et al. 2001]	Provisioning additional resources at different levels (this action can be extended to adding/removing any resources, such as servers)
	Restarting / redeployment [Candea et al. 2006]	Restarting/ rebooting (macro- or micro-) or redeployment of system entities at different levels mainly due to faults/failures

Дисциплине које подржавају истраживања у области само-адаптивних софтвера су [Salehie&Tahvildari, 2009]: софтверско инжењерство, вештачка интелигенција, теорија и инжењерство контроле, мрежно и дистрибуирано рачунарство. Анализа реализованих пројеката, укључујући и индустријске пројекте извршена је у односу на кључне особине и процесе само-адаптивних система [Salehie&Tahvildari, 2009]. У односу на интерне карактеристике издваја се пројекат Rainbow, а у односу на подршку за четири процеса пројекат Self-Adaptive.

### 2.4.2.2. Адаптивни и адаптибилни информациони системи

У раду [Maurino et al, 2003] описан је приступ адаптацији информационог система у контексту е-Сервиса. Постојећа решења адаптивности односила су се на прилагођавање сервиса улазним подацима који су достављани путем дистрибуционих канала. Дистрибуциони канал карактерише скуп сервиса који су на располагању, као и технолошке карактеристике имплементације. Стање система описано је у сваком тренутку н-торком атрибута (карактеристика) наведеног система. Сваки атрибут има опсег вредности у домену свог функционисања, а у технолошком смислу може бити: видљив или да има могућност да буде контролисан. У наведеном раду предложен је нови приступ где се с адаптирају дистрибуциони канали у односу на постојеће сервисе. У наведеном раду предложен је приступ разликовању логичке и технолошке адаптивности. Технолошка адаптивност настоји да промени неке карактеристике како би се прилагодила ограничењима, а логичка адаптивност доставља комплетно нову н-торку карактеристика. На слици 2.4.2.2.1. представљена је предложена архитектура система.



Слика 2.4.2.2.1. Архитектура адаптивног информационог е-Сервис система, према Maurino et al [Maurino et al, 2003]

Платформа за композицију е-Сервиса задужена је за пријем захтева од клијента и обезбеђивање покретања одговарајућег е-сервиса. Платформа интеграције задужена је за прикупљање ограничења од стране клијента, е-сервиса и контекста и на основу тога дефинисање карактеристика квалитета услуге (quality of service). Платформа за резонување („Reflective platform“) задужена је за мониторинг колико сам е-сервис извршава своје активности у складу са дефинисаним нивоом квалитета и за адаптирање дистрибуционог канала у складу са захтевима квалитета. Различити корисници могу имати различите нивое, тј. интервале задовољивости квалитета. Зато је скуп ограничења дефинисан кроз скуп карактеристика компоненте и доње и горње границе дозвољених вредности:

$$T_{c_i}^j = \langle v_{min_i}, v_{max_i}, attribute, component \rangle$$

Слика 2.4.2.2.2. Формула која представља параметре вредновања квалитета атрибута одређених компоненти на основу дефинисаног интервала прихватљивости вредности према Maurino et al [Maurino et al, 2003]

Платформа за резонување се састоји из модула: Монитор (мерење вредности карактеристика), Адаптер (вредновање мерених вредности у односу на потребан ниво вредности карактеристика), Манагер (реализује промене иницирањем одговарајућих софтверских компоненти). У наведеном контексту, „адаптирати значи минимизовати одступања између захтева сервиса и доступности канала уз максимизирање квалитета услуге.“ Ради реализације ових захтева, функција валидације врши упоређивање текућих вредности карактеристика са потребним нивоом, а функција адаптације врши корекције.

У оквиру рада [Houben, 2005] описано је стање у области адаптивних web-базираних информационих система (AWIS), у контексту адаптивних хипермедијалних система, моделовања корисника, персонализације итд. Технолошки, адаптација се заснива на примени правила која дефинишу услове под којим се презентују информације, а наведена правила могу бити контекст-независна (али зависна од могућности технологије уређаја) и контекст-зависна (зависна од процеса у којем се извршавају). Контекст-зависна правила се примењују у e-learning системима где се у одређеном тренутку достављају садржаји ученику у складу са његовим тренутним стањем (његовим нивоом знања и приказаних способности у одређеном тренутку). Према [Houben, 2005], имплементација адаптације заснива се на *метаподацима и правилима* која се односе на садржај, корисника, пословни контекст итд. који може бити описан коришћењем *онтолошких записа семантичког web-а* (RDF, OWL). У адаптацији потребно је користити различита правила и различито вредновати разне факторе. У оквиру адаптације изузетно је важно узети у обзир и мерење и задовољавање квалитета, посебно квалитета модела и проверити конзистентност модела колико је у складу са спецификацијом, као и верификовати употребљивост решења. У техничком смислу, у оквиру имплементације AWIS система потребно је водити рачуна о подршци мобилности, кеширању, репликацији, поновној искористивости (reusability) сервиса и компоненти, односно примени дизајн патерна и реализације персонализације, тј. адаптације апликације кориснику („user adaptation“).

У раду [Carpiello et al, 2006] описан је приступ управљању контекстом у оквиру мултиканалног адаптивног web-базираног информационог система (MAIS). Услуге кориснику се динамички генеришу у односу на карактеристике (категорију) корисника, карактеристике уређаја који корисник користи (укључујући мобилне уређаје) и расположивост сервиса који су реализовани у различитим технологијама и доступни од стране различитих услужних ентитета. Наведени приступ користи онтологије ради дефинисања контекста:

- Домен-зависног и домен-независног профила корисника кроз листу парова: атрибут-вредност, који могу бити експлицитно (од стране корисника) или

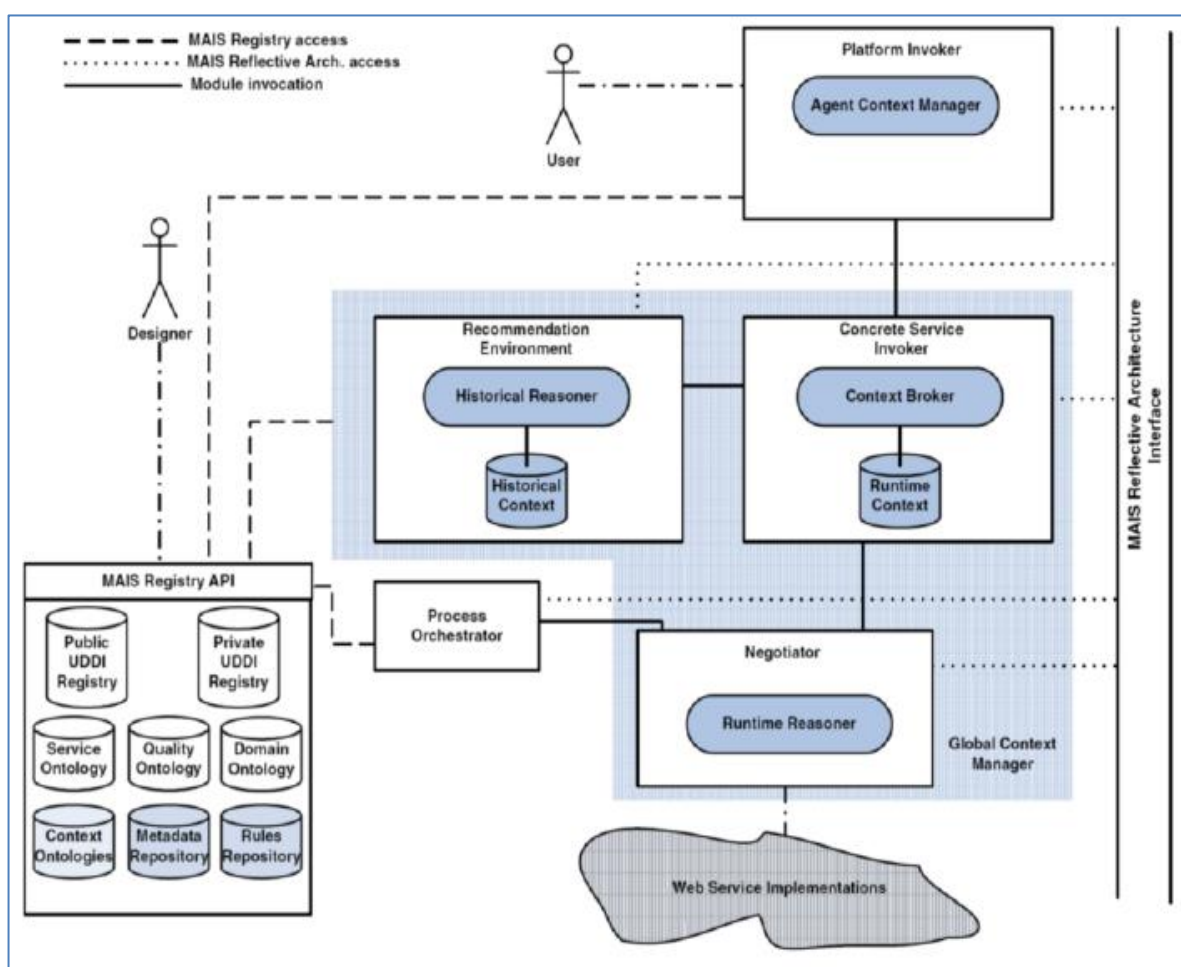
имплицитно дефинисане (аутоматским резонувањем дефинише се класа корисника на основу вредности атрибута).

- Окружење корисника – географска локација, услови амбијента, време, акције и интеракција са простором који окружује корисника
- Канала интеракције корисника са платформом која обезбеђује сервисе – уређај, мреже, мрежни интерфејс, апликациони протокол.

Управљање контекстом реализовано је коришћењем компоненти:

- Repository – садржи метаподатке, правила и онтологије
- Context Manager – процесира резонување на основу историјских и извршних (run-time) података
- Context Agent – омогућава примену контекст заснованог процесирања у оквиру удаљених уређаја, прикупљањем података о контексту на страни корисника.

Приликом примене, корисник бира апстрактне сервисе, који се трансформишу у конкретне сервисе у складу са контекстом. На наредној представљена је архитектура MAIS система.

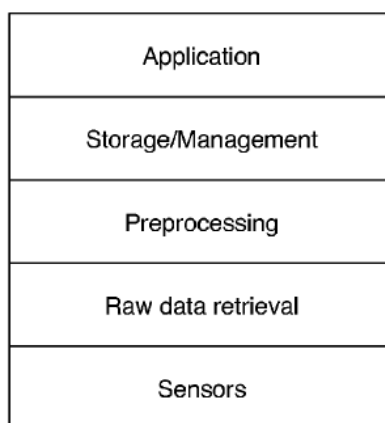


Слика 2.4.2.2.3. Архитектура мултиканалног адаптивног web-базираног информационог система (MAIS) система према Carpiello et al [Carpiello et al, 2006]

У раду [Baldauf et al, 2007] представљено је истраживање које се односи на адаптивност контекст-свесних („context-aware“) система. У раду [Schilit&Theimer, 1994] описан је први пут термин *контекст-свесности* кроз податке о локацији, идентитетима људи из окружења, објеката и промена тих објеката. У раду [Brown, 1996] дефинисан је појам контекста као елементе окружења корисника којих је свестан рачунар који тај корисник користи, односно [Dey& Abowd, 2000] као „билокоја информација која може бити коришћена да опише ситуацију ентитета (особа, простора или објеката) који се сматрају релевантним у интеракцији између корисника и апликације, укључујући и

информације о самом кориснику и апликацији." У раду [Hofer et al, 2003] извршена је категоризација контекста на физички (екстерни) контекст (који се може објективно измерити хардверским сензорима – локација, светло, звук итд.) и логички (интерни) контекст (који се односи на карактеристике корисника (могу се евидентирати на основу мониторинга корисниковог понашања – циљеви, задаци, радни контекст, пословни процеси, корисниково емоционално стање). Сензори могу бити: физички (хардверски), виртуелни (прикупљају податке из апликација или сервиса) и логички (комбинују податке из хардверских и виртуалних сензора како би решавали сложеније претраге и задатке).

У раду [Baldauf et al, 2007] представљени су различити приступи технолошкој имплементацији контекст-свесних система: директни приступ сензорским подацима од стране апликације, трослојна апликација са одвајањем сензора и прикупљања података од процесирања у средњем слоју презентовања у корисничком интерфејсу, до контекст сервера, који омогућавају истовремени приступ сензорским подацима од стране више различитих дистрибуираних апликација. Одвајањем детектовања и коришћења података о вишеслојном приступу омогућава проширивост (extensibility) и поновну искористивост (reusability) система.



Слика 2.4.2.2.4. Архитектура контекст-свесних система, према Baldauf et al [Baldauf et al, 2007]

Према [Baldauf et al, 2007], контекст модели представљају записе о подацима о контекстним атрибутима у форми која је могуће да буде процесирана од стране рачунара. Неки од најчешће коришћених контекстних модела су: 1) кључ-вредносни модел, 2) Markup-scheme модел хијерархијски уређених података (нпр. RDF) 3) графички модел (UML), 4) објектно-орјентисани модел, 5) Модел базира на логици (погодан за аутоматско резонување), 5) онтолошки засновани модел. Према [Baldauf et al, 2007], онтологије се могу користити као контекст-онтологије, које имају као атомарни елемент пар: контекстни тип (мерну величину) и контекстну вредност (мерну вредност). Онтологије се могу користити у комбинацији са базом знања како би се аутоматским резонувањем издвајали закључци. Пример контекстног правила:

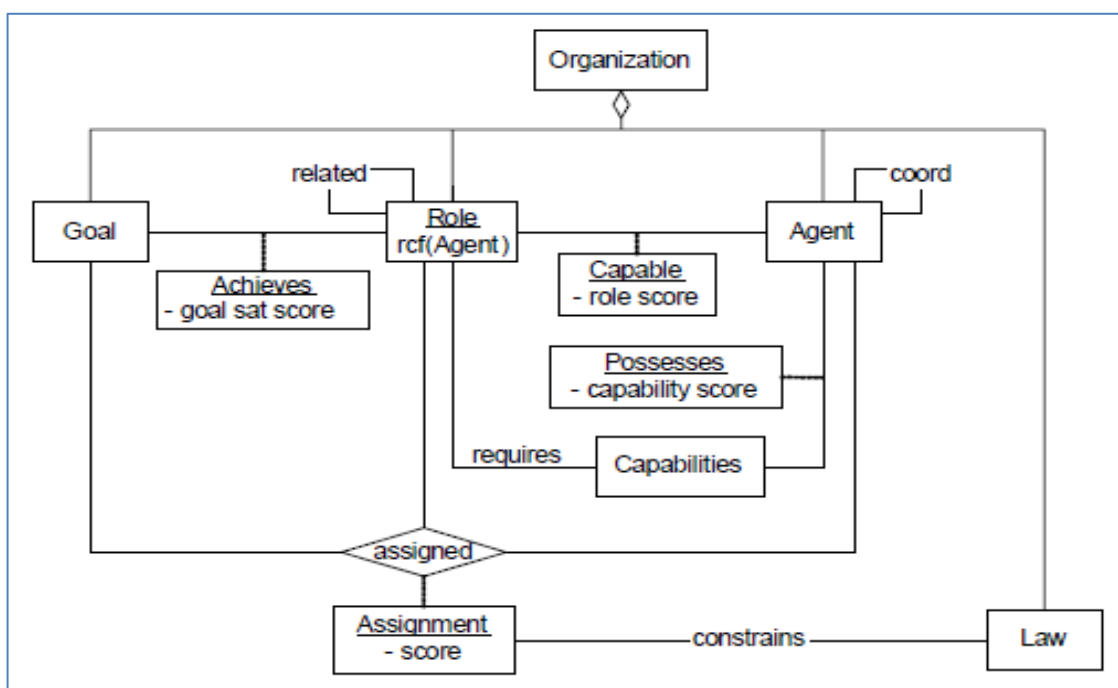
*Context(Number of people, Room 2401, >, 4) AND Context(Application, Powerpoint, is, Running) ⇒ Context(Social Activity, Room 2401, Is, Presentation).*

Адаптивни информациони системи примењени су у различитим областима. У оквиру војно-стратешких апликација у контексту прикупљања и процесирања података о току борби на бојном пољу, адаптивни информациони систем је креиран и описан у [Matson& DeLoach, 2005]. Наведени систем је базиран на мулти-агентном приступу. У наведеном приступу, организациони модел састоји се од: *подсистема структуре, скупа стања и функција транзиције.*

Подсистем структуре састоји се из:

- Скупа циљева које тим настоји да постигне
- Скуп улога које морају да се одвијају како би се постигли наведени циљеви

- Скуп способности (capabilities) потребних да би се одвијале одговарајуће улоге; ако би се односиле на агенте, способности су: сензорске способности, способности актуатора и способности процесирања („computational“)
- Скуп правила или закона која представљају ограничења организације, односно у контексту агената, односе се на додељивање задатака агентима.



Слика 2.4.2.2.5. Структурни модел организације адаптивног информационог система, према Matson & DeLoach [Matson & DeLoach, 2005]

У оквиру медицинских информационих система, адаптивност информационог система [Lenz & Kuhn, 2004] здравствене установе је разматрана у контексту модел-базираног вишеслојног развоја информационог система, заснованог на примени генератора кода, како би се подржао концепт еволуирајућег информационог система.

У раду [Edwards & Pengelly, 1997] описана је примена симулације у истраживању могућности за само-адаптивношћу информационог система. Адаптирање система су разматрано је у контексту прилагођавања људима и пословним процесима кроз реорганизацију и поновно планирање, што би се реализовало у техничком смислу на учитавање новог шаблона функционисања система.

### 2.4.3. Истраживања у области адаптивних и адаптивних дистрибуираних информационих система

#### 2.4.3.1. Приступ и архитектуре адаптивних и адаптивних дистрибуираних информационих система

У раду [DAISY] описани су приступи развоју информационих система и адаптивних дистрибуираних система. Дат је модел адаптивног дистрибуираног информационог система DAISY (Distributed and Adaptive Information System), који представља оквир за истраживања у овој области, односно у оквиру истоименог пројекта. У наведеном раду описане су промене развојних парадигми у контексту развоја информационих система, које су довеле до дефинисања система DAISY:

1. Од развоја система до интеграције система – савремени информациони системи настају интеграцијом готових хардверских и софтверских решења (COTS – Commercial Off The shelf Software) и потребно је ради интеграције да се омогући аутоматска адаптивност софтвера на различита технолошка окружења (минимизујући потребу да те промене реализују корисници, који често нису довољно обучени).



2. Од програм-центричног ка информационо-садржај-центричним системима – са појавом www све више су апликације повезане у циљу обезбеђивања што бољих услуга кроз бољи дизајн, информативност, ефикасност и атрактивност, уз нагласак на потреби бољег одржавања.

3. Од самосталних апликација до отворених дистрибуираних система – с појавом рачунарских мрежа и интернета, апликације и информациони системи се реализују у дистрибуираном окружењу с циљем да извршавање апликације никад не може да престане, због већег броја рачунара који увек могу да преузму део функционалности. У таквим системима јавља се проблем безбедности. Потребно је направити такву адаптивност система да се систем сам мења без заустављања функционисања.

4. Од подешљивог корисничког интерфејса до адаптивног корисничког интерфејса – кориснички интерфејс може да се подеси од стране корисника који могу бити професионални корисници из области информационих технологија, али и особе које нису стручне у датој области. Зато је циљ омогућити аутоматску адаптивност за ту другу категорију корисника.

5. Од временски одређеног процесирања до процесирања вођеног крајњим корисником – поједине апликације није лако унапред имплементирати, већ треба да буде омогућена њихова селекција и интеграција у складу са потребама реалног пословног процеса.

У оквиру рада [Steen, 2012] најчешће архитектуре дистрибуираних информационих система представљају вишеслојне системе где сваки слој функционише на посебној машини. У оквиру наведене архитектуре, адаптивност у средњем слоју („middleware“) је у досадашњим истраживањима реализовано као:

- Одвајање аспеката („Separation of concerns“) – настојање да се додатне функционалности одвоје и касније уклопе у заједничку имплементацију.
- Резоновање у току процесирања („Computational reflection“) – рачунарски програм врши самоконтролу („inspect itself“) у току извршавања и адаптира се/мења параметре рада (settings) динамички ако је потребно.
- Компонент-базирани дизајн – организација дистрибуиране апликације кроз компоненте које могу динамички бити замењене ако је потребно

У раду [Pathak et al, 2007] описана је примена приступа комплексних адаптивних система (Complex Adaptive System - CAS) у оквиру реализације мреже за набавку (Supply Network) где је потребно реализовати динамичку повезаност више различитих организација.

У раду [Ben-Shaul et al, 2000] описан је самоадаптивни систем FarGO као интернет-базирани систем чија се самоадаптација односи на структурну самоадаптацију динамичком интер-компонентном структуром.

#### **2.4.3.2. Адаптивно процесирање дистрибуираних база података**

У оквиру области истраживања која се односи на адаптивне базе података, једно од истраживања бави се адаптивном репликацијом података [Wolfson& Kalpakis, 1997]. У наведеном истраживању репликација података заснива се на кеширању, које треба да обезбеди боље перформансе дистрибуираног система, а адаптивност се заснива на алгоритму који мери трошкове репликације и на основу тога одлучује о начину репликације.

Други аспект адаптивности у односу на базе података односи се на адаптивно претраживање и филтрирање података. У раду [Huang et al, 2001] описан је систем за дистрибуирану претрагу података користећи механизам заснова на адаптивној мета-претрази. Наведени систем обједињује податке из разних извора, интегрише и даје могућност постављања различитих упита, уз пратеће моделовање корисника и адаптацију корисниковим преференцијама, односно динамички се адаптира кориснички интерфејс и извори података у односу на упите.

У раду [Liu et al, 2005] описане су технике адаптивности које се односе на подршку процесирању континуираних упита у дистрибуираном окружењу. Адаптивност је реализована на нивоу оператора, на нивоу стања, планирања run-time упита.

У раду [Klemm et al, 2005] представљен је адаптивни систем за дистрибуиране peer-to-peer системе, који мониторише индексе који су коришћени у претраживању података. У наведеном истраживању креирана је адаптивна hash табела.

#### **2.4.4. Технолошка решења адаптивности и адаптивности у дистрибуираним информационим системима**

##### **2.4.4.1. Адаптивност моделовањем корисника**

Према [Zukerman& Albrecht, 2001], моделовање корисника укључује закључивање о невидљивим информацијама о кориснику на основу видљивих, али најчешће некомплетних, информација о корисниковим акцијама или исказима. У оквиру наведеног процеса, због некомплетности информација, процес моделовања корисника се реализује у оквиру одређеног степена неодређености. Један од циљева моделовања корисника је препознавање плана активности корисника, како би се могли креирати закључци о корисниковим преференцама, циљевима и будућим акцијама и локацијама. У оквиру рада [Zukerman& Albrecht, 2001] дат је преглед техника креирања модела корисника:

1. Класично (ручно) креирање базе знања [Carberry, 2001], које има недостатке: велико коришћење ресурса за прикупљање и припрему, немогућност проширења и адаптирања

2. Предиктивни статистички модели [Larson, 1969] - заснивају се на предиктивном статистичком моделу корисника (чиме се предвиђа понашање корисника: циљеви, акције и преференце) заснованог на статистици над подацима о кориснику. Најчешће коришћени предиктивни статистички модели су: садржај-базирани (полазно начело: корисник се у сличним околностима увек понаша на сличан начин) и колаборативни (полазно начело: корисник се у групи у сличним околностима увек понаша на сличан начин).

3. Технике вештачке интелигенције које припадају подршци предиктивних статистичких модела категоришу се у областима машинског учења и резоновања у околностима неодређености. Машинско учење се заснива на тренинг скупу, на основу којег систем учи и тест скупу, на основу којег се проверавају перформансе система. Неки приступи су:

- Линеарни модели – базирају се на квантификованој суми познатих вредности како би се израчунала квантитативна вредност непознате.
- TFIDF (Term Frequency Inverse Document Frequency) модел – садржај-базирани системи који препоручују документе кориснику базирано на другим сличним документима од интереса
- Модели Маркова –наступајући догађај се предвиђа на основу вероватноће тог догађаја на основу на основу података о прошлим догађајима
- Методе класификације података – подаци се повезују и формирају се групе сродних података, нпр. техником cluster mining [Perkowitz&Etzioni, 2000]
- Индукција правила
- Неуронске мреже
- Дрво одлучивања
- Бајесове мреже – усмерени графови, чији су чворови повезани узрочно-последичним везама, а представљају променљиве и њима је додељена вероватноћа наступања у односу на вредности променљивих са којима су повезани у родитељ-дете односу.

#### **2.4.4.2. Адаптивност пословних процеса и примена пословних правила у адаптивним информационим системима**

У раду [Pestic&Aalst 2006] описан је декларативни приступ флексибилности у управљању пословним процесима. Декларативни приступ подразумева да је за дефинисање пословног процеса потребно одредити шта треба да буде урађено, а не како. У оквиру наведеног приступа реализован је прототип софтвера и језик моделовања *ConDec* који омогућава моделовање и извршавање динамичких промена пословних процеса.

У раду [Reichert et al, 2005] описан је пројекат ADEPT, у оквиру ког је развијен адаптивни процес-свесни информациони систем. Промене пословног процеса се помоћу наведеног система могу вршити у структурном делу (промене типа процеса) или на нивоу инстанце процеса (решавајући појединачни специфичан случај).

У раду [Khriss et al, 2008] описан је предлог технолошке реализације адаптивности информационог система на промене у пословним процесима. Промене информационог система као подршке пословних процеса се могу односити на време, обухват и перспективу. У односу на време, промене могу бити у дизајн режиму и извршном режиму. У односу на обухват, промене се односе на величину – промене тока процеса или специфичног случаја и пратећих трансакција. Када је перспектива у питању, промена се може односити на процес, податке или сервис. Промене процеса се односе на додавање и брисање процеса, као промену редоследа извршавања. Промене података се односе на структуру и усмерење порука између сервиса и структуру сложенијих израза у оквиру контролних процеса. Промене сервиса који обезбеђују подршку неком пословном процесу се могу односити на промену типа и добављача сервиса, а посебно могу бити динамичке промене везе ка *web servisima*. У овом раду описан је *Change Protocol for Collaboration (CPC)* који се може користити за усклађивање пословних процеса између пословних партнера. Промена шеме пословног процеса и увођење нове шеме реализована је применом *BPEL2Java (B2J) engine* обезбеђене од стране *Eclipse Service Tools Project*. У оквиру примене *CPC*, примењена су и одређена правила која се односе на стања која одређују покретање одређених активности.

Према [Wang, 2005], развој само-адаптивног софтвера је инспирисан потребом за успешним одржавањем софтвера који функционише у окружењу где постоји низ догађаја, који може утицати на успешност функционисања софтвера. Само-адаптивни софтвер треба да садржи 3 најважније компоненте за подршку: мониторингу, одлучивању и акцијама (активностима). У раду [Wang, 2005] представљен је приступ развоју само-адаптивног софтвера базираном на примени правила кроз посебно дефинисан Модел правила. У наведеном моделу разликују се догађаји, параметри и правила и они су јасно дефинисани у оквиру XML записа:

- Догађаји – над њима се врши мониторинг ради детектовања промена-представљају промене стања из окружења или из самог система које могу да утичу на успешност извршавања основних процеса и на које систем треба да се адаптира. Догађаји могу бити грешке извршавања програма, потенцијални напади и неауторизовани приступи, измене у окружењу (пропусна моћ мреже, број конектованих клијената), издвајање нових улазних чињеница адаптирањем на корисникове потребе. Механизми којима се постиже реаговање на наведене догађаје су подржани на разним нивоима: оперативни систем, база података, програмски језик итд.
- Параметри – треба да буду промењени када наступи неки од догађаја на које треба реаговати; треба разликовати имплицитне и експлицитне параметре. Различити типови параметара се могу мењати у оквиру различитих нивоа софтвера. Неки параметри могу добијати директне унапред одређене вредности, док неки параметри
- Правила – креирају се на основу догађаја, параметара и логике. Основна форма правила је: „АКО услов ОНДА акција“. Услови су дефинисани на основу догађаја,

подуслова који се односе на стања и логичких оператора И, ИЛИ или НЕ. Догађаји изазивају покретање извршавања правила, а стања заправо представљају акумулацију догађаја. Акције су дефинисане на основу параметара, односно односе се на операције. Већина операција, у само-адаптивним системима, извршава промену параметара. Правила су у оквиру [Wang, 2005] имплементирана коришћењем XML хијерархијске структуре где се издваја сегмент услова (догађај кроз мерење карактеристике компоненте и стање) и скуп акција.

```

<?xml version=1.0' ?>
.....
<Rule>
  <Condition>
    <Events>
      <Component>
        <Name>A</Name>
        <ClientNumber>
          <UpperLimit>100</UpperLimit>
        </ClientNumber>
      </Component>
    </Events>
    <States>
      <Relation>AND</Relation>
      <CPU>
        <Available>0.1</Available>
      </CPU>
      <RAM>
        <Available >2</Available>
      </RAM>
    </States>
  </Condition>
  <Action>
    <Parameters>
      <Component>
        <Name>A</Name>
        <InstanceNumber>120</InstanceNumber>
      </Component>
    </Parameters>
  </Action>
</Rule>
.....

if ( ( A.GetClientNumber() > 100 )
    &&( CPU.Available() >= 0.1
        && RAM.Available() >= 2 ) ){
A.SetInstanceNumber(120);
}

```

Листинг 2.4.4.3.1. Пример XML записа са правилима која омогућавају самоадаптивност софтвера и еквивалент у објектно-орјентисаној програмској имплементацији, према Wang [Wang, 2005]

Предложени модел правила је имплементиран у оквиру J2EE апликативног сервера. Механизам примене правила („Rule Engine“) је кључни елемент система који треба да омогући повезивање података од рецептора (дела који ради мониторинг стања и догађаја) са акцијама које су повезане, путем правила, са променом стања или наступањем догађаја. Извршавање акција се односи на промену параметара система. Елементи из XML правила се мапирају у одговарајуће процедуре, односно методе одговарајуће класе која имплементира одговарајуће процедуре које се односе на мерење вредности стања и догађаја, односно процедуре које се односе на акције, односно промене параметара система.

### 2.4.4.3. Адаптибилност у сервисно-орјентисаним архитектурама

У раду [Khriss et al, 2008] разматране су могуће ситуације и области промена у контексту промена пословних процеса, посебно у оквиру колаборације више организација и њихових информационих система. Као погодна технологија предложена је примена сервисно-орјентисаних архитектура заснована на Web сервисима, који се динамички могу позивати и укључивати у колаборативне радне токове. За потребе дефинисања тока пословних процеса у контексту примене Web сервиса, користи се Business Process Execution Language for Web Services (WS-BPEL), а за приказ начина позива web сервиса, користи се Web Services Choreography Description Language (WS-CDL), односно UML дијаграм секвенци кроз приказ сценарија. У смислу подршке променама у пословним процесима, дефинисан је протокол којим се дефинише уговор између носиоца пословног процеса (master) и подређених партнерских система (slave), где master обавештава, а подређени систем прихвата или одбија примену нових пословних процеса. У оквиру овог рада дат је и упоредни приказ адаптивности у односу на промене пословних процеса и утицај на примену различитих web сервиса у оквиру постојећих професионалних решења (Microsoft BizTalk Server, Oracle BPEL Process Manager, IBM WebSphere Process Server, Hewlett Packard eFlow), као и неколико прототипских решења у оквиру истраживачких резултата.

У раду [Charfi&Mezini, 2004] анализирани су недостаци једног од најзаступљенијих језика за композицију пословних процеса применом web сервиса (BPEL4WS -Business Process Execution Language for Web Services) у погледу модуларизације и аспект-орјентисане примене пословних правила која треба да се извршавају над више веб сервиса, као и у погледу омогућавања динамичких измена композиције веб сервиса у току извршавања (односно адаптибилности на промене).

```
<process name = "echoString" .../>
  <variables>
    <variable name="request" messageType="StringMessageType"/>
  </variables>
  <partners>
    <partner name="caller" serviceLinkType="tns:echoSLT"/>
  </partners>
  <sequence name="EchoSequence">
    <receive partner="caller" portType="tns:echoPT"
      operation="echo" variable="request"
      createInstance="yes" .../>
    <reply partner="caller" portType="tns:echoPT"
      operation="echo" variable="request"
      name="EchoReply"/>
  </sequence>
</process>
```

Листинг 2.4.4.3.1. Пример описа процеса применом BPEL4WS, према Charfi&Mezini [Charfi&Mezini, 2004]

Као алтернативно решење, у овом раду се предлаже подршка примене аспект-орјентисаних принципа програмирања (уз илустрацију елементима језика AspectJ) у омогућавању динамичке композиције web сервиса орјентисане на пословне процесе и аспекте кроз проширење BPEL4WS под именом AO4BPEL. С обзиром да су BPEL процеси представљени као XML документи, у процесирању ових XML докумената се користи XPath упитни језик.

```

<aspect name="Counting">
  <partnerLinks>
    <partnerLink name="JavaExecWSLink" .../>
  </partnerLinks>
  <variables>
    <variable name="invokeMethodRequest" .../>
  </variables>
  <pointcutandadvice type="after">
    <pointcut name="Lufthansa Invocations">
      //process//invoke[@portType = "LufthansaPT" and
        @operation = "searchFlight"]
    </pointcut>
    <advice>
      <sequence>
        <assign>
          <copy>
            <from>increaseCounter</from>
            <to variable="invokeMethodRequest" part="methodName" />
          </copy>...
        </assign>
        <invoke partnerLink="JavaExecWSLink" portType="JavaExecPT"
          operation="invokeMethod"
          inputVariable="invokeMethodRequest" />
      </sequence>
    </advice>
  </pointcutandadvice>
</aspect>

```

```

<aspect name="AddCarRental">
  <partnerLinks>
    <partnerLink name="carRentalPortal" .../>
  </partnerLinks>
  <variables>
    <variable name="getCarRequest" .../>
    <variable name="getCarResponse" .../>
  </variables>
  <pointcutandadvice type="after">
    <pointcut name="accommodation procurement">
      //process[@name="getTravelPrCs"]//sequence[@name="FlightHotel"]
      //invoke[@portType = "HotelPT" and @operation = "findRoom"] or
      //process[@name="getFlightPrCs"]//flow[@name="FlightSearchFlow"]
    </pointcut>
    <advice>
      <invoke partnerLink="CarRentPortal" portType="carRentPT"
        operation="getCar" inputVariable="getCarRequest"
        ouputVariable="getCarResponse" />
      <assign>
        ...
      </assign>
    </advice>
  </pointcutandadvice>
</aspect>

```

Листинг 2.4.4.3.2. Пример описа два аспекта у адаптацији процеса применом АО4ВРЕL, према Charfi&Mezini [Charfi&Mezini, 2004]

У докторској дисертацији [Zafar, 2008] разматрана су постојећа решења динамичких web сервиса, међу којима су и резултати истраживања креирања динамичких веб сервиса применом пословних правила. У овој дисертацији описан је приступ развоју динамички адаптивних (адаптибилност веб сервиса у току извршавања - Run-time) веб сервиса применом Петријевих мрежа високог нивоа.

#### **2.4.4.4. Адаптибилност компоненти и објеката дистрибуираних система**

У раду [Phung-Khac et al, 2007] описан је приступ координацији у оквиру само-адаптивног система дистрибуираних софтверских компоненти. У дистрибуираним системима разликујемо функционалне и комуникационе компоненте (омогућавају повезивање функционалних компоненти) и адаптационе менаџере који омогућавају њихово функционисање уз примену одговарајућих правила. Како би се реализовао комплетан само-адаптивни софтверски систем, потребно је поред постојећих функционалних и комуникационих компоненти, реализовати и модуле: посматрач (observer) – региструје промене у окружењу, доносилац одлуке (decider) – доноси одлуку о извршавању адаптације, оптимизатор – одлучује о избору најпогодније варијанте адаптације.

У раду [Colman&Han, 2005] описан је проблем координације дистрибуираних софтверских компоненти и потребу да постоји посебан подсистем који би био задужен за координацију, контролу и управљање у оквиру интеракције софтверских компоненти за потребе конкретног апликационог домена. У овом раду предложен је приступ, односно радни оквир ROAD (Role-oriented adaptive design), где су елементи којима се координира заправо улоге које извршавају објекти. Улоге имају своја задужења у оквиру целине система и могу бити додељене или уклоњене са објеката. Разликују се функционалне улоге (реализују доменске функције и процесе апликације) и менаџмент улоге (реализују повезивање између функционалних улога путем уговора, баве се креирањем и мониторингом уговора). ROAD уговор представља заправо асоцијациона класа која изражава обавезе које уговорне стране имају једна према другој, а реализоване су кроз вредности атрибута овакве класе, као што је тип уговора, временски оквир извршавања, степен задовољавања услова уговора (који се динамички мења).

У раду [Hudson&King, 1989] описан је објектно-релациони модел података и систем CACTIS, који омогућава динамичку реконфигурацију структуре, односно релација између ентитета која се може извршавати у току извршавања апликације.

У раду [Yoder et al, 2001] описује софтверске архитектуре које су адаптивне у току извршавања на промене захтева корисника. Такве архитектуре се називају рефлективне или мета архитектуре. У овом раду разматране су посебне врсте оваквих архитектура које се називају „Тип-Инстанца патерн“ или „Адаптивне објект-модел архитектуре“. У оквиру рада је презентован софтвер као декстоп апликација која путем интуитивног корисничког интерфејса омогућава унос конкретних пословних ентитета, правила и ограничења, који се аутоматски трансформишу у класе програмског кода.

У раду [David&Ledoux, 2006] разматрана је адаптација софтвера кроз приступ аспект-орјентисаном програмирању. У овом раду представљен је SAFRAN модел (Self-Adaptive FRactal compoNents) који представља основ за креирање самоадаптивних софтверских апликација. Како би се омогућило дефинисање елемената аспект-орјентисане структуре ради постизања софтверске самоадаптивности, у овом раду описан је језик FScript који је првенствено намењен за коришћење у оквиру система SAFRAN.

```
// Changes a cache's replacement strategy.
action select-strategy(cache, strat) = {
  // Gets the cache's client interface to the strategy
  itf := $cache/interface::strategy;
  if (bound($itf)) { // Is it already bound to a server interface?
    // Unbind it and stop the now unused component.
    previous := $itf/binding::*;
    unbind($itf);
    stop($previous/component::*);
  }
  // Binds the cache client interface to the
  // appropriate server interface on $strat.
  bind($itf, $strat/interface::replacement-strategy);
  // Make sure the strategy component is started.
  start($strat);
}
```

Листинг 2.4.4.4.1. Део FScript акција која се може користити за адаптацију компоненте у складу са раније дефинисаном стратегијом, према David&Ledoux [David&Ledoux, 2006]

У раду [David&Ledoux, 2006] адаптациони процес је заснован на реактивности на догађаје у складу са раније дефинисаним реактивним правилима и политици примене правила која наведена правила удружује:

```
when <event> if <condition> do <action>
-----
policy example = {
  rule { when <event1> if <cond1> do <action1> }
  rule { when <event2> if <cond2> do <action2> }
  ...
}
```

Листинг 2.4.4.4.2. Реактивно адаптационо правило и скуп правила дефинисан политиком, према David&Ledoux [David&Ledoux, 2006]

Имплементација политике и правила реализује се помоћу интерфејс класа које омогућавају учитавање скупа правила дефинисаних политиком и примену правила над одређеним компонентама, као што је приказано на следећој слици.

```
public interface AdaptationController {
  void attachFcPolicy(AdaptationPolicy policy);
  void detachFcPolicy(AdaptationPolicy policy);
  AdaptationPolicy[] getFcPolicies();
}
```

Листинг 2.4.4.4.3. Примена политике као скупа адаптационих правила у оквиру интерфејс класе Адаптациони Контролер, према David&Ledoux [David&Ledoux, 2006]

#### 2.4.4.4. Адаптивност софтверских агената

У раду [Fujita et al, 1998] описана је потреба да дистрибуирани информациони системи имају могућности адаптације на различите промене од стране корисникових захтева и потреба за одређеним сервисима, као и оперативних услова окружења (квалитета и врсте расположивих сервиса) у којем дистрибуирани систем функционише. У овом раду описане су могућности адаптације у оквиру адаптивног дистрибуираног система путем: а) подешавања рада (Tuning) где свака компонента дистрибуираног система има оперативне параметре чијом изменом се мењају функционалне карактеристике и понашање наведене компоненте, б) замена компоненте другом, в) реорганизација компоненти. У овом раду предложена је агент-базирана архитектура адаптивбилног дистрибуираног информационог система, где су предложени следећи типови агената: сервисни агент, менаџмент агент, кориснички агент, агент сензора платформе.



У раду [Decker&Sycara, 1997] описана је потреба за реализацијом и коришћењем интелигентних адаптивних софтверских агената. Аутори описују мулти-агентну инфраструктуру дистрибуираног система названог RETSINA (Reusable Task Structure-based Intelligent Network Agents) који омогућава адаптације (на нивоу појединачног агента и целог агентног система) у оквиру отвореног интернет окружења. Систем се састоји из 3 типа агента: а) Интерфејс агент – реализује интеракцију са корисником, прима корисникове спецификације и испоручује резултате; Агент за задатке (Task agent) – помаже корисницима у извршавању задатака припремањем плана решавања проблема; 3) Информациони агент – обезбеђује интелигентни приступ хетерогеном скупу извора података. У овом раду описане су ситуације и технике реализације адаптивности информационих агената.

У раду [Green et al, 1997] описани су резултати истраживања у области софтверских агената. Једна од области се односи на решења адаптивног корисничког интерфејса креирањем модела корисника применом техника машинског учења и применом интелигентних интерфејс агената (који треба да „ухвате“ понашање корисника, како би прилагодили софтвер његовим намерама и преференцијама). У наведеном раду описана је технологија дистрибуираних агената и мулти-агентних система уз опис проблема координације, преговарања и комуникација. Описана су истраживања интелигентних мобилних агената и њихове примене.

С обзиром да су најчешће примене агената управо у прикупљању података са дистрибуираних локација, у раду [Fan&Gauch, 1999] је разматран проблем прикупљања података са више дистрибуираних извора применом адаптивних агената. У наведеном раду описан је развој интелигентног адаптивног алата за претрагу Web садржаја који лоцира потребне податке кориснику, али се и прилагођава променама у динамичком web окружењу.

#### **2.4.4.5. Адаптивност мобилних апликација**

У раду [Kunz&Black, 1999] разматране су архитектуре за адаптивне мобилне апликације. Разматран је клијент-прокси-сервер модел и начин како компоненте овог модела могу да се прилагоде путем миграције прокси сервера. У раду [Efstratiou et al, 2000] разматране су алтернативне архитектуре за подршку адаптивности мобилних апликација. У наведеном раду предложен је приступ издвајању контекста, односно модула средњег слоја који треба да води рачуна о подацима из мониторинга уређаја и да их анализира у контексту примене утицаја на средњи апликациони слој.

У раду [Davies et al, 1996] разматрана је адаптивност мобилних апликација на промене које се дешавају првенствено у квалитету комуникационе инфраструктуре. Потреба за одржавањем нивоа квалитета рада мобилних апликација у условима променљивог квалитета инфраструктурне подршке довела је до потребе развоја адаптивних механизма. У овом раду описан је механизам удаљеног позива процедуре (Remote Call Protocol) који је назван QEX и који је у овом раду описан као технолошко решење обезбеђивања повратне информације мобилној апликацији о томе да ли је дошло до промена квалитета комуникационе инфраструктуре. Наведено решење део је ширег система ANSAware развојне платформе за дистрибуиране системе.

У раду [Davies et al, 1997] критички су разматрени резултати истраживања у области платформи мобилних дистрибуираних система. Анализиран је систем Mobile DCE, MOST платформа и Rover. У наведеном раду предлаже се систем Limbo, који се заснива на Linda платформи, односно заснива се на приступу вишеструких простора n-торки података, који су уређени хијерархијски и експлицитно у односу на атрибуте квалитета сервиса, а реализована је применом система агената који обезбеђују мониторинг квалитета сервиса, креирање нових простора n-торки и размену n-торки података.

У раду [Sato, 2000] описан је радни оквир за конструкцију мобилних агената. Предложени радни оквир омогућује хијерархијско уређивање и груписање мобилних агената. Развијен је систем MobileSpaces коришћењем Java програмског језика. Развијени систем исказује особине екстензибилности (проширивости) и адаптивности, омогућавањем прилагођавања функција у току извршавања програма.

У раду [Augustin et al, 2002] описана је софтверска архитектура адаптивних и дистрибуираних мобилних апликација развијена у оквиру ISAM пројекта. Архитектура система предложена у оквиру ISAM пројекта омогућава адаптацију у три димензије: временска, просторна и персонална. Основна карактеристика адаптације је колаборативно понашање адаптације. Адаптација је омогућена у току учења и у току извршавања мобилне апликације.

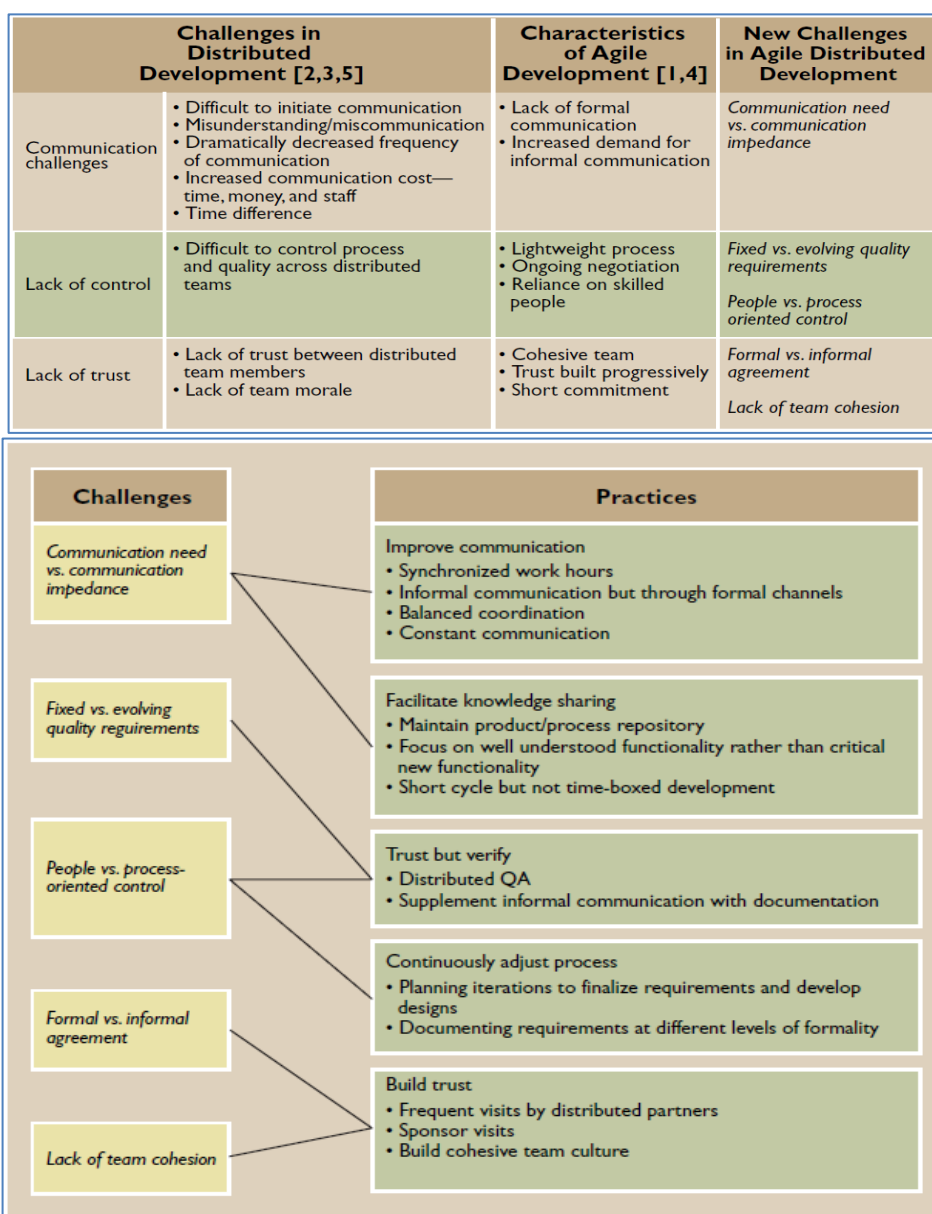
У раду [Arhippainen&Tahti, 2003] описана је емпиријска евалуација корисникових искустава у коришћењу два прототипа адаптивних мобилних апликација. Искуства корисника су снимана као говорни интервјуи пре и после коришћења прототипа апликација и запажања су бележена. Прва адаптивна апликација је контекст-свесна апликација која прилагођава начин рада у односу на профил корисника (гласноћу рада мобилног уређаја и слично), као и да повезује податке из белешки, локације састанака и тренутне локације и реализује аутоматски подсетник. Друга адаптивна апликација се адаптира у односу на локацију корисника и позиционира га на интерактивној мапи.

## 2.5. ИСТРАЖИВАЊА, ТЕХНИЧКА РЕШЕЊА И ИСКУСТВА У КОМБИНОВАНИМ ОБЛАСТИМА

### 2.5.1. Агилни развој софтвера у дистрибуираном окружењу

У раду [Moore&Barnett, 2004] сугерисано је да се дистрибуирани развој и примена агилних метода не може применити код свих софтверских пројеката, тј. код развоја комплексних или стратешки критичних софтвера примена агилних метода у дистрибуираном развоју може бити неодговарајућа. Према [Lee et al, 2006], примена агилних метода у дистрибуираном окружењу посебно мора пажљиво да се прилагоди како би се постигла довољна дисциплина примене.

У раду [Balasubramaniam et al, 2006] разматране су могућности дистрибуираног развоја софтвера са становишта агилности. Анализирани су проблеми и изазови дистрибуираног развоја, агилног развоја и дистрибуираног агилног развоја.



Слика 2.5.1.1. Проблеми агилног дистрибуираног развоја софтвера и најчешћа практична решења [Balasubramaniam et al, 2006]

У раду [Holmstrom et al, 2006] разматране су три димензије удаљености у процесима глобалног развоја софтвера: временска, географска и социокултурна. Ови типови удаљености утичу на контролу, координацију и комуникацију. У наведеном раду предложене су активности које припадају карактеристикама агилног развоја софтвера које могу да помогну у превазилажењу наведених проблема удаљености у дистрибуираном развоју софтвера.

У раду [Shrivastava&Date, 2010] разматране су сличности између проблема дистрибуираног и агилног развоја, као и предности у комбиновању агилног и дистрибуираног развоја софтвера. Предложене су технике за повећање ефикасности дистрибуираних агилних тимова: унапређење комуникације, примењивати личне посете тимова и личну комуникацију, дистрибуција тима уз разматрање разлика временских зона, фокус на тимског ментора (team coaching), подела радних задатака у контексту корисникових прича као целина, а не орјентација на компоненте система, управљање документацијом и транспарентношћу коришћењем посебних алата, коришћење алата за различите аспекте: друштвено умрежавање, алати комуникације, алати софтверске конфигурације (репозиторијуми и алати контроле верзија), алати и базе података за праћење грешака, центри знања, колаборативна развојна окружења итд.

У раду [Khalid et al, 2014] описана је анализа погодности и допринос примене агилних метода развоја мобилног софтвера. У оквиру развоја мобилних апликација, веома је важна подршка променама које се односе на брзе промене технологија мобилних уређаја и одговарајућих софтверских платформи, а такође и промене захтева корисника и стратегије тржишта захтевају подршку променама. У наведеном раду дат је преглед агилних метода које су прилагођене развоју мобилних апликација: Mobile-D, RaPiD7, Hybrid Methodology Design, MASAM (Mobile Application software agile Methodology), SleSS.

### **2.5.2. Агилно управљање софтверским пројектима у дистрибуираном окружењу**

Посебне методологије управљања дистрибуираним софтверским пројектима односе се на примену агилног приступа [Lee&Yong, 2009], у оквиру којег је посебно изучаван SCRUM приступ [Lee&Yong, 2009][Sutherland et al, 2007]. У оквиру рада [Lee&Yong, 2009] описана је студија случаја са практичним решењима примене SCRUM приступа у међународном развоју My Yahoo! софтвера. Изазови/проблеми у развоју односили су се на: комуникације, контролу, поверење. Проблеме су решавали следећим техникама: флексибилност процеса развоја, појачање самосталности тимова, дељени радни простори за размену података, коришћењем свакодневне комуникације уз помоћ комуникацијских алата, обука и менторство.

У оквиру рада [Sutherland et al, 2007] представљена је студија случаја примене SCRUM приступа у међународном развоју софтвера у сарадњи фирме SirsiDyrix и outsourcing firme StarSoft Development Labs, Inc., која ангажује професионалце из Русије, Украјине и источне Европе. У наведеном раду описане су основе продуктивности система дистрибуираних Scrum тимова: примена подршке интегрисаног система за размену резултата рада (Integrated Library System), организација дистрибуираних Scrum тимова уз интеграцију са елементима XP као што је програмирање у пару, дневни састанци, централни репозиторијуми аутоматско генерисање (builds) извршног кода.

У раду [Chan&Chung, 2002] описан је приступ интеграцији пројектног менаџмента (који је првенствено намењен за пројекте који се реализују на једној локацији) и процесног менаџмента (који се односи на процес развоја софтвера који треба да се реализује у дистрибуираном окружењу). Наведени приступ интеграцији резултовао је развојем одговарајућег алата IPPM који омогућава интеграцију процеса развоја софтвера на дистрибуираним локацијама и процеса управљања пројектом. Наведени алат је имплементиран и коришћен од стране неколико развојних тимова у Кини и Хонг Конгу.

У раду [Bose, 2008] извршена анализа 12 студија случаја успешно реализованих пројеката из софтверске индустрије (референцираних у пратећој литератури у раду [Bose, 2008]) применом агилног приступа у дистрибуираном окружењу. Заједничка карактеристика за све пројекте је да се односе на реализацију мањих софтверских система. Анализа резултата наведеног истраживања извршена је у односу на 2 групе карактеристика:

1. Тип агилне методе, начин комуникације, селекција кадрова, радна култура, разлике у временским зонама, поверење, управљање знањем

2. Карактеристике из четири вредности и 12 принципа Agile Manifesto

Неки од резултата наведених анализа:

- Најчешће коришћене агилне методе су: SCRUM, XP и FDD.
- Комуникације: синхроне (видеоконференције, Skype, IRC, On-line chat/IM, позиви путем Blackberry, телефонски позиви), асинхроне (е-маил, SMS, неструктуриране Wiki или TWiki странице за представљање story boards), open source groupware коришћен за управљање развојним простором, ради демонстрирања урађеног дела (sprint) користи се дељење радне површине (desktop sharing), личне посете једном у 3 месеца. Одговорност за комуникацију са удаљеним тимовима на пројектном менаџеру и старијим менаџерима који дискутују о критичним деловима. Најфреквентнији on-line chat, умерено коришћење телефонских позива и видеоконференција.
- Управљање знањем – коришћење CVS репозиторијума као заједничке локације за снимање података о захтевима, дизајну, изворном коду и пратећим документима, мали број докумената, коришћење заједничке базе изворног кода и система за вишелокацијску контролу, web конференције коришћене за виртуално „писање по табли“ (virtual white boarding), коришћење дијаграма напретка („burn down chart“), Bugzilla коришћен као репозиторијум за евиденцију грешака, Twiki коришћен као централни репозиторијум, уместо детаљних описа случајева коришћења, користе се једноставни описи случајева коришћења и корисникове приче („user stories“), статус пројекта се евидентира у бази података, користи се систем обавештавања о проблемима и обрада приоритета, коришћење корисникових прича („user stories“) уместо обичних спецификација захтева корисника, Креирање web страница за сваки пројекат, помоћу којих се може пратити статус и напредак пројекта, различити алати коришћени за праћење задатака и времена, контролу проблема („Star Team“) и управљање кодом, дељење података о тестирању помоћу заједничке командне табле („dashboard“), документи пројекта снимљени у Wiki, коришћење система за контролу изворног кода, систем евиденције заостатака („backlog“), документовање дискусионих токова, захтеви записани у форми случајева коришћења, прототипа корисничког интерфејса и документације о стандардима.

### 2.5.3. Методе и технике контроле квалитета и фактори успеха у дистрибуираном агилном развоју софтвера и пројектном менаџменту

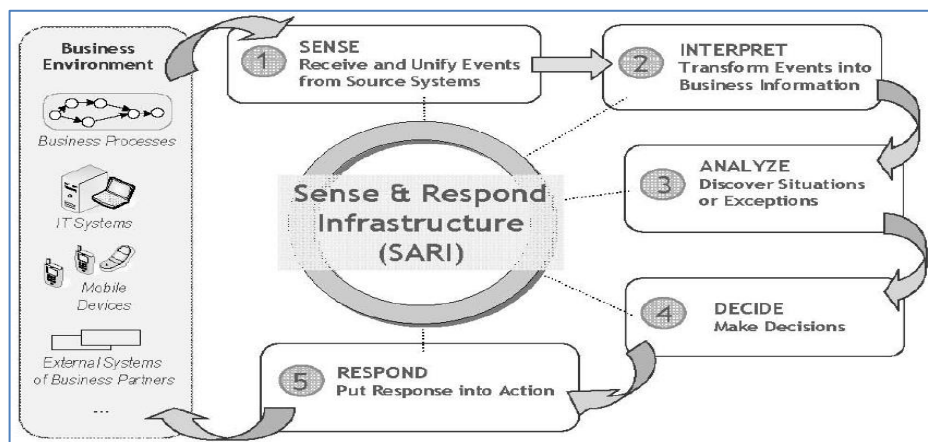
У раду [Huo et al, 2004] анализиран је аспект квалитета у оквиру развоја софтвера применом агилних метода. Упоредене су активности у реализацији провере квалитета у оквиру waterfall модела и агилних модела развоја софтвера. Резултати компарације представљени су следећом табелом.

Табела 2.5.3.1. Компарација контроле квалитета у оквиру Waterfall и Агилних метода према Huo et al [Huo et al, 2004]

Приступ / фаза	Метода контроле квалитета софтвера	Приступ	Метода контроле квалитета софтвера
Waterfall		Агилне методе развоја	
Дефиниција захтева	Ревизија захтева Реализација прототипа Валидација модела	- Дефинисање корисникове приче - Планирање испоруке	Обезбеђење квалитета захтева и артефакта дизајна: ----- Метафора система
Дизајн система и	Упитници		

софтвера	Чек листе Валидација метрикама Сценарио-базирана валидација Провера модела		Анализа архитектуралних елемената издвајања (spike) Одговор корисника
Имплементација и тестирање софтверских модула	Ревизија кода Инспекција кода Пролаз кроз код	Планирање итерације Развој кода Креирање тестова софтверских модула	Обезбеђење квалитета кода: ----- Рефакторисање Програмирање у пару Састанци CRC, поједностављивање проблема
Интеграција и тестирање система	Симулација Симболично извршавање Тестирање интеграције Тестирање прихватљивости	Континуална интеграција Тестирање прихватљивости Мање испоруке	Обезбеђење квалитета кода: ----- 100% пролаз на тестирању модула Повратна информација од корисника
Рад система и одржавање	Контрола захтева за изменама		

У оквиру [Wahyudin & Tjoa, 2007] описан је приступ мониторингу догађаја у оквиру софтверских пројеката отвореног кода (Open Source Software Projects). Увођење аутоматизованог извештавања о прогресу пројекта заснива се на „Sense and Respond“ инфраструктури и приступу.



Слика 2.5.3.1. „Sense and Respond“ инфраструктура [Wahyudin & Tjoa, 2007]

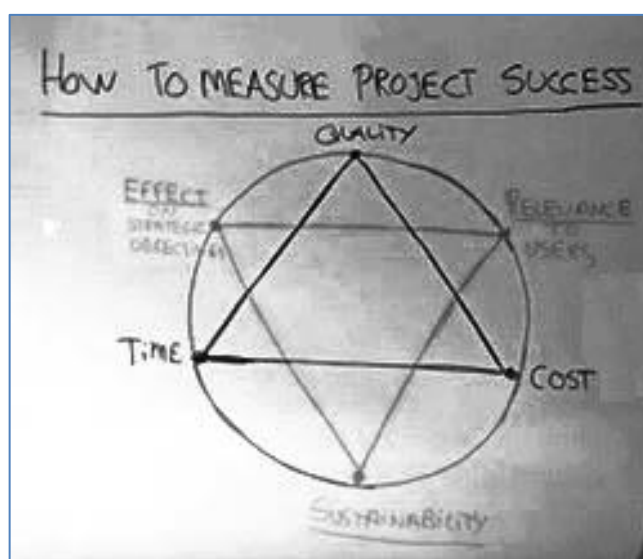
У раду [Bourgault et al, 2002] разматра се проблем мерења перформанси и контроле дистрибуиране реализације пројеката, са специјалним фокусом на метрике перформанси. Мерење је разматрано у контексту 3 димензије: ланац пројектне вредности, Balanced Scorecard, захтеви корисника.

У [Bruegge & Dutoit, 1996] описано је емпиријско истраживање могућности коришћења софтверских метрика над комуникационим артефактима генерисаних и снимљених у оквиру алата за групни рад. Демонстрирано је емпиријски да мерење комуникационих артефакта може да пружи бољи увид у квалитет процеса и производа него метрике које се заснивају на мерењима резултата – функционалних поена, линија кода, речи у документацију).

У раду [Chow & Das-Vu, 2008] анализирани су критични фактори успеха агилних софтверских пројеката, кроз анализу фактора неуспеха и успеха анализом литературе, као и путем анкетања чланова Агилне Алијансе.

#### 2.5.4. Примена **Balanced Scorecard** у ИТ и пројектном менаџменту

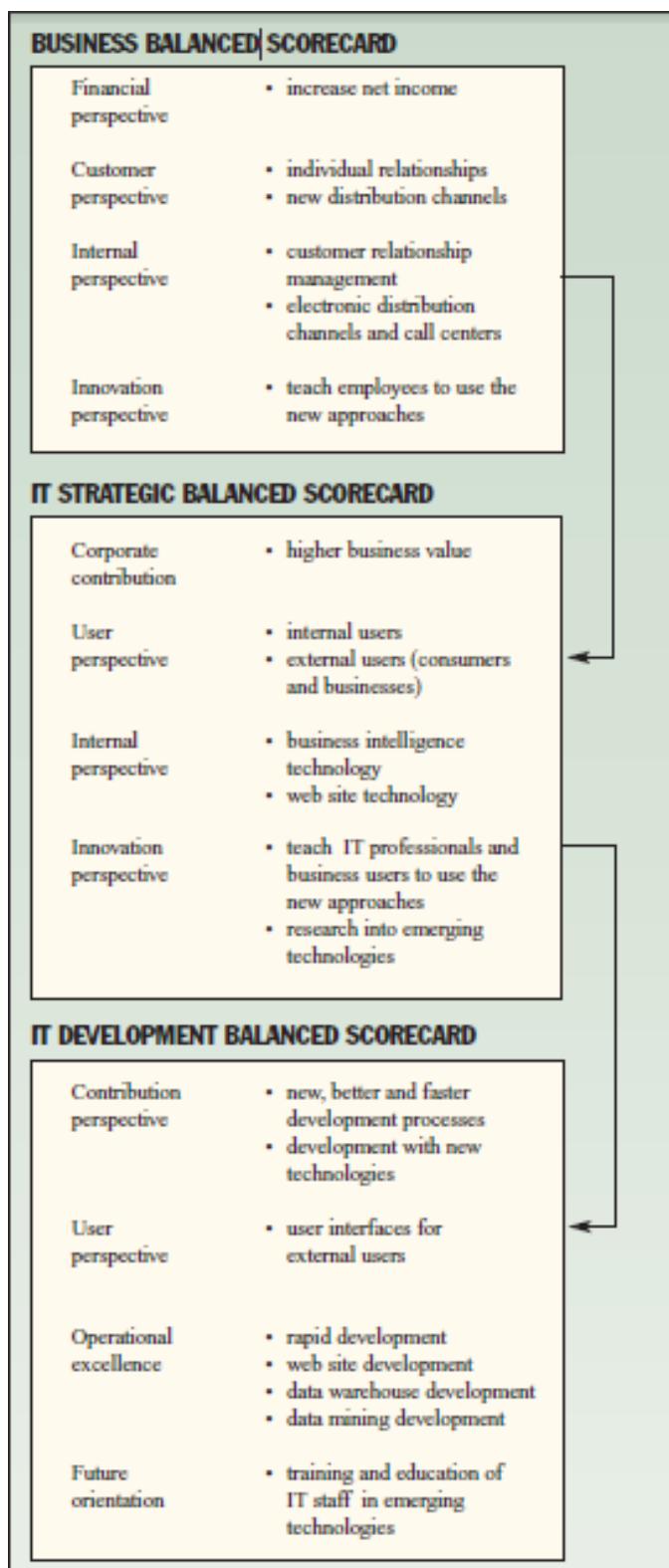
Савремени приступи у пројектном менаџменту у оквиру професионалног окружења разматрају могућности проширења критеријума успеха пројекта, а самим тиме и дефинишу шири скуп метрика којима би се мерио ниво постигнућа у контексту успеха пројекта. У оквиру [projectmanager] представљен је кратак туторијал намењен професионалном особљу у области пројектног менаџмента којим се елаборира потреба за проширењем традиционалног „гвозденог троугла“ управљања пројектима (квалитет, време и трошкови) који се сматра оперативним нивоом одлучивања о пројекту. Проширење представља додатни троугао који се односи на стратегијски аспект одлучивања о успеху пројекта у контексту шире организације у којој се примењује, а односи се на критеријуме: ефекти на пословање (стратешке циљеве), релевантност за кориснике и одрживост (дуготрајност решења). На слици 2.5.4.1. представљен је део презентације са web sajta [www.projectmanager.com](http://www.projectmanager.com) [projectmanager] где је представљена допуна критеријума успеха пројекта, односно мерења успеха пројекта.



Слика 2.5.4.1. Допуна гвозденог троугла управљања пројектима критеријумима који се односе на стратегијске аспекте, према [www.projectmanager.com](http://www.projectmanager.com) [projectmanager]

Balanced Scorecard (BSC) представља савремени приступ стратегијском менаџменту, односно мерењима успеха пословања, који се у истраживањима и професионалној пракси примењује и у контексту управљања ИТ пројектима. У раду [Brock et al, 2003] дат је модел примене Balanced Scorecard у оквиру управљања ИТ пројектним менаџментом.

У раду [ITPM, 2001] и [ITPM, 2002] описана су искуства примене Balanced Scorecard у оквиру ИТ сектора јавне управе. У раду [Grembergen, 2000] описана је примена Balanced Scorecard у оквиру пословног окружења, ИТ стратегијског контекста и ИТ развојног контекста, кроз уобичајене 4 категорије BSC и одговарајуће циљеве који се у различитим областима примене могу дефинисати применом BSC оквира.



Слика 2.5.4.2. Примене Balanced Scorecard и одговарајући циљеви и области мерења у пословном, ИТ стратешком и ИТ развојном контексту, према [Grembergen, 2000]

У оквиру [Keyes, 2005] описана је примена Balanced Scorecard у оквиру усклађивања информационих технологија у оквиру стратегије организације. У оквиру [Keyes, 2011] приказан је приступ примене Balanced Scorecard у оквиру пројектног менаџмента и размотрени су различити аспекти и области примене, а посебно је дат предлог кључних индикатора перформанси који су изведени из захтева Balanced Scorecard као општег оквира а који се односе на мерење успешности пројекта општег типа, као и пројеката колаборативног типа.



Табела 2.5.4.1. Кључни индикатори перформанси стандардног пројекта у односу на Balanced Scorecard методологију, према Keyes [Keyes, 2011]

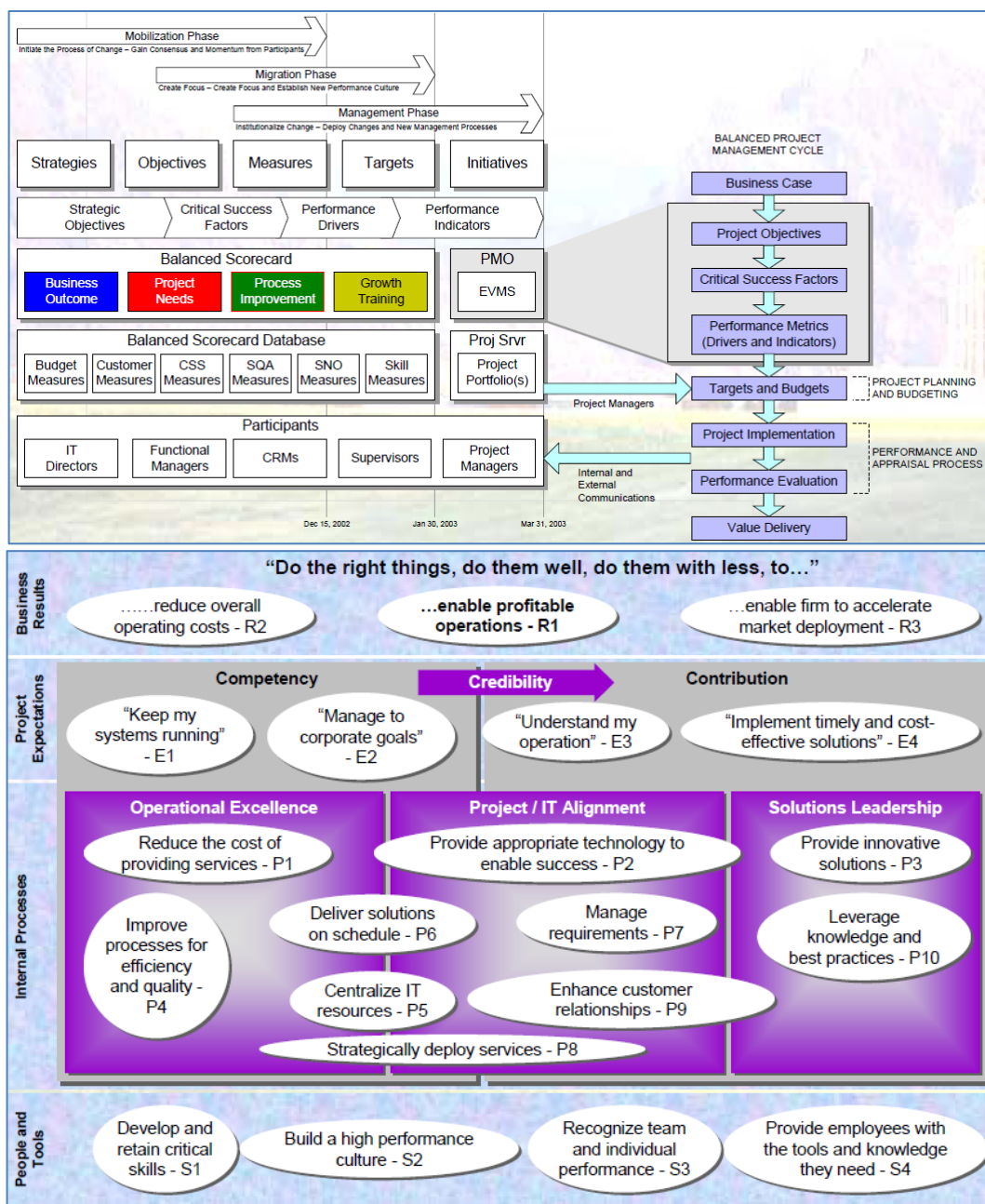
<i>Balanced Scorecard Perspective</i>	<i>Drivers</i>	<i>KPIs</i>
Finances	Project budget Increase of business value Multiproject categorization Project management	Human resources Share of sales Profit margin Savings ROI Expenditure
Customer	Customer satisfaction	Cost overrun Number of customer audits Change management Process stability
Process	Adherence to schedules Innovation enhancement Minimizing risks Optimization of project structure Quality	Adherence to delivery dates Lessons learned Number of patent applications External labor Quality indices Duration of change management Product maturity Percentage of overhead Number of internal audits Project risk analysis
Development	Employee satisfaction Employee qualification enhancement	Rate of employee fluctuation Travel costs Overtime Index of professional experience Continuing education costs

Табела 2.5.4.2. Кључни индикатори перформанси колаборативног пројекта у односу на Balanced Scorecard методологију, према Keyes [Keyes, 2011]

<i>Balanced Scorecard Perspective</i>	<i>Drivers</i>	<i>KPIs</i>
Finances/Project	Project cost Increase of business value Categorization into CP management Project maturity	Product costs Production costs Cost overruns Savings Productivity index Turnover Risk distribution Profit margin Feature stability Product maturity index
Process	Adherence to schedules Innovation enhancement Minimizing risks Adherence to collaboration process Quality	Variance to schedule Changes before and after design freeze Duration until defects removed Number and duration of product changes Number of postprocessing changes Continuous improvement process Project risk analysis Maturity of collaboration process Frequency of product tests Defect frequency Quality indices
Collaboration	Communication Collaboration	Number of team workshops Checklists Degree of communication efficiency Collaborative lessons learned Maturity of collaboration Degree of lessons learned realization
Development	Team satisfaction Team qualification enhancement Trust between team members	Employee fluctuation Project-focused continuing education Employee qualification

У оквиру рада [Grembergen&Amelinckx, 2002] разматране су могућности примене Balanced Scorecard, као приступа мерењу перформанси, у оквиру евалуације е-Бизнис пројеката. Предложен је концепт е-Business balanced scorecard. У оквиру рада [Dobrović et al, 2008] разматрају се могућности примене Balanced Scorecard у оквиру јавног сектора у примени решења е-Government.

У раду [Alleman, 2003] описан је приступ примени Balanced Scorecard у креирању пројектно фокусиране ИТ организације (Слика 2.5.4.3.)



Слика 2.5.4.3. Примена Balanced Scorecard у оквиру пројектно орјентисане организације према [Alleman, 2003]

У раду [Asosheh et al, 2010] је предложен приступ интеграцији метода „Data Envelopment Analysis“ и „Balanced Scorecard“ у евалуацији и селекцији пројеката из области информационих технологија. Применљивост приступа илустрована је примерима пројеката из Министарства науке, истраживања и технологије државе Иран.

### **2.5.5. Информациони системи у пројектном менаџменту**

У раду [Raymond& Bergeron, 2008] приказано је истраживање утицаја примене информационог система пројектног менаџмента (Project Management Information System – PMIS) на успех пројекта. Истраживање је спроведено анкетањем 224 пројектних менаџера и консултаната. Утврђено је да примена PMIS директно не утиче на успех пројекта, већ индиректно, обезбеђујући квалитетне информације пројектним менаџерима. Да би PMIS имао директнији утицај на елементе успеха пројекта (буџет, рокове, реализацију задатака према спецификацији), треба да је високо софистициран и да обезбеђује информације високог квалитета.

У раду [wThomsen] описан је информациони систем за подршку пројектном менаџменту кроз дефиницију пројекта, елемената тима, стандардне и прилагодљиве документе, извештаје и радне токове. Описане су вредности које доприноси и унапређује информациони систем пројектног менаџмента у контексту успеха пројекта и односа пројекта према организационом окружењу. Описани су изазови и потенцијални проблеми реализације информационог система за подршку управљању пројектима који се односе на подршку групном и инкременталном раду, подршку документ менаџменту, подршка интероперабилности апликација итд.

У раду [Siles, 2004] описан је имплементирани информациони систем за подршку управљању пројектима у оквиру организације CARE, назван DME-IS. Основне карактеристике развијеног система: укључује квалитативне и квантитативне податке и податке о повратном одговору у оквиру континуираног вредновања и процена кроз све фазе пројекта, омогућава систематски мониторинг процеса и резултата, омогућава подршку идентификацију потенцијалних проблема и иницира предлоге решења и корективне акције, прикупља, анализира, складишти и доставља информације корисне за доношење одлука, флексибилан је у односу на захтеве, даје подршку принципима транспарентности, поузданости, инклузије и партиципације, где учесници у пројекту имају могућност учешћа и контроле у оквиру доношења одлука.

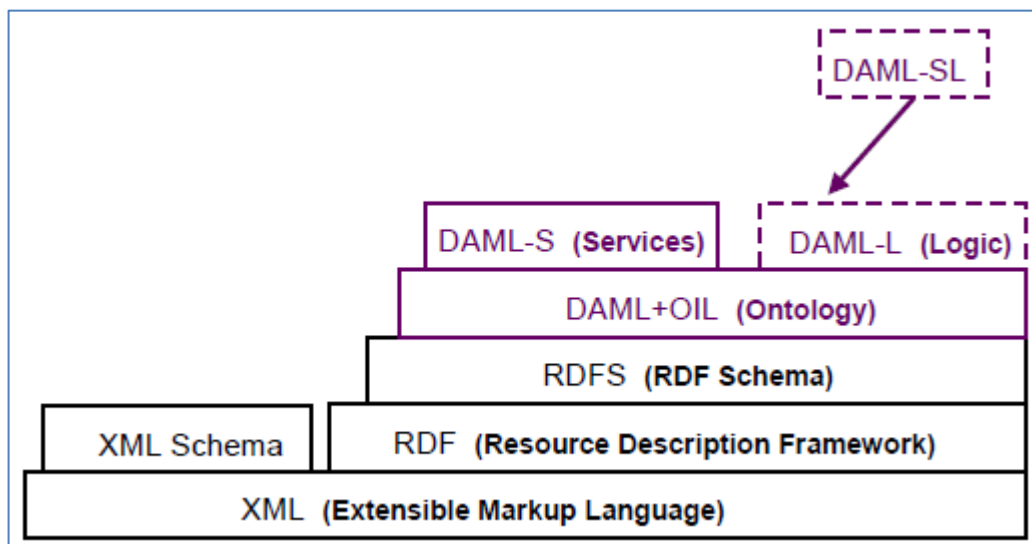
У раду [Alshawi&Ingirige, 2003] разматрају се изазови активности пројектног менаџмента, ограничења постојеће праксе пројектног менаџмента, као и утицај интернета и пословних модела на пројектни менаџмент. У наведеном раду описана је потреба за развојем web базиране подршке пројектном менаџменту и разматране фазе реализације пројекта где web-базирани софтвер за пројектни менаџмент може да пружи пуну подршку. Анализирана су доступна web-базирана решења подршке пројектном менаџменту која су орјентисана на специјализовану област подршке управљању пројектима у грађевинарству.

### **2.5.5. Истраживања динамичких онтолошки заснованих web сервиса и софтверских агената и примена web сервиса у пројектном менаџменту**

Web сервиси се користе у оквиру имплементације софтверске подршке пословним процесима [Leumann et al, 2002]. С обзиром на динамичност самих пословних процеса, односно измене начина извршавања и тока активности, као и повезаности више пословних ентитета, потребно је да web сервиси који реализују елементе подршке пословним процесима буду адаптивни на промене и да се могу динамички позивати у складу са потребама и квалитетом услуге коју пружају. У оквиру истраживања која се односе на web сервисе, једна група истраживања односи се на могућности динамичке селекције и повезивања web сервиса. У истраживањима и реализованим прототипским решењима, најчешће су захтеви за динамичким и семантички заснованим повезивањем web сервиса реализовани интеграцијом web сервиса, софтверских агената и онтологија. [Greenwood&Calisti, 2004]

У раду [Narayanan&McIlraith, 2002] описани су приступи семантичким web сервисима, кроз историјски развој технологија. Описани су основни приступи, као што је DAML-S,

односно DARPA Agent Markup Language for Services, креиран под покровитељством DARPA организације 2001 године. У раду [Narayanan&McIlraith, 2002] предложена је интеграција семантичких web сервиса и предикатског рачуна првог реда у циљу процесирања логике (коришћење Ситуационог рачуна – језика предикатског рачуна првог реда који треба да омогући резонување о акцијама и променама), а такође је предложена дистрибуирана операционална семантика представљена Петријевим мрежама (повезивањем ситуационог рачуна са формализмима Петријевих мрежа).



Слика 2.5.5.1. Развој семантичких web сервиса, према [Narayanan&McIlraith, 2002]

У раду [Sycara et al, 2003] је у циљу омогућавања аутоматизованог откривања, интеракције и композиције семантичких web сервиса предложена употреба онтологија којима се описује шта web сервис ради и како се може користити. Пример такве онтологије описан је кроз реализацију DAML-S прототипа. У раду [Wu et al, 2003] описан је софтвер SHOP2 који омогућава аутоматску композицију web сервиса уз подршку хијерархијском мрежном планирању рада, који се заснива на повезаности са процесним моделом и семантиком у оквиру ситуационог рачуна.

У раду [Baader et al, 2005] описан је приступ коришћењу дескриптивне логике као подршке резонувању, која се односи на web сервисе. Језик дескриптивне логике који је коришћен у овом раду је ALCQIO. Главни допринос овог рада је у анализи како избор дескриптивног језика утиче на комплексност задатака резонувања које се односи на web сервисе. Основни проблеми резонувања које се односи на web сервисе представљене језиком дескриптивне логике односе се на извршивост (испуњеност предуслова за извршавање web сервиса) и пројекцију (одржавање услова окружења пре и после извршавања web сервиса). У раду [Salaun et al, 2004] размотрене су могућности интеграције процесне алгебре и коришћења web сервиса применом BPEL језика. Главни мотив коришћења апстрактних језика је у коришћењу аутоматизованих алата који би омогућили аутоматско резонување над web сервисима у циљу провере да ли систем ради у складу са захтевима и да ли ради коректно. Посебно су од интереса за анализу композиције постојећих сервиса и упоређивање да ли таква композиција одговара апстрактној форми представљања спецификације потребног функционисања. У овом раду разматрана су два основна проблема са становишта резонувања над web сервисима: проблем кореографије (компатибилност web сервиса, односно у могућности да web сервиси могу да се повежу у циљу заједничког рада), проблем оркестрације, односно проблем решавања комплексних задатака коришћењем, односно креирањем новог сервиса који се зове оркестратор, а циљ му је да омогући размену порука између других web сервиса.

У раду [Maxmilien&Singh, 2004] разматран је проблем динамичке селекције и повезивања web сервиса базираних на подацима о квалитету сервиса који неки web

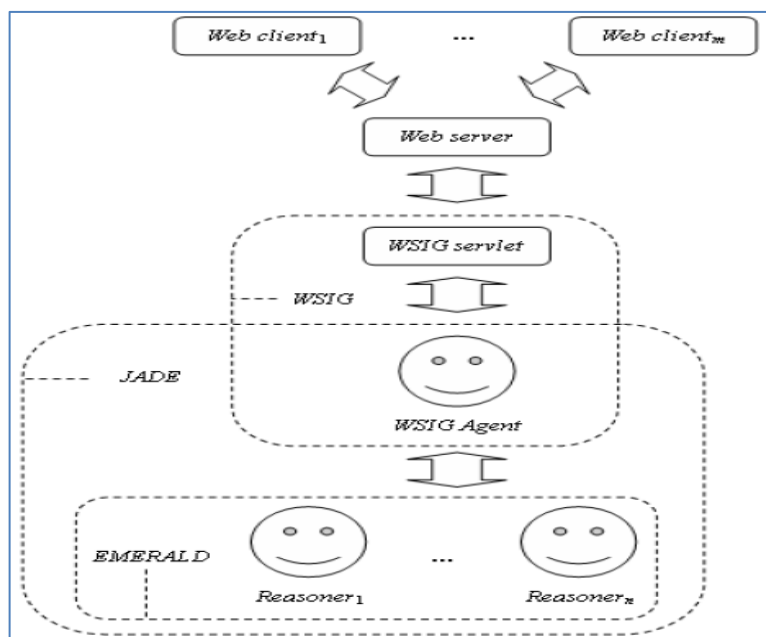
сервис омогућава. За потребе очитавања података о квалитету сервиса и подршке динамичкој аутономној конфигурацији и реконфигурацији, у овом раду се предлаже коришћење софтверских агената. Такође, предложено је коришћење онтологија које треба да омогуће опис захтеваног знања о елементима квалитета услуга који се траже од стране web сервиса. Политика примене web сервиса укључује податке о web сервисима, онтологији, политици квалитета изражена кроз ограничења (листинг 2.5.5.1).

```

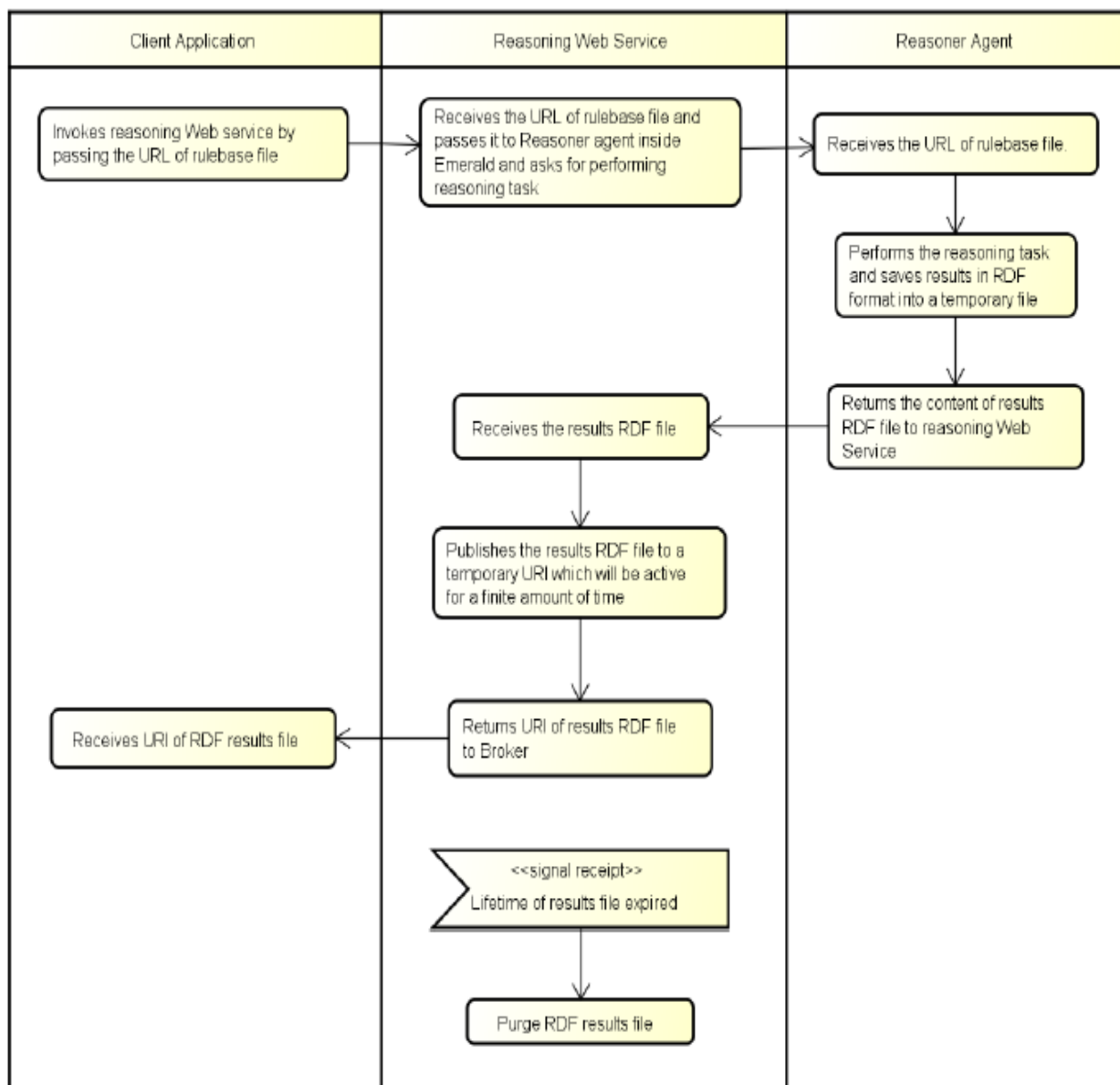
<WsPolicy name=
  "insuranceServiceProvider1" ...>
  <Services><Service name="ispl"
    interfaceUri
    ="http://.../insurance?wsdl"/>
  ... </Services>
  <Ontologies>
  <Ontology name="InsuranceOnt"
    uri="http://.../insurance-
    qos.owl"/>
  ... </Ontologies>
  <QoSPolicy ontologyName="InsuranceOnt"
    serviceName="ispl" methods="*">
  <QoS name="PremiumPrice"><qValue>
  <unit>USD</unit><min>2000</min>
  <max>6000</max><typical>4000</typical>
  </qValue></QoS>
  <QoS name="Deduction"><qValue>
  <unit>USD</unit><min>50</min>
  <max>1000</max><typical>500</typical>
  </qValue></QoS>
  </QoSPolicy>
</WsPolicy>
    
```

Листинг 2.5.5.1. Политика примене web сервиса уз примену онтологија, агената и политике квалитета приказом ограничења, према [Maxmilien&Singh, 2004]

У раду [Badica et al, 2014] описано је проширење могућности система агент-базираног резонувања EMERALD како би били доступни за покретање Web сервиса. Коришћењем система EMERALD омогућено је резонување у оквиру коришћења web сервиса, а применом JADE (Java Agent Development Environment) омогућено је креирање и рад са софтверским агентима (слика 2.5.5.2 и 2.5.5.3).



Слика 2.5.5.2. Архитектура система за резонување применом агената и web сервиса према Badica et al [Badica et al, 2014]



Слика 2.5.5.3. Улоге и активности у оквиру система резоновања путем web сервиса и агената према Badica et al [Badica et al, 2014]

Истраживања примене web сервиса орјентисана су најчешће на технолошке аспекте и могућности унапређења, али нису довољно заступљена у описима конкретних примена у реализацији конкретних проблемских области. У истраживањима која се односе на web сервиса дати су илустративни једноставни примери примене у реалним окружењима, али радови који би описали примену web сервиса у пројектном менаџменту нису заступљени.

Ипак, примена web сервиса у контексту подршке пројектном менаџменту реализована је у оквиру професионалних решења. У наставку је дат приказ анализе неколико професионалних решења наведеног типа у контексту анализе примене web сервиса. Резултати анализе су представљени у табели 2.5.5.1.

Табела 2.5.5.1. Анализа примене web сервиса у оквиру професионалних апликација подршке пројектном менаџменту

<b>НАЗИВ АЛАТА</b>	<b>ОПИС ПРИМЕНЕ WEB СЕРВИСА</b>
iPlanWare	Интеграција са другим апликацијама (MS Project)
ProjectInsight	Реализација основних функција уноса и издвајања података о базичним подацима о пројекту, задацима, трошковима итд. Интеграција са другим апликацијама (.NET)
Projektron	Читање и упис података у базу података која се односи на елементе управљања пројектима
ProjectOpen	Web сервис базиран на REST API технологији Омогућавају да друге апликације могу приступати подацима и објектима Project Open система, ради читања и уписа података.
ProjectManager	Интеграција апликација није омогућена путем web сервиса, већ коришћењем система Zapier за општу интероперабилност различитих апликација

Очигледно, у оквиру постојећих професионалних решења софтверских алата за подршку управљању пројектима, web сервис се користе за реализацију интероперабилности других апликација према развијеном алату, омогућавајући интеграцију података из наведеног алата са подацима других софтверских алата.



## **3. МОДЕЛ ПРЕДЛОЖЕНОГ РЕШЕЊА**

### **3.1. ОПИС ПРОБЛЕМА**

#### **3.1.1. Опис посла**

Опис посла представља текст којим се описује пословни домен и ток пословних процеса за који је потребно реализовати подршку информационог система. У овом одељку описан је процес развоја софтвера као пословни процес од интереса. Процес развоја софтвера зависи од усвојене методологије развоја софтвера и типа софтвера. У наставку дат је општи приказ активности у развоју софтвера, уз поједине елементе који се односе на развој специфичног софтвера у области информационог система организације.

Процес развоја новог софтверског решења започиње дефинисањем потребе за неким софтвером у оквиру организације клијента. Клијент контактира софтверску фирму и у интеракцији са одговарајућим представницима софтверске фирме дефинишу се захтеви. Након договора о почетку посла, представници софтверске фирме анализирају постојећу документацију и прописе из наведене области, како би се упознали са семантиком проблема. У договору са клијентом, дефинише се снимак стања којим се утврђују постојећа решења код клијента и дефинишу проблеми. Након тога детаљно се описује спецификација захтева корисника. Развојни тим софтверске фирме започиње са дизајном апликације кроз дефинисање архитектуре и модела (модел пословних процеса, модел софтверских функција, модел података). Започиње израда базе података и дизајн корисничког интерфејса. Прототип апликације се у сарадњи са корисником детаљније анализира и на основу смерница за унапређење, креира се апликација. У оквиру више итерација у сарадњи са корисником дефинишу се прецизнији захтеви и унапређује апликација до нивоа покривања свих потребних функција софтвера. Следи тестирање са реалним подацима, у сарадњи са корисником. Коначно, након испоруке, следи обука и пробни рад, где се у реалним условима софтвер тестира и дорађује.

#### **3.1.2. Снимак стања**

Снимак стања се односи на приказ постојећег стања информационог система у конкретној организацији, специфицирајући недостатке и проблеме. У овом одељку циљне организације су факултети (пример: Технички факултет „Михајло Пупин“ у Зрењанину) и софтверске фирме.

На Техничком факултету „Михајло Пупин“ у Зрењанину не постоји информациони систем подршке развоју софтвера, посебно софтвера у дистрибуираном окружењу. Овакав информациони систем је потребан да би омогућио подршку праћењу успеха рада студената на семинарским радовима у току израде. Основ за овај податак о стању је непосредно искуство кандидата у раду у наведеној институцији.

У софтверским фирмама користе се различити софтвери за подршку тимском раду и праћење реализације софтверских пројеката. Поједине фирме развијају своје системе за праћење реализације софтверских пројеката, али се најчешће користе бесплатна или комерцијална решења. Основ за ово тврђење о стању је у резултатима анкете која је спроведена са запосленима у ИТ сектору, чији су резултати приказани у одељку 5.3.

#### **3.1.3. Спецификација захтева корисника**

Шира дефиниција спецификације захтева корисника настала је на основу анализе проблема реализованих пројеката (одељак 1.2.2), анализе постојећих истраживања, технолошких решења и искустава из ИТ сектора (одељак 2).

Потребно је реализовати информациони систем који ће омогућити:

---

- Праћење реализације софтвера у складу са стандардном методологијом управљања пројектима и развоја софтвера
- Праћење реализације софтвера у свим фазама развоја софтвера
- Праћење успешности реализације софтвера базирано на софтверским метрикама и пословним правилима
- Праћење реализације софтвера који се реализације у дистрибуираном окружењу
- Визуализацију података у праћењу успешности реализације софтверског пројекта
- Адаптивност таквог информационог система у односу на типове софтвера, методологије рада и технолошка окружења.

### 3.1.3.1. Функционално-технолошки теоријски модел софтверске подршке адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката

У наставку је приказан функционално-технолошки теоријски модел софтверске подршке адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката. Модел је заснован на општим теоријским областима и технолошким областима имплементације описаним у оквиру методолошког оквира истраживања, уводног теоријског разматрања и анализе постојећих истраживања.

Табела 3.1.3.1. Функционално-технолошки теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката

<b>ОПШТЕ ТЕОРИЈСКЕ ОБЛАСТИ</b>	<b>ТЕХНОЛОШКЕ ОБЛАСТИ</b>
<b>Развој информационог система</b>	<b>Дистрибуирани информациони системи</b>
Функција – прикупљање података	Дистрибуција података (базе података, XML)
Функција – верификација података	Дистрибуција апликативне логике – софтверских компоненти
Функција – чување података	Дистрибуција корисничког интерфејса
Размена података	<b>Адаптивни и Адаптибилни системи</b>
Уређивање, обрада података	Реаговање на промене
Презентовање података	Промене у окружењу
<b>Методологија развоја софтвера</b>	Промене интерних стања
Захтеви	Промене изменом структуре
Дизајн	Промене изменом понашања
Конструкција	Детекција промене
Тестирање	Одлучивање о акцији
Интеграција	Акција
Квалитет – верификација у односу на захтеве корисника и стандарде	Самоконфигурирање уз одлучивање о промени понашања и/или структуре
Економски аспект – трошкови, добити (Earned Value)	Самоизлечење
<b>Методологија пројектног менаџмента – РМВОК</b>	Самооптимизација
Интеграција	Самозаштита
Обухват	Свесност сопственог стања
Време	Свесност окружења - контекста
Трошкови	<b>Системи за управљање пословним правилима, аутоматско резонување и онтологије</b>
Квалитет	Рад са екстерно чуваним правилима
Људски ресурси	Аутоматска примена правила
Комуникације	Примена онтологија
Ризици	
Набавке	

Заинтересоване стране	
Процес – дефинисање циља	
Процес – планирање	
Процес – организација	
Процес – Контрола	
<b>Успех пројекта</b>	
Успех пројектног менаџмента – процеса, ресурса, трошкова	
Успех производа – квалитет производа	
Мерење података	
Евалуација података	
Сумирање података	
Визуализација сумираних података	
<b>Управљање перформансама пословних процеса</b>	
Дефинисање циља, стратегије	
Планирање	
Анализа, мониторинг	
Предузимање корективних акција	

### 3.1.4. Анализа документације

«Поређењем информација које пружа систем извештавања о стварном стању у реализацији пројекта и планираних величина, добијају се основни елементи за управљање реализацијом пројекта, односно за преузимање одговарајућих управљачких акција које омогућавају да се стварна реализација на терену одвија у складу са претходно утврђеним плановима.» [Jovanovic, 2006]

Свака методологија управљања пројектима дефинише и развија свој систем документације којима се врши структурирано евидентирање и приказ свих релевантних података у току процеса управљања пројектом, односно у току животног циклуса пројекта. На националном (за поједине земље), а и међународном нивоу развијени су стандарди којима је прописана методологија и документација у оквиру управљања пројектима.

«Извештаји се углавном ослањају на рачунарску обраду и скуп извештаја који даје коришћени програмски пакет за управљање пројектом. Они су посебно формирани и прилагођени за коришћење на одговарајућем организационом нивоу, зависно од потребе и компетенција одговарајућих органа и појединаца у процесу управљања реализацијом пројекта. Стандардни програмски пакети за управљање пројектом најчешће нуде следеће извештаје:

- термински извештаји,
- гантограми, извештај о кључним догађајима,
- извештај о стању реализације,
- извештај о напредовању радова,
- извештај о трошковима итд.» [Jovanovic, 2006]

У оквиру PRINCE 2 методологије управљања пројектима, користе се следећи најважнији општи образци докумената [Yeong, 2007]:

- |                                    |                                 |
|------------------------------------|---------------------------------|
| 1. Acceptance Criteria             | 15. Product Description         |
| 2. Business Case                   | 16. Project Approach            |
| 3. Checkpoint Report               | 17. Project Brief               |
| 4. Communication Plan              | 18. Project Initiation Document |
| 5. End Project Report              | 19. Project Issue               |
| 6. End Stage Report                | 20. Project Mandate             |
| 7. Exception Report                | 21. Project Plan                |
| 8. Follow-on Action Recommendation | 22. Project Quality Plan        |
| 9. Highlight Report                | 23. Quality Log                 |
| 10. Issue Log                      | 24. Request for Change Form     |
| 11. Lessons Learned Report         | 25. Risk Log                    |

12. Off Specification Form  
13. Post Project Review  
14. Product Checklist

26. Stage Plan  
27. Work Package

Према верзији ПРИНЦЕ2:2009 из 2009. године, производи/резултати (који се односе на документацију) ове методологије су представљени у Табели 3.1.4.1. и 3.1.4.2.

Табела 3.1.4.1. Основни резултати (производи) примене PRINCE2: 2009 [PRINCE 2 templates]

Prince2: 2009 Product	Old product or contributor
<b>BASELINE</b>	
Project Brief	Project Brief, Project Approach
Project Initiation Documentation	Project Initiation Document
Business Case	Business Case
Communication Management Strategy	Communications Plan
Quality Management Strategy	Project Quality Plan
Risk Management Strategy	None
Configuration Management Strategy	Configuration Management Plan
Project Product Description	Customer Quality Expectations, Acceptance Criteria
Work Package	Work Package
Benefits Review Plan	Post Project Review Plan
Plan (Project, Stage and Team)	Project, Stage, Exception Plan
Product Description	Product Description

Табела 3.1.4.2. Записи и извештаји према методологији PRINCE2: 2009 [PRINCE 2 templates]

Prince2: 2009 Product	Old product or contributor
<b>RECORDS</b>	
Daily Log	Daily Log
Lessons Log	Lessons Learned Log
Issue Register	Issue Log
Risk Register	Risk Log
Quality Register	Quality Log
Configuration Records	Configuration Item Records
<b>REPORTS</b>	
Highlight Report	Highlight Report
Checkpoint Report	Checkpoint Report
Exception Report	Exception Report
Issue Report	Project Issue/Request for Change/Off Spec
Lessons Report	Lessons Learned Report
Product Status Account	Product Status Account
End Stage Report	End Stage Report
End Project Report	End Project Report Follow-on Action Recommendation

На основу ПРИНЦЕ2 методологије, многе организације су развиле обрасце којима омогућују примену структурираног приступа управљању докумената у пројектима. Поједини документи ПРИНЦЕ2 су дати као обрасци/темплејти и детаљније описани у [PRINCE 2 templates]. У наставку је дат кратак опис појединих врста претходно наведених докумената.

- *Business case* (пословни проблем) - Документује оправданост пројекта, базирано на трошковима развоја и претпостављене користи, добити (бенефит) које треба да се постигну. Обезбеђује иницијални приказ различитих опција које су на располагању, усмерава процес доношења одлуке и користи се континуирано да би се усклађивао напредак пројекта са циљевима и задацима пословања организације, као и постављеним перформансама. Користи се да би се проверавала успешност пројекта и омогућила упућеност менаџмента у ток реализације пројекта.
- *Project Brief* (Пројектни приказ) - Обезбеђује опис шта ће бити рађено у оквиру пројекта. Представља основ за Документ иницирања пројекта (ПИД). Даје усмерење и обухват пројекта и чини уговор између пројект менаџмент тима и пројектног одбора. Било које измене у пројектном приказу потребно је реферисати пројектном одбору.
- *Project Initiation Document* (Документ иницирања пројекта) - Овај документ је проширење Пројектног приказа. Треба да укључује детаље о пројектном тиму, регистар ризика и детаље о томе како ће пројекат бити контролисан и како ће се њиме управљати. ПИД документ окупља све кључне

информације потребне да би пројекат започео И омогућује да сви који су укључени у пројекат буду о свим релевантним информацијама обавештени.

- *Quality Log* (Запис о квалитету) - Оно што се испоручује (деливераблес) пројектом треба да одговара сврси. У контексту пројекта, уграђивање квалитета не значи увек високог квалитета, већ најчешће оно што је испоручено или произведено треба да ради оно што је предвиђено – планирано да ради. Процес развоја може да скрива многе грешке, али битне су карактеристике крајњег производа. Запис о квалитету приказује које излазне резултате (испоруке – деливераблес) ће бити испитиване, према којим критеријумима И ко ће извршити контролу квалитета према тим критеријумима.
- *Progress report* (Извештај о напретку пројекта) - У циљу доношења одлука на основу информација и спровођења контроле над пројектом, пројектни менаџер треба да зна шта се десило и да буде у могућности да пореди то у односу на то шта је требало да се деси. Од виталног је значаја да постоји стабилан, ажуран и редован ток информација са радних токова до пројектног менаџера. Овај извештај то омогућује.
- *Highlight report* (Извештај о кључним елементима) - Да би се обезбедили сумарни подаци пројектном одбору везани за статус поједине фазе у пројекту у интервалима како је дефинисано ПИД документом, користи се овај извештај. Овај извештај обезбеђује детаље о тренутном статусу временског плана рада (schedule), статусу буџета, сумарни приказ проблема и ризика, статус завршеног посла и посла који треба да буде завршен током следећег периода времена.
- *Benefits realisation plan* (План реализације користи-добити) - Овај план се користи да би се пратила реализација користи кроз пројекат или програм. Многи од предвиђених користи неће почети да се материјализују све до тренутка након завршетка пројекта. Зато је битно да се власништво над користима од пројекта, односно план реализације користи од пројекта одржава након и изван завршетка пројекта све до комплетне реализације користи. Пројекат или програм треба да омогући време за постимплементациону анализу менаџмента и реализације користи и компарацију са пословним проблемом.
- *Project Quality plan* (План квалитета пројекта) - План квалитета пројекта је најчешће део ПИД документа. Сврха овог документа је да дефинише технике И стандарде квалитета који треба да буду примењени на различите одговорности за постизање захтеваног нивоа квалитета, у току пројекта.
- *Risk Log* (Листа ризика или Регистар ризика)- Управљање ризицима је кључ успешне испоруке. Ризик је било који документ који изазива да пројекат не буде успешно окончан. Да би се управљало ризицима, потребно је да буду идентификовани, вредновани у смислу могућности да наступе и вредновани у смислу утицаја који могу да имају на пројекат. Креирање и комплетирање регистра ризика може да помогне идентификацији, управљању или елиминацији ризика. Листа ризика се користи да би се излистали идентификовани ризици који се односе на пројекат или програм и резултати њихове анализе и евалуације. Информација о статусу ризика је такође укључена. Овај документ се може називати и Регистар ризика. Ови детаљи могу бити коришћени за праћење и мониторинг успешности менаџмента као део активности испоруке захтеваних, предвиђених користи.
- *Stakeholder Map* (Листа заинтересованих страна)- Овај документ служи за приказ свих заинтересованих страна интерно или екстерно у односу на пројекат или програм; може се односити на појединце или групе у оквиру или изван пословног система. Сврха овог документа је да обезбеди да се интереси свих интересних група (заинтересованих страна, стакхолдерс) су обухваћени програмом, покривени циљевима пројекта, а такође обезбеђује њихову информисаност о пројекту или програму и повратну информацију о задовољству (феедбацк).
- *Issues log* (Листа проблема) - У ПРИНЦЕ2 пројектни проблеми (пројект иссуес) су било које ситуације које изазивају пажњу на пројекту и захтевају одговор. Сваки проблем треба да добије јединствени идентификациони број и да буде вреднован у смислу утицаја који изазива на производ, утицаја на рад, трошкове и ризике, пројектни план и пословни проблем. Проблеми могу имати позитивни и негативни утицај на пројекат и понекад је потребно је да се пројектни одбор обавести о њима. Одржавање листе проблема током пројекта даје могућност да се управља свим проблемима који се јављају у току пројекта.
- *Change request log* (Листа захтева за променама) - Често се јављају промене у пројекту и препоручљиво је да се имплементира систем који прати све потенцијалне промене и обезбеђује да, тамо где је неопходно, вреднује ефекте које може имати на испоруку. Праћење процеса значи да о променама и њиховом утицају треба да буде обавештен, на разумљив начин, пројектни спонзор, заинтересоване стране и други чланови пројектног тима.
- *End of Project Report* (извештај о завршетку пројекта) - Извештај о завршетку пројекта саставља пројектни менаџер и доставља пројектном власнику са детаљима о томе у којој мери је пројекат испунио захтеве и извршавао се у складу са ПИД: Такође треба да садржи информације о резултатима поређења са оригиналним планираним трошковима, временским распоредом активности и дозвољеним ризицима, са пословним проблемом, као и коначну верзију пројектног плана.

## 3.2. МОДЕЛИ У ПРОЦЕСУ РАЗВОЈА ПРЕДЛОЖЕНОГ СИСТЕМА

У наставку ће бити приказани уобичајени модели који се користе у процесу моделовања пословних процеса, спецификације и дизајна информационог система, делом представљени стандардним језиком моделовања Unified Modeling Language (UML), односно одговарајућим дијаграмима. Међународна организација ISO описала је UML стандардни језик моделовања у оквиру ISO/IEC 19505:2012 Information Technology—Object Management Group Unified Modeling Language (OMG UML).

На основу описа проблема, снимка стања и спецификације захтева корисника дефинише се модел пословних процеса (Модел пословних процеса или Дијаграм тока података - Business Process Model или Data Flow Diagram), који се затим преводи у модел софтверских функција (UML-ов Дијаграм случајева коришћења - USE CASE diagram) и модел података (Концептуални модел података - Conceptual Data Model). Коначно, дефинише се и архитектура система (UML-ови дијаграми: Дијаграм компоненти - Component Diagram и Дијаграм размештаја - Deployment Diagram). Приказани дијаграми су креирани у оквиру CASE (Computer Aided Software Engineering) алата Power Designer.

### 3.2.1. Теоријско-истраживачка основа за креирање предложених модела развоја система

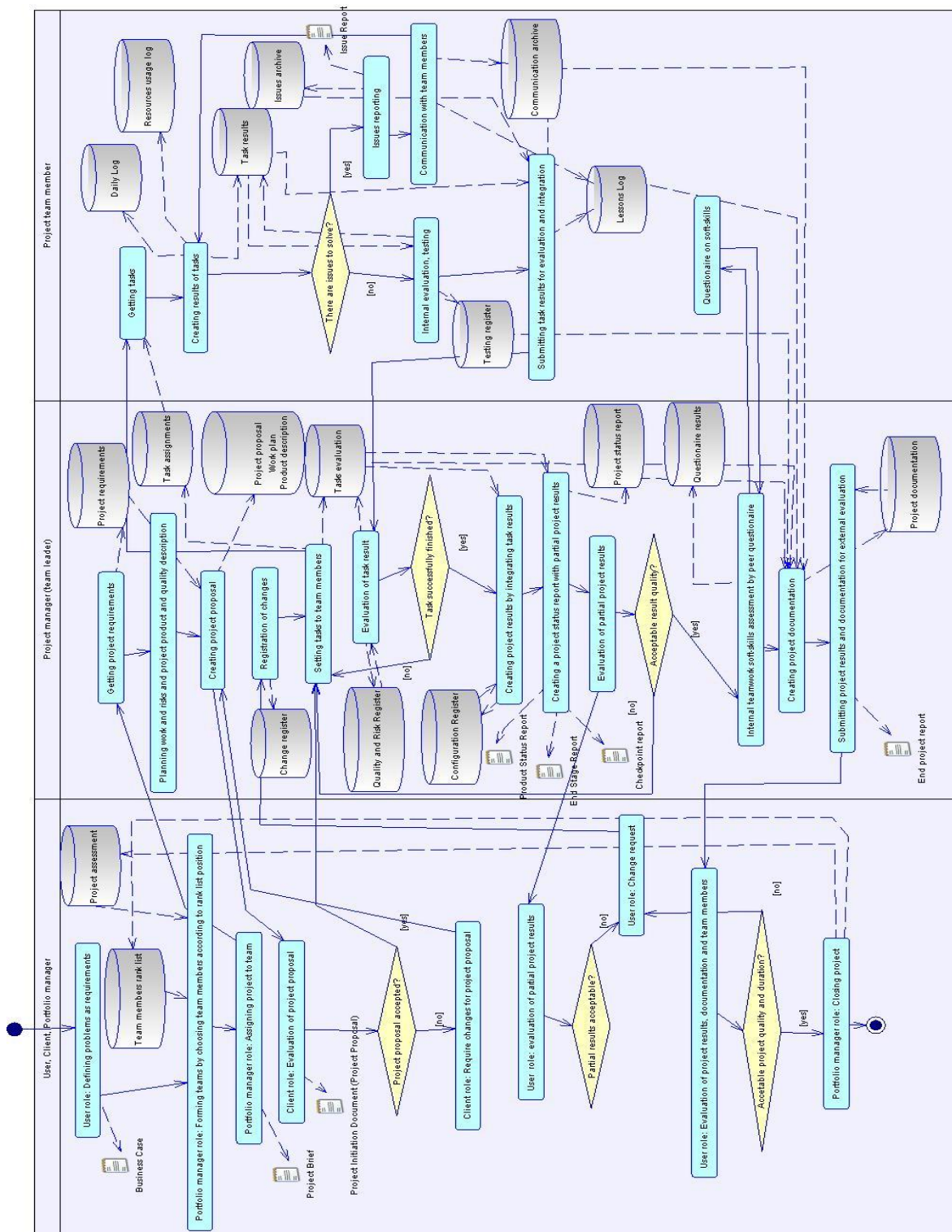
Циљ креирања модела предложеног система и каснија имплементација није у обухвату функција које су већ реализоване у постојећим професионалним технолошким решењима, већ научни и практични допринос развојем нових функционалности, заснованих на теоријским основама и интеграцији постојећих метода и развоју и примени нових метода и приступа. Модел предложеног система базиран је првенствено на претходним теоријским основама, односно резултатима претходних истраживања у овој области, изражене првенствено кроз одговарајуће методологије. У креирању модела предложеног система основе представљају методологије и приступи представљени табелом 3.2.1.1., где је поред категорије и конкретног типа модела наведена и семантичка област примене која је заступљена у моделу и референца на објављени рад, где је презентован предложени приступ интеграцији модела или нове методе или методологије.

Табела 3.2.1.1. Теоријско-истраживачке основе за креирање предложених модела развоја система

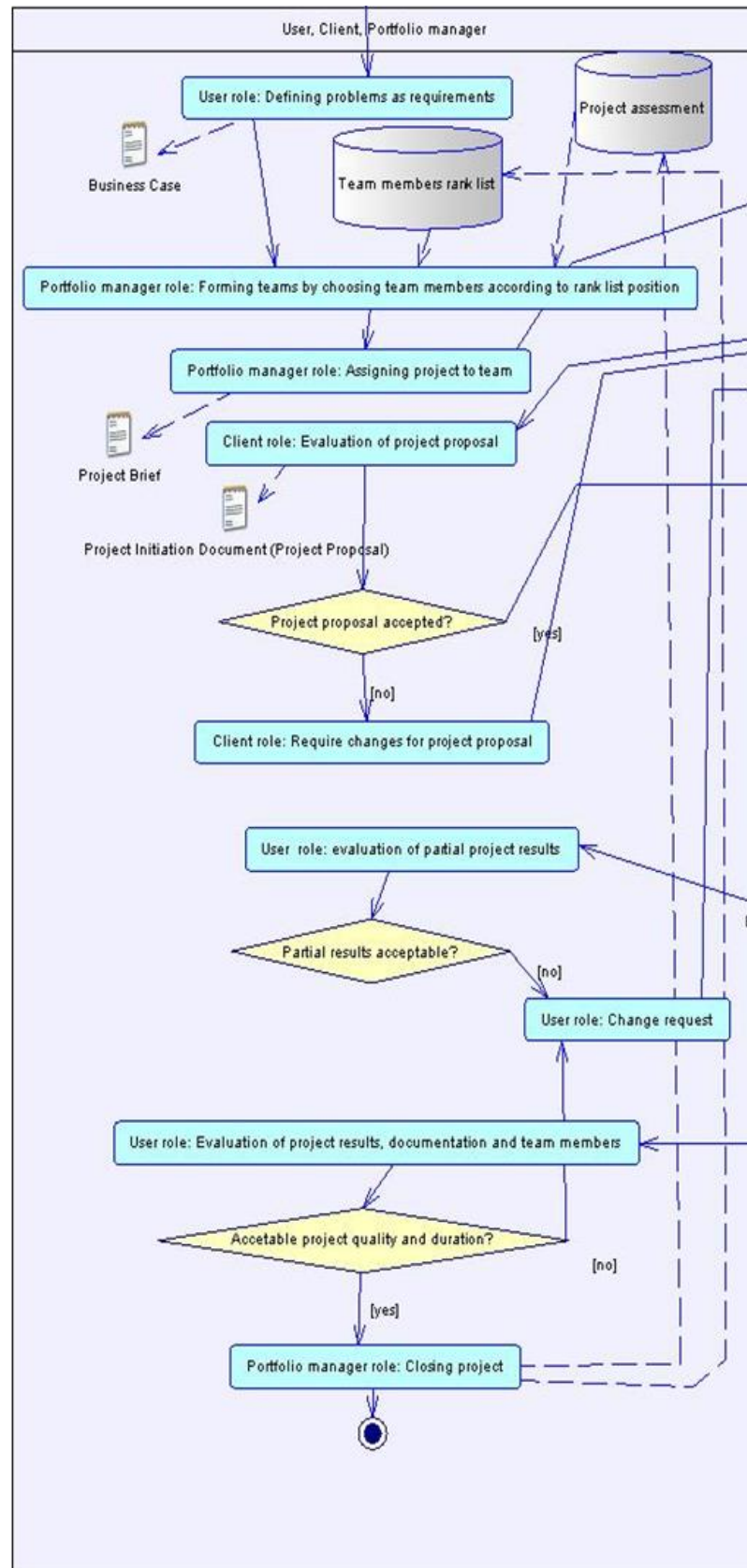
КАТЕГОРИЈА МОДЕЛА	ТИП МОДЕЛА	Семантичка ОБЛАСТ ПРИМЕНЕ	МЕТОДОЛОГИЈА / Приступ	Референца
Модел пословних процеса	Business Process Model	Управљање пројектима	PMBOK PRINCE 2	[Kazi et al, JEMC, 2011]
		Животни циклус развоја софтвера	Waterfall Agilni pristup	
Модел софтверских функција	USE CASE модел (дијаграм) из UML-а	-	Пресликавање примитивних пословних процеса у софтверске функције табелом пресликавања	[Kazi et al, AIIT, 2012a]
Модел података	Концептуални модел	-	Методологија интегрисаног приступа примени различитих метода концептуалног моделовања	[Kazi et al, Metalurgia, 2012] [Kazi et al, MIPRO, 2014]

### 3.2.2. Модел пословних процеса базиран на PRINCE2 методологији управљања пројектима

Модел пословних процеса, који прати ток извршавања активности и описује бележење релевантних података о систему, приказан је на следећој слици. Активности и складишта података описана су у овом моделу на основу активности и документације PRINCE2 методологије управљања пројектима.

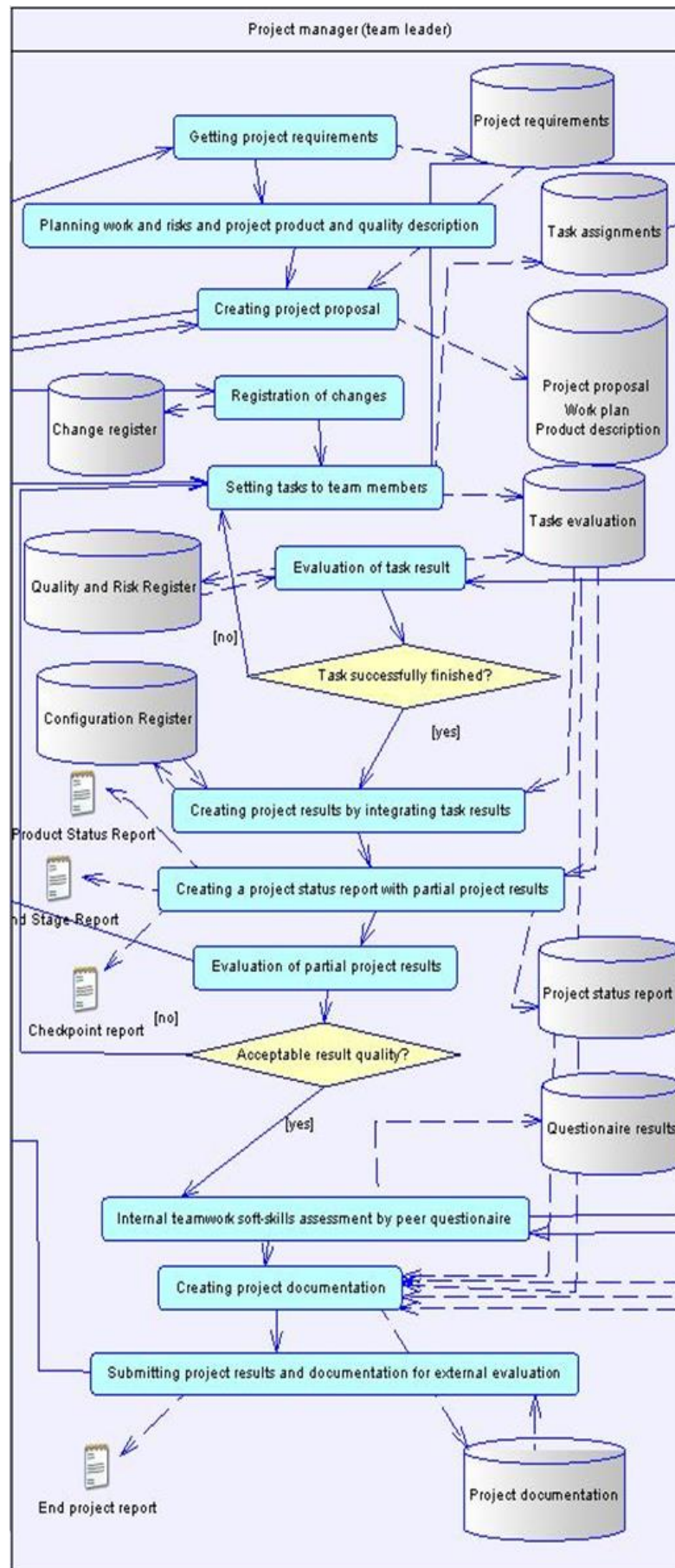


Слика 3.2.2.1. Бизнис процес модел за приказ модела процеса тока активности у управљању пројектом према PRINCE 2 и Balanced Scorecard методологији [Kazi et al, SISY, 2012]

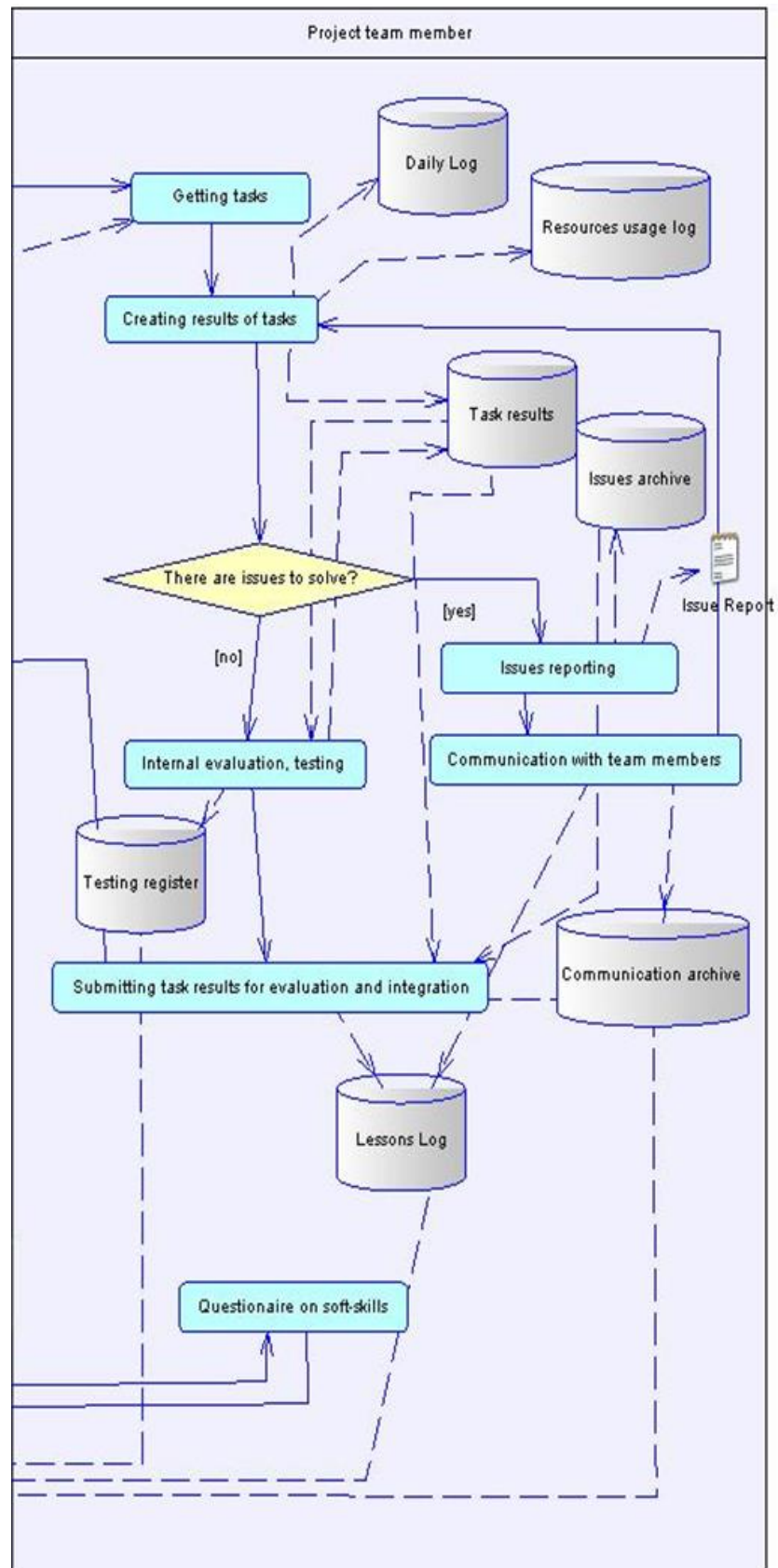


Слика 3.2.2.2. Део бизнис процес модела који се односи на улоге корисника, клијента и портфолио менаџера [Kazi et al, SISY, 2012]





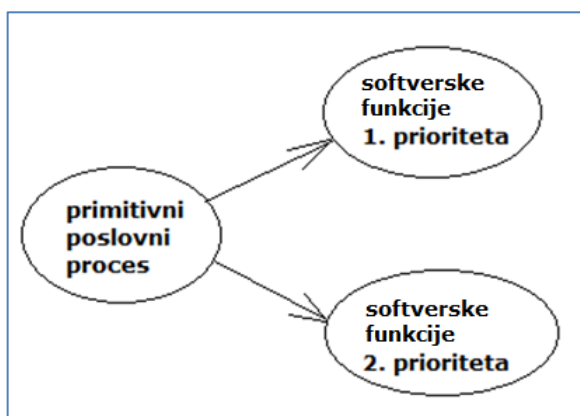
Слика 3.2.2.3. Део бизнис процес модела који се односи на улогу пројектног менаџера или тим-лидера [Kazi et al, SISY, 2012]



Слика 3.2.2.4. Део бизнис процес модела који се односи на улогу члана пројектног тима [Kazi et al, SISY, 2012]

### 3.2.5. Креирање модела софтверских функција на основу пресликавања примитивних пословних процеса

У оквиру [Kazi et al, AИТ, 2012a] описан је приступ дизајну софтверских функција на основу пресликавања примитивних пословних процеса из модела пословних процеса. У оквиру наведеног рада предложена је примена табеле пресликавања, где једном примитивном процесу може одговарати више софтверских функција (типа уноса, табеларног приказа, штампе докумената итд.) које се уређују у 2 колоне по приоритетима (1. Приоритет – софтверске функције које непосредно подржавају пословни процес, 2. Приоритет – софтверске функције које представљају додатне функције које су корисне, али не обезбеђују директно решавање пословне потребе).



Слика 3.2.5.1. Приступ пресликавању примитивних пословних процеса у софтверске функције 1. и 2. приоритета

Општи приказ табеле пресликавања примитивног пословног процеса у софтверске функције дат је у наставку:

Табела 3.2.5.1. Општи модел табеле пресликавања примитивних пословних процеса у софтверске функције

Примитивни пословни процес	Радно место реализације пословног процеса	Софтверска функција 1. Приоритета	Софтверска функција 2. Приоритета	Софтверска функција предуслов са функцију 1. приоритета	Профил корисника софтверске функције (Actor)	Тип апликације (Desktop, C/S, LAN, Web, Mobilna)
1	2	3	4	5	6	7

Начин примене методе пресликавања примитивних пословних процеса у софтверске функције описан је у наставку. На основу једног примитивног процеса може бити додељено више софтверских функција у свакој од колоне (3,4,5) које представљају софтверску подршку реализацији одговарајућег пословног процеса.

На основу издвајања примитивних процеса из модела пословних процеса (Business Process Model, Модел процеса Структурне систем анализе – дијаграм тока података и сл.) називи примитивних процеса се уписују сваки појединачно у посебан ред табеле у колону (1). У колону (2) уписујемо назив радног места који у организационом систему реализује активности наведеног примитивног пословног процеса. У колону (3) уписујемо називе софтверских функција које директно подржавају реализацију примитивног процеса („need to have“), док у колону (4) уписујемо називе софтверских функција које представљају додатне функционалне могућности програма које су корисне, али нису нужне у циљу решавања одговарајућег пословног процеса („nice to have“). У колону (5) уписујемо називе софтверских функција које представљају предуслов за извршавање софтверских функција 1. Приоритета. То су најчешће

софтверске функције које се односе на унос базичних (шифарничких) података или су то софтверске функције које по логичком следу тока пословних процеса претходе одговарајућој софтверској функцији. У колону (6) уписујемо назив профила корисника у софтверском решењу који ће имати могућност да користи софтверске функције 1. и 2. Приоритета. За исти примитивни пословни процес може бити више профила корисника при чему сваки има свој део софтверских функција које су усклађене са одговарајућим примитивним пословним процесом (у наредном примеру видети пример примитивног процеса „Креирање пројектног тима“). У последњу колону (7) уписујемо тип софтверске апликације (Desktop, C/S LAN – клијент сервер у мрежном окружењу, Web апликација или мобилна апликација) која ће представљати технолошки основ реализације софтверских функција за наведени примитивни пословни процес. Дакле, за сваки примитивни процес могуће је осмислити подршку у оквиру различитих технолошких решења, тако да заједнички предлог решења за целу табелу се може састојати из више различитих технолошких основа реализације. Чак штавише, за један примитивни процес може бити понуђено више алтернативних скупова софтверских функција и више различитих технолошких имплементационих основа. Коначно, креирамо UML-ов USE CASE дијаграм на основу података о софтверским функцијама из табеле пресликавања. Иако је изворна идеја о случајевима коришћења у USE CASE дијаграма заправо представљање пословних процеса у комбинацији са применом софтверског (или уопште ИТ) решења [Jacobson et al, 1999] [Booch et al, 1999], у овом предлогу приступа строго се одваја моделовање пословних процеса и софтверских функција, које се представљају у USE CASE дијаграму.

Примена наведеног приступа пресликавању примитивних пословних процеса у софтверске функције илустрована је у оквиру [Kazi et al, UDZ, 2013]. У наставку је дат приказ примене табеле пресликавања примитивних пословних процеса (дефинисаних у претходно приказаном моделу пословних процеса) у софтверске функције, реализовано на основу [Kazi et al, AIIT, 2012a]. Као технологија реализације за целокупну табелу усвојена је Web апликација.

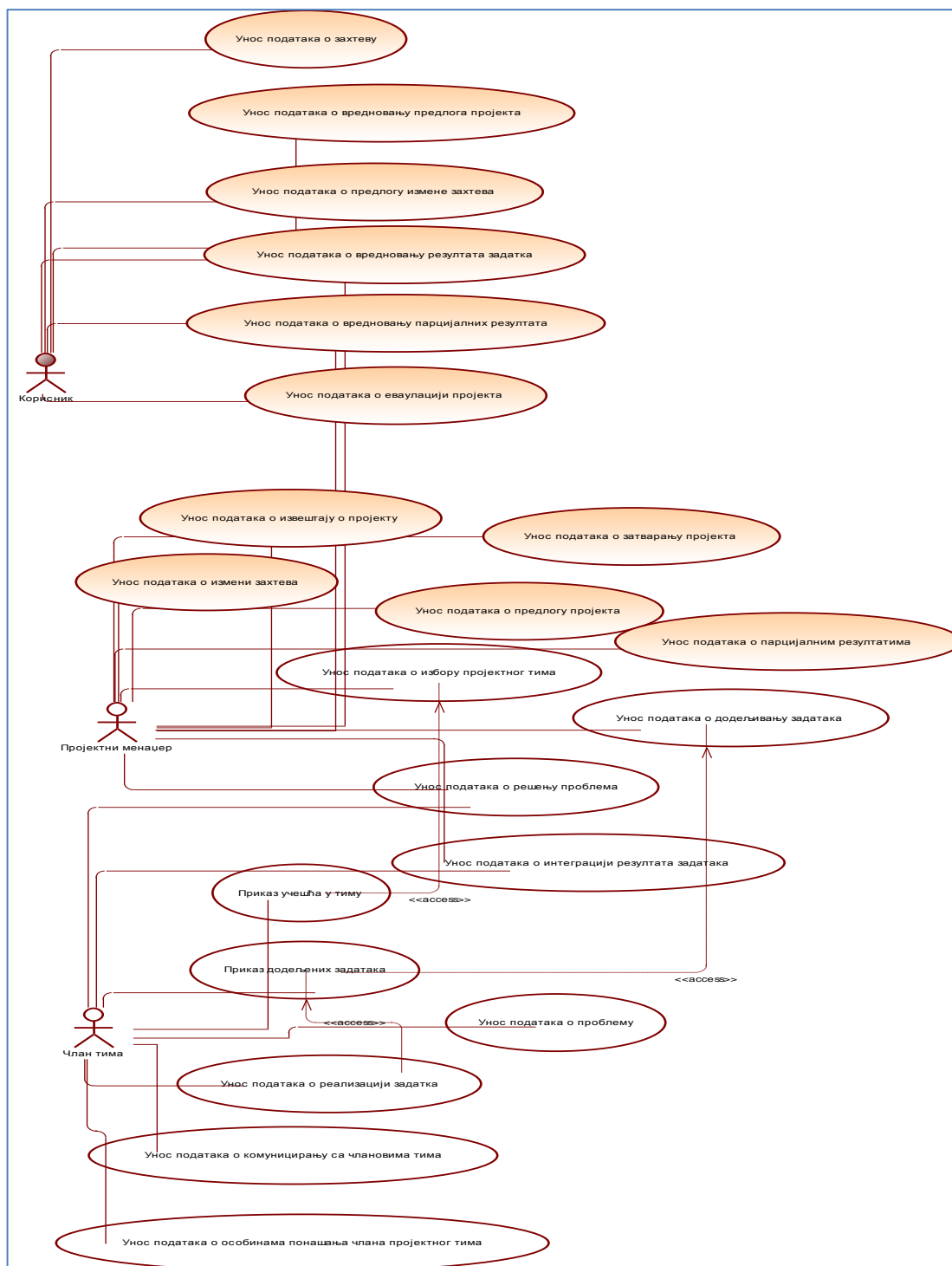
Табела 3.2.5.2. Пример примене табеле пресликавања пословних процеса у софтверске функције

ПРИМИТИВНИ ПОСЛОВНИ ПРОЦЕС	УЧЕСНИК ПОСЛОВНОГ ПРОЦЕСА	СОФТВЕРСКА ФУНКЦИЈА 1. ПРИОРИТЕТА	СОФТВЕРСКА ФУНКЦИЈА 2. ПРИОРИТЕТА	СОФТВЕРСКА ФУНКЦИЈА ПРЕДУСЛОВ	ПРОФИЛ КОРИСНИКА
Дефинисање проблема као захтев	Корисник	Унос података о захтеву корисника	Табеларни приказ захтева корисника Штампање листе захтева корисника	Унос података о корисницима	Корисник
Креирање предлога пројекта	Пројектни менаџер	Унос података о предлогу пројекта	Штампање предлога пројекта	Унос података о захтеву корисника	Пројектни менаџер
Креирање пројектног тима	Пројектни менаџер	Унос података о избору пројектног тима	Табеларни приказ чланова тима Штампање списка чланова пројектног тима	Унос података о предлогу пројекта	Пројектни менаџер
		Табеларни приказ учешћа у тиму	Штампање списка пројектата члана тима	Унос података о избору пројектног тима	Члан пројектног тима
Вредновање предлога пројекта	Корисник	Унос података о вредновању предлога пројекта	Табеларни приказ пројектата и филтрирање према статусу вредновања	Унос података о предлогу пројекта	Корисник
Предлагање измена захтева	Корисник	Унос података о предлогу измене захтева	Табеларни приказ измена захтева	Унос података о захтеву корисника	Корисник

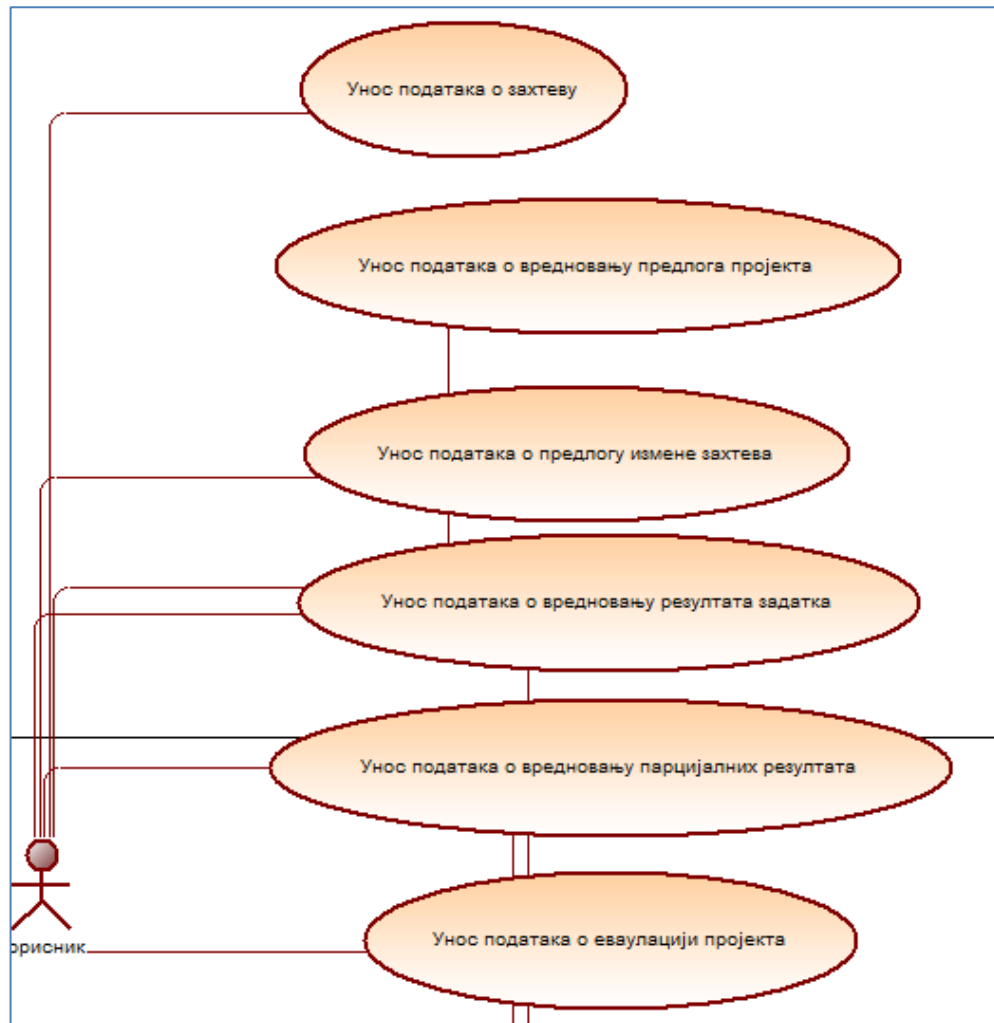
Евидентирање измена захтева	Пројектни менаџер	Унос података о измени захтева	Табеларни приказ усвојених измена захтева	Унос података о захтеву корисника	Пројектни менаџер
Додељивање задатка	Пројектни менаџер	Унос података о додељивању задатка	Табеларни приказ додељених задатака	Унос података о избору пројектног тима	Пројектни менаџер
		-	Табеларни приказ додељених задатака	Унос података о додељивању задатка	Члан пројектног тима
Реализација задатка	Члан пројектног тима	Унос података о реализацији задатка	Табеларни приказ реализованих задатака	Унос података о додељивању задатка	Члан пројектног тима
Пријављивање проблема	Члан пројектног тима	Унос података о новом проблему	Табеларни приказ проблема	Унос података о додељивању задатка	Члан пројектног тима
Решавање проблема	Пројектни менаџер	Унос података о решењу проблема	Табеларни приказ решавања проблема	Унос података о новом проблему	Пројектни менаџер Члан пројектног тима
Комуницирање са члановима тима	Члан пројектног тима	Унос података о комуницирању са члановима тима	Табеларни приказ порука у комуникацији	Унос података о избору пројектног тима	Члан пројектног тима
Вредновање резултата задатка	Корисник	Унос података о вредновању резултата задатка	Табеларни приказ података о задацима са подацима о вредновању	Унос података о реализацији задатка	Корисник Пројектни менаџер
Интеграција резултата задатака	Пројектни менаџер	Унос података о интеграцији резултата задатка	Табеларни приказ података о интеграцији резултата	Унос података о вредновању резултата задатка	Пројектни менаџер Члан пројектног тима
Креирање извештаја о парцијалним резултатима	Пројектни менаџер	Унос података о парцијалним резултатима Штампа извештаја о парцијалним резултатима	Табеларни приказ парцијалних резултата	Унос података о реализацији задатка	Пројектни менаџер
Верификација парцијалног резултата	Корисник	Унос података о верификацији парцијалног резултата	Табеларни приказ верификованих парцијалних резултата	Унос података о парцијалним резултатима	Корисник Пројектни менаџер
Утврђивање особина понашања учесника	Члан пројектног тима	Унос података о особинама понашања учесника	Табеларни статистички приказ података о понашању учесника	Унос података о избору пројектног тима	Члан пројектног тима
Креирање извештаја о пројекту	Пројектни менаџер	Унос података о извештају о пројекту Штампање извештаја о пројекту	Табеларни приказ пројеката	Унос података о евалуацији пројекта	Пројектни менаџер
Евалуација пројекта	Корисник	Унос података о евалуацији пројекта	Табеларни приказ података о пројектима са подацима о евалуацији	Унос података о интеграцији резултата задатка	Корисник
Затварање	Пројектни	Унос података	Табеларни	Унос података	Пројектни

пројекта	менаџер	о затварању пројекта	приказ података о активним и затвореним пројектима	о евалуацији пројекта	менаџер
----------	---------	----------------------	--	-----------------------	---------

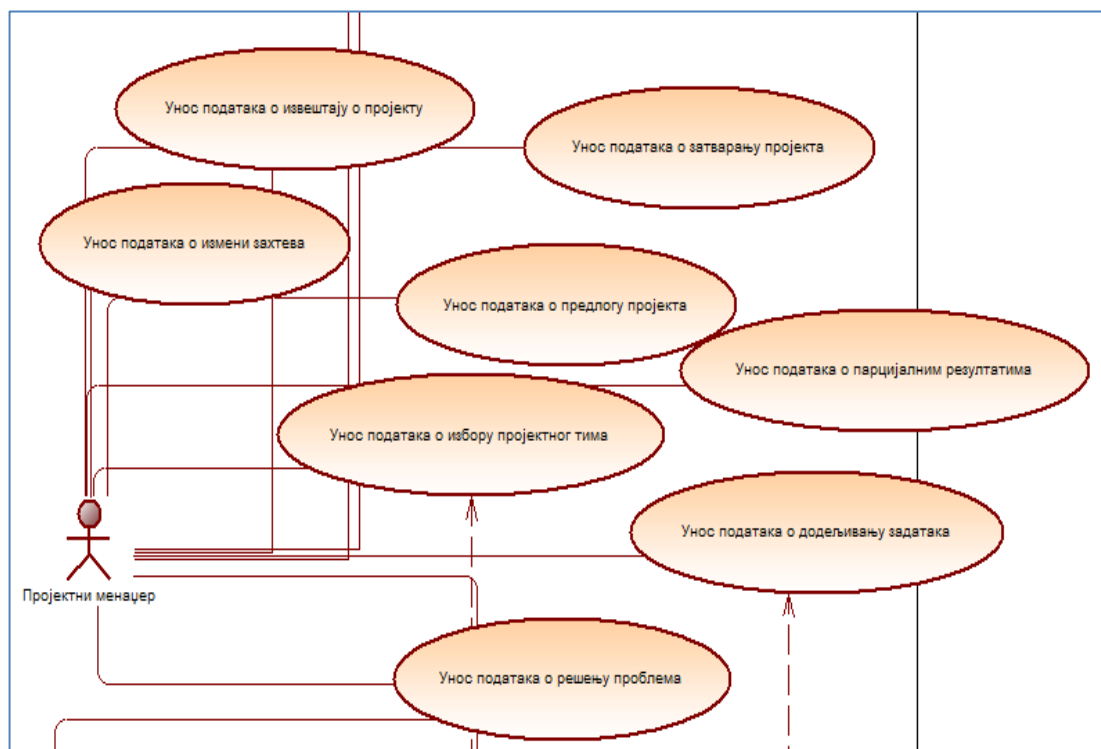
Дијаграм случајева коришћења представља софтверске функције, њихову међузависност и додељеност поједином профилу корисника. У наставку ће бити приказане само суштинске функције уноса података на заједничком и појединачним дијаграмима за сваки профил корисника.



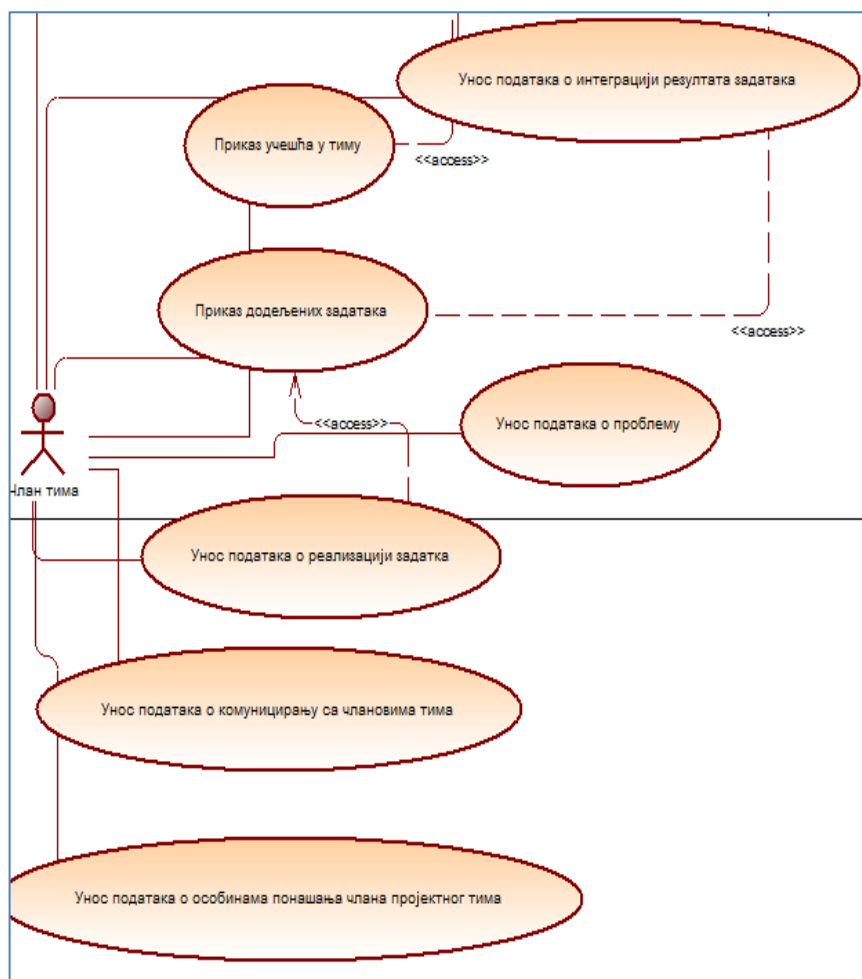
Слика 3.2.5.2. Дијаграм случајева коришћења са приказом софтверских функција



Слика 3.2.5.3. Дијаграм случајева коришћења за профил Корисник



Слика 3.2.5.4. Дијаграм случајева коришћења за профил Пројектни менаџер



Слика 3.2.5.5. Дијаграм случајева коришћења за профил Члан тима

### 3.2.6. Складишта података за нормализацију и концептуални модел података

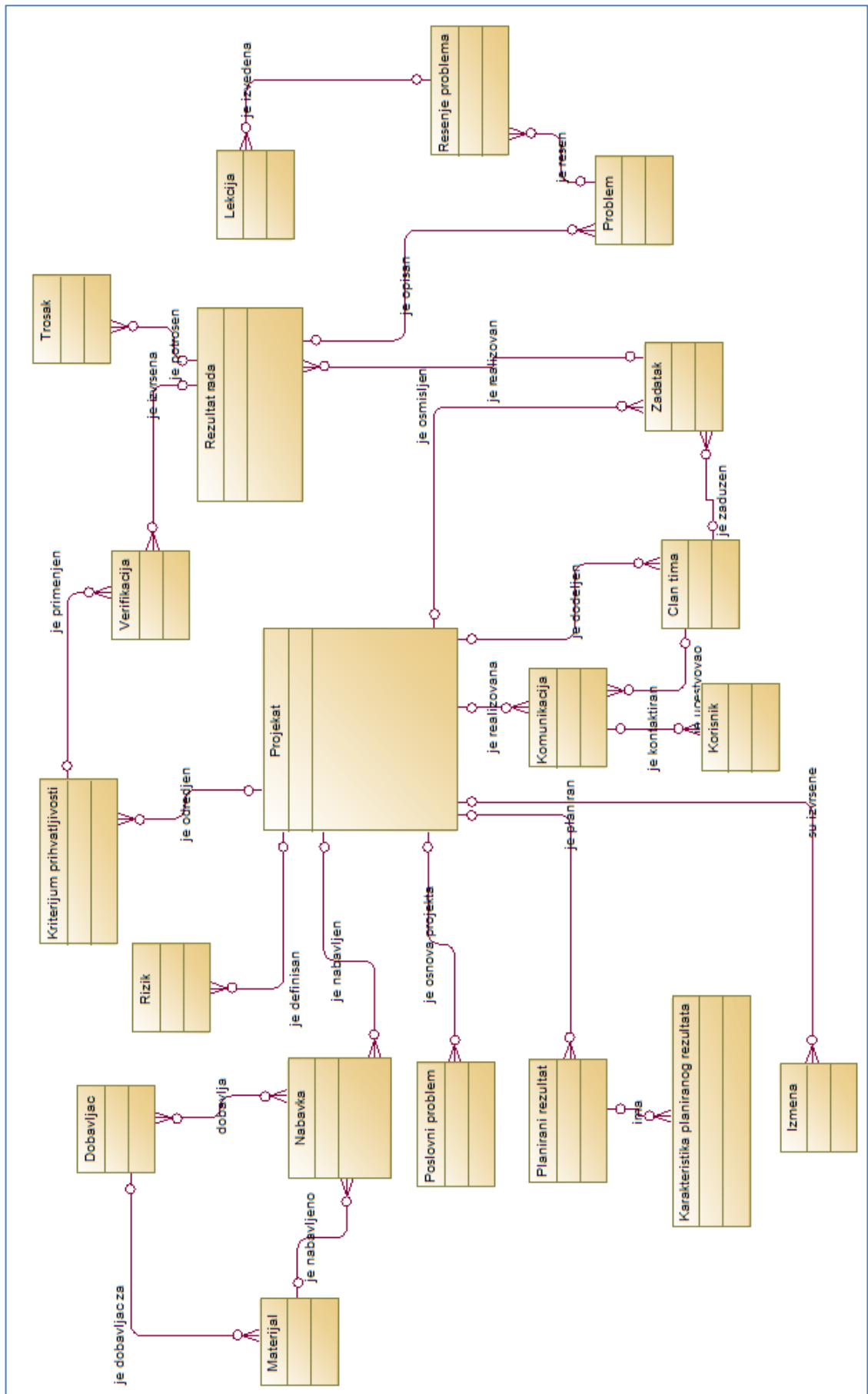
На основу претходно приказаног дијаграма модела пословних процеса, архиве података (складишта података) које треба да буду нормализоване у концептуални модел података представљене су табелом 3.2.6.1. Нормализација складишта података као нови метод креирања концептуалног модела података је приказана у [Kazi et al, Metalurgia, 2012] и [Kazi et al, MIPRO, 2014].

Табела 3.2.6.1. Списак складишта података за нормализацију

Ранг члана тима	Захтеви корисника	Извештај о статусу пројекта	Архива проблема
Оцена пројекта	Додељивање задатака	Резултати упитника о понашању	Регистар тестирања резултата
Регистар промена	Предлог пројекта	Пројектна документација	Архива комуникације
Регистар ризика	Радни план	Дневник рада	Архива лекција
Регистар квалитета	Опис производа	Регистар коришћења ресурса	
Регистар конфигурације	Вредновање задатака	Евиденција резултата задатака	

Списак елементарних података који чине структуру претходно описаних складишта података зависе од усвојене методологије развоја софтвера и управљања пројектима, али општи концептуални модел података који је дат у наставку може представљати заједничку основу. Концептуални модел података (Слика 3.2.6.1.) креиран је на основу PMBOK и PRINCE2 методологије за управљање пројектима општег типа.

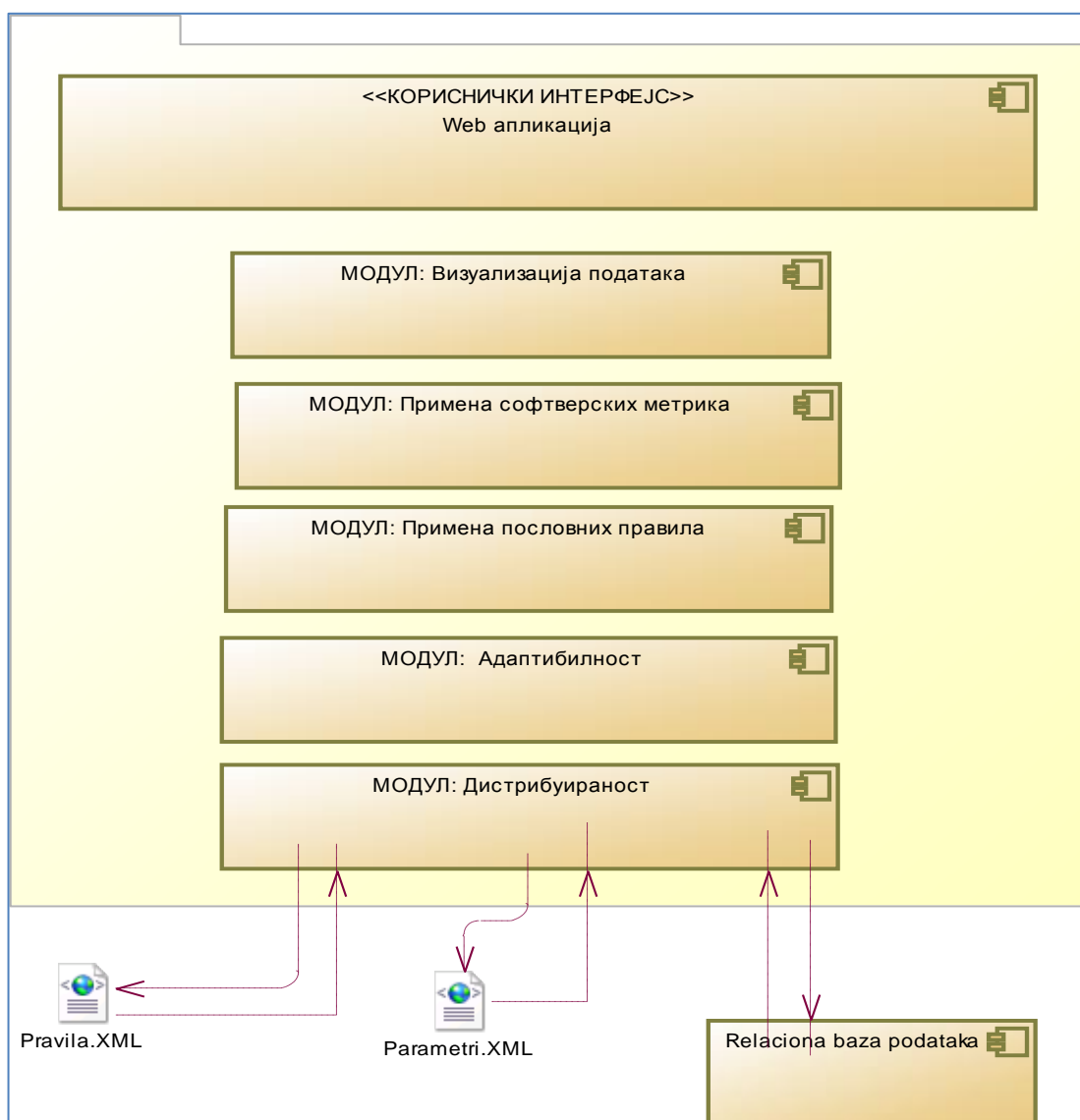




Слика 3.2.6.1. ER дијаграм – концептуални модел података

### 3.2.7. Модел компоненти и размештаја у опису архитектуре предложеног система

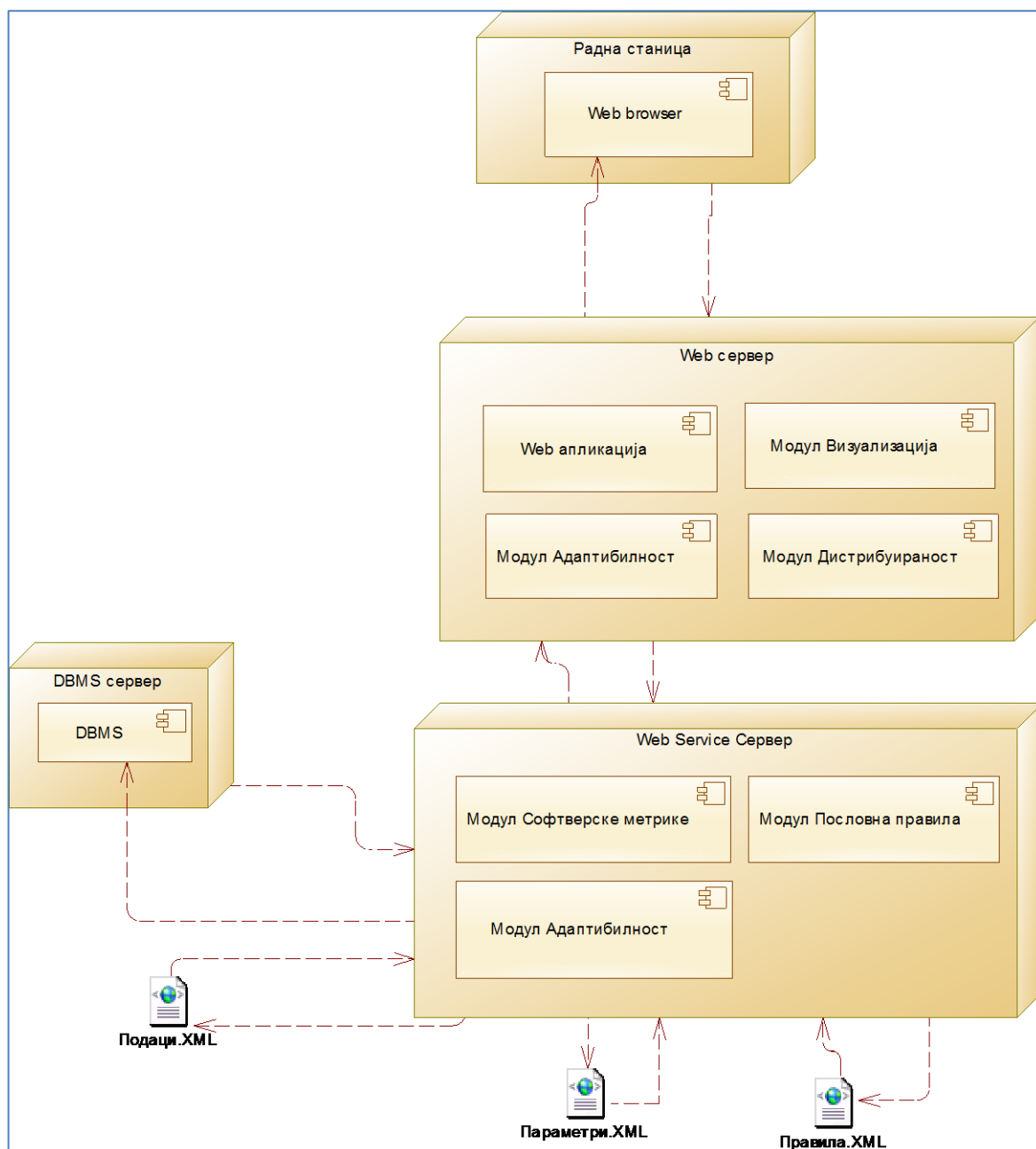
Архитектура предложеног система састоји се од модула, представљених дијаграмом компоненти на слици 3.2.7.1., односно дијаграмом размештаја на слици 3.2.7.2.



Слика 3.2.7.1. Дијаграм компоненти у приказу архитектуре предложеног система

Планирани систем састоји се из Web апликације која омогућава евидентирање података о процесима и карактеристикама софтверског производа у току реализације софтверског пројекта. Снимање података врши се у оквиру релационе базе података. Подршка визуализацији података треба да представља модул у оквиру Web апликације којим се визуално представља напредак сваког пројекта. Посебан модул омогућава примену пословних правила и софтверских метрика, који се могу мењати у зависности од типа пројекта и технологије израде, тако да се постиже адаптивност система. Дистрибуираност информационог система постиже се дистрибуираним изворима података – XML фајловима и релационим базама података, као и дистрибуираношћу процесирања, које се постиже применом Web сервиса.

Дијаграм размештаја описује планиран размештај софтверских компоненти у оквиру рачунарске мреже, као што је дато на слици 3.2.7.2.



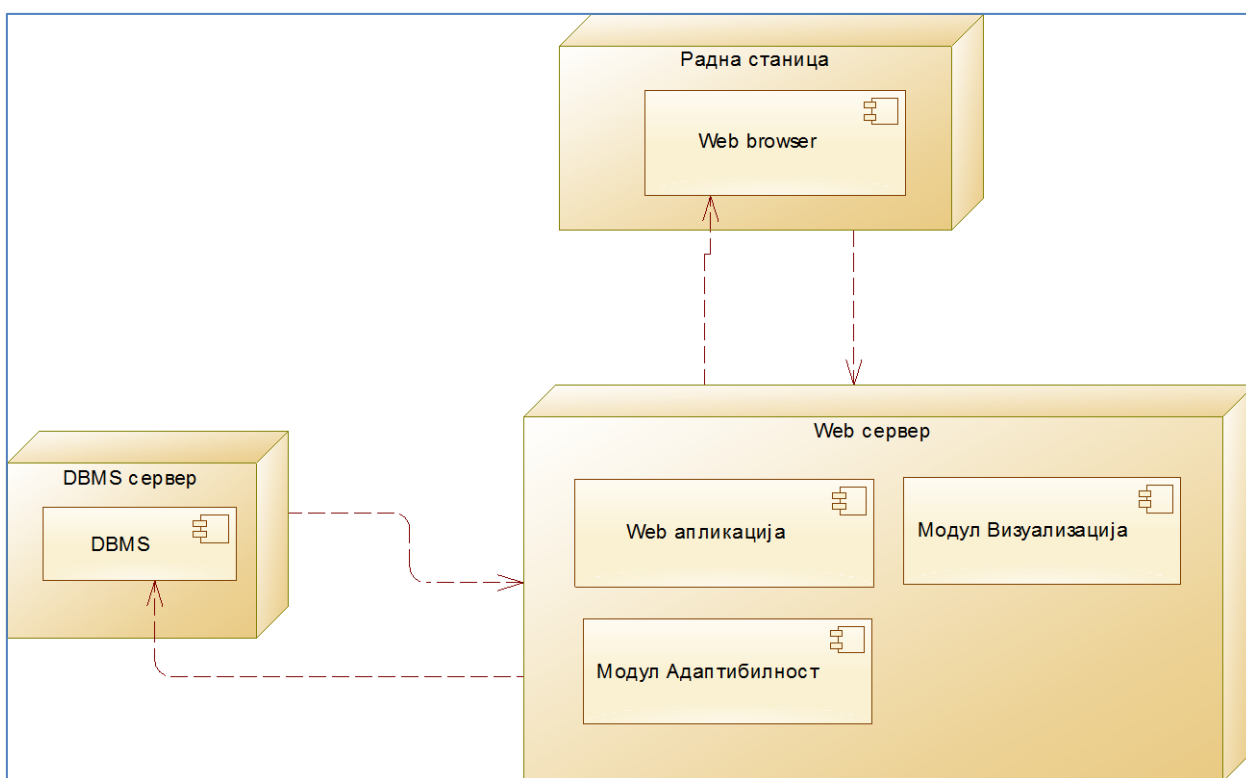
Слика 3.2.7.2. Дијаграм размештаја предложеног система

Планирани систем треба да се састоји из 4 типа чворова:

1. DBMS сервер – рачунар који садржи систем за управљање базом података (DBMS), као и саму базу података (комплетне структуре или хоризонтално или вертикално фрагментирану)
2. Web service сервер – рачунар на ком се извршава Web service који омогућава дефинисање и смештање параметара рада, пословних правила и софтверских метрика, које се могу примењивати позивом из Web апликације, чиме се постиже адаптивност, а садржи и функције за приступ бази података
3. Web сервер – садржи основну Web апликацију, која има модуле за консултовање Web сервиса (подршку дистрибуираности), као и модул за подршку адаптивности коришћењем Web сервиса који омогућавају примену софтверских метрика и пословних правила. Такође, садржи и модул за визуализацију података о праћењу прогреса процеса развоја софтвера.

4. Радна станица – садржи Web browser ради омогућавања коришћења web апликације.

Почетни прототип (приказан на слици 3.2.6.3.) је у складу са агилним приступом реализован тако да обухвата основне функције у првој итерацији развоја (имплементација је описана у одељку 4.). Наведени почетни прототип је тестиран у емпиријском истраживању у оквиру наставног окружења (одељак 5.2.) и од стране ИТ кадрова (одељак 5.3.). У каснијем унапређењу почетни прототип је надограђен тако да има технолошке елементе имплементације у складу са сликом 3.2.7.2. и побољшану функционалност у складу са сликом 3.2.7.1.



Слика 3.2.7.3. Дијаграм размештаја за почетни прототип који садржи основне функције, развијен у првој итерацији у складу са агилним приступом развоја

### 3.3. МЕТРИЧКИ МОДЕЛИ ЗА ЕВАЛУАЦИЈУ АРТЕФАКТА РАЗВОЈА СОФТВЕРА У ОБЛАСТИ РАЗВОЈА ИНФОРМАЦИОНОГ СИСТЕМА

#### 3.3.1. Теоријско-истраживачка основа предложених модела

Управљање реализацијом софтвера, тј. управљање софтверским пројектом у оквиру процеса контроле и мониторинга захтева мерење карактеристика самог процеса, али и квалитета резултата рада. Наравно, скуп резултата рада се разликује код различитих типова софтвера и софтверских пројеката. У наставку је дат преглед метричких модела и теоријско-истраживачких основа предложених модела, који се односе на евалуацију артефакта, односно резултата у процесу развоја софтвера у области развоја информационог система – модела пословних процеса, модела софтверских функција, модела података и софтверске апликације. У табели 3.3.1.1. представљени су предложени метрички модели, методологија или приступ који представља основ за креирање наведеног метричког модела и референца на рад где је наведени метрички модел презентован или су описани његови елементи.

Табела 3.3.1.1. Теоријско-истраживачка основа предложених модела за евалуацију артефакта развоја софтвера у области развоја информационог система

МЕТРИЧКИ МОДЕЛ – сврха	ОСНОВНА МЕТОДОЛОГИЈА /Приступ	Референца
Евалуација модела пословних процеса	Синтаксни, семантички и прагматички аспект [Belle J-P, 2006]	Елементи и хеуристичка упутства [Radulovic et al, UDZ, 2011]
Евалуација модела софтверских функција		Елементи и хеуристичка упутства [Kazi et al, UDZ, 2013]
Евалуација концептуалног модела података		Метрички модел [Kazi et al, UDZ, 2013] [Kazi et al, Metalurgia, 2012] Аутоматизација евалуације концептуалног модела [Kazi et al, ТТЕМ, 2012] [Kazi et al, ARAB, 2016]
Евалуација софтверске апликације (у оквиру развоја информационог система)	Стандарди квалитета софтвера ISO 9126, ISO 2010 Критеријуми селекције софтверског пакета [Jadhav&Sonar, 2009] Модел за евалуацију информационог система [Whitworth&Zaic, 2003] [Irani, 2002]	Елементи и хеуристичка упутства [Kazi et al, UDZ, 2013]

#### 3.3.2. Метрички модел за евалуацију модела пословних процеса

Истраживања квалитета модела пословних процеса реализована су у контексту синтаксног, семантичког и прагматичког аспекта. У раду [Belle J-P, 2006] дефинисан је метрички модел за евалуацију модела пословних процеса, али се предложени приступ може применити на моделе других категорија. У раду [Vanderfeesten et al, 2008] извршена је примена елемената метрика у области софтверског инжењерства које се могу применити на евалуацију модела процеса.

У овом одељку описан је предлог приступа евалуацији модела пословних процеса. Најчешће коришћени модели пословних процеса су Дијаграм тока података са пратећим речником података структурне систем анализе (DTP) и Business Process Model

(BPM). За оба наведена типа модела дат је предлог скупа метрика које се односе на поједине аспекте вредновања квалитета модела пословних процеса.

Табела 3.3.2.1. Метрички модел за евалуацију модела пословних процеса

<b>КРИТЕРИЈУМ ВРЕДНОВАЊА</b>	<b>ДТР</b>	<b>ВРМ</b>
<b>НАЗИВИ ЕЛЕМЕНАТА – правилни, прецизни</b>	+	+
• Процес	+	+
• Ток података	+	+
• Складиште података	+	+
• Систем из окружења (entitet)	+	
• Организациона јединица („swim lane“)		+
<b>ОРГАНИЗАЦИЈА ЕЛЕМЕНАТА</b>	+	
• Стабло процеса – припадност, међузависност	+	
• Стабло процеса – хронолошка уређеност	+	
• Повезаност елемената - правило баланса	+	
• Повезаност елемената – интерни токови	+	
• Повезаност елемената – повезаност процеса са складиштима података	+	+
• Повезаност елемената – повезаност са процеса са системима из окружења	+	
• Хронолошки распоред процеса		+
• Распоред процеса, токова и складишта података унутар одговарајуће „пливачке линије“ организационе јединице (учесника процеса)		+
<b>РЕЧНИК ПОДАТАКА</b>	+	+
• Потпуност скупа елементарних података	+	+
• Правилни називи елементарних података	+	+
• Правилан распоред елементарних података - у токовима података	+	+
• Правилан распоред елементарних података – у складиштима података	+	+
• Правилан синтаксни опис структуре тока података	+	+
• Правилан синтаксни опис структуре складишта података	+	+

### 3.3.3. Метрички модел за евалуацију модела софтверских функција

Моделом софтверских функција представљамо функционални аспект дизајна будућег софтверског решења. Модел софтверских функција може представљати и основ за процену величине, односно трајања софтверског пројекта [Nassif, 2012]. За представљање ове категорије модела најчешће се користи UML USE CASE дијаграм и пратећа спецификација. У овом раду се предлаже проширење модела софтверских функција табелом пресликавања из примитивних пословних процеса у софтверске функције, што представља основ креирања USE CASE дијаграма.

Табела 3.3.3.1. Метрички модел за евалуацију модела софтверских функција

<b>ОБЛАСТ</b>	<b>КАРАКТЕРИСТИКА</b>
ТАБЕЛА ПРЕСЛИКАВАЊА	Назив софтверске функције прецизан и технички коректан (назив као софтверска функција)
	Распоред – правилно одређен размештај софтверских функција по категоријама: 1. Приоритет, 2. Приоритет, Предуслов
	Потпуност- потпун скуп софтверских функција првог приоритета
	Правилно одређен профил корисника у односу на радно место примитивног пословног процеса и одговарајућих софтверских функција којима може да приступа
ДИЈАГРАМ USE CASE	Правилна повезаност случаја коришћења са профилем корисника
	Правилна међусобна повезаност случајева коришћења
	Правилно коришћење стереотипа код веза међузависности

	случајева коришћења
СПЕЦИФИКАЦИЈА СЛУЧАЈА КОРИШЋЕЊА	Правилно одређен предуслов Правилно описани кораци активности (action steps) Правилно одређене тачке проширења Правилно одређени изузеци Правилно одређен постуслов (резултат)

### 3.3.4. Метрички модел за евалуацију концептуалног модела података

Концептуални модел података представља један од најважнијих артефакта процеса развоја софтвера, који настаје у фази моделовања, односно дизајна. Концептуални модели података представљају основ за креирање софтверског решења у области информационих система и софтвера других категорија чија се имплементација заснива на примени база података. Такође, концептуални модели података представљају основ за процену величине софтвера, односно трајања софтверског пројекта. [Tan et al, 2009] [Atak&Gencel, 2006] С обзиром на значај концептуалних модела података, истраживања у области метода креирања и евалуације концептуалних модела података су изузетно важна.

#### 3.3.4.1. Методологија за интеграцију метода концептуалног моделовања података

У оквиру рада [Kazi et al, Metalurgia, 2012] предложен је и емпиријски верификован приступ и метрички модел за евалуацију концептуалних модела података насталих применом појединачних метода концептуалног моделовања, као и применом интеграције појединачних метода концептуалног моделовања података у оквиру процеса развоја информационих система.

Табела 3.2.4.1.1. Улога концептуалног моделовања података у оквиру интегралног процеса развоја софтвера информационог система [Kazi et al, Metalurgia, 2012]

Inf. system development phase	Input material	Conceptual Modeling Method
Requirements collection	Text that describe client requirements	Grammar analysis of text
Business process description	Business process textual description	Grammar analysis of text
Business process modeling	Primitive process	Sub-model creation
	Attributes	"Assigning attribute to entity"
	Data flows	Syntax decomposition, Normalization
	Data stores	Syntax decomposition, Normalization
	XML message	schema analysis, sub-model
System design	Use case specification	Grammar analysis of text, sub-model creation
	Actors/ user roles	Partial modeling for each external schema

2. Детаљнији приказ интеграције метода формулисан као нова методологија која укључује интеграцију појединачних метода концептуалног моделовања представљена је у [Kazi et al, MIPRO, 2014].

Табела 3.2.4.1.2. Методологија концептуалног моделовања применом интеграције метода концептуалног моделовања у оквиру развоја софтвера информационог система [Kazi et al, MIPRO, 2014]

<b>Order of methods application</b>	<b>INPUT - material</b>	<b>Method</b>	<b>OUTPUT – result</b>
1	Textual description of business process	Grammar analysis of business domain textual description	1 <sup>st</sup> draft model with: Entities, relationships
2	List of data dictionary data items	Assignment of entities to attributes from data dictionary	2 <sup>nd</sup> draft model with : Entities with attributes assigned
3	Syntax structure of each data store from data flow diagram	Data store decomposition by structure notation analysis	3 <sup>rd</sup> draft model with Relationships, attributes, ; cardinalities of relationships, IS-A hierarchy relationships
4	Data store structure, i.e. list of data items for each data store	Data store normalization	4 <sup>th</sup> draft model with : Entities, relationships, attributes, identifiers, functional dependencies within entities, non-transitive functional dependencies within entities
<b>Integrated/Central draft model</b>			
5	Integrated draft model	Checking correctness and performing corrections	Correct model
6	Correct model, Table (created according to [5]) of primitive processes assignment to Use cases	Checking completeness and corrections	Correct and complete model
<b>Correct and complete model</b>			

### 3.2.4.2. Метрички модел за евалуацију концептуалних модела података

У раду [Kazi et al, Metalurgia, 2012] приказан је метрички модел за вредновање карактеристика квалитета концептуалног модела података. Метрички модел је емпиријски верификован над примерима (реализованих над истом семантиком у оквиру заједничке студије случаја) концептуалних модела који су настали применом појединачних метода и интеграције метода концептуалног моделовања.

Комплетнији скуп метрика који се односи на карактеристике квалитета концептуалног модела дат је у овој докторској дисертацији, као проширење предложеног метричког модела из [Kazi et al, Metalurgia, 2012] и [Kazi et al, UDZ, 2013].

Табела 3.2.4.2.1. Метрички модел за евалуацију концептуалног модела података са синтаксног, семантичког и прагматичког аспекта (проширен у односу на [Kazi et al, Metalurgia, 2012] и [Kazi et al, UDZ, 2013])

<b>GRUPA ELEMENATA</b>	<b>GRUPA OSOBINA</b>	<b>Potrebna osobina - METRIKA</b>
CELINA MODELA	Semantika	Semantički u skladu sa problemom
	Usklađenost sa datim materijalom	Pokrivenost opisa posla
		Pokrivenost importovanih elementarnih podataka
		Pokrivenost skladišta podataka sa DTP
	Kompletnost	Kompletnost skupa entiteta
Kompletnost skupa atributa		



	Jezička nedvosmislenost	Kompletnost skupa poveznika
		Nepostojanje sinonima u nazivima entiteta i atributa
	Прагматика	Nepostojanje homonima u nazivima entiteta i atributa
		Моргућност креирања SQL упита
ENTITET	Naziv	Једноставност ажурирања података у оквиру трансакција
		Imenica ili glagolska imenica
		Izražava sustinske objekte i događaje
		Ne izražava nazive dokumenata
	Entitet i atributi	Jednina u nazivu
		Ima identifikaciono obeležje ili je identifikaciono zavisn od drugih entiteta
		Dodeljen atribut odgovarajućem entitetu
ATRIBUT	Naziv	(2NF) Ne postoje brojni odnosi atributa 1:M u entitetu
		(3NF) Ne postoje tranzitivne zavisnosti atributa u entitetu
	Atributi i tipovi /domeni podataka	(1NF) Jednina u nazivu
		(1NF) Nema naziv kao struktura
		Odgovarajući tip podatka
	Karakteristike	Atribut nema za tip podatka domen (uzi skup vrednosti nego sto je standardni tip podatka), vec je izvucen kao sifarnik
		Atributi se sa istim nazivom ne ponavljaju (nema redundanse)
POVEZNIK	Naziv	Nema izračunljivih atributa, vec samo analitickih vrednosti medju atributima na osnovu kojih se vrše izracunavanja
		Poveznik ima naziv
	Nacin povezivanja entiteta	Naziv kao glagol
		Povezani direktno entiteti koji treba da budu povezani
		Entiteti su povezani hronoloskim sledom međuzavisnih događaja
		Povezani su entiteti u gerund odnosu, ako je potrebno
	Podesavanja osobina	Ako poveznik ima važne attribute, pretvoren je u entitet
		Pravilno podešen kardinalitet
		Pravilno podešena strana na vezi 1:M
		Pravilno određeno da li ima opravdanja za vezu 1:1 ili je jedan entitet
Pravilno određeno u vezi 1:1 ko je dominant		
Pravilno određeno u vezi M:M da li ima važnih atributa ili ostaje M:M		
Pravilno podesena identifikaciona zavisnost		
Pravilno podešena is-a hijerarhija (gde je nadredjeni, bar jedan podređeni, migrira samo ID)		

### 3.3.5. Метрички модел за евалуацију софтверске апликације

У наставку ће бити приказан метрички модел за евалуацију софтверске апликације као артефакта у процесу развоја софтвера информационог система. Метрички модел је описан кроз карактеристике из стандарда ISO 2010 и ISO 9126 (прва колона), истраживања [Jadhav&Sonar, 2009], [Whitworth&Zaic, 2003], [Irani, 2002] и имплементационе детаље реализације захтеване карактеристике, која омогућује конкретну проверу постојања одговарајуће карактеристике, односно имплементацију решења у складу са наведеним захтевом.

Табела 3.3.5.1. Метрички модел за евалуацију софтверске апликације као артефакта у процесу развоја софтвера информационог система

<b>ГРУПА ЗАХТЕВА / ЗАХТЕВ КВАЛИТЕТА</b>	<b>ИМПЛЕМЕНТАЦИОНИ ДЕТАЉИ РЕАЛИЗАЦИЈЕ ЗАХТЕВАНЕ КАРАКТЕРИСТИКЕ</b>
<b>ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	
Функционалност софтвера	Софтвер функционише
Функционална усклађеност са потребама	Усклађеност решења са захтевима (случајеви коришћења USE CASE или „user stories“), потребама и очекивањима корисника
Подршка основним функцијама	
<ul style="list-style-type: none"> <li>Унос и аквизиција података</li> </ul>	Унос података подршком за различите уређаје
<ul style="list-style-type: none"> <li>Ажурирање података</li> </ul>	брисање, измена
<ul style="list-style-type: none"> <li>Верификација података</li> </ul>	Контрола исправности података приликом уноса/измене
<ul style="list-style-type: none"> <li>Чување података</li> </ul>	База података, различити формати (XML), дистрибуиране базе података, бекап уређаји
<ul style="list-style-type: none"> <li>Размена података</li> </ul>	Интероперабилност – експорт података, учитавање података других формата
<ul style="list-style-type: none"> <li>Обрада података</li> </ul>	Сумарна и статистичка обрада података
<ul style="list-style-type: none"> <li>Презентовање података</li> </ul>	Графичко представљање, Визуализација података, Графикони, Табеларни прикази
<ul style="list-style-type: none"> <li>Претрага</li> </ul>	Филтрирање података, сортирање, навигација
<b>КВАЛИТЕТ ПОДАТАКА</b>	
<ul style="list-style-type: none"> <li>Комплетност</li> </ul>	Усклађеност базе података, екранских форми и извештаја са базом података, захтевима корисника
<ul style="list-style-type: none"> <li>Прецизност</li> </ul>	Одређени типови података, ограничења
<ul style="list-style-type: none"> <li>Конзистентност</li> </ul>	Усклађеност података из различитих табела базе података (транзакције)
<ul style="list-style-type: none"> <li>Доступност</li> </ul>	Поузданост апликације у LAN, развој web и мобилних апликација повећава доступност
<ul style="list-style-type: none"> <li>Поузданост - кредибилитет</li> </ul>	Извор података – аквизиција, директни унос Профили корисника, Овлашћени приступ,
<b>НЕФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	
<ul style="list-style-type: none"> <li>Погодност за одржавање</li> </ul>	Издвајање шифарника, довољан ниво апстракције дизајна базе података, модуларизација, вишеслојно програмирање
<ul style="list-style-type: none"> <li>Модуларност</li> </ul>	Издвајање модула – Dynamic Link Libraries, Web Services
<ul style="list-style-type: none"> <li>Конзистентност</li> </ul>	Усклађеност делова апликације
<ul style="list-style-type: none"> <li>Поновна искористивост</li> </ul>	Модули са општим решењима, семантички независним, Генератори класа, Генератори корисничког интерфејса
<ul style="list-style-type: none"> <li>Оптимално коришћење ресурса</li> </ul>	Брзина рада кода, минимизација процесорског времена и меморијских ресурса
<ul style="list-style-type: none"> <li>Поузданост</li> </ul>	Верификација података, Коректност формула код израчунавања
<ul style="list-style-type: none"> <li>Валидност</li> </ul>	Минимизација грешака функционисања
<ul style="list-style-type: none"> <li>Стабилност</li> </ul>	Верификација података, Бекап података
<ul style="list-style-type: none"> <li>Безбедност података</li> </ul>	Профили корисника, Овлашћени приступ, криптографија
<ul style="list-style-type: none"> <li>Прецизност</li> </ul>	Тачност приликом обраде података (израчунавања, интеграција итд.)
<ul style="list-style-type: none"> <li>Лакоћа коришћења</li> </ul>	Интуитивни кориснички интерфејс, Аутоматизми, Изглед корисничког интерфејса у складу са

	навикама (менталним моделима) корисника
<ul style="list-style-type: none"> <li>Лакоћа учења коришћења апликације</li> </ul>	Минимизација потребних корака у коришћењу апликације за решавање неког пословног задатка
<ul style="list-style-type: none"> <li>Прилагодљивост различитим оперативним окружењима</li> </ul>	Технолошко решење које може бити коришћено на различитим оперативним системима рачунара и мобилних уређаја, у оквиру различитих web browser-a
<ul style="list-style-type: none"> <li>Интероперабилност са другим апликацијама</li> </ul>	<p>Могућност да коегзистира са другим апликацијама</p> <p>Могућност да учитава податке из других апликација</p> <p>Могућност да доставља податке другим апликацијама</p> <p>Могућност да аутоматски интегрише свој рад са радом у оквиру других апликација разменом података</p>
<ul style="list-style-type: none"> <li>Могућност прилагођавања различитим захтевима – персонализација</li> </ul>	Измене структуре података, изгледа извештаја, изгледа корисничког интерфејса
<ul style="list-style-type: none"> <li>Проширивост</li> </ul>	Могућност додавања нових функционалности уз интеграцију са постојећим
<ul style="list-style-type: none"> <li>Ергономска погодност</li> </ul>	Усклађеност боја са потребама корисника, величина фонта, распоред контрола, подршка за различите уређаје за улаз (тастатура, миш, читачи картица) и излаз података (екран, штамшач...)
<ul style="list-style-type: none"> <li>Подршка за продуктивност</li> </ul>	Аутоматизми, Минимизација потребних корака у коришћењу апликације за решавање неког пословног задатка
<ul style="list-style-type: none"> <li>Управљивост</li> </ul>	Могућност да се врше измене у структури и начину функционисања апликације ради прилагођавања потребама пословања – кастомизација
<ul style="list-style-type: none"> <li>Усклађеност са стандардима</li> </ul>	ИТ, семантички стандарди примене
<ul style="list-style-type: none"> <li>Квалитет програмског кода</li> </ul>	Читљивост, рефакторисање, модуларност, коментари
<ul style="list-style-type: none"> <li>Документовано решење</li> </ul>	Корисничка и техничка документација

### 3.4. МЕТРИЧКИ МОДЕЛИ ЗА ПРОЦЕНУ ТРАЈАЊА И МОНИТОРИНГ УСПЕХА ПРОЦЕСА У ОКВИРУ ДИСТРИБУИРАНОГ АГИЛНОГ СОФТВЕРСКОГ ПРОЈЕКТА

#### 3.4.1. Теоријско-истраживачка основа предложених модела за процену и мониторинг успеха процеса развоја софтвера

У табели 3.4.1.1. представљени су приказани називи предложених метричких модела, методологија или приступ који представља основ за креирање наведеног метричког модела и референца на рад где је наведени метрички модел презентован или су описани његови елементи.

Табела 3.4.1.1. Теоријско-истраживачке основе модела за процену и мониторинг успеха процеса развоја софтвера

МЕТРИЧКИ МОДЕЛ	ПРЕТХОДНА ИСТРАЖИВАЊА	Референца
Мониторинг успешности процеса развоја у оквиру софтверског пројекта	Полазна претпоставка: фактори успеха пројекта у току различитих фаза и за различите учеснике/заинтересоване стране у пројекту се разликују [Hartman&Ashrafi, 2002]	
	Критеријуми успеха: [Atkinson, 1999],[DeLone& McLean, 1992],[Crowston et al, 2003],[Shenhar et al, 2001],[Westhuizen& Fitzgerald, 2005], [Berntsson-Svensson&Aurum, 2006], [Procaccino& Verner, 2006]	-
	Фактори успеха: [Jurison, 1999], [Berntsson-Svensson&Aurum, 2006],[Eveleens& Verhoef, 2010], [Hartman&Ashrafi, 2002]	-
	Ризици: [Jovanovic, 2006] [Boehm, 1991] [Whittaker, 1999] [Eveleens& Verhoef, 2010]	-
	Стратегијски менаџмент: [Atkinson, 1999] Balanced Scorecard приступ стратегијског менаџмента [Kaplan&Norton, 1992] Примена Balanced Scorecard у пројектном менаџменту [Niebecker et al, 2008]	[Kazi et al, JEMC, 2011] [Kazi&Radulovic, MIPRO, 2011] [Kazi et al, SISY, 2012]
	Goal-Question-Metric [Basili& Weiss, 1984]	-
	Агилни приступ [AgileManifesto, 2001] Фактори успеха агилних софтверских пројеката [Chow&Das-Buu, 2008]	-
	Проблеми дистрибуираног развоја софтвера: <ul style="list-style-type: none"> <li>Табела 2.1.1.1.1. (стр. 64) – списак прегледних радова и проблема дистрибуираног развоја софтвера</li> <li>Табела 9.2.1. (прилог) – списак референци истраживања појединачних проблема дистрибуираног развоја софтвера</li> </ul>	[Kazi et al, AIIT, 2014] [Kazi et al, COMENG, 2014]
Модел-базирана процена трајања софтверског пројекта	Процена трајања пројекта на основу карактеристика пројекта пре почетка реализације [Khatibi&Jawawi, 2011],на основу броја елемената софтвера и сложености [Reifer, 2000], Елементи управљања пројектом као део модела процене (не само инжењерске метрике) [Abdel-Hamid& Madnick, 1987]	-
	Процена на основу модела пословних процеса [Vogelzang, 2006][Issa&Rub, 2007][Monsalve et al, 2011]	
	Процена путем USE CASE Points [Nassif, 2012] [Clemmons, 2006]	-
	Процена на основу анализе структуре података: <ul style="list-style-type: none"> <li>* Концептуални модел података [Tan et al, 2009]</li> <li>* Релациони модел података [Mishra et al, 2010]</li> <li>* Објектни модел података [Živković et al, 2005]</li> </ul>	-

### 3.4.2. Метрички модели за мониторинг успешности процеса у оквиру дистрибуираног агилног софтверског пројекта

Метрички модел за мониторинг успешности процеса у оквиру дистрибуираног софтверског пројекта је настао на основу систематизације и интеграције резултата истраживања у области анализе фактора и ризика успеха/неуспеха софтверских пројеката, а посебно дистрибуираних и агилних софтверских пројеката. Анализом претходних истраживања утврђено је да се исте категорије проблема и фактори успеха/неуспеха и ризика једнако јављају у оквиру истраживања у области софтверских пројеката, агилних и дистрибуираних пројеката, с тим да је у оквиру агилних софтверских пројеката нагласак на карактеристикама процеса које се односе на продукцију решења, док је код дистрибуираних софтверских пројеката нагласак на комуникацијама, координацији, интеграцији итд. Дакле, тип методологије и технологије реализације пројекта не утиче суштински на избор врста карактеристика у мониторингу, већ само доприноси издвајању појединих карактеристика као значајнијих.

У складу са ставом [Hartman&Ashrafi, 2002] да се разликују фактори успеха и карактеристике од значаја у мониторингу софтверског пројекта у зависности од учесника или заинтересоване стране, у оквиру предлога метричког модела за мониторинг успешности софтверског пројекта разликујемо 3 нивоа разматрања успеха:

- стратешки (односи се на поглед вишег менаџмента организације на пројекат),
- тактички (односи се на поглед пројектног менаџмента на пројекат)
- оперативни (односи се на поглед чланова развојног тима на пројекат).

На основу базичних елемената из теорије система (систем, окружење, елементи, процеси, улази, излази), у наставку је дат предлог општег модела за дефинисање критеријума за мониторинг успеха активности на реализацији пројекта општег типа који се одвија у оквиру организационог система. У предложеном моделу разликује се стратешки, тактички и оперативни скуп критеријума успешности који се односе на окружење (клијент, добављач, надзорни органи) и интерни систем, односно базичне групе процеса – основна делатност (доприноси сврси постојања), помоћна делатност (бави се ресурсима који су потребни основној делатности) и управљачка делатност.

Табела 3.4.2.1. Општи модел за дефинисање критеријума за мониторинг успеха активности на реализацији пројекта који се одвија у оквиру организационог система

ЦИЉЕВИ	ПРОШЛОСТ – реализовати уговорено САДАШЊНОСТ – ефикасно/ефективно функционисање БУДУЋНОСТ – раст, развој и учење					
	ОКРУЖЕЊЕ			ИНТЕРНО – СИСТЕМ		
	КЛИЈЕНТ	ДОБАВЉАЧ	НАДЗОРНИ ОРГАНИ	ОСНОВНА делатност	ПОМОЋНА делатност	УПРАВЉАЧКА делатност
СТРАТЕГИЈСКИ ЦИЉ Ниво организације						
ТАКТИЧКИ ЦИЉ Ниво пројекта						
ОПЕРАТИВНИ ЦИЉ Ниво задатка						

На основу предложеног општег приступа стратешком, тактичком и оперативном управљању на основу одговарајућег скупа критеријума успеха и метрика, у циљу избора критеријума и метрика, може се усвојити:

- Стратешки ниво - Balanced Scorecard методологија као основ,

- Тактички ниво - методологија пројектног менаџмента базирану на PMBOK,
- Оперативни ниво - методологија агилног приступа развоју софтвера.

Наравно, у току процеса реализације софтверског пројекта нису у сваком тренутку сви нивои и категорије/области подједнако важни и заступљени, у складу са ставовима [Hartman&Ashrafi, 2002]. У табели 3.4.2.2. предложен је општи метрички модел за мониторинг успеха процеса софтверског пројекта. Иако је Balanced Scorecard оквир који може бити проширен или модификован ради прилагођавања конкретне проблематице, у предложеном метричком моделу користе се изворне четири категорије критеријума. PMBOK важећи стандард односи се на 10 области, а у приказу оперативног нивоа заснованог на агилном приступу, у метричком моделу су у табели 3.4.2.2. приказане само кључне речи из основних вредности и принципа Agile Manifesto.

Табела 3.4.2.2. Општи метрички модел и методолошка заснованост општег метричког модела за мониторинг успеха процеса софтверског пројекта

НИВО / методологија	КАТЕГОРИЈА - области
СТРАТЕШКИ (Balanced Scorecard)	S1 - Корисник S2 - Интерни процес S3 - Финансије S4 - Учење и развој
ТАКТИЧКИ (Project Management - PMBOK)	T1- Управљање интеграцијом у пројекту T2 - Управљање обухватом пројекта T3 - Управљање временом у пројекту T4 - Управљање трошковима пројекта T5 - Управљање квалитетом у пројекту T6 - Управљање људским ресурсима пројекта T7 - Управљање комуникацијама у пројекту T8 - Управљање ризиком у пројекту T9 - Управљање набавкама у пројекту T10 - Управљање заинтересованим странама пројекта
ОПЕРАТИВНИ (Агилни)	Вредности: AV1 - Појединци и интеракције AV2 - Софтвер који функционише AV3 - Сарадња са клијентом AV4 - Одговарати на промене ----- Принципи: AP1 – адовољити клијента кроз ране и континуиране испоруке корисног софтвера. AP2 – измене захтева AP3 – Испоручити радни софтвер често AP4 - Људи из пословног окружења и људи у развоју морају радити заједно AP5 – мотивисани појединци AP6 - конверзација „лицем у лице“. AP7 - Софтвер који функционише AP8 –одрживи развој. AP9 –константну комуникацију бесконачно. AP10 –техничку изврност и добар дизајн AP11 – Једноставност посла, AP12 –самоорганизујућих тимова. AP13 –тим настоји бити ефективнији,усклађује понашање

На основу претходног приказа метричког модела, може се видети да су поједине карактеристике различитих нивоа разматрања повезане у смислу да реализацијом тактичких и оперативних циљева пројекта реализујемо и стратешке циљеве организације.

Табела 3.4.2.3. Матрица повезаности стратешких, тактичких и оперативних компоненти управљања софтверским пројектом на основу општег метричког модела за мониторинг софтверског процеса

СТРАТЕШКА КАРАКТЕРИСТИКА	ТАКТИЧКА И ОПЕРАТИВНА КАРАКТЕРИСТИКА
S1 – корисник	T10,AV1,AV3,AV4 AV2, AP7, AP1, AP4
S2 – интерни процес	T1, AP3 T2, AV4, AP2, AP11 T3 T5, AP8,AP10 T6, AV1,AP5, AP12 T7, AP4, AP6, AP9 T8 T9
S3 – финансије	T4 T9
S4 – учење и развој	AP13

У оквиру прилога (одељак 9.10) дата је анализа реализованих истраживања у области успеха, неуспеха и ризика реализације софтверског пројекта, дистрибуираног и агилног софтверског пројекта категоризацијом одговарајућих фактора према категоријама које су дефинисане претходним метричким моделом (табела 3.4.2.2. Општи метрички модел).

У наставку (табела 3.4.2.4. и 3.4.2.5) предложен је метрички модел за вредновање процеса извршавања софтверског пројекта који за основ има Balanced Scorecard оквир. У оквиру наведеног модела дефинисане су категорије, перспективе, циљеви, мере и извори података за реализацију мерења.

Табела 3.4.2.4. Проширени метрички модел за мониторинг софтверског пројекта – категорија процеса (проширен у односу на [Kazi et al, SISY, 2012])

Категорија	Перспектива	Циљ	Мера	Извор података за мерење	
Процес	Квалитет чланова тима	Избор квалитетних чланова тима	Позиција члана тима на ранг листи	Ранг листа чланова тима	
		Понашање члана тима позитивно оцењено	Број поена на оцењивању понашања у упитнику	Резултати упитника	
	Предлог пројекта	Предлог пројекта у складу са захтевима	Број захтева за изменама предлога пројекта	Регистар пројектних предлога Регистар промена	
	Радни задаци	Оствариви задаци у предлогу пројекта	Процент имплементираних софтверских функција	Процент имплементираних подршке елементима структуре података (табеле у бази података, XML)	Резултати задатака
			Постављање остваривих задатака	Број проблема по задатку	Резултати задатака, Регистар проблема
		Задаци завршени у предвиђеном времену и са предвиђеним квалитетом	Број задатака имплементираних дневно-недељно-месечно	Резултати задатака	
			Број задатака реализованих у специфицираном времену	Резултати задатака	
			Број задатака завршених са предвиђеним квалитетом	Резултати задатака	
			Број функција имплементираних дневно	Дневник рада	
	Број имплементираних случајева коришћења	Резултати задатка			

			Број софтверских функција имплементираних са захтеваним квалитетом	Резултати задатка	
			Број табела у бази података покривених софтверском подршком	Резултати задатка	
			Број покривених „user stories“ софтверском подршком		
		Не преоптерећивати чланове тима задацима	Број задатака по члану тима	Архива доделе задатака	
	Проблеми	Минимум проблема		Број проблема	Архива проблема
				Број проблема по задатку	
				Број проблема по члану тима	
				Број нерешених проблема	
	Комуникација	Минимум комуникације		Број порука	Архива комуникације
		Максимум јасности задатка		Број порука у вези разјашњавања задатка	
Максимална кохезија тима – узајамна помоћ			Број проблема решених захваљујући порукама између чланова тима		

Табела 3.4.2.5. Метрички модел за вредновање софтверског пројекта – категорије производа, финансија, учења и напредовања [Kazi et al, SISY, 2012]

Категорија	Перспектива	Циљ	Мера	Извор података за мерење
Производ	Парцијални резултати	Парцилани резултати у складу предлогом пројекта	Број захтева за изменама у вези парцијалних резултата	Вредновање задатка
		Парцијални резултати у складу захтевима корисника и очекивањима	Број захтева за изменама у вези парцијалних резултата	Регистар промена
	Интегрисан и производ	Сви модули да буду интегрисани	Процент модула који су интегрисани	Регистар конфигурације
		Комплетна функционалност реализована	Процент реализованих софтверских функција у односу на захтевани број софтверских функција	Предлог пројекта Резултати задатка
		Минимум грешака	Број тест случајева извршених Број грешака решених	Регистар тестирања
	Финансије	Људи	Минимум ангажовања радне снаге	Број људи ангажовано
Минимум радног оптерећивања по задатку			Број радних сати радног ангамана по задатку	Дневник рада
Минимум трошкова путовања			Адреса становања члана тима (путни трошкови)	Ранг листа чланова тима
Максимална продуктивност			Старост члана тима	Ранг листа чланова тима
Опрема		Минимално коришћење опреме	Број сати рада рачунара и других рачунарских уређаја	Регистар коришћења ресурса
Материјал		Минимум коришћења материјала	Број и тип материјала који је коришћен (папир, DVD/CD) по задатку	Регистар коришћења ресурса



	Трошкови комуникације	Минимум комуницирања	Број порука, број телефонских позива, трајање позива	Архива комуникација
Учење	Проблеми	Минимум понављања истих проблема	Број понављајућих проблема (поредећи према претходним задацима, члановима тима или пројектима)	Архива проблема
	Вештине понашања	Најбоље вештине понашања чланова тима	Број поена остварених на упитнику о понашању	Запис о понашању чланова тима
	Лекције на основу искуства	Максимум научених лекција	Број лекција у архиви лекција	Запис о лекцијама
Напредовање (Раст)	Чланови тима	Максимизирање позиције члана тима на ранг листи	Позиција члана тима на ранг листи – упоређивање промена током времена	Ранг листа чланова тима
		Максималан број успешно завршених пројектних задатака	Број успешно реализованих задатака од стране чланова тима	Ранг листа чланова тима
		Максималан број успешно завршених пројеката	Број успешно завршених пројеката од стране чланова тима	Ранг листа чланова тима

Табела 3.4.2.6. представља метрички модел за оперативни ниво планирања и праћења успеха реализације софтверског пројекта у области развоја информационих система. Кључни елементи за мониторинг представљају: број реализованих софтверских функција за подршку случајевима коришћења, број корисникових захтева (или „user stories“), број покривених табела података из базе података. Предложени метрички модел заснива се на кључним карактеристикама агилног приступа – оријентацији на софтвер који функционише у складу са захтевима корисника.

Табела 3.4.2.6. Метрички модел за оперативни ниво планирања и праћења успеха реализације софтверског пројекта у области развоја информационих система

<b>ДАТУМ:</b>	Дато / Реализовано / Планирано / Задато				
<b>ФАЗЕ</b>	<b>КОНЦЕПТ РЕШЕЊА</b>			<b>ЗАДАЦИ</b>	
ЕЛЕМЕНТ	примитивних процеса	Екранских форми	табела у бази података	„user stories“	задатака
	софтверских функција	Извештаја	Други формати фајлова	спецификација захтева	
	случајева коришћења	Процедура обраде			
	Профила корисника				
	ОСТАЛО				
	Тестирање	Документовање	Инсталација	Обука	Верификација од стране корисника /менаџмента

Идеја примене наведеног модела је у нумеричком представљању елемената спецификације и решења, уз пратеће текстуалне описе конкретних елемената којима се даје аналитички и детаљни приказ датих, планираних и реализованих елемената. Предложеним приступом је предвиђено да се у одређеним интервалима контролних тачака (у оквиру SCRUM – сваког дана) бележи улаз од стране корисника – спецификације захтева, „user stories“ или верификације, а затим се дефинишу задаци у оквиру сваког сегмента у оквиру концепта решења. Укупан број задатака за све сегменте концепта решења се уписује у оквиру ћелије „задатака“.

Погоднија форма за примену мониторинга процеса реализације софтвера у развоју информационих система дата је у виду табеле другачијег изгледа:

Датум:											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Наведене форме табеларног приказа резултата могу се користити у појединачном приказу (резултати активности само у оквиру наведеног датума) и кумулативном приказу где се виде резултати свих дотадашњих активности и непокривене области. Пример примене наведених модела дат је у прилогу.

### 3.4.3. Метрички модели за модел-базирану процену трајања софтверског пројекта у области развоја информационог система

#### 3.4.3.1. Анализа недостатака постојећих модел-базираних приступа процени величине, трајања и трошкова реализације софтверског пројекта у развоју информационог система

Анализом доступне литературе, постојећи приступи модел-базираној процени величине, трајања и трошкова реализације софтверског пројекта укључују приступе и методе засноване на анализи пословних процеса, UML модела, а посебно модела случаја коришћења, као и модела података (концептуалног, објектног и релационог). У наставку ће бити дат кратак преглед метода у оквиру наведених приступа:

Рад [Issa&Rub, 2007]: описан је приступ креирању модела пословних процеса коришћењем посебне RAD (Role Activity Diagram) нотације, издвајање пословних процеса који се могу аутоматизовати и на основу њих издвајање случаја коришћења, израчунавање потребног ангажовања запослених („effort“) на основу броја случајева коришћења и укључивање других параметара пројекта ради креирања студије изводљивости.

- Позитивни аспекти рада: приступ заснованости на пословним процесима, приступ извођењу случаја коришћења на основу могућности аутоматизације, формуле за израчунавање укупног ангажовања на основу броја случаја коришћења
- Недостаци рада: недовољна прецизност формуле за израчунавање ангажовања – нема разликовања сложености реализације случаја коришћења, није дефинисан начин пресликавања пословних процеса у случајеве коришћења (поистовећују се у извесној мери), нису јасно издвојене софтверске функције (користи се концепт случаја коришћења)

Рад [Vogelzang, 2006]: описује COSMIC-FFP као универзални метод (базиран на стандарду ISO/IEC 14143) за мерење функционалне величине софтвера који је једнако примењив за разне врсте софтвера. Метод се заснива на приступу декомпозиције функционалних захтева корисника на атомарне акције „покретања података“ („data movement“ са подврстама: entry – улаз података од стране корисника, write - снимање, read - читање, exit – приказ података кориснику) и „манипулације подацима“ („data manipulations“). Свака активност покретања података добија вредност 1 Cfsu (COSMIC functional sizing unit).

- Позитивни аспекти рада: Формула за израчунавање времена потребног да се изврши испорука софтвера, формула за израчунавање трошкова
- Негативни аспекти рада: Не одваја јасно функционално-зависне и технолошко-имплементационо зависне елементе и утицаје у формули за процену трајања,

Вредности параметара у формулама су фиксне и добијене су на основу претходних искустава (Степен = 0.2 за једну продукциону линију или 0.37 за две), само 2 продукционе линије, не узима техничке и пројектне услове за израчунавање, већ само функционални аспект и број производних линија, поједностављена формула за израчунавање трошкова на основу фиксног дела и дела који зависи од броја функција софтвера. Није јасна улога и величина  $C_{fsu}$  за део манипулације подацима – обраду података.

Рад [Monsalve et al, 2011] описује BPMN (Business Process Modeling Notation) – стандардну нотацију OMG, Qualigram нотацију која повезује стратешки ниво, процесни и радни ниво активности, као и COSMIC приступ функционалном мерењу величине апликације. Описује правила и упутства правилног моделовања у наведеним нотацијама, врши компарацију правила и издвајање правила која су независна од типа нотације, предлаже приступ прилагођавању нотација моделовања пословних процеса како би биле погодне за COSMIC мерење.

- Предности рада: детаљна хеуристичка упутства за правилно моделовање пословних процеса и уопштење правила независна од нотације, пример примене мерења коришћењем COSMIC приступа
- Недостаци рада: приказ примера мерења са једноставним софтверским операцијама које се односе само на „померања“ података, али не и на манипулацију (процесирање).

Поред појединачних, заједнички недостаци свих радова су у фокусирању само на пословне процесе унутар модела пословних процеса, док се остали елементи модела пословних процеса не узимају у обзир, а такође се не узима у обзир ни сложеност пословних процеса.

У оквиру дела који се односи на дизајн софтверског решења, уобичајено је креирање одговарајућих модела дизајна. У области функционалног дизајна, најчешће се користе дијаграми случајева коришћења, а у области дизајна података најчешће се креирају концептуални, релациони и објектни модели података. Такође, често се поред анализе модела у оквиру формула за израчунавање трајања пројекта укључују и имплементационо-технолошки фактори, односно карактеристике.

Анализом елемената дијаграма случаја коришћења у циљу процене трајања софтверског пројекта [Nassif, 2012], којим се специфицира функционални аспект софтвера, као и у комбинацији тих података са подацима карактеристика техничко-имплементационог развојног окружења, дефинисани су тзв. USE CASE Points у оквиру [Clemmons, 2006].

- Позитивни аспект рада [Clemmons, 2006]: дефинисање формуле за израчунавање USE CASE Points на основу сложености случаја коришћења (броја акција у оквиру свих сценарија и броја класа које се користе у имплементацији наведеног случаја коришћења) и сложености актора (типа – други софтверски систем или корисник), а такође и на основу фактора техничке комплексности и комплексности развојног окружења (заправо елемената управљања пројектом – ресурси итд).
- Недостаци рада: у оквиру фактора техничке комплексности и комплексности развојног окружења, нису јасно дефинисане подгрупе фактора. Типови случајева коришћења подељени су у 3 групе према сложености и тежински фактори за одговарајуће групе имају фиксне вредности.

У оквиру предлога процене трајања софтверског пројекта на основу модела података, у раду [Tan et al, 2009] дат је предлог процене могуће величине софтверског пројекта (изражене у KLOC – број хиљада линија кода) на основу анализе броја елемената концептуалног модела података, у раду [Mishra et al, 2010] разматрају се метрике за процену ангажовања („effort“) на основу структуре релационе базе података, а у раду [Živković et al, 2005] описана је компарација приступа процени FPA (Functional Point Analysis) величине пројекта на основу елемената објектно-орјентисаних модела.

### 3.4.3.2. Метрички модел за процену трајања пројекта будућег софтверског решења на основу модела пословних процеса

Метрички модел за процену трајања пројекта будућег софтверског решења на основу модела пословних процеса који је предложен у овом одељку заснива се на анализи и вредновању карактеристика елемената модела пословног процеса, као и на пресликавању пословних процеса у потенцијалне софтверске функције. Предложени метрички модел има карактеристике којима се превазилазе недостаци раније представљених приступа:

1. Јасно су разграничени пословни процеси и софтверске функције, а не поистовећују се пословни процеси са случајевима коришћења
2. Заснован је на пословним процесима, за које се одређују основне карактеристике:
  - А) да ли може да се аутоматизује
  - Б) да ли је типа улаза, процесирања или излаза у односу на окружење
  - В) ниво сложености процеса – одређен је бројем токова података и складишта података са којима је процес повезан
3. Дефинисана су правила пресликавања елемената модела пословних процеса у елементе дизајна софтверске апликације, на основу чега се може добити иницијални скуп софтверских функција и одредити функционална величина и трајање софтверског пројекта
4. Заснован је и на свим елементима бизнис процес модела и анализи њихових карактеристика, не само на пословним процесима
5. Параметри у формулама за израчунавање трајања нису фиксне величине
6. Не разматрају се елементи имплементације, као што је број број потенцијалних учесника у имплементацији
7. Узимају се у обзир сви типови софтверских функција, а не само базичне функције COSMIC „померања“ података.
8. У оквиру анализе пословних процеса, узимају се у даљој обради само процеси који могу да се аутоматизују.
9. У оквиру анализе пословних процеса, узимају се у даљој обради само процеси који су у оквиру граница обухвата пројекта, док се екстерни процеси не обрађују.

У наставку је дат предлог метричког модела за процену трајања софтверског пројекта на основу анализе карактеристика модела пословних процеса и правила пресликавања елемената модела пословних процеса у елементе дизајна софтверског решења.

Табела 3.4.3.1.1. Метрички модел са предлогом тежинских фактора за процену трајања пројекта будућег софтверског решења на основу модела пословних процеса

Бизнис процес модел			
ПРОЦЕС	ТОК ПОДАТАКА	СКЛАДИШТЕ ПОДАТАКА	ОРГАНИЗАЦИОНА ЈЕДИНИЦА или СИСТЕМ ИЗ ОКРУЖЕЊА
Да ли је у оквиру граница обухвата или је из окружења ----- Да ли може да се аутоматизује ----- УЛАЗНИ – коеф. Тежине: 1 ИЗЛАЗНИ - коеф. Тежине: 2 ОБРАДА – коеф. Тежине: 3	број елементарних података, "do 10" – коефицијент 1 ">10" – коефицијент 2	број елементарних података, "do 10" – коефицијент 1 ">10" – коефицијент 2	Sadrzi: Број процеса: Број складишта: Број токова података:
Број токова података са којима је процес повезан 1 – коефицијент 1 ">1" - коефицијент 2	повезаност елементарних података – синтаксна анализа jednostavan - 1 podstrukture 1:M -2	повезаност елементарних података – синтаксна анализа jednostavan - 1 podstrukture 1:M -2	Tip: Интерна (унутар организације) Екстерна (ван организације, ако припада обухвату)

			пројекта)
Број складишта података са којима је процес повезан 1 – коефицијент 1 ">1" – коефицијент 2			

Правила пресликавања елемената пословних процеса у елементе дизајна софтверске апликације [Kazi et al, АИТ, 2012a]:

- процеси из модела пословних процеса пресликавају се у софтверске функције
- токови података се пресликавају у колекције података приликом уноса, излаза и записа података
- складишта података се декомпонују у табеле базе података [Kazi et al, Metalurgia, 2012]
- организационе јединице се пресликавају у профиле корисника

У наставку ће бити описан приступ процени трајања софтверског пројекта у делу реализације софтвера, односно само имплементације, без тестирања, обуке, инсталације и других пратећих активности у развоју софтвера. С обзиром да се ради о специфичној врсти софтвера (пословно орјентисан софтвер, односно софтвер развоја информационог система), време потребно за реализацију софтвера је може се израчунати на основу основног приступа:

ВРЕМЕ ЗА ИЗРАДУ БАЗЕ ПОДАТАКА И ДРУГИХ ФОРМАТА ЗАПИСА +  
ВРЕМЕ ЗА ДИЗАЈН И ПРОГРАМИРАЊЕ ЕКРАНСКИХ ФОРМИ АПЛИКАЦИЈЕ +  
ВРЕМЕ ЗА ПОДРШКУ РАЗЛИЧИТИМ ПРОФИЛИМА КОРИСНИКА (МЕНИЈИ, СЕСИЈЕ ИТД,)

Време за израду базе података и других формата записа може се извести из података о складиштима података, време за дизајн и програмирање екранских форми апликације на основу података о процесима, а време за подршку профилима корисника на основу података о организационим јединицама.

Наравно, приликом процене времена трајања софтверског пројекта треба узети у обзир технолошко-имплементационе могућности (разне варијанте аутоматизације процеса креирања саме апликације) и ограничења, као и особине самог пројекта са организационе стране, а првенствено квалитет и мотивисаност кадрова у реализацији, као и расположивост и квалитет ресурса (алата) и сарадње са корисником. Поједине карактеристике целокупног пројектног окружења развоја могу да допринесу убрзању, али и успоравању укупног процеса реализације софтверског пројекта.

У наставку ће бити представљен део метричког модела који је изражен кроз формулу за израчунавање трајања процеса имплементације софтвера. Наведена формула изражава суштинске зависности трајања од функционалних карактеристика система који се развија, а који се у иницијалној форми може проценити на основу карактеристика модела пословних процеса.

$$MPPT = SPT + PT + OJT$$

Слика 3.4.3.1.1. Формула за израчунавање трајања софтверског пројекта на основу структуре модела пословних процеса

При чему је:

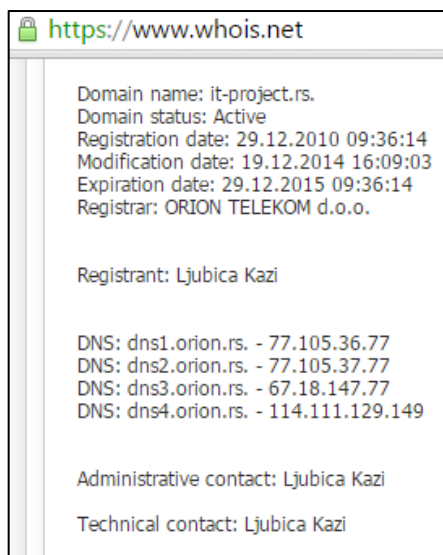
MPPT – трајање процеса имплементације софтвера на основу анализе модела пословних процеса  
SPT - трајање процеса имплементације дела софтвера који се односи на имплементацију структуре података, на основу на основу анализе складишта података  
PT - трајање процеса имплементације дела софтвера који се односи на на имплементацију екранских форми апликације, на основу анализе карактеристика процеса  
OJT - трајање процеса имплементације дела софтвера који се односи на на имплементацију подршке профилима корисника, на основу анализе карактеристика организационих јединица

## 4. ОПИС ИМПЛЕМЕНТАЦИЈЕ ПОЧЕТНОГ ПРОТОТИПА

### 4.1. Основни поступци имплементације прототипа

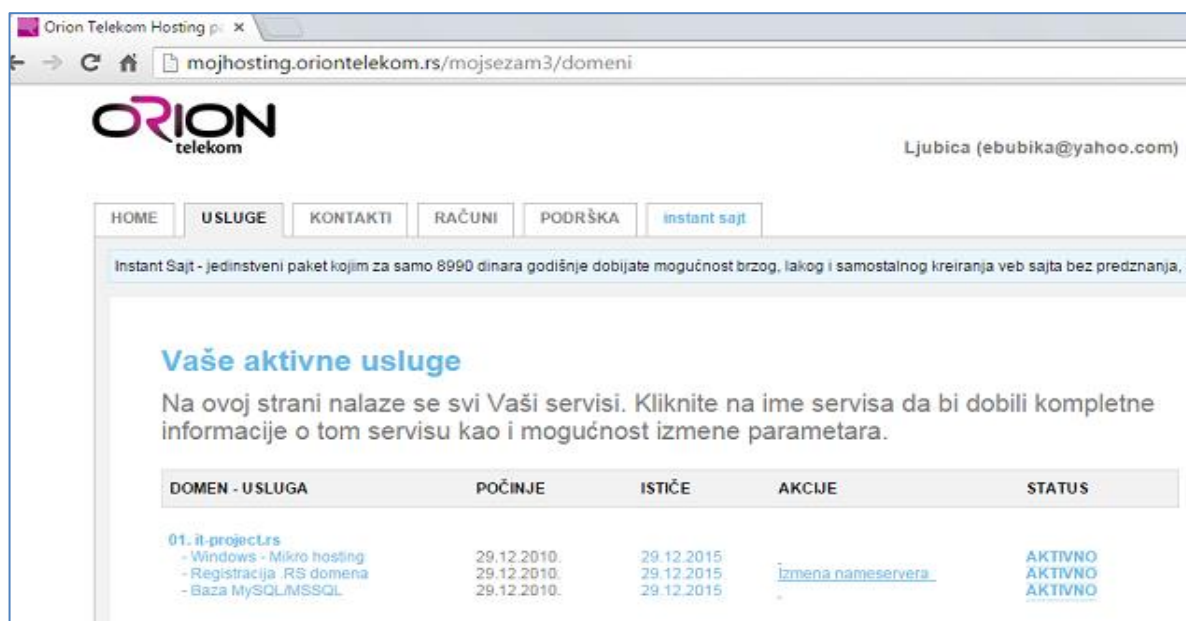
#### 4.1.1. Hosting и администрација web сајта

ASPX web апликација је реализована у оквиру MS Visual Studio радног окружења. Процес је настављен регистрацијом web домена [www.it-project.rs](http://www.it-project.rs) код организације ORION TELEKOM.



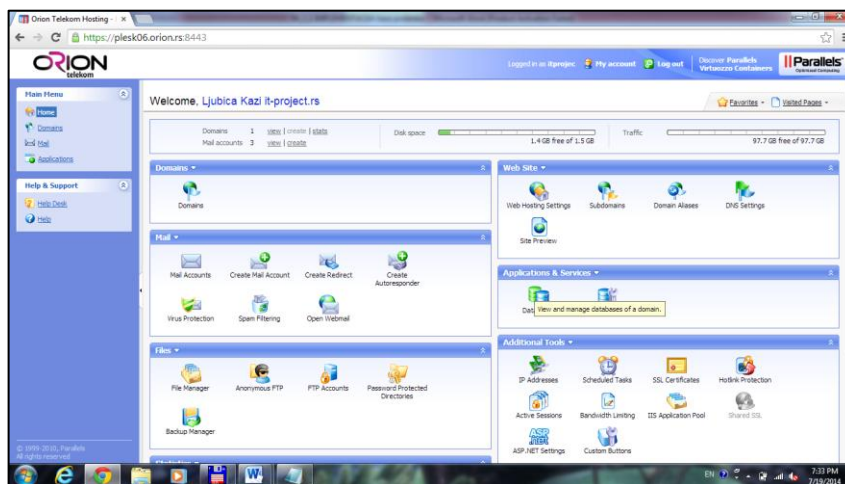
Слика 4.1.1.1. Подаци о регистрацији домена [www.it-project.rs](http://www.it-project.rs), доступни на сајту [www.whois.net](http://www.whois.net)

Истовремено са закупом домена, изабрана је и услуга хостинга web апликације у Microsoft пакету услуга (MS ASPX/ MS SQL Server) такође код организације ORION TELEKOM.



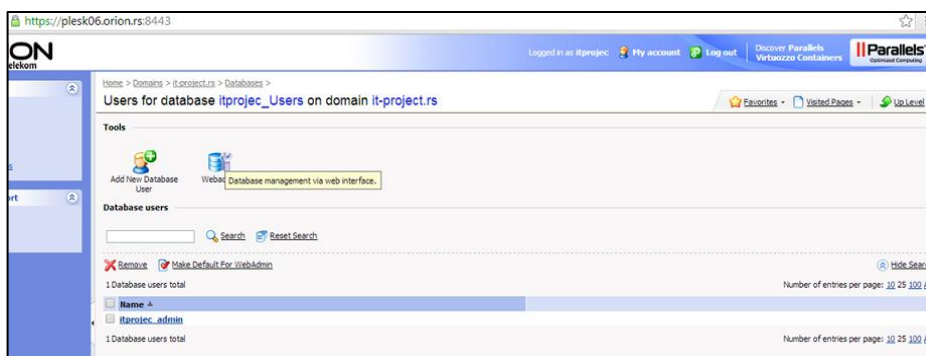
Слика 4.1.1.2. Подаци о хостингу web апликације домена [www.it-project.rs](http://www.it-project.rs)

У оквиру on-line подршке Orion Telekom hosting компаније, након пријављивања путем [www.it-project.rs/plesk](http://www.it-project.rs/plesk) странице, добијамо административни панел који представља подршку за рад са фајловима апликације и базом података.



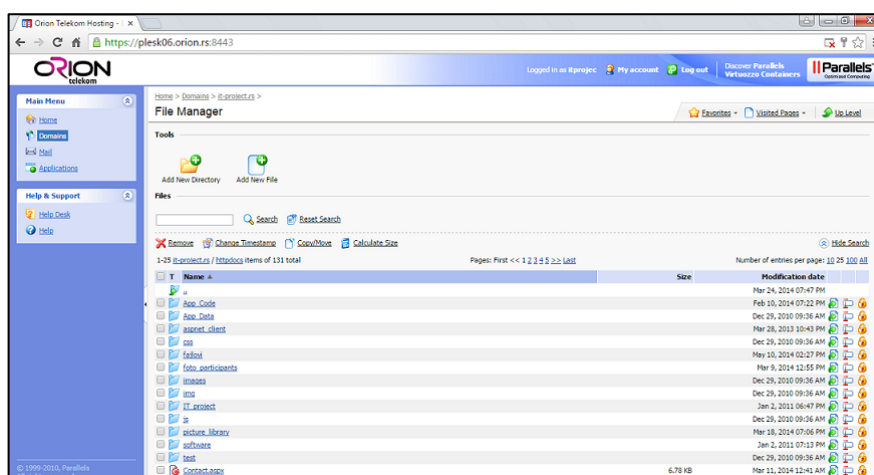
Слика 4.1.1.3. Административни plesk панел web апликације домена www.it-project.rs

За потребе апликације, реализована је MS SQL Server база података, користећи опције у оквиру plesk администраторског модула (Слика 4).



Слика 4.1.1.4. Опција за рад са базом података у оквиру одељка plesk администрације

Коришћењем опције File Manager (Слика 5.), урађен је upload појединачних фајлова реализоване ASPX апликације.

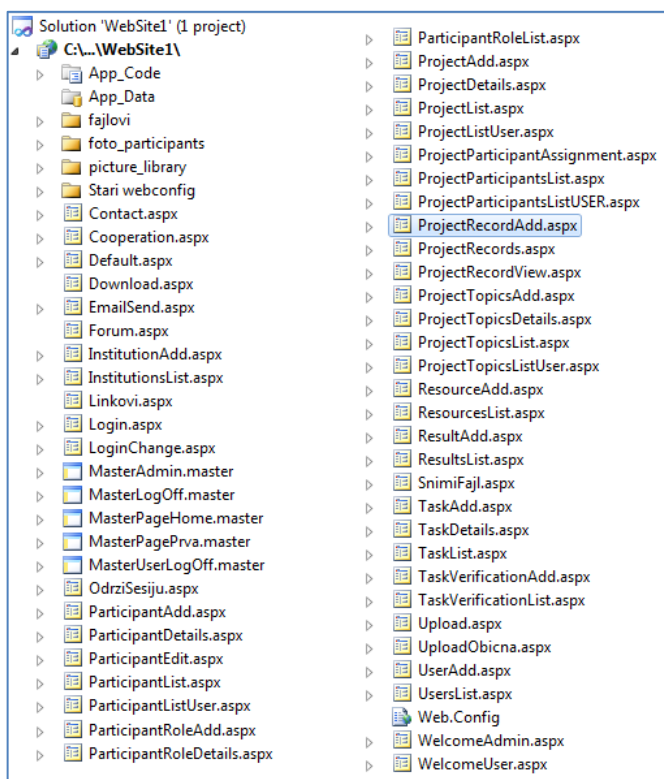


Слика 4.1.1.5. Опције File Manager-а за постављање фајлова апликације на web server

Коришћењем опције FTP (File Transfer Protocol) из програма Total Commander реализован је upload већег броја фајлова реализоване ASPX апликације.

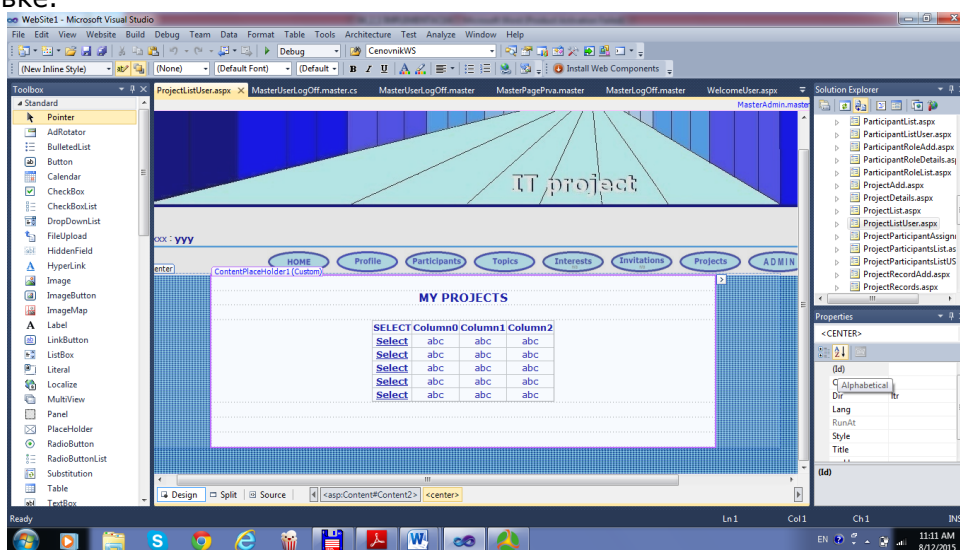
#### 4.1.2. Креирање web апликације у развојном окружењу

У оквиру развојног окружења Visual Studio .NET креирана је ASPX апликација. Слика 4.1.2.1. приказује списак свих фајлова ASPX апликације.



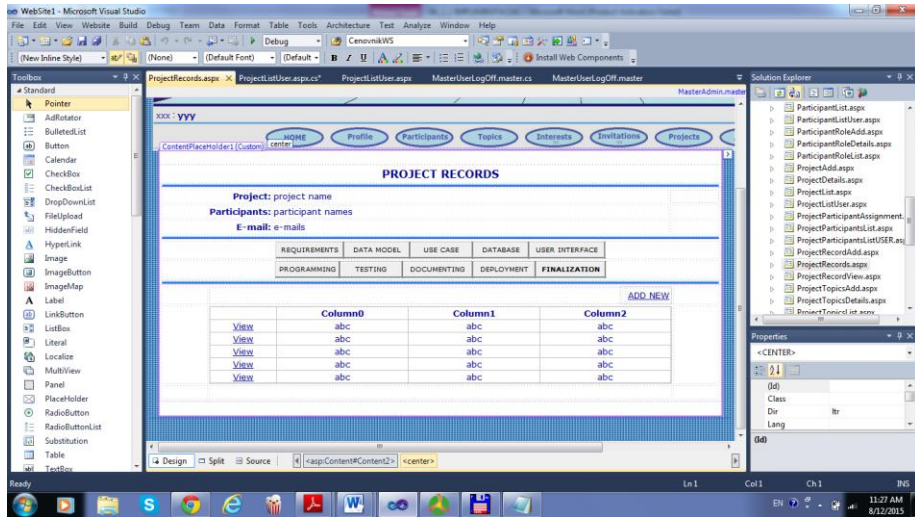
Слика 4.1.2.1. Списак фајлова прототипа апликације www.it-project.rs

Слика 4.1.2.2. приказује web форму (у дизајн режиму) са списком пројеката за пријављеног корисника, слика 3. приказује форму са списком радних ставки у оквиру одабраног пројекта, слика 4. приказује форму за унос радне ставке, а слика 5. приказује део изворног кода у развојном окружењу који се односи на унос података радне ставке.

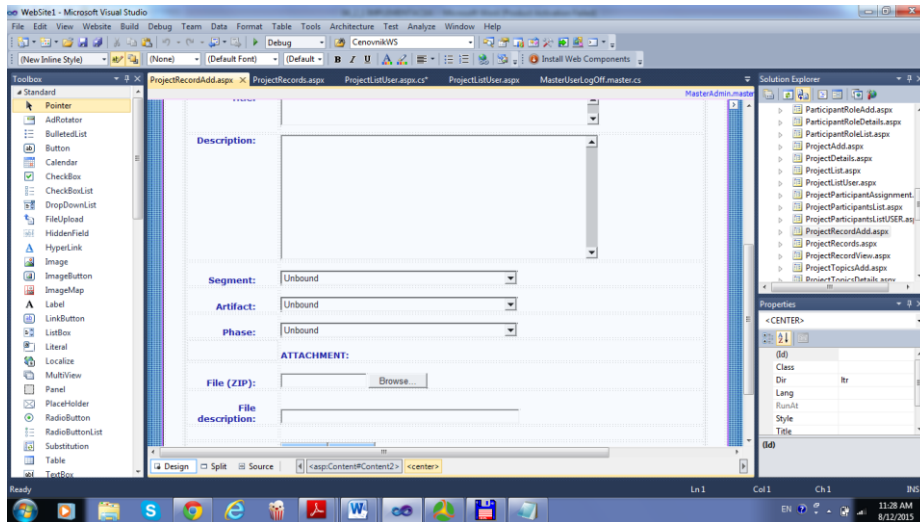


Слика 4.1.2.2. Форма са списком пројеката у дизајн режиму у оквиру развојног окружења

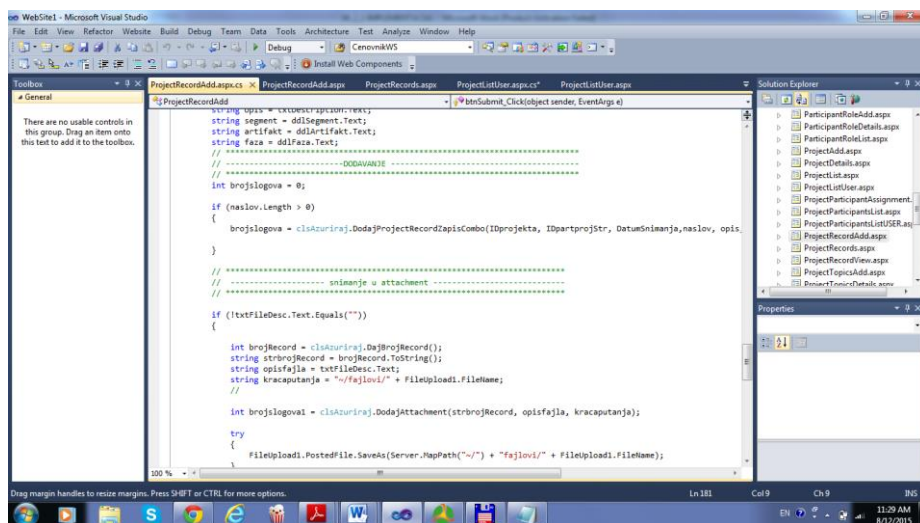




Слика 4.1.2.3. Форма са списком радних ставки у оквиру изабраног пројекта, у дизајн режиму у оквиру развојног окружења



Слика 4.1.2.4. Форма за унос радне ставке, у дизајн режиму у оквиру развојног окружења

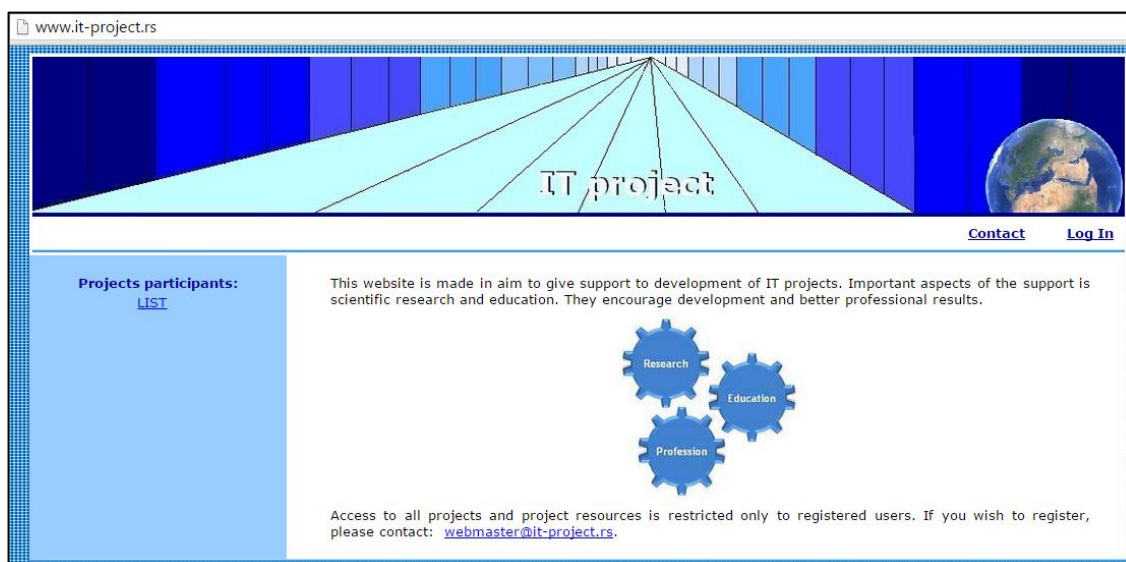


Слика 4.1.2.5. Изворни код за унос података радне ставке, у развојном окружењу

## 4.2. Карактеристике имплементираних прототипа

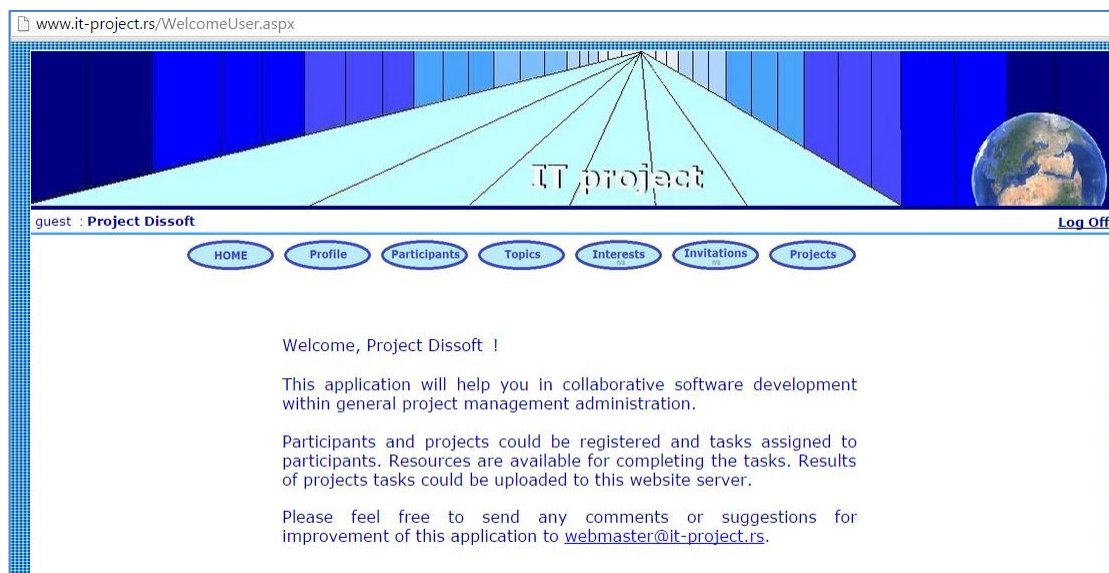
### 4.2.1. Опис карактеристика имплементираних прототипа

Функционалне карактеристике имплементираних прототипа система, који се састоји из ASPX web апликације и MS SQL Server базе података, а описан је у [Kazi et al, ITRO, 2014].



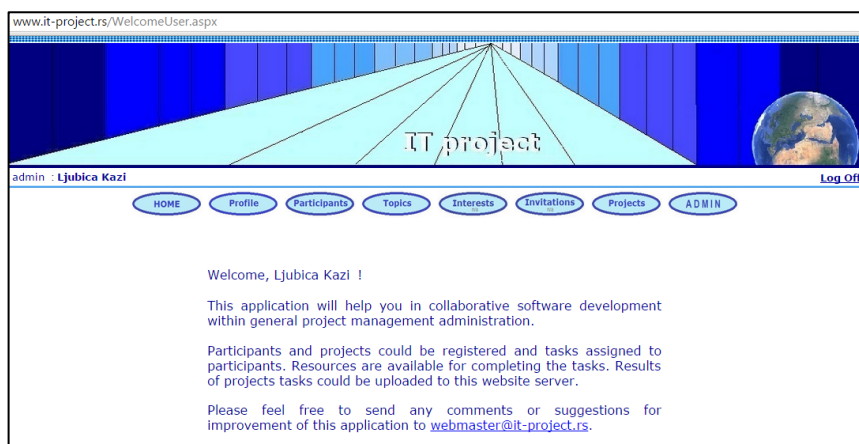
Слика 4.2.1.1. Почетна страница web апликације домена www.it-project.rs

Након пријављивања на систем (корисничко име: „dissoft@it-project.rs“, шифра: „ABC“) добијамо почетни екран са менијем апликације, као што је приказано на слици 4.2.1.2..



Слика 4.2.1.2. Почетна страница web апликације www.it-project.rs за рад са менијем за корисника

Након пријављивања корисника са статусом администратора, добијамо додатну ставку менија „ADMIN“ (Слика 4.2.1.3.).



Слика 4.2.1.3. Почетна страница web апликације www.it-project.rs за рад са менијем за администратора

Имплементирано решење има следеће могућности и карактеристике:

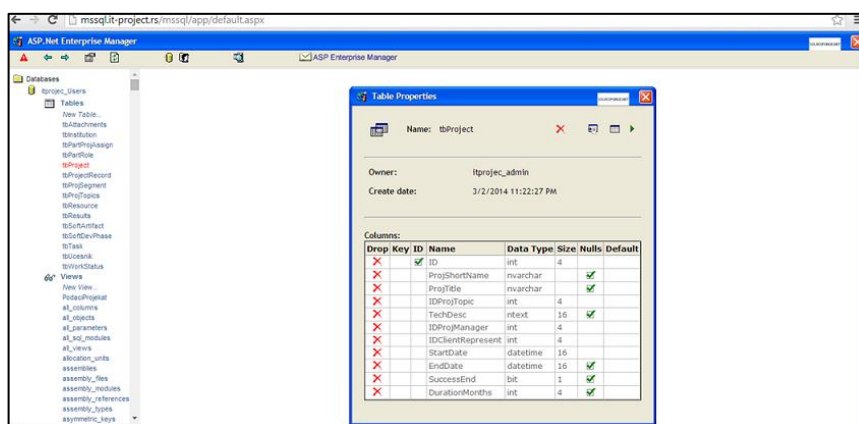
1. Комплетна апликација је реализована тако да је на корисничком интерфејсу коришћен енглески језик, у циљу подршке међународном дистрибуираном развоју
2. Апликација омогућава персонализован рад, што значи да сваки корисник има увид у податке које се односе на њега као корисника, односно као учесника на пројектима. Посебно се разликује 2 профила корисника: обичан корисник („USER“) и администратор „ADMIN“. Корисник који има статус администратора има могућност уноса шифарничких података, потребних за рад целокупне апликације.
3. Подршку за софтверске функције које се односе на унос, табеларни приказ и филтрирање података, као и програмску подршку upload функцији.

Прототип система је реализован у 2 облика:

1. Подршка Waterfall приступу реализацији и управљању софтверским пројектом, са детаљном разрадом свих елемената, активности и података у реализацији софтверског пројекта овог типа (приступ путем „ADMIN“ опције)
2. Подршка Agilном приступу реализацији и управљању софтверским пројектом, са опцијама које омогућавају бржи унос и приказ суштинских података (основни приступ за све кориснике, користећи „PROJECTS“ опцију).

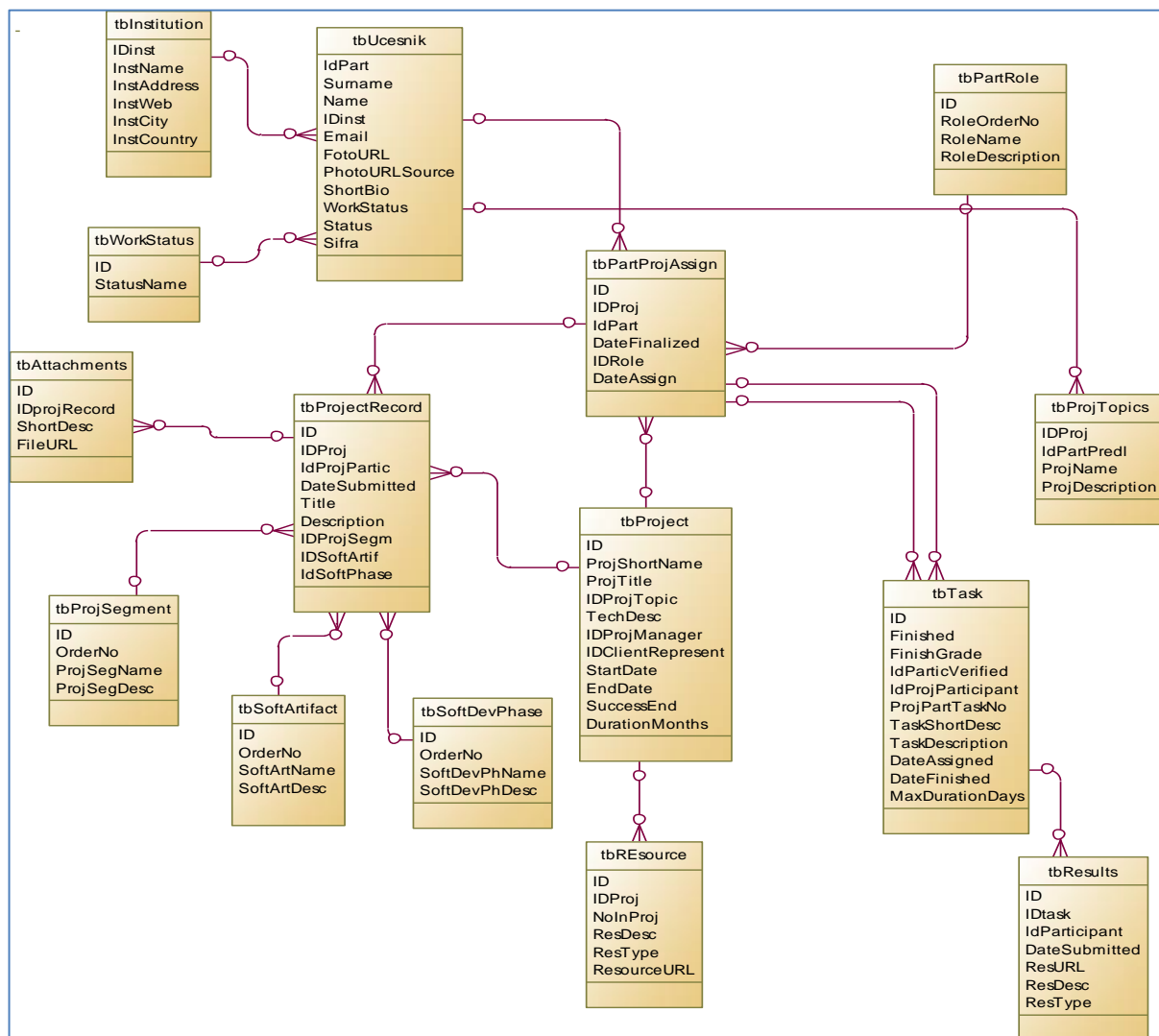
#### 4.2.2. Имплементирана база података

На слици 4.2.2.1. приказан је одељак plesk администрације који се односи на рад са базом података са списком табела базе података. Слика 9. приказује структуру табеле tbProject.



Слика 4.2.2.1. Структура табеле tbProject у оквиру одељка plesk администрације web апликације www.it-project.rs

Структура комплетне базе података приказана је на следећој слици (комбинована нотација ER и релационог модела):



Слика 4.2.2.2. Шема базе података реализоване апликације, приказана комбинованом нотацијом ЕР дијаграма и релационе шеме

У табели 4.2.2.1. приказана је структура табела које чине реализовану базу података.

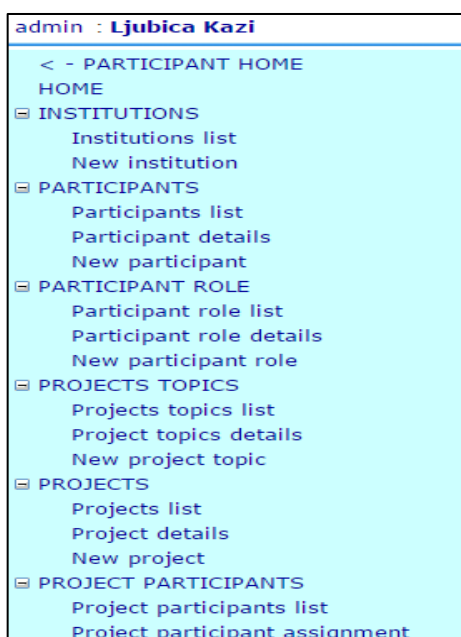
Табела 4.2.2.1. Структура табела базе података реализованог прототипа

<p><b><u>tbAttachments</u></b>                      ID int pk                      IdprojRecord int                      ShortDesc - nvarchar null                      FileURL - nvarchar</p>	<p><b><u>tbInstitution</u></b>                      IDInst int pk                      InstName nvarchar                      InstAddress nvarchar                      InstWeb nvarchar                      InstCity nvarchar                      InstCountry nvarchar</p>	<p><b><u>tbPartProjAssign</u></b>                      ID int pk                      IDProj int                      Idpart int                      DateFinalized date time null                      IDRole int                      DateAssign date time</p>
<p><b><u>tbPartRole</u></b>                      ID int pk                      RoleOrderNo int                      RoleName nvarchar                      RoleDescription nvarchar</p>	<p><b><u>tbProject</u></b>                      ID int pk                      ProjShortName nvarchar null                      ProjTitle nvarchar null                      IDProjTopic int                      TechDesc text null                      IDProjManager int                      IDClientRepresent int                      StartDate datetime                      EndDate datetime null</p>	<p><b><u>tbProjectRecord</u></b>                      ID int pk                      IDproj int                      IdPojPartic int                      DateSubmitted datetime                      Title nvarchar                      Description ntext                      IDProjSegm int                      IDSoftArtif int                      IDSoftPhase int</p>

	SuccessEnd bit null DurationMonths int null	
<b>tbProjSegment</b> ID int pk OrderNo int ProjSegName nvarchar null ProjSegDesc ntext	<b>tbProjTopics</b> IDProj int pk IdpartPredl int null ProjName nvarchar ProjDescription text	<b>tbResource</b> ID int pk IDproj int NoInProj int ResDesc nvarchar ResType nvarchar ResourceURL ntext
<b>tbResults</b> ID int pk IDtask int IdParticipant int DateSubmitted datetime ResURL ntext ResDesc nvarchar ResType nvarchar	<b>tbSoftArtifact</b> ID int pk OrderNo int SoftArtName nvarchar SoftArtDesc ntext	<b>tbSoftDevPhase</b> ID int ok OrderNo int SofDevPhName nvarchar SofDevPhDesc ntext
<b>tbTask</b> ID int pk Finished bit FinishGrade int IdparticVerified int IdProjparticipant int ProjPartTaskNo int null TaskShortDesc nvarchar null TaskDescription ntext DateAssigned datetime DateFinished datetime null MaxDurationDays int	<b>tbUcesnik</b> IDpart int pk Surname nvarchar Name nvarchar Idinst int null Email nvarchar FotoURL nvarchar null PhotoURLSource nvarchar ShortBio text WorkStatus int null Status nvarchar null Sifra nvarchar	<b>tbWorkStatus</b> ID int pk StatusName nvarchar

#### 4.2.3. Административни модул

У оквиру административног дела, омогућен је унос шифарничких података који се користе у оба модалитета апликације.



Слика 4.2.3.1. Ставке менија администрације

У оквиру администрације могуће је уносити податке, приказати детаље према филтеру или табеларно приказати (уз филтер) податке о институцијама, учесницима, улогама учесника, пројектним темама, пројектима и доделити учеснике пројекту.

**LIST OF INSTITUTIONS**

Institution:

INSTITUTION	ADDRESS	CITY	COUNTRY	WEB
Atmiya institute of technology and science	Yogidham Gurukul, Kalawad Road, Rajkot, 360005, India	Rajkot	India	http://atmiya.net/
Irkutsk State Technical University	83, Lermontov street 664074 Irkutsk, Russia	Irkutsk	Russia	http://www.istu.edu/
Parth Systems	3-1293,Zaveri Road,Navsari - 396 445, Gujarat, India	Navsari	India	parthsystems.biz
University of Mumbai, R. D. National College	Linking Road, Bandra (W). Bandra, Mumbai 400 050. Maharashtra, India	Mumbai	India	www.rdnational.edu
University of Novi Sad, Technical faculty "Mihailo Pupin"	Djure Djakovica bb, 23000 Zrenjanin, Serbia	Zrenjanin	Serbia	www.tfzr.uns.ac.rs

Слика 4.2.3.2. Екран - табеларни приказ институција где раде или студирају учесници пројеката

**LIST OF PROJECTS PARTICIPANTS**

Surname:

Institution:

Work status:

NAME	SUR	EMAIL	Status
Djordje	Ala	@tfzr.rs	student
Erika	A	.ac.rs	administrative worker
Madhusudan	Bhatt	mmbhatt@gmail.com	teaching staff
Dragan	Blanusa	draganblanusa@gmail.com	student
Aleksandar	Bugaric	buzga91@hotmail.com	student
Evgeny	Cherkashin	eugene@irnok.net	teaching staff
Naisargee	Chotaliya	gava.chotaliya@gmail.com	student
Dragan	Cockalo	cole@tfzr.uns.ac.rs	teaching staff
PROJECTs	DISSOFT	dissoft@it-project.rs	administrative worker
Jasmina	Djukic	jasmina@tfzr.uns.ac.rs	administrative worker
Olivera	Dobrosavljev	biblio@tfzr.uns.ac.rs	administrative worker
Nikola	Dragan	nikoladragan@gmail.com	student
Miroslav	Eremic	elemmiki@gmail.com	student

Слика 4.2.3.3. Екран - табеларни приказ учесника, уз филтер према радном статусу

No	Role Name	Role Description
0	Client representative	Defines client's requirements and checks quality of delivery
0	Consultant	Defines details about problem field in semantic area, could have roles similar to client representative
1	Program manager	Organizes development program that consists of multiple project. Coordinates with clients regarding projects approval and verification. Coordinates work of each project manager and coordinates integration of each project into a complex solution.
2	Project manager	Coordinates all participants and directs toward finishing project within specified quality, scope, time and costs
3	Project team coordinator	Coordinates team or sub-team within a project
4	Projectant	Create models of system proposal
5	Database designer and administrator	Designs database, implements it within particular DBMS and does administration of using database
5	Developer	Creates database, user interface and does programming of the business logic of an application
5	Database designer	Designs database and implements it within a particular DBMS
5	Database administrator	Does administration of using a database within a particular DBMS
6	User interface designer	Designs user interface of an application
6	Front-end (user interface) programmer	Implements user interface of an application
6	User interface designer and programmer	Graphically designs user interface and implements it within a particular user interface environment (desktop, web, mobile...)
7	Programmer	Implements program
8	Tester	Performs testing on programs
9	Technical writer	Creating project's documentation

Слика 4.2.3.4. Екран - табеларни приказ улога учесника у пројекту, с описима задужења

**PROJECT TOPIC DETAILS**

<b>Topic keywords:</b>	<input type="text"/>	<input type="button" value="Load"/>
<b>Topic name:</b>	Alumni web (Web stranica za evidenciju diplk ▾)	<input type="button" value="Load"/>
<b>TOPIC name:</b>	Alumni web (Web stranica za evidenciju diplomiranih studenata)	
<b>Description:</b>	Web application for alumni record. Alumni of a university school could upload photos and text regarding their careers and memories related to their study time. They could also submit some suggestions for school quality improvements.	

Слика 4.2.3.5. Екран - приказ детаља у области пројектних тема

**PROJECT DETAILS**

<b>Project title:</b>	Alumni record 1.0	<input type="button" value="Load"/>
<b>Short name:</b>	Alumni 1.0	
<b>Topics:</b>	Alumni web (Web stranica za evidenciju diplomiranih studenata)	
<b>Topics description:</b>	Web application for alumni record. Alumni of a university school could upload photos and text regarding their careers and memories related to their study time. They could also submit some suggestions for school quality improvements.	
<b>Technical description:</b>	ASPX + SQL Server	
<b>Project manager:</b>	Glusac Dragana	
<b>Proj. man. institution:</b>	University of Novi Sad, Technical faculty "Mihajlo Pupin", Zrenjanin, Serbia	

Слика 4.2.3.6. Екран - приказ детаља пројекта



**LIST OF PROJECTS**

Topics:

Manager:

Client:

ID	Pr. short name	Project title	Topics	Proj. manager	Client repr.	Proj. start date	Duration (months)
6	Alumni 1.0	Alumni record 1.0	Alumni web (Web stranica za evidenciju diplomiranih studenata)	Dragana Glusac	Erika Antal	March 15, 2014	3
8	CIP online 1.0	On-line CIP records 1.0	On-line CIP (on-line biblioteka katalogizacija CIP)	Olivera Dobrosavljev	Olivera Dobrosavljev	March 15, 2014	3
11	Comp Admin 1.0	Computer system administration record 1.0	School computer system administration record (Evidencija administracije racunarskog sistema skole)	Miodrag Ivkovic	Vladimir Karuovic	March 15, 2014	3
19	ConfOrg 1.0	Conference organizer 1.0	Conference organization web portal (Web portal za organizaciju)	Branko Markoski	Branko Markoski	March 15, 2014	3

Слика 4.2.3.7. Екран - табеларни приказ пројеката

**IT project**

[Log Off](#)

**PROJECT PARTICIPANT ASSIGNMENT**

Project:

Participant:

Role:

Assign. Date:    (dd/mm/yyyy)

Слика 4.2.3.8. Екран- унос података о додељивању учесника пројекту

#### 4.2.4. Елементи подршке Waterfall приступа

У оквиру Waterfall приступа, омогућен је детаљнији унос података о задацима, ресурсима, верификацији урађеног задатка и верификацији пројекта.

**NEW PROJECT TASK**

**Project:** choose...

**Participant:**

**Part. proj. assignment:**

**Task short description:**

**Task description:**

**Date assigned:**    (dd / mm / yyyy)

**Assigned duration:**  (days)

Слика 4.2.4.1. Екран - унос података о задатку

**LIST OF PROJECT TASKS**

**Project:** choose...

**Participant:** choose...

Project	Participant	Assignment	Task	Task start date	Max duration (days)	Finalized
Alumni record 1.0	Milos Simic	Developer	Programiranje	January 1, 2003	3	No

Слика 4.2.4.2. Екран - табеларни приказ података о задацима пројеката

**NEW RESOURCE**

**Project:** choose... ▼

**File Upload:** Choose File No file chosen

**Resource type:** choose... ▼

**Resource description:**

Слика 4.2.4.3. Екран - унос података о ресурсу пројекта

**LIST OF PROJECT RESOURCES**

**Project:** choose... ▼

PROJECT	Resource No	Description	Type
Alumni record 1.0	1	Dokument	document

Слика 4.2.4.4. Екран - табеларни приказ података о ресурсима пројекта

**NEW TASK RESULT**

**Project:** choose... ▼

**Participant assigned to task:** ▼

**Part. proj. assignment:**

**Task short description:** ▼

**Task No:**

**Task description:**

**Date assigned:**

**Assigned duration:**  (days)

**File Upload:** Choose File No file chosen

**Participant implemented result:** ▼

**Submission date:**    (dd/mm/yyyy)

**Result type:** choose... ▼

**Result description:**

Слика 4.2.4.5. Екран - унос података о резултату реализованом у односу на задатак

**LIST OF TASKS RESULTS**

Project:

Project	Participant	Assignment	Task	Result	Result type	Submission date
Alumni record 1.0	Milos Simic	Developer	Programiranje	uradjeno sve	document	February 2, 2013

Слика 4.2.4.6. Екран - табеларни приказ података о резултатима задатака

**NEW TASK RESULTS VERIFICATION**

Project:

Participant:

Part. proj. assignment:

Unfinished Task:

Participant's Task No:

Task description:

Date assigned:

Assigned duration:  (days)

List of task results:

Finished:  Yes  No

Date verified:  /  /  (dd/mm/yyyy)

Finish grade:  / 5

Project participant verified:

Слика 4.2.4.7. Екран - унос података о верификацији резултата задатка

**TASKS RESULTS VERIFICATION LIST**

Project:

Participant:

Project	Participant	Assignment	Task	Task start date	Task end date	Finalized
Alumni record 1.0	Milos Simic	Developer	Programiranje	January 1, 2013	March 3, 2013	Yes

Слика 4.2.4.8. Екран - табеларни приказ података о верификацији резултата задатака

#### 4.2.5. Елементи подршке мултипројектном раду, агилном приступу, визуализацији и адаптивности

У оквиру реализације другог модалитета рада, тј. подршке за агилни приступ управљању софтверским пројектом, основна промена је у уопштавању података и поједностављивању уноса, као и прегледнијем интегрисаном приказу података. Подршка агилном приступу подразумева мањи обим података за унос и бољу прегледност података удруживањем података различитих категорија у оквиру заједничког табеларног приказа и могућност да се реализација пројеката одвија и евидентира у оквиру различитих фаза произвољним редоследом.

Подршка мултипројектном раду подразумева да један корисник апликације може бити додељен као учесник више пројеката. Након пријављивања и учитавања главног менија, бирамо опцију „Projects“, након чега добијамо списак пројеката на којима је ангажован наведени учесник (Слика 4.2.5.1.).

The screenshot shows a web browser window with the URL www.it-projects.rs/ProjectListUser.aspx. The page has a blue header with the text 'IT project' and a globe icon. Below the header, there is a navigation menu with buttons for HOME, Profile, Participants, Topics, Interests, Invitations, and Projects. The main content area is titled 'MY PROJECTS' and contains a table with the following data:

SELECT ID	Project title	Proj. start date	Duration (months)	MY ASSIGNMENT
Select 29	SAMPLE PROJECT	July 23, 2014	3	Developer

Слика 4.2.5.1. Екран - табеларни приказ пројеката пријављеног учесника

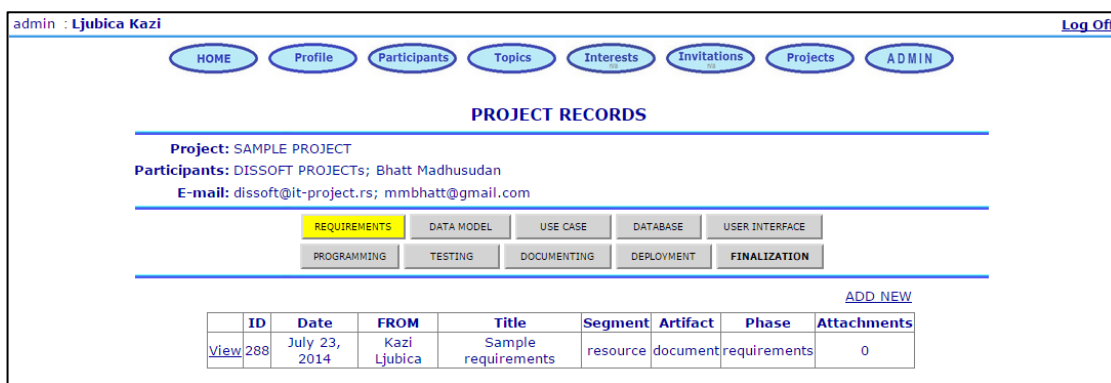
У реализованој апликацији, корисник који је у статусу администратора такође може бити ангажован на пројектима (Слика 4.2.5.2.).

The screenshot shows a web browser window with the URL www.it-projects.rs/ProjectListAdmin.aspx. The page has a blue header with the text 'IT project' and a globe icon. Below the header, there is a navigation menu with buttons for HOME, Profile, Participants, Topics, Interests, Invitations, Projects, and ADMIN. The main content area is titled 'MY PROJECTS' and contains a table with the following data:

SELECT ID	Project title	Proj. start date	Duration (months)	MY ASSIGNMENT
Select 6	Alumni record 1.0	March 15, 2014	3	Consultant
Select 11	Computer system administration record 1.0	March 15, 2014	3	Consultant
Select 19	Conference organizer 1.0	March 15, 2014	3	Consultant
Select 22	E-mail stud info 1.0	April 2, 2014	3	Consultant
Select 26	Example project 1.0	June 1, 2014	6	Consultant
Select 23	Excel interoper 1.0	April 3, 2014	3	Consultant
Select 10	Faculty employees record 1.0	March 15, 2014	3	Consultant
Select 7	Final exam organizer 1.0	March 15, 2014	3	Consultant
Select 18	Forum 1.0	March 15, 2014	3	Consultant
Select 28	Int. Sci. Journal Organizer	July 15, 2014	4	Consultant
Select 24	Invoice 1.0	April 8, 2014	3	Consultant
Select 15	Legal documents 1.0	March 15, 2014	3	Consultant
Select 20	Library users 1.0	April 1, 2014	2	Consultant
Select 14	Online Bookstore 1.0	March 15, 2014	3	Consultant
Select 8	On-line CIP records 1.0	March 15, 2014	3	Consultant
Select 17	Pre-exam archive 1.0	March 15, 2014	3	Consultant
Select 27	Reservations 1.0	June 10, 2014	3	Consultant
Select 29	SAMPLE PROJECT	July 23, 2014	3	Consultant
Select 9	SCI fund record 1.0	March 15, 2014	3	Consultant
Select 16	Sessions archive 1.0	March 15, 2014	3	Consultant
Select 25	Student Behaviour 1.0	May 25, 2014	3	Consultant
Select 12	Web evaluation of teaching 1.0	March 15, 2014	3	Consultant
Select 13	Work organizer 1.0	March 15, 2014	3	Consultant

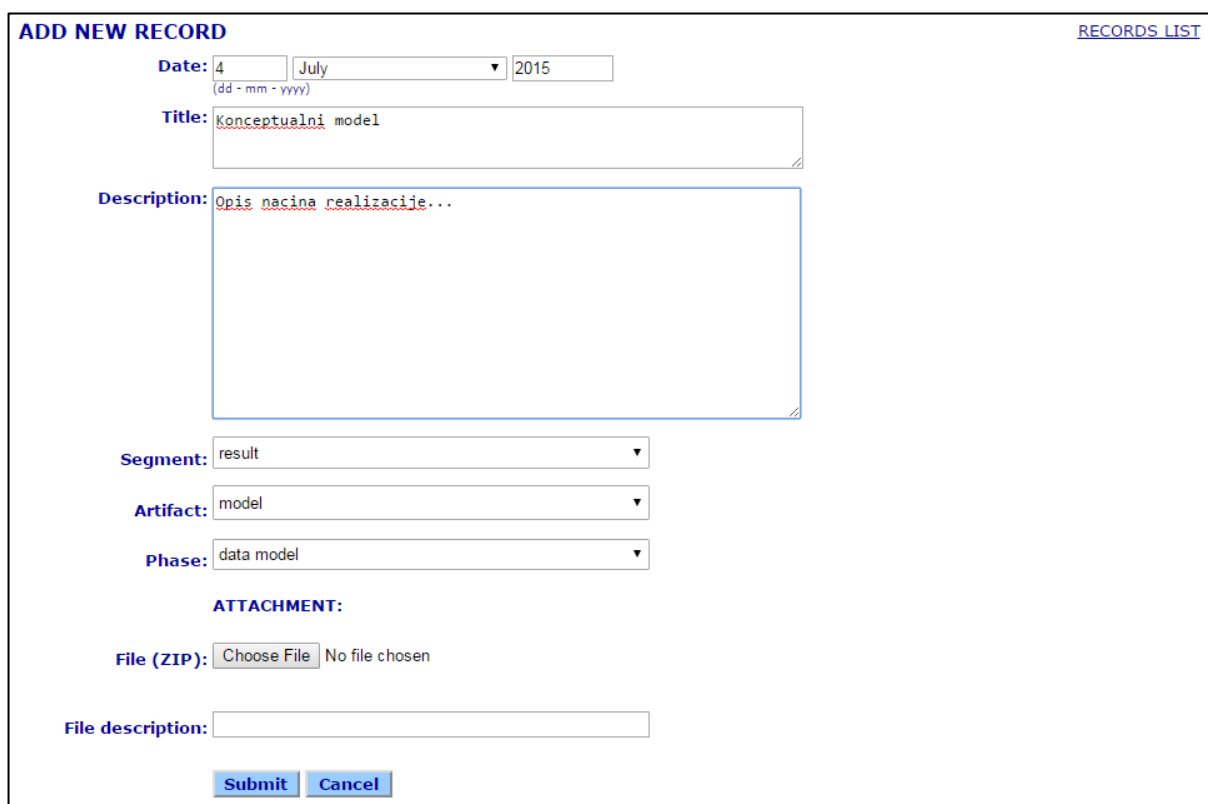
Слика 4.2.5.2. Екран - табеларни приказ пројеката пријављеног администратора

Избором пројекта, отвара се централни део за визуализацију статуса пројекта, аналитички приказ резултата и покретање уноса ставке у току реализације пројекта.



Слика 4.2.5.3. Екран - табеларни приказ ставки у току реализације пројекта

Избором опције „ADD NEW“ добијамо форму за унос нове ставке:



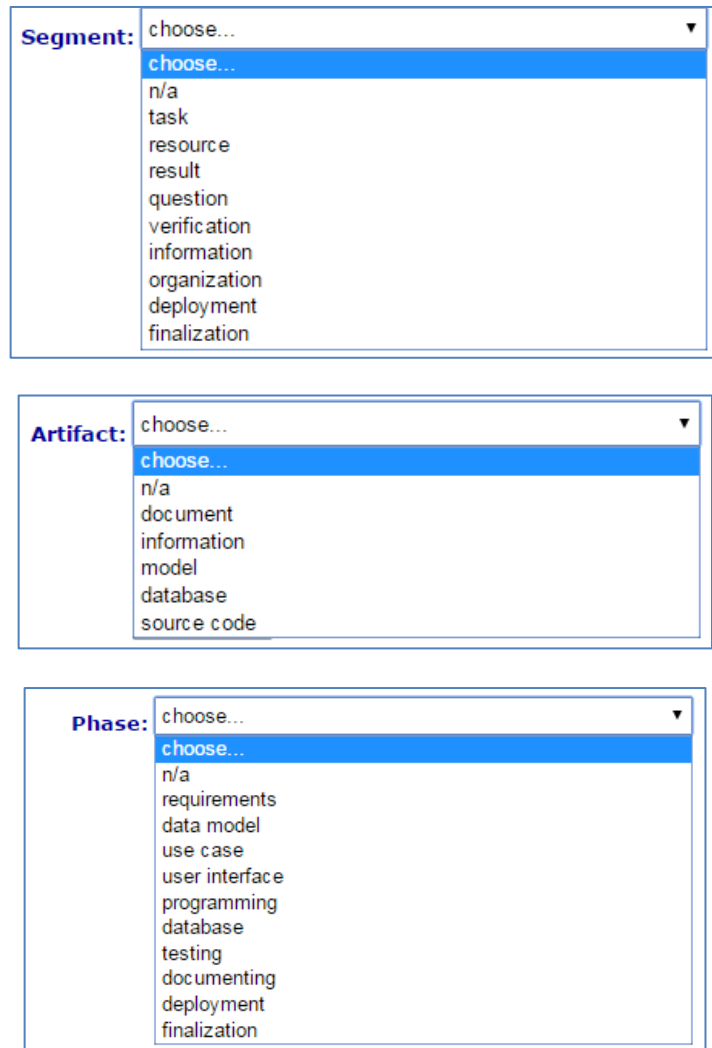
Слика 4.2.5.4. Екран - унос нове ставке на списак ставки тока реализације пројекта

Избором опције „Record List“ добијамо листу ставки тока реализације пројекта са новом ставком.

Основна подршка визуализацији је реализована кроз приказ основних фаза развоја софтвера од почетне спецификације захтева до финализације, као и кроз промену боје одговарајућег сегмента-фазе када се реализују активности у оквиру одговарајуће фазе. Уколико је унет податак о сегменту који до тада није рађен, одговарајући део у визуалном приказу статуса ће „засветлети“, тј. добити „жуту боју“, што значи да је за тај сегмент започела реализација.

Основни елементи подршке адаптивности реализовани су кроз уопштавање фаза, сегмената и артефакта, који могу бити прилагођени у односу на различите методологије, технологије и типове пројеката. Уопштавање је постигнуто коришћењем шифарника за: сегмент, артефакт и фазу развоја. Увођењем наведених шифарника,

омогућава се подршка прилагођавању различитим методологијама развоја и технологијама имплементације – тј. различитим фазама, сегментима и артефактима.



Слика 4.2.5.5. Шифарници у уносу ставке тока пројекта – сегмент, артефакт и фаза

## 5. ЕМПИРИЈСКА ИСТРАЖИВАЊА

### 5.1. План реализације емпиријских истраживања

#### 5.1.1. Методолошке основе емпиријских истраживања

Према [Kitchenham et al, 1997], у евалуацији метода и алата у области софтверског инжењерства најчешће се утврђују и мере:

1. Ефекти примене наведене методе или алата, квантитативним мерењем предности коришћења наведене методе или алата
2. У којој мери наведени метод или алат одговара потребама решавања одређене класе проблема или типа организације, квалитативним утврђивањем (најчешће личним ставовима корисника) да ли наведени систем поседује одговарајуће карактеристике у складу са потребама и стандардима.

У наведеном контексту, најчешће коришћене методе су [Kitchenham et al, 1997]:

- формални експеримент – коришћење метода и алата од стране софтверских инжењера за извршавање једног или више задатака
- студија случаја – примена методе или алата који се анализира за конкретан пројекат једне организације и опис искуства након коришћења
- анкетирање – учесници испитивања који су раније користили неке друге методе или алате треба да искажу своје ставове у вези нове методе или алата.

Наведене методе могу се користити као [Kitchenham et al, 1997]:

- методе квантитативне евалуације – базиране на утврђивању особина које се могу квантитативно измерити
- методе квалитативне методе – базиране на утврђивању захтева за особинама које наведени алат или метода треба да има и поређењу са карактеристикама које стварно има; најчешће се базира на субјективним ставовима учесника
- хибридне методе (испитивање мишљења експерата уз квалитативну анализу ефеката примене система, бенчмаркинг). Бенчмаркинг представља примену низа различитих тестова над методама или алатима који се анализирају, где је избор тестова субјективан, али мерне величине и вредности су објективни; може да се примењује за објективно упоређивање карактеристика различитих алата.

У оквиру емпиријских истраживања приказаних у овом одељку, намера је одговорити на истраживачка питања и утврдити статус потврђености хипотеза истраживања. У оквиру плана реализације емпиријских истраживања дат је кратак преглед начина како би се могли утврдити одговори на истраживачка питања и статус потврђености хипотеза.

#### 5.1.2. План емпиријског истраживања у односу на истраживачка питања

Истраживачка питања дефинишу шта је потребно истражити и усмеравају истраживање у одређеном правцу. У наставку ће бити дата *истраживачка питања*:

1. Која су искуства у дистрибуираном развоју и управљању софтверским пројектима?
  - Који начин рада се примењује у развоју софтвера у смислу тимских и локацијских карактеристика?
  - Који алати се користе у тимском раду и управљању софтверским пројектима?
  - Који су фактори успеха и неуспеха, односно који је утицај људског, организационог и технолошког фактора на успех софтверског пројекта?
2. Да ли постоји потреба за коришћењем и развојем информационог система за подршку дистрибуираном развоју софтвера, односно управљању софтверским пројектима у дистрибуираном окружењу? Какве би биле карактеристике таквог система и које су најзначајније?



*План реализације емпиријског истраживања* ради добијања одговора на наведене две групе питања односи се на примену анонимног анкетирања.

3. Да ли су предложени метрички модели за евалуацију артефакта у развоју софтвера, мониторинг успеха и процену трајања софтверског пројекта применљиви у пракси?

*План реализације емпиријског истраживања* ради добијања одговора на ово питање је примена метричких модела над конкретним артефактима развоја софтвера, као и подацима из реализације конкретних софтверских пројеката.

### **5.1.3. План емпиријских истраживања у односу на евалуацију употребљивости развијеног прототипа**

У оквиру емпиријског истраживања употребљивости развијеног прототипа (реализовани web сајт који је развијен за потребе информационог система за управљање пројектима у дистрибуираном развоју софтвера) потребно је утврдити одговоре на следећа питања:

1. Које су предности наведеног система?
2. Кији су недостаци наведеног система?
3. Које су могућности унапређења наведеног система?
4. Које су предности наведеног система у односу на сличне системе који се користе у реалној пракси?

*План истраживања* у овој области обухвата:

1. Примена анонимног анкетирања.
2. Анализом и упоређивањем карактеристика постојећих професионалних решења (бенчмаркинг).

### **5.1.4. План емпиријских истраживања у односу на утврђивање статуса потврђености хипотезе и подхипотеза**

У оквиру методолошког оквира истраживања у уводном поглављу (одељак 1.1.4), дефинисана је основна хипотеза и подхипотезе. Основна хипотеза се може поделити на неколико делова, односно приказати кроз неколико реченица, тј. тврђења која треба доказати. Парцијалним утврђивањем статуса потврђености тих делова хипотезе може се добити статус утврђивања хипотезе. Такође, за сваку од подхипотеза може се дефинисати начин утврђивања потврђивања подхипотезе.

#### **5.1.4.1. План истраживања статуса потврђености основне хипотезе**

У наставку је дат приказ појединачних делова хипотезе, који се може засебно разматрати у контексту потврде статуса потврђености.

1. „Могуће је креирати теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима“.

*План утврђивања статуса потврђености:* Креирањем теоријски заснованог модела могуће је потврдити да је могуће креирати теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима.

2. „Креирани теоријски модел омогућава имплементацију наведеног система.“

*План утврђивања статуса потврђености:* Имплементирањем наведеног система могуће је потврдити да теоријски модел омогућава имплементацију наведеног система.

3. „Имплементирани систем може да се користи у оквиру наставног процеса високошколске установе у области развоја софтвера и управљања софтверским

пројектима.“, односно истраживачко питање: „Да ли систем може да се користи у наставној пракси?“

*План утврђивања статуса потврђености:* Применом система у оквиру реализације софтверских пројеката за потребе наставе у високом образовању, могуће је потврдити да се развијени систем може користити у оквиру наставног процеса високошколске установе у области развоја софтвера и управљања софтверским пројектима.

4. „Имплементирани систем може да обезбеди подршку за управљање софтверским пројектима у реалној пракси.“, односно истраживачко питање: „Да ли систем може да се користи у реалној професионалној пракси?“

*План утврђивања статуса потврђености:*

1. Пробном применом имплементiranог система од стране ИТ кадрова уз пратеће анонимно анкетирање
2. Коришћењем система у реализацији софтвера који би по својим карактеристикама могли да се сматрају професионалним решењима или који су намењени решавању проблема реалне праксе или примени у некој конкретној организацији.

#### **5.1.4.2. План истраживања статуса потврђености подхипотеза**

##### **АНАЛИЗА 1. ПОДХИПОТЕЗЕ**

„Коришћење адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима смањује време реализације софтверског пројекта.“

*План утврђивања статуса потврђености:*

1. Пробном применом имплементiranог система од стране ИТ кадрова уз пратеће анонимно анкетирање
2. Конкретном применом система и анализом резултата са становишта времена потребног за коришћење система, односно времена реализације софтверског пројекта

##### **АНАЛИЗА 2. ПОДХИПОТЕЗЕ**

1. ДЕО – „Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим типовима софтвера“.

*План утврђивања статуса потврђености:*

Конкретном применом система у оквиру праћења реализације софтвера различитог типа.

2. ДЕО – „Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим типовима софтверских пројеката“.

*План утврђивања статуса потврђености:*

Конкретном применом система у оквиру праћења реализације софтверских пројеката различитог типа.

##### **АНАЛИЗА 3. ПОДХИПОТЕЗЕ**

„Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим технологијама имплементације софтвера и реализације софтверских пројеката.“

*План утврђивања статуса потврђености:*

Конкретном применом система у оквиру праћења реализације софтверских пројеката, где су софтвери реализовани коришћењем различитих технологија.

##### **АНАЛИЗА 4. ПОДХИПОТЕЗЕ**

1. ДЕО – „Адаптивбилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се користи у оквиру наставног процеса у области образовања развоја софтвера и управљања софтверским пројектима.“

*План утврђивања статуса потврђености (идентичан као за 3. део хипотезе):* Применом система у оквиру реализације софтверских пројеката за потребе наставе у високом образовању.

2. ДЕО - „Применом система се повећава ефикасност наставе у области софтверског инжењерства и развоја информационих система.“

*План утврђивања статуса потврђености (идентичан као за 1. подхипотеза):*

1. Пробном применом имплементираног система од стране ИТ кадрова уз пратеће анонимно анкетирање
2. Конкретном применом система и анализом резултата.

3. ДЕО – „Применом система повећава се квалитет реализованих софтверских пројеката у наставном процесу.“

*План утврђивања статуса потврђености:*

1. Пробном применом имплементираног система од стране ИТ кадрова уз пратеће анонимно анкетирање
2. Конкретном применом система и анализом резултата.

### **5.1.5. Сумарни приказ планираних техника и метода истраживања**

Сумарно су приказане планиране технике и методе истраживања које се односе на истраживачка питања, истраживање употребљивости решења и потврду хипотезе и подхипотеза:

1. *Емпиријско истраживање коришћењем наведеног система* и утврђивањем његових предности и недостатака, као и доприноса. Емпиријско истраживање треба да се спроведе у оквиру коришћења система праћењем реализације студентских софтверских пројеката у оквиру реализације предиспитних обавеза у настави, као и у оквиру реализације професионалних решења који треба да су употребљиви у пракси. Посебно треба обратити пажњу да узорку припадају софтверска решења различитих технологија, типова софтвера и софтверских пројеката.
2. *Анкетирање* са узорком «ИТ кадар», односно особе које су завршиле студије у области информационих технологија, запослене у овој области, које су учествовале у развоју софтвера, учествовале у софтверским пројектима, користиле алате за подршку тимском раду у реализацији софтвера и алате за управљање софтверским пројектима.
3. *Анализа постојећих решења*, односно алата за подршку тимском раду у реализацији софтвера и алата за управљање софтверским пројектима, као и компарација карактеристика постојећих решења.
4. *Примена метричких модела* над артефактима конкретних софтверских пројеката и подацима о реализацији процеса у оквиру реализације софтверских пројеката. Примену је могуће реализовати у оквиру развоја софтвера у области информационих система, анализом студентских радова и пројеката. Артефакти обухватају моделе анализе пословних процеса и модела дизајна, као и саме софтверске апликације.

У оквиру појединих емпиријских истраживања биће приказана анализа резултата са становишта истраживачких питања и хипотеза, односно са становишта делова основне хипотезе и делова подхипотеза. Детаљни подаци којима се доказују поједини делови хипотезе и подхипотеза биће дати у прилогу.

У појединим емпиријским истраживањима, где је у фокусу функционални аспект реализованог система, користи се почетни прототип који је имплементиран као подршка основним функцијама у администрацији управљања софтверским пројектима. У одговарајућим одељцима анализе потврђености хипотезе и подхипотеза, под „имплементирани систем“ сматрамо почетни прототип софтверске подршке адаптивном дистрибуираном информационом систему за подршку управљању реализацијом софтверских пројеката, који омогућава подршку за основну функционалност.

## **5.2. ЕМПИРИЈСКО ИСТРАЖИВАЊЕ МОГУЋНОСТИ КОРИШЋЕЊА ПРОТОТИПА У НАСТАВИ И ПРОФЕСИОНАЛНОМ УСАВРШАВАЊУ СТУДЕНАТА**

### **5.2.1. Циљеви и методе емпиријског истраживања**

Циљеви емпиријског истраживања су:

- утврђивање могућности примене система у наставним условима
- утврђивање могућности примене система у реализацији софтверских решења за потребе реалне праксе
- могућности примене са различитим:
  - типовима софтвера
  - типова софтверских пројеката
  - примењених технологија имплементације софтвера
- провера статуса потврђености дела главне хипотезе и делова подхипотеза које се односе на претходно описане циљеве емпиријског истраживања.

Емпиријско истраживање, које ће бити описано у овом одељку, односи се на испитивање могућности и карактеристика коришћења прототипа у наставним условима и у оквиру професионалног усавршавања студената у оквиру унапређења информационог система факултета. Конкретном применом прототипа над реалним софтверским пројектима тестиране су могућности за применљивост прототипа у наставној пракси и реалној пракси. Прототип система је коришћен у свим фазама развоја софтвера. Критеријум завршетка студентског софтверског пројекта је представљао покривеност свих функционалних захтева, поузданост рада и задовољавајући степен аутоматизације у циљу потребног нивоа „user friendly“ корисничког интерфејса.

### **5.2.2. Узорак истраживања**

Узорак истраживања представљају радови студената, као и процес реализације студентских софтверских пројеката унапређења појединих сегмената информационог система Техничког факултета „Михајло Пупин“ из Зрењанина из школске 2013/14, као и школске 2014/15 године.

#### **5.2.2.1. Процес формирања узорка истраживања**

У складу са трендом ка повећању заинтересованости студената за реализацијом практичних радова у испуњавању предиспитних обавеза путем реализације конкретних пројеката (семинарских радова) уместо путем колоквијума који се реализују у рачунарској учионици [Kazi et al, MIPRO, 2012], на Техничком факултету „Михајло Пупин“ у Зрењанину је крајем 2012. године покренута иницијатива од стране студентата 4. године смера Информационе технологије – инжењерство о могућности да студенти у оквиру наставног предмета Информациони системи 2 реализују у оквиру предиспитних обавеза софтвере који би по својим првенствено функционалним карактеристикама биле на нивоу блиском употребљивости, односно блиске професионалним решењима. На овај начин би се знање и практична искуства студената значајно унапредиле и омогућиле усклађивање са потребама за знањем и практичним вештинама које су потребне за решавање проблема у реализацији софтвера реалне праксе у привреди. Претходни дуги низ година су се одговарајуће активности већ спроводиле, али парцијално:

1. илустративно-демонстративним методама где су презентована готова професионална решења на теоријским и практичним часовима,
2. елементима реализације који су демонстрирани на теоријским часовима и увежбавани на практичним часовима, а практично знање проверавано у оквиру колоквијума који су реализовани у рачунарској учионици,

3. семинарским и дипломским радовима из области које су се односила на практична решења инспирисана потребама реалне праксе.

На основу иницијативе студената, фебруара 2013. године за потребе реализације практичне наставе осмишљен је почетни списак тема које би се односиле на реализацију софтвера за потребе унапређења информационог система Техничког факултета „Михајло Пупин“ у Зрењанину. Фебруара 2014. године студенти основних и мастер студија Техничког факултета „Михајло Пупин“ у Зрењанину добили су могућност да своје предиспитне обавезе из предмета Информациони системи 2 (основне студије) и Дистрибуирани информациони системи (мастер студије) реализују одабиром неких од понуђених тема које се односе на софтвере за подршку унапређењу информационог система Техничког факултета „Михајло Пупин“ у Зрењанину, а почев од маја 2014. године и да реализују своје предиспитне обавезе из предмета Стручна пракса.

С обзиром да су студенти покренули иницијативу и прихватили учешће у реализацији наведених софтвера, а такође с обзиром да је прототип web апликације био употребљив марта 2014. године у језгру функционалности, јавила се могућност да се евидентирање тока извршавања реализованих софтверских пројеката студената врши коришћењем наведеног прототипа. У периоду од марта 2014. до октобра 2014. године коришћен је претходно приказани прототип ASPX web апликације као подршка у процесу управљања софтверским пројектима које су реализовали студенти, а односе се на софтвере унапређења информационог система Техничког факултета „Михајло Пупин“. Две реализоване web апликације су након тог периода постављене на web сајт факултета и унапредиле функционисање факултета (Alumni део web сајта факултета и Део web сајта факултета који омогућава on-line поручивање књига из скриптарнице). Наведене две апликације су описане у одељку 9.6. У оквиру [Kazi et al, АИТ, 2014b] описане су активности на унапређењу информационог система Техничког факултета „Михајло Пупин“ у Зрењанину у сарадњи са студентима.

На основу успешних резултата претходног периода и интересовања других студената за учешће у наведеним активностима, Наставно-научно веће Техничког факултета „Михајло Пупин“ из Зрењанина је новембра 2014. године одобрило интерни пројекат „Професионално усавршавање студената у оквиру унапређења информационог система факултета“, који је омогућио пријављивање и учешће нових студената основних и мастер студија.

#### **5.2.2.2. Карактеристике узорка истраживања**

Узорак истраживања представљају активности и резултати рада 23 студената основних и мастер студија у школској 2013/14 у оквиру реализације укупно 19 реализованих софтверских пројеката, реализованих под менторством Мр Љубице Кази, у циљу реализације предиспитних обавеза у практичном делу, које се односе на наставне предмете основних и мастер студија инжењерског профила информационог технологија на Техничком факултету „Михајло Пупин“ у Зрењанину.

Такође, узорак представљају активности и резултати рада 13 студената у школској 2014/15 години у оквиру реализације укупно 11 реализованих софтверских пројеката (приказани су само пројекти који су завршени до 8.9.2015) ради реализације предиспитних обавеза у практичном делу, које се односе на наставне предмете основних и мастер студија инжењерског профила информационог технологија на Техничком факултету „Михајло Пупин“ у Зрењанину, које истовремено припадају резултатима интерног пројекта Техничког факултета Михајло Пупин „Професионално усавршавање студената у оквиру унапређења информационог система Факултета“ под руководством Мр Љубице Кази.

За обе групе узорака важи да су студенти се самостално и добровољно јавили да реализују софтверске пројекте у области унапређења информационог система

Техничког факултета „Михајло Пупин“ у Зрењанину. Студенти су имали могућност да слободно одаберу једну од понуђених тема, начин рада (појединачно или у тиму, као и чланове тима) и технологију имплементације. Рад на софтверском пројекту из реалне праксе представљао је основни мотив, али и могућност да се поједини делови решења реализују кроз предиспитне обавезе појединих наставних предмета. Студенти су били мотивисани да реализују софтверске пројекте који се односе на реалан организациони систем, а који им је релативно близак у смислу познавања пословне проблематике. Такође, студенти су мотивисани јер су радом на наведеним пројектима реализовали испуњавање предиспитних обавеза у практичном делу који се односе на поједине наставне предмете, уз могућност да реализована решења прошире у оквиру завршног (дипломског или мастер) рада.

На почетном састанку, који је одржан са студентима у току обе наведене школске године, представљен је приступ развоју у 4 корака односно нивоа сложености апликације (семинарски рад, стручна пракса, дипломски рад, употребљиво решење). На основу наведених предлога, студенти су се определили на почетку рада за ниво до којег желе и могу да реализују своја софтверска решења. Радом на одабраним софтверским пројектима студенти су постепено реализовали све сложенија софтверска решења. Активности у реализацији софтвера биле организоване модуларно, постепеним усложњавањем и додавањем софтверских функција. У оквиру реализације софтверских пројеката, најчешће су у оквиру обавеза на предметима Информациони системи 2 и Дистрибуирани информациони системи студенти реализовали језгро апликације (унос, табеларни приказ, филтрирање), а проширење функционалности (аутентификација, извештаји, напредне претраге и статистичке обраде података) реализовано је у оквиру стручне праксе за исти софтверски пројекат. Коначно, у оквиру завршног рада проширен је обухват података и додате су нове софтверске функције, а такође додате су и функције које се односе на валидацију података и аутоматизме. Студенти који су желели након завршетка дипломског рада да реализују додатне карактеристике софтвера ради постизања нивоа употребљивости и ради омогућавања инсталирања и коришћења софтвера на Техничком факултету „Михајло Пупин“ у Зрењанину, додатно су се укључили у унапређење софтвера првенствено унапређењем корисничког интерфејса софтвера у смислу организације података и убрзања рада аутоматизмима и додавањем напредних опција извештавања, статистике и претраге података.

### 5.2.3. Резултати истраживања

У овом одељку је дат приказ дела списка софтверских пројеката који су реализовани уз подршку управљању софтверским пројектима коришћењем развијеног прототипа.

SELECT	ID	Project title	Proj. start date	Duration (months)	MY ASSIGNMENT
Select	6	Alumni record 1.0	March 15, 2014	3	Consultant
Select	11	Computer system administration record 1.0	March 15, 2014	3	Consultant
Select	19	Conference organizer 1.0	March 15, 2014	3	Consultant
Select	22	E-mail stud info 1.0	April 2, 2014	3	Consultant
Select	26	Example project 1.0	June 1, 2014	6	Consultant
Select	23	Excel interoper 1.0	April 3, 2014	3	Consultant
Select	10	Faculty employees record 1.0	March 15, 2014	3	Consultant
Select	7	Final exam organizer 1.0	March 15, 2014	3	Consultant
Select	18	Forum 1.0	March 15, 2014	3	Consultant
Select	28	Int. Sci. Journal Organizer	July 15, 2014	4	Consultant
Select	24	Invoice 1.0	April 8, 2014	3	Consultant
Select	15	Legal documents 1.0	March 15, 2014	3	Consultant

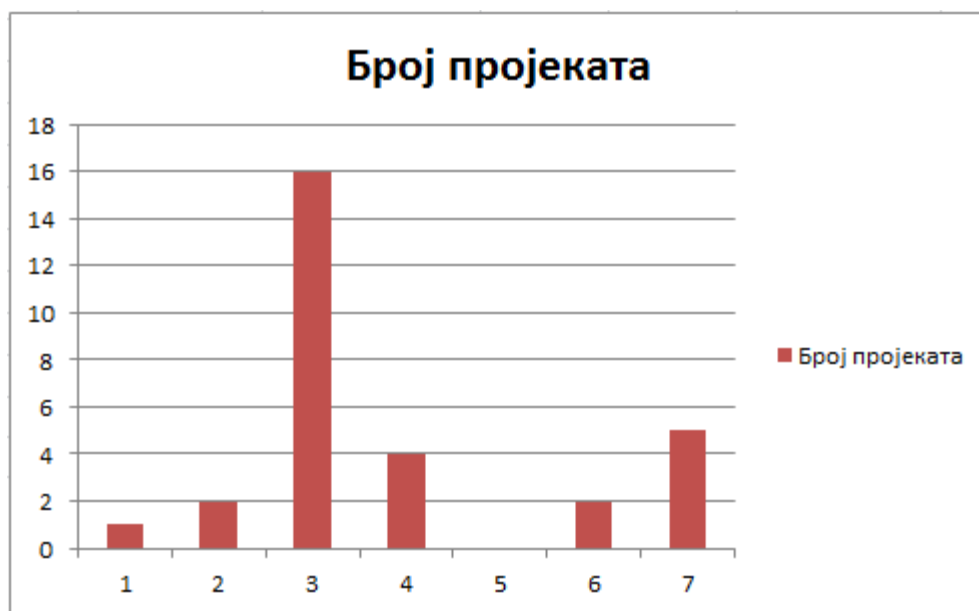
Слика 5.2.3.1. Приказ дела табеларног приказа пројеката у оквиру коришћења прототипа [www.it-project.rs](http://www.it-project.rs)

Комплетан списак свих пројеката из емпиријског истраживања дат је у прилогу. Карактеристике и други детаљни подаци о реализованим пројектима такође су дати у прилогу. У овом одељку приказани су сумарни статистички и графички прикази података о резултатима примене прототипа у наставној пракси и у оквиру пројекта професионалног усавршавања студената.

Број пројеката према броју месеци трајања појединачног софтверског пројекта представљен је следећом табелом и графиком. Највећи број пројеката реализован је у трајању од 3 месеца (укупно 15 пројеката). Подаци из прилога са конкретним датумима почетка и завршетка пројеката представљају основ ове статистике. Датум почетка пројекта представља датум одабира теме (у школској 2013/14 години), односно датум потписивања приступнице пројекту (у школској 2014/15 години). Датумом завршетка пројекта означен је датум последње евиденције која се односи на наведени пројекат. Ти датуми се односе на датум предаје завршеног и тестираног софтверског решења.

Табела 5.2.3.1. Број месеци и број софтверских пројеката који је трајао одговарајући број месеци у оквиру узорка емпиријског истраживања

Број месеци	Број пројеката
1	1
2	2
3	16
4	4
5	0
6	2
7	5



Слика 5.2.3.2. Графички приказ броја пројеката према броју месеци трајања

Број реализованих пројеката према технологији представљен је наредном табелом (табела 5.2.3.2). Евидентна је доминација ASPX (16 пројеката) и PHP (11 пројеката) пројеката у односу на пројекте примене framework-а (ASPX MVC – 1 пројекат и PHP Symphony – 1 пројекат). Такође, реализован је и један пројекат типа Windows апликације коришћењем C# програмског језика.

Табела 5.2.3.2. Приказ примењених технологија и броја пројеката реализованих примењеном технологијом у оквиру студентских софтверских пројеката

Технологија	Број пројеката
ASPX	16
ASPX MVC	1
PHP	11
PHP Symp	1
Win C#	1

У табели 5.2.3.3. је приказана статистика о категоризацији реализованих софтвера. UNSPSC категоризација типова софтвера односи се првенствено на намену. Према наведеној категоризацији, у оквиру овог истраживања највише софтверских пројеката је реализовано у категорији „Business Function Specific Software“ (15) и категорији „Enterprise Resource Planning ERP Software“ (13). Категорији „Business Function Specific Software“ припадају софтверска решења која се односе на факултетску проблематику, док „Enterprise Resource Planning ERP Software“ представља софтвер који може бити примењен у билокојој организацији. У следећој табели такође је приказана статистика категорија софтвера, према Turban et al. Највећи број реализованих софтвера према намени су типа ASN (апликативни софтвер специфичне намене) (27) док је само 2 типа AON (апликативни софтвер опште намене). У односу на категорију корисника, највише је софтвера типа КК (намењен категорији корисника), док је два софтвера К типа (тип софтвера за конкретног корисника) и један софтвер типа ОК (опште намене). Коначно, приказана је и статистика категоризације софтвера према Nguyen-у, где је највећи број софтвера типа NAR (софтвер по наруџби) (21), док је типа КОМ (комерцијални софтвер) укупно 9. Под софтвером по наруџби подразумева се софтвер за конкретну пословну примену, док се под комерцијалним софтвером сматра софтвер који може имати ширу општу примену, у оквиру и независно од организационих система.

Табела 5.2.3.3. Приказ броја реализованих софтвера према категоризацији софтвера

UNSPSC	UNSPSC тип софтвера	Број пројеката
432315	Business Function Specific Software	15
432316	Enterprise Resource Planning ERP software	13
432335	Information Exchange software	2
Turban	Број пројеката	
ASN	27	
AON	2	
OK	1	
KK	27	
K	2	
Nguyen	Број пројеката	
NAR	21	
KOM	9	

Број пројеката према категоризацији у односу на типове софтверских пројеката представљена је у наставку. Према Shenhar & Dvir категоризацији типова софтверских пројеката, која се односи на ниво претходног познавања примењене



технологије у пројекту, у овом истраживању реализовано је 2 пројекта типа МТ - „Medium-Tech Projects“ (примена framework), док су остали пројекти типа ЛТ - „Low-Tech Projects“. Према Berntsson-Svensson&Aurum класификацији софтверских пројеката, према активности највећи број реализованих пројеката је типа ТАН – реализација новог софтвера (29), док је само један софтвер типа ТАУ – унапређење постојећег софтвера. У односу на клијента, укупно 18 пројеката је типа КО – клијент оријентисани, док је 12 пројеката типа ТО. Према Nassif&Boehm класификацији софтверских пројеката, највећи број је софтверских пројеката типа О – мањи тимови и флексибилни захтеви, док је мањи број пројеката типа SD – мешовити тимови, нивои знања и захтеви. Према Reifer класификацији софтверских пројеката, сви пројекти (29) осим једног(који је типа Т - традиционални) су типа W – web апликације. Према Cadle&Yeates класификацији која се специфично односи на софтверске пројекте у области развоја информационих система, сви пројекти осим једног (који је типа UPS – унапређење постојећег софтвера) су типа RNS – развој новог софтвера.

Табела 5.2.3.4. Приказ броја реализованих софтвера према категоризацији софтверских пројеката

<b>Shenhar &amp; Dvir</b>	<b>Број пројеката</b>
LT	28
MT	2
<b>Berntsson-Svensson &amp; Aurum</b>	<b>Број пројеката</b>
TAN	29
TAU	1
KO	18
TO	12
<b>Nassif &amp; Boehm</b>	<b>Број пројеката</b>
O	24
SD	5
<b>Reifer</b>	<b>Број пројеката</b>
W	29
T	1
<b>Cadle &amp; Yeates (IS)</b>	<b>Број пројеката</b>
RNS	29
UPS	1

У наставку је дат приказ статистичких података о улози реализованих софтвера у наставном окружењу. Према наведеној статистици, подједнак је број пројеката који су реализовани у оквиру основних (15) и мастер студија (19), у оквиру наставних предмета Информациони системи 2 (14), Дистрибуирани информациони системи (13), Стручна пракса (16), док су у школској 2014/15 реализоване предиспитне обавезе на мастер студијама из предмета Електронско пословање (1) и Управљање ИТ пројектима (1).

Табела 5.2.3.5. Статистика о улози реализованих софтверских пројеката у наставном окружењу

Ниво студија	Број пројеката
О	19
М	15
Наставни предмет	Број пројеката
IS2	14
DIS	13
SP	16
UITP	1
EP	1
Завршни рад	Број пројеката
DA	19
DA-PLAN	8
NE	9

Софтверски пројекти су реализовани у појединачном и тимском раду. Наредна табела приказује статистику о начину рада на пројектима. Највећи број пројеката (24) су пројекти где студенти раде појединачно, 2 пројекта су специфична јер су повезана (у табели 5.2.3.6. обележено је са 1\* да су то учесници који су задужени за свој појединачни пројекат, али га раде тимски у сарадњи са другим учесником чији је засебан пројекат повезан са првим), док је 2 пројекта са 2 учесника и 2 пројекта са 3 учесника.

Табела 5.2.3.6. Статистика о броју пројеката и броју учесника по пројекту

Број учесника	Број пројеката
1	24
1*	2
2	2
3	2

У контексту употребљивости решења, дата је статистика броја употребљивих решења која су реализована у оквиру студентских пројеката. Студентски пројекти из школске 2013/14 године резултовали су инсталирањем 2 софтверска пројекта, док је 9 пројекта спремно за инсталирање. Укупно 5 студентских пројеката из школске 2014/15 године је на нивоу употребљивости. Сумарно: реализовано је 29 пројеката на почетном нивоу сложености, 16 пројеката на 2. Нивоу, 27 пројеката на 3 нивоу и 16 пројеката на 4. нивоу сложености.

Табела 5.2.3.7. Статистика о применљивости решења и нивоима сложености софтверских решења

Применљиво решење		Постављено	Спремно
шк 2013/14		2	9
шк 2014/15		0	5
1 ниво - наставни предмет	2 ниво - стручна пракса	3 ниво - завршни рад	4 ниво - применљиво решење
29	16	27	16

#### **5.2.4. Анализа резултата са становишта истраживачких питања и хипотеза**

##### **ХИПОТЕЗА - 3. ДЕО**

*„Имплементирани систем може да се користи у оквиру наставног процеса високошколске установе у области развоја софтвера и управљања софтверским пројектима.“*

##### **4. ПОДХИПОТЕЗА – 1. ДЕО**

*„Имплементирани систем може да се користи у оквиру наставног процеса у области образовања развоја софтвера и управљања софтверским пројектима.“*

ОДГОВОР: Овај део главне хипотезе, односно подхипотезе је потврђен, с обзиром на чињеницу да је имплементирани систем коришћен у оквиру подршке реализацији софтверских пројеката који су реализовани у оквиру испуњавања предиспитних обавеза из 5 наставних предмета – Дистрибуирани информациони системи, Информациони системи 2, Стручна пракса, Електронско пословање, Управљање ИТ пројектима.

##### **ХИПОТЕЗА - 4. ДЕО**

*„Имплементирани систем може да обезбеди подршку за управљање софтверским пројектима у реалној пракси.“*

##### **ИСТРАЖИВАКО ПИТАЊЕ**

*„Да ли систем може да се користи у реалној професионалној пракси?“*

ОДГОВОР: Овај део главне хипотезе је потврђен, односно одговор на истраживачко питање је да се наведени систем може користити у реалној пракси. Потврђивање овог дела хипотезе заснива се на чињеници да је развијени систем успешно коришћен за подршку управљању пројектима реализације два софтвера („Alumni web апликација“ и „On-line систем за поручивање књига из скриптарнице“) који су октобра 2014. године успешно интегрисани у реални информациони систем, тј. web сајт Техничког факултета „Михајло Пупин“ у Зрењанину. Приказ наведена два софтвера, као и подаци о коришћењу наведене апликације ([www.it-project.rs](http://www.it-project.rs)) у оквиру реализације наведена два софтвера дат је у прилогу.

##### **2. ПОДХИПОТЕЗА – 1. ДЕО**

*„Имплементирани систем може да се прилагоди различитим типовима софтвера.“*

ОДГОВОР: Овај део друге подхипотезе је потврђен. У оквиру примене развијеног система реализовани су софтвери различитог типа према класификацијама UNSPSC, Turban и Nguyen.

##### **2. ПОДХИПОТЕЗА – 2. ДЕО**

*„Имплементирани систем може да се прилагоди различитим типовима софтверских пројеката.“*

ОДГОВОР: Овај део друге подхипотезе је потврђен. У оквиру примене развијеног система реализовани су софтверски пројекти различитог типа према класификацијама Shenhar&Dvir, Berntsson-Svensson&Aurum, Nassif&Boehm, Reifer, Cadle&Yeates (IS).

##### **3. ПОДХИПОТЕЗА**

*„Имплементирани систем може да се прилагоди различитим технологијама имплементације софтвера и реализације софтверских пројеката.“*

ОДГОВОР: Овај део друге подхипотезе је потврђен. У оквиру примене развијеног система реализовани су софтвери различитог типа у односу на технологије реализације: ASPX - ASPX web апликација + MS SQL Server база података; ASPX MVC - ASPX MVC web апликација + MS SQL Server база података; PHP - PHP web апликација + MY SQL база података; PHP Symp - PHP (Symfony framework) web апликација + MY SQL база података; Win C# - Windows Forms C# + MS SQL Server.

## **5.3. Пробно коришћење прототипа од стране ИТ кадрова и испитивање њихових ставова анкетирањем**

### **5.3.1. Методологија истраживања**

Методологија истраживања састоји се из утврђивања циља, креирање инструмента истраживања, одабир узорка, прикупљања података, верификација потпуности и коректности података, корекције података, вредновање подобности узорка и статистичка обрада.

#### **5.3.1.1. Дефинисање циља истраживања**

У оквиру истраживања ставова анкетирањем, циљ је утврдити:

- чињенице о знању и искуству испитаника у области развоја софтвера и учешћа у софтверским пројектима, посебно у области дистрибуираног тимског развоја софтвера,
- искуства испитаника која се односе на начин реализације софтвера у организационом смислу,
- који алати су најчешће коришћени у пракси у реализацији софтверских пројекта у домену тимског рада и управљања софтверским пројектима
- ставове испитаника о томе да ли је потребно користити или развити информациони систем за потребе праћења реализације софтверског пројекта
- какве карактеристике треба да има информациони систем за подршку управљању софтверским пројектима који се одвијају у дистрибуираном окружењу,
- ставове о употребљивости развијеног прототипа софтверске подршке управљању софтверским пројектима који се развијају у дистрибуираном окружењу,
- предности, недостатке и могућности унапређења развијеног прототипа.

#### **5.3.1.2. Припрема инструмента истраживања**

Као инструмент истраживања припремљена је анкета у форми MS Word документа (на српском и енглеском језику; текст анкета у Прилогу), која се састоји из питања затвореног и отвореног типа. Ради обезбеђивања што веће прецизности одговора, поред питања затвореног типа (да, не, не знам), дата су и питања где су испитаници за исту тему испитивања требали да искажу податке и нумерички. Намерна редунданса у питањима и очекиваним одговорима је креирана због повећања поузданости одговора. Питања отвореног типа омогућавају образлагање ставова датих питањима затвореног типа.

Такође, питања затвореног типа (типа заокруживања) су проширена питањима типа рангирања, односно вредновања значаја фактора и карактеристика. Питања у анкети се односе на опште ставове које се односе на проблеме дистрибуираног развоја софтвера и управљања софтверским пројектима у дистрибуираном окружењу, као и ставове у вези реализованог прототипа (доступног на адреси *www.it-project.rs*). Да би испитаник могао да одговори на питања у вези прототипа, треба да приступи web сајту и пријави се (сви испитаници су добили заједничке приступне податке – *корисничко име: dissoft@it-project.rs*, *шифра: ABC*, односно – *корисничко име: proba@it-project.rs*, *шифра: proba*), као и да пробно користи опције наведеног прототипа.

#### **5.3.1.3. Одабир узорка и прикупљање података**

Ради припреме узорка истраживања, контактирани су потенцијални учесници путем друштвених мрежа Facebook и LinkedIn и одабрани су случајним избором међу колегама, из Србије и иностранства, али који су по образовању у области

информационих технологија, запослени у области ИТ и-или су учествовали у реализацији софтвера и софтверских пројеката. Посебно је фокус био на особама које су завршиле студије у области ИТ и које су запослене у области информационих технологија. Дакле, студенти нису контактирани. Након добијања потврде за пристанак да желе да учествују у анкетирању, анкете су слате е-маилом, а мањи број анкетних листова у папирној форми подељен је у директном контакту са организацијама од значаја за истраживање. Највећи број попуњених анкета су добијене е-маилом у форми попуњеног MS Word и попуњеног PDF обрасца. Мањи број анкета (укупно 16) попуњено је и достављено у папирној форми. Процес контактирања потенцијалних учесника анкете и прикупљање анкета трајао је од 5.7.2015. до 26.8.2015. године.

#### **5.3.1.4. Обрада података**

Подаци из свих примљених попуњених анкета су унети и обрађени користећи MS Excel. Извршена је анализа потпуности и коректности уноса података у анкету. У појединим анкетама испитаници нису одговорили на свако питање. У појединим анкетама су унети нетачни подаци:

- a) нису у складу одговори типа избора и пратећи редундантни нумерички подаци);
- b) одговори на поједина питања нису коректни (нпр. за алат за управљање пројектима наведен је Power Designer);
- c) одговори на поједина питања типа категоризације нису коректно урађени (нпр. софтвер из области књиговодства или ERP није разврстан у категорију софтвера информационог система, већ у категорију друго).

У настојању да се каснија статистичка анализа заснива на тачним подацима, извршена је корекција, односно узимање у обзир тачних података:

- a) коришћени су као тачни нумерички подаци
- b) приказани су сви подаци, уз напомену који подаци се сматрају нетачним
- c) извршена је правилна категоризација софтвера.

У анализи коректности унетих података такође је примећено неразумевање појединих питања, а поједини испитаници нису могли да се успешно пријаве на реализованом web сајту за коришћење прототипа. Некомплетност података из анкета представља чињеницу и треба је узети у обзир приликом извођења закључака.

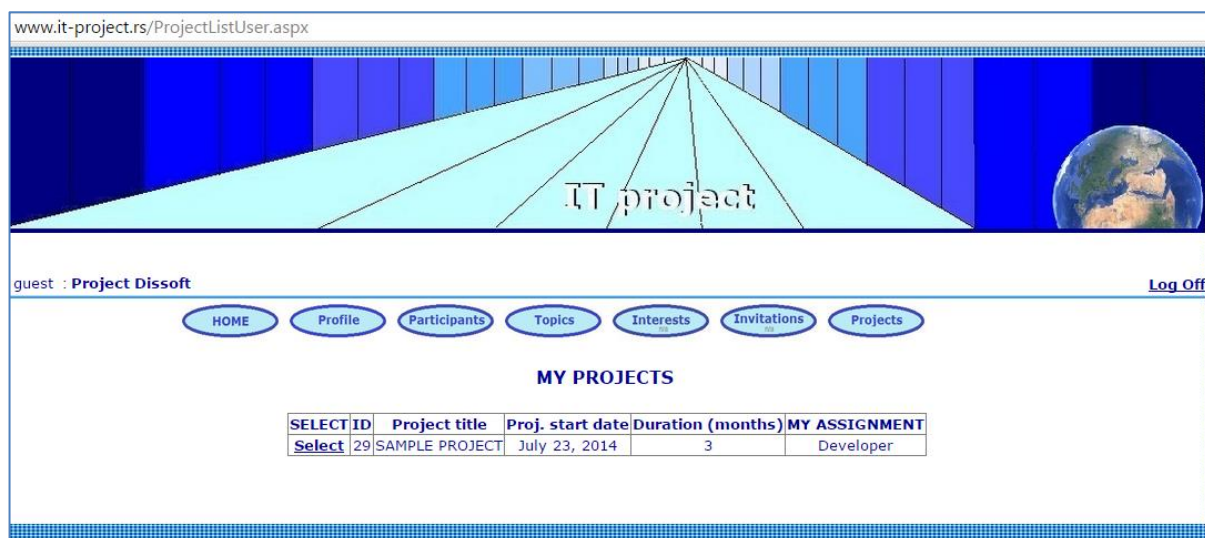
Над подацима из почетног узорка извршена је валидација узорка, у смислу вредновања подобности (eligibility) сваке анкете за даљу обраду, у односу на биографске податке испитаника.

У оквиру припреме статистичке обраде података у оквиру MS Excel алата, за неке податке, који припадају одговорима на питања отвореног типа – нумеричког и описног, извршено је груписање и категоризација (нпр. година рођења, назив занимања и друго). Такође, реализовано је пребројавање података по категоријама и израда графикана.

#### **5.3.2. Резултати истраживања**

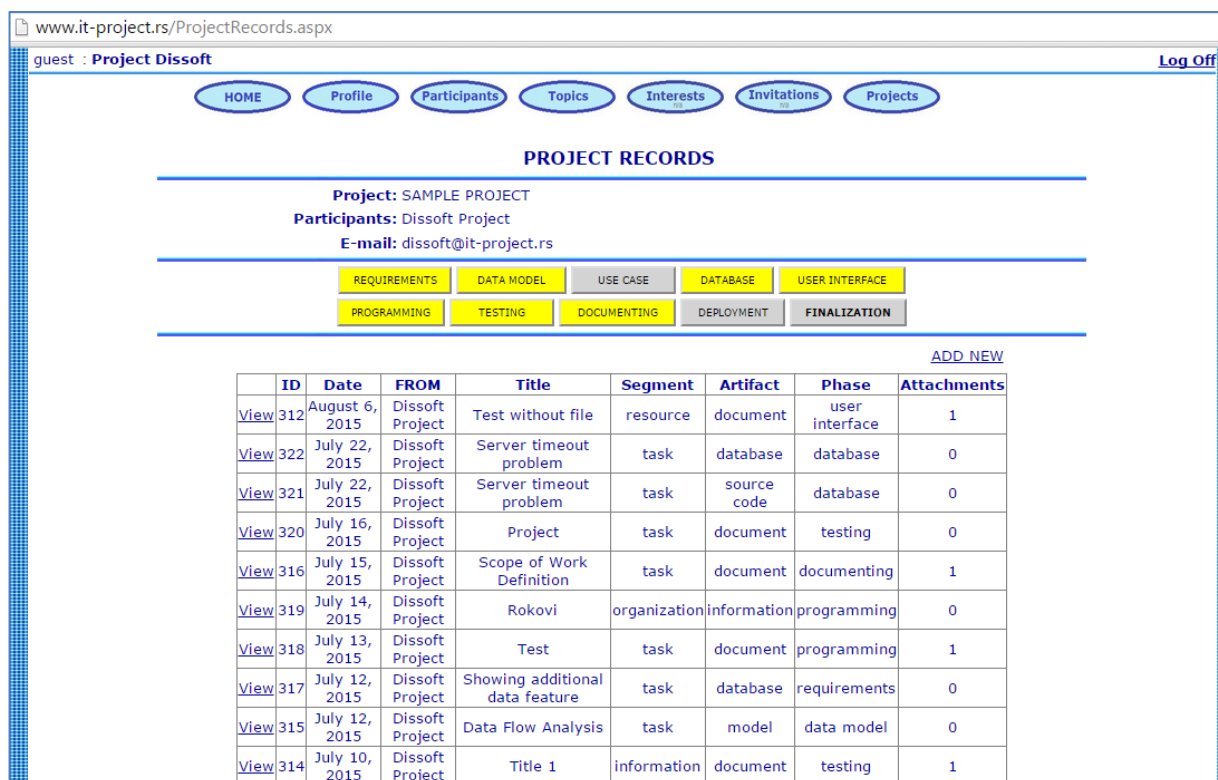
##### **5.3.2.1. Приказ примера резултата пробног коришћења прототипа од стране ИТ кадрова**

Након пријављивања на систем (корисничко име:"dissoft@it-project.rs", шифра:"ABC"), корисник има могућност да види списак пројеката. У овом случају, пробном налогу у статусу „гост“ је додељен само један пробни пројекат.



Слика 5.3.2.1.1. Екран - списак пројеката за пробни налог апликације

Избором линка „select“ омогућен је приказ и унос радних ставки у оквиру наведеног пробног пројекта. Учесници анкете су уносили у оквиру заједничког пробног налога податке, како би тестирали могућности апликације (Слика 5.3.2.2.).



Слика 5.3.2.1.2. Екран - Табеларни приказ списка радних ставки у оквиру пробног пројекта

Једна од пробних ставки са насловом „Showing additional data feature“, која је унета од стране учесника анкетања, описује потребне измене софтвера у односу на захтеве „пробног“ клијента који се одражава на додавање елементарног податка у базу података, што се одражава и на друге елементе решења – кориснички интерфејс, слој пословне логике, документацију, тест-скрипт итд.

guest : Project Dissoft Log Off

[HOME](#) [Profile](#) [Participants](#) [Topics](#) [Interests](#) [Invitations](#) [Projects](#)

**PROJECT RECORD**

Project: SAMPLE PROJECT  
Participants: Dissoft Project

---

**PROJECT RECORD - selected item view** [RECORDS LIST](#)

Date:

From:

Title:

Description: 

Customer has additional requirement for showing some data on users page.  
 This requirement affect:  
 - Main functional documentation  
 - Testing tool script and testing documentation  
 - Database model (adding of stored procedure in main sql creation script)  
 - Business layer (Class 1, Class 5 and Class X)  
 - Interface (Page 3 and Page 12)

Segment:

Artifact:

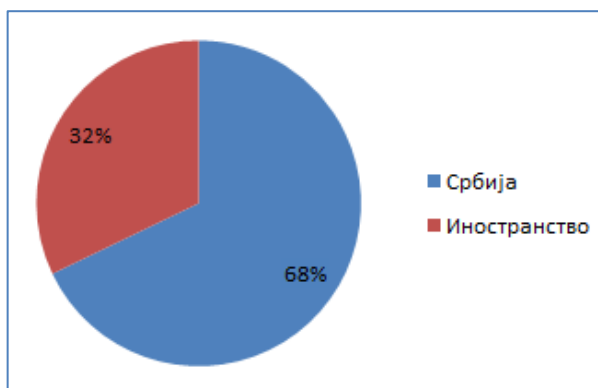
Слика 5.3.2.1.3. Екран - приказ детаља једне пробне радне ставке, која садржи више подставки

### 5.3.2.2. Анализа биографских података почетног узорка истраживања

Почетни узорак чини 165 прикупљених анкета, које је попунио исти број испитаника. У овом одељку биће анализирани одговори на питања која представљају:

- a) основне демографске податке (личне податке, податке о школовању),
- b) стручне биографске податке (искуству у реализацији софтвера, учешће у софтверским пројектима)
- c) ускостручне биографске податке (искуства у тимском дистрибуираном раду у реализацији софтвера, искуства са коришћењем алата за подршку тимском раду и управљању пројектима).

У односу на државу седишта организације где су запослени (за незапослене – држава где станују), укупно је у анкетавању учествовало 53 учесника из 21 стране државе и 112 учесника из Србије. У наставку је дат графикон са приказом процентуалног учешћа партиципаната на основу земље седишта организације где су запослени, а за незапослене испитанике узета је у разматрање држава становања.



Слика 5.3.2.2.1. Процентуални графички приказ броја учесника према државама седишта организације где су запослени или где станију

Табела 5.3.2.2.1. Преглед држава и броја учесника анкете

<b>ДРЖАВА</b>	<b>БРОЈ УЧЕСНИКА</b>
Аустралија	1
Бугарска	1
Црна Гора	1
Данска	1
Финска	1
Француска	3
Грчка	1
Холандија	6
Хрватска	1
Индија	2
Ирска	1
Италија	2
Канада	4
Мађарска	1
Македонија	1
Немачка	4
Русија	3
Словенија	1
Србија	112
Шведска	2
Уједињено Краљевство	7
Сједињене Америчке Државе	9

Табела 5.3.2.2.2. Сумарни приказ основних демографских података испитаника у анкетирању

<b>Година рођења</b>		<b>Пол</b>	
1960-1969	2	М	112
1970-1979	84	Ж	53
1980-1989	71		
1990-1999	5	<b>Област ИТ образовања (назив занимања према завршеном смеру студија)</b>	
		Професорски	63
		Инжењерски	74
<b>Ниво завршеног школовања</b>		Није из ИТ области	3
Средња школа	3		
Висока школа	1		
Факултет	92		
Мастер	44	<b>Тип институције запослења</b>	
Магистар	9	Образовна институција	36
Доктор наука	16	Јавно предузеће или државна управа	21
		ИТ фирма	72
<b>Статус запослења</b>		Привреда	29
Студент	0	Државна фирма	1
Незапослен	5	ИТ фирма, Јавно предузеће	1
Запослен	160		



Табела 5.3.2.2.3. Сумарни приказ стручних биографских података испитаника у анкетирању

Радно место - категорија		Учешће у реализацији софтвера		Учешће у софтверском пројекту	
Општи информатичар	10	Да	137	Да	124
Систем администратор	10	Не	28	Не	41
Консултант	3				
		Укупан број реализованих софтвера		Укупан број пројеката учествовао	
Бизнис аналитичар	3				
Пројектант	7	0	5	0	5
Софтвер девелопер	25	1 до 4	36	1 до 4	53
Програмер	18	5 до 9	24	5 до 9	21
Администратор базе података	4	Преко 10	73	Преко 10	46
Графичар	8				
		Број реализованих професионалних софтвера у употреби		Број година искуства програмирања	
Тестер	4				
Неинформатичар	12	0	7	0	4
Менаџер	19	1 до 4	60	1 до 4	34
Пројектни менаџер	7	5 до 9	16	5 до 9	32
Наставник	23	Преко 10	40	Преко 10	69
Истраживач	4				

Конкретни називи радних места испитаника и њихова категоризација дата је у табели 9.7.1. у прилогу.

У наставку ће бити дат статистички приказ стручних биографских података који се односе на типове софтвера са одговарајућим бројем учесника који одговара наведеној категорији података (табела 5.3.3.4.) као и ускостручних података о искуствима испитаника у тимском дистрибуираном раду у развоју софтвера и коришћењу одговарајућих алата за подршку тимском раду и управљању пројектима (табела 5.3.3.5).

Табела 5.3.2.2.4. Сумарни приказ стручних биографских података испитаника у анкетирању који се односе на типове реализованих софтвера

Софтвер у области информационог система организације		Web sajt	
0	3	0	5
1	10	1	15
2 до 5	43	2 до 5	35
6 до 10	20	6 до 10	17
Преко 10	14	Преко 10	13
Реализовао	26	Реализовао	22
Мултимедијални софтвер		Други софтвери	
0	9	0	7
1	12	1	7
2 до 5	17	2 до 5	25
6 до 10	6	6 до 10	5
Преко 10	4	Преко 10	11
Реализовао	10	Реализовао	17
Категорије других софтвера		Категорије других софтвера	
Игре	1	Софтвер за	18

		управљање уређајима	
Рачунарске мреже	1	Графика	5
Алати за управљање документима и пројектима	1	Мобилне апликације	4
Системски софтвер	1	Научно-истраживачки прототип	4
Софтвер за безбедност	1	Услужни модули	3
Анализа података	2	Образовни софтвер	2

Табела 5.3.2.2.5. Сумарни приказ ускостручних биографских података испитаника - искуства у тимском дистрибуираном раду у развоју софтвера и у коришћењу алата за подршку тимском раду и управљању пројектима

<b>Начин реализације софтверских пројеката</b>			
Тимски	112	Колокацијски	51
Појединачно	26	Дистрибуирано	62
Тимски и појединачно	7	Колокацијски и дистрибуирано	6
<b>Коришћење алата – за тимски рад</b>		<b>Коришћење алата – за управљање пројектима</b>	
Да	87	Да	71
Не	61	Не	73

### 5.3.2.3. Анализа критеријума и метода за утврђивање репрезентативности узорка и опис метода ELIG-RANK и ELIM-F

За потребе овог истраживања, валидним партиципантом у овом истраживању можемо сматрати особу која има знање и искуство у области развоја софтвера и управљања софтверским пројектима, а посебно у области дистрибуираног тимског рада у развоју софтвера и коришћења алата за подршку тимском дистрибуираном раду у развоју софтвера и коришћења алата за управљање софтверским пројектима.

У истраживањима путем анкете најчешће се примењује одбацавање анкета уколико партиципант не поседује одговарајуће особине (није погодан, репрезентативан, „eligible“). Међутим, у овом истраживању није могуће једноставно одредити критеријум репрезентативности узорка, где ће ова тврдња бити илустрована примерима података из спроведене анкете. На пример, поставља се питање колико ниво и врста завршеног школовања представља релевантан критеријум за вредновање испитаника и његове анкете.

а) Да ли треба одбацити анкете учесника са завршеном средњом школом? Строго гледано, требало би одбацити анкете учесника са завршеном средњом школом с обзиром да немају довољан ниво формалног стручног образовања. Међутим неки од учесника (у овом узорку сви) са завршеном средњом школом су један део свог живота студирали и стекли извесна стручна знања, а такође имају изузетно висок ниво професионалног искуства у развоју софтвера у тимском дистрибуираном окружењу и висок ниво познавања одговарајућих софтверских алата за подршку тимском дистрибуираном развоју софтвера. Дакле, такве анкете су веома драгоцене и не би требале бити одбачене. Такође, испитаници који су завршили средњу школу нису разумели неке елементе анкете управо из разлога што нису прошли комплетно школовање у овој области и из наведених разлога нису правилно одговорили на поједина питања. У овом контексту, вредност оваквих анкета се доводи у питање.

б) С друге стране, у оквиру анкетираних учесника, један број испитаника поседује формални ниво образовања – доктор наука из области информатике.

Међутим, на основу анкете, велики број испитаника у овој категорији нема искуства у развоју софтвера, није учествовао у софтверским пројектима, није радио у тимском дистрибуираном развоју софтвера и није користио алате за подршку тимском раду и управљању пројектима. Из наведених разлога се поставља питање – да ли одбацити анкете ових испитаника?

с) У односу на формално образовање, поставља се питање који утицај има ова карактеристика испитаника на значај у овом истраживању, с обзиром да у анкети није предвиђено питање којим би испитаници описали да ли су у оквиру свог формалног образовања имали наставу из области управљања софтверским пројектима или дистрибуираног тимског развоја софтвера. Претпоставка је да су испитаници, који поседују формално образовање у инжењерској области информатике и рачунарства, имали одговарајуће сродне наставне предмете у току студија, па би се испитаници са инжењерског профила сматрали „погоднијим“ узорком.

Претходна дискусија указује на проблем избора критеријума „подобности“ („eligibility“) члана популације (тј. појединца или његове анкете) како би био сматран елементом узорка. У раду [Judson et al, 2013] описана је потреба да се утврди елигибилност елемента популације у истраживањима повезаности података из два истраживачка центра. У наведеном раду предложен је приступ додељивању одређене вредности сваком критеријуму за селекцију елемената узорка, где се вредност израчунава на основу вероватноће наступања одређених вредности. У раду [Battistin&Rettore, 2008] разматране су граничне ситуације припадности члана популације некој групи према одређеном критеријуму и потреба да се користе правила елигибилности уз примену фази логике.

У оквиру овог истраживања, можемо изабрати један од три могућа приступа, односно методе:

1. метод (назив „Елиминациони метод“, кратак назив: ELIM) - Дефинисањем и применом критеријума вредновања елемената узорка (односно вредновања учесника анкете), одређене анкете би се одбациле (остају као погодне само анкете чији испитаници задовољавају све критеријуме). У даљој статистичкој обради узимале у обзир само „погодне“ анкете. У наставку ће бити дефинисани критеријуми за први приступ, који ће бити примењени у овом истраживању:

- K1 - Ниво завршеног школовања: минимално висока школа
- K2 - Област ИТ образовања: инжењерски
- K3 - Тип институције запослења: све осим образовне институције
- K4 - Статус запослења: запослен
- K5 - Радно место: Програмер, Софтвер девелопер, општи информатичар, консултант, менаџер, пројектни менаџер, истраживач, пројектант
- K6 - Учешће у реализацији софтвера: да
- K7 - Учешће у софтверском пројекту: да
- K8 - Укупан број реализованих софтвера: од 5 и више
- K9 - Укупан број софтверских пројеката на којима је учествовао: од 2 и више
- K10 - Број реализованих професионалних софтвера у употреби: од 2 и више
- K11 - Број година искуства програмирања: од 5 и више
- K12 - Број софтвера из области информационих система: од 2 и више
- K13 - Број софтвера типа web сајта: од 2 и више
- K14 - Тимски рад: да
- K15 - Дистрибуирани рад: да
- K16 - Коришћење алата за тимски рад: да
- K17 - Коришћење алата за управљање пројектима: да

2. метод (назив: „Елигибиност рангирањем“, кратак назив: ELIG-RANK) Применом рангирања чланова популације, где би се узеле у обзир све анкете, али би се за сваку анкету, на основу општих, стручних и ужестручних података утврдио ниво (број поена или проценат) „поверења“ односно „значаја“ одговарајуће анкете. Дакле, свака анкета би имала одређену нумеричку вредност и резултати те анкете би се узимале у обзир у

заједничкој статистичкој обради са одређеним нумеричким или процентуалним нивоом вредновања. У наставку ће бити дефинисане вредности које се додељују одређеној врсти података из сваке категорије, а које се могу доделити конкретним вредностима из анкета.

*Општа биографија - школовање*

- OBS1 - Ниво завршеног школовања: средња школа (1), висока школа (2), факултет (3), мастер (4), магистар (5), докторат (6)
- OBS2 - Област ИТ образовања: није из ИТ (0), професорски (1), инжењерски (2)

*Општа биографија - запослење*

- OBZ1 -Тип институције запослења: образовна институција (1), остало (2)
- OBZ2- Статус запослења: незапослен (1), запослен (2)
- OBZ3- Радно место: Програмер (3), Софтвер девелопер (3), општи информатичар (2), консултант (4), менаџер (4), пројектни менаџер (5), истраживач (6), систем администратор (2), бизнис аналитичар (2), пројектант (5), администратор базе података(2), графичар (2), тестер 2), неинформатичар (0), наставник (1)

*Стручна биографија - софтвер*

- SBS1- Учешће у реализацији софтвера: да(1), не (0)
- SBS2- Укупан број реализованих софтвера: 0 (0), 1-4 (1), 5-9 (2), преко 10 (3)
- SBS3- Број реализованих професионалних софтвера у употреби: 0 (0), 1-4 (1), 5-9 (2), преко 10 (3)
- SBS4- Број година искуства програмирања: 0 (0), 1-4 (1), 5-9 (2), преко 10 (3)
- SBS5- Број софтвера из области информационаих система: 0 (0), 1 (1), 2-5 (2), 6-10(3), преко 10 (4), реализовао (2)
- СБС6- Број софтвера тима web сајта: 0 (0), 1 (1), 2-5 (2), 6-10(3), преко 10 (4), реализовао (2)

*Стручна биографија – софтверски пројекат*

- SBP1- Учешће у софтверском пројекту: да (1), не (0)
- SBP 2- Укупан број софтверских пројеката на којима је учествовао: 0(0), 1-4 (1), 5-0 (2), преко 10 (3)

*Ускостручна биографија*

- USB1- Тимски рад: да(1), не (0)
- USB2- Дистрибуирани рад: да(1), не (0)
- USB3- Коришћење алата за тимски рад: да(1), не (0)
- USB4- Коришћење алата за управљање пројектима: да(1), не (0)

Укупан ELIG-RANK појединачне анкете се израчунава на основу збира поена за сваку групу питања, при чему свака група питања има свој тежински фактор, односно ниво значаја и утицаја на коначан ELIG-RANK резултат. Формула на основу које се израчунава ELIG-RANK за сваког испитаника у анкетирању је представљена у наставку:

$$\begin{aligned} \text{ELIG-RANK} = & \\ & \text{OBSf} * \sum (\text{OBS}_i) + \text{OBZf} * \sum (\text{OBZ}_j) + \\ & \text{SBSf} * \sum (\text{SBS}_k) + \text{SBPf} * \sum (\text{SBP}_l) + \\ & \text{USBf} * \sum (\text{USB}_m) \end{aligned}$$

Слика 5.3.2.3.1. Формула за израчунавање ELIG-RANK

При чему је:

- OBSf = тежински фактор за податке из опште биографије који се односе на школовање и у овом истраживању ће им бити додељена вредност 1
- OBZf= тежински фактор за податке из опште биографије који се односе на запослење и у овом истраживању ће им бити додељена вредност 2
- SBSf= тежински фактор за податке из стручне биографије који се односе на развој софтвера и у овом истраживању ће им бити додељена вредност 3
- SBPf= тежински фактор за податке из стручне биографије који се односе на учешће у софтверским пројектима и у овом истраживању ће им бити додељена вредност 4

- USBf= тежински фактор за податке из ускостручне биографије и у овом истраживању ће им бити додељена вредност 5

Применом ELIG-RANK приступа, подобним ће се сматрати испитаник и одговарајућа анкета уколико је укупан проценат ELIG-RANK поена већи од 50%.

3. метод (назив „Елиминациони-флексибилни метод“, кратак назив: ELIM-F) – применом критеријума за вредновање анкета у циљу елиминације анкета које нису погодне (дефинисаних методом ELIM), за сваку анкету одређује се укупан збир поена и проценат у односу броја поена и максималног броја поена. На основу процента извршава се сортирање података и издвајају се (за потребе примене методе ELIM) анкете које имају као коначни резултат 100% заступљеност особина у односу на свих 17 критеријума. У наставку примењује се проширење методе ELIM (односно метода ELIM-F) тако што се подаци о осталим анкетама групишу у 2 групе – анкете које имају за коначан резултат више од 50% и анкете које имају мање од 50%. **Као коначан узорак у даљим анализама узима се група анкета која имају више од 50% поена.**

Анализа резултата анкете заправо се односи на анализу одговора на питања од броја 4 до броја 15. У оквиру овог истраживања биће примењен следећи приступ:

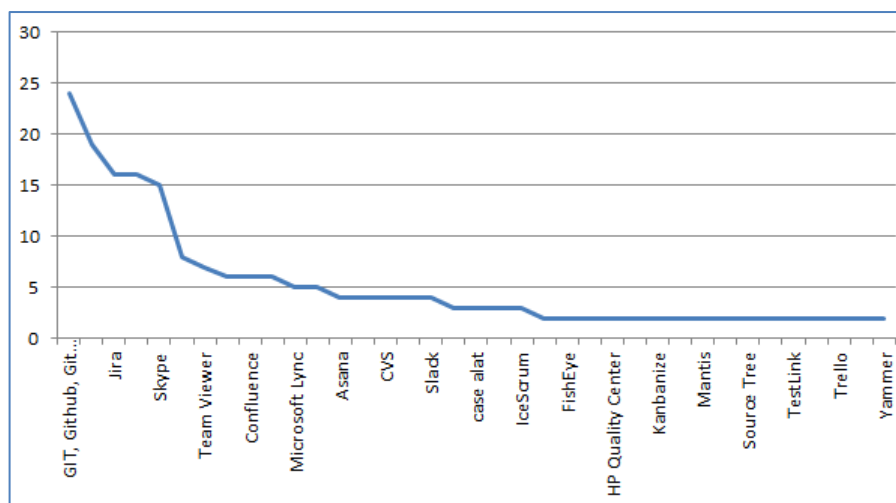
1. За питања отвореног типа (4 б и 5 б) биће извршена систематизација резултата свих анкета
2. за остала питања биће примењена метода ELIM-F.

У оквиру прилога представљена је табела резултата примене ELIM-F методе на све анкете, где су посебно обележени подаци о анкетама које су „освојиле“ 100% бодова (укупно 5 анкета). **Према резултатима примене ELIM-F методе, од укупно 165 анкета, 128 анкета има више од 50% поена и подаци из тих анкета ће се приказати сумарно у наставку.**

#### 5.3.2.4. Сумарни приказ одговора у вези најчешће коришћених алата за подршку тимском раду и софтверском пројектном менаџменту

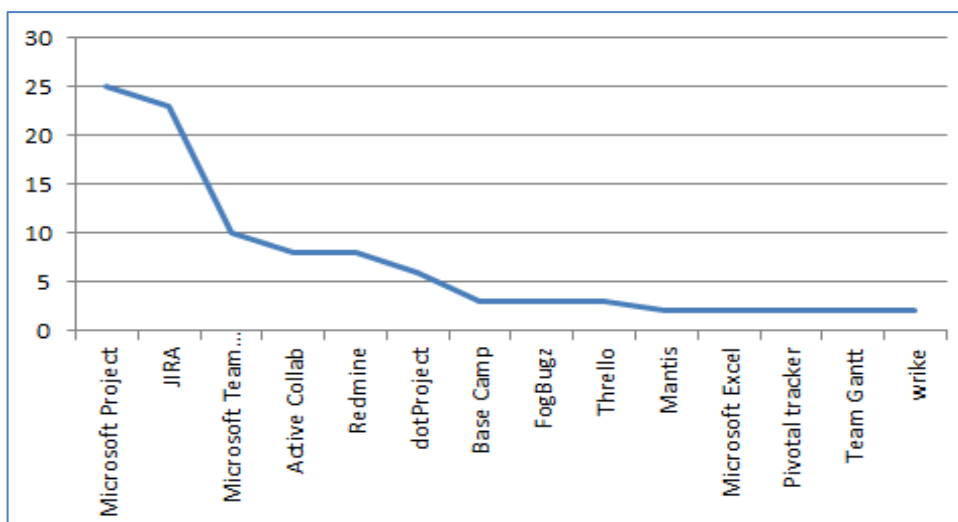
У наставку су дати графикони са подацима из свих анкета (без примене ELIM-F) о укупном броју одговора на питања која се односе на алате. У оквиру графикана су приказани само најзаступљенији одговори, док су у прилогу дати детаљни подаци о свим одговорима.

##### ➤ **Питање 4 б) Које алате за подршку тимском раду сте користили?**



Слика 5.3.2.4.1. Графички приказ броја одговора у вези коришћења алата за подршку тимском раду

➤ **Питање 5 б) Које алате за подршку управљању пројектом сте користили?**



Слика 5.3.2.4.2. Графички приказ броја одговора у вези коришћења алата за подршку пројектном менаџменту

Детаљан списак свих наведених алата са бројем одговора испитаника дат је у прилогу.

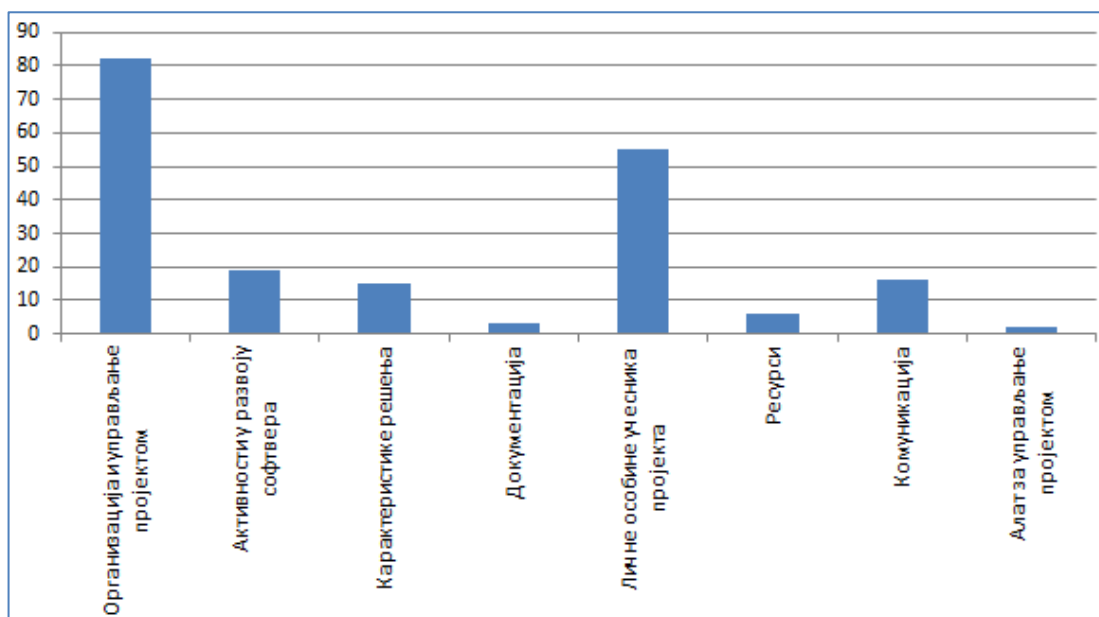
**5.3.2.5. Сумарни приказ одговора у вези фактора утицаја на успех и неуспех софтверског пројекта**

➤ **Питање 6 - Шта по вашем мишљењу и искуству највише утиче на успех софтверског пројекта?**

У оквиру одељка са прилозима дата је табела са комплетном листом свих група фактора, на основу којих су груписани конкретни фактори који утичу на успех софтверског пројекта. Такође, приказани су и појединачни комплетни одговори. У овом одељку дат је сумарни приказ фактора успеха.

Табела 5.3.2.5.1. Сумарни приказ броја одговора према групама фактора који утичу на успех софтверског пројекта

ГРУПА ФАКТОРА	БРОЈ ОДГОВОРА
Организација и управљање пројектом	82
Активности у развоју софтвера	19
Карактеристике решења	15
Документација	3
Личне особине учесника пројекта	55
Ресурси	6
Комуникација	16
Алат за управљање пројектом	2

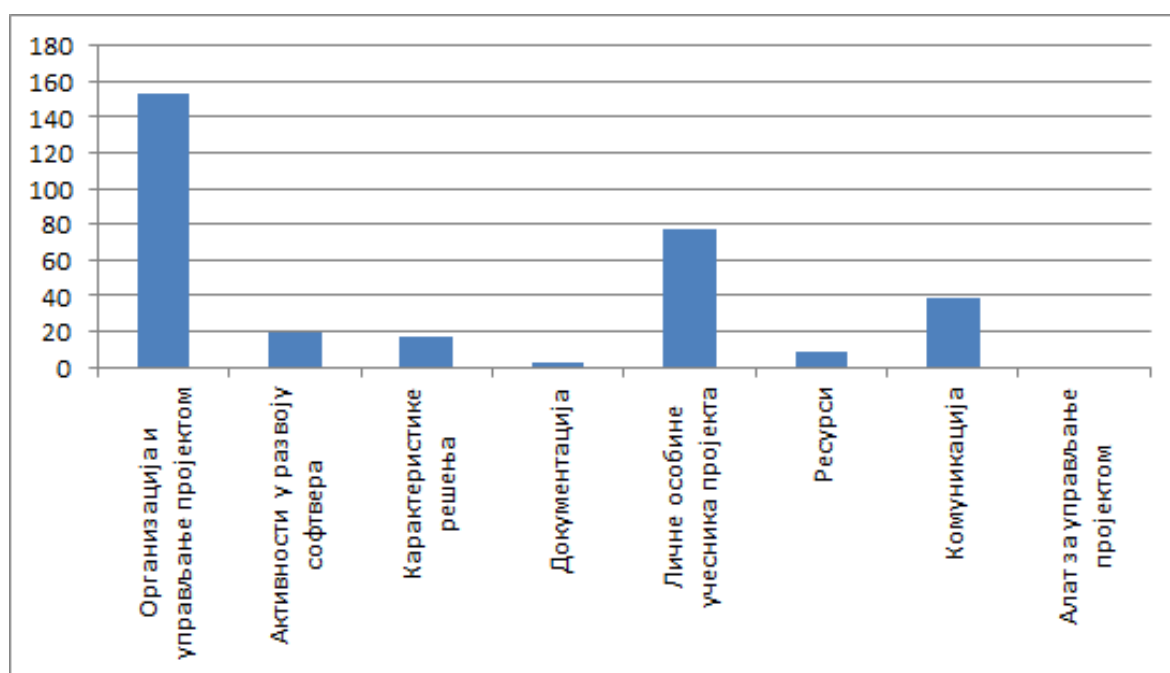


Слика 5.3.2.5.1. Графички приказ броја одговора у вези фактора утицаја на успех пројекта

➤ **Питање 7 - Шта по вашем мишљењу и искуству највише утиче на неуспех софтверског пројекта?**

Табела 5.3.2.5.2. Сумарни приказ броја одговора према групама фактора који утичу на неуспех софтверског пројекта

ГРУПА ФАКТОРА	БРОЈ ОДГОВОРА
Организација и управљање пројектом	153
Активности у развоју софтвера	19
Карактеристике решења	17
Документација	3
Личне особине учесника пројекта	77
Ресурси	8
Комуникација	39
Алат за управљање пројектом	0

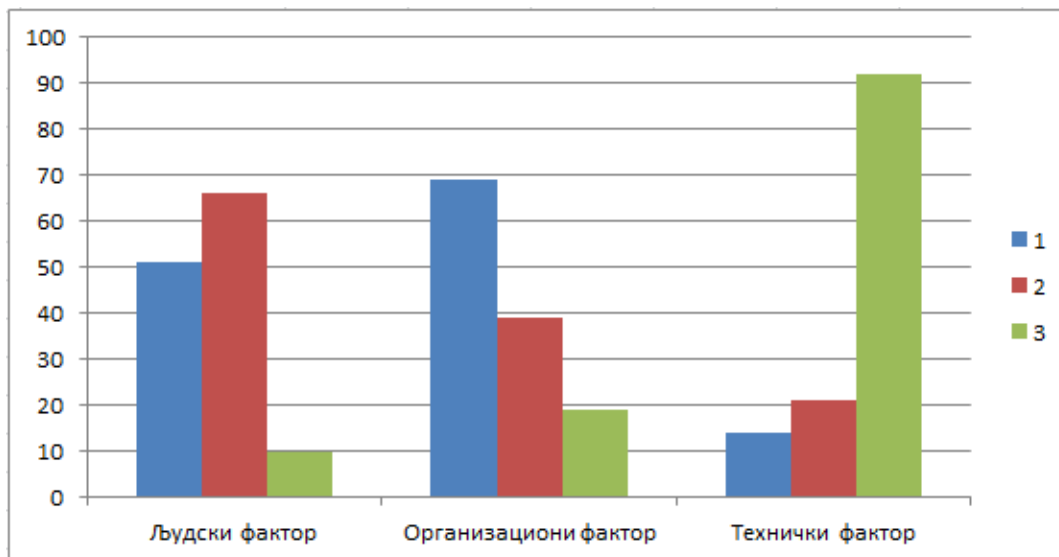


Слика 5.3.2.5.2. Графички приказ броја одговора у вези фактора утицаја на неуспех пројекта

- **Питање 8 – Рангирати значај фактора утицаја на успех пројекта (1- најважнији, 2 – средње важан, 3 – најмање важан):**

Табела 5.3.2.5.3. Сумарни приказ броја одговора у рангирању фактора утицаја на успех пројекта

Рангирање фактора	Људски фактор	Организациони фактор	Технолошки фактор
1- Најважнији	51	69	14
2 – Средње важан	66	39	21
3 - Најмање важан	10	19	92



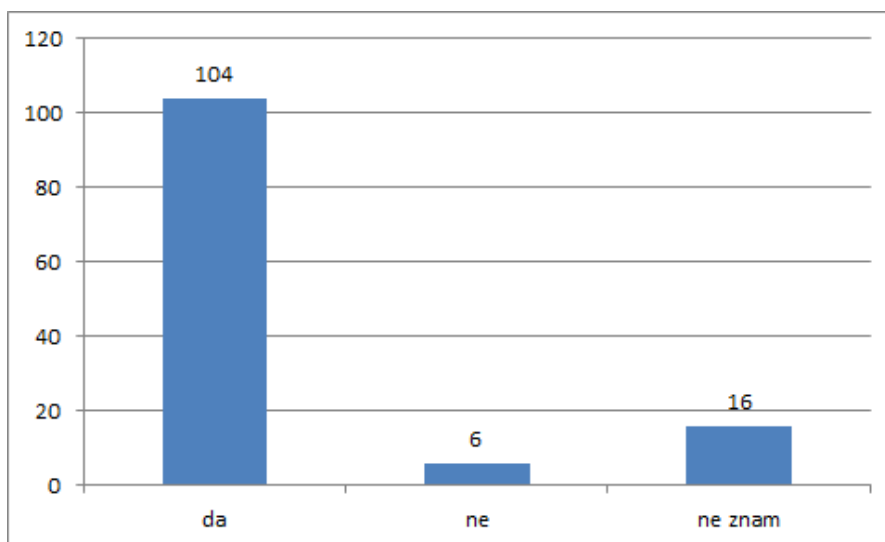
Слика 5.3.2.5.3. Графички приказ броја одговора у вези рангирања фактора утицаја на успех софтверског пројекта

Упоређивањем сумарних резултата који се односе на одговоре на питања отвореног типа (6 и 7) и питања типа рангирања (8) можемо закључити да испитаници сматрају да је најважнији фактор који утиче на успех или неуспех пројекта организациони фактор, затим људски фактор, док за технолошки фактор испитаници сматрају да најмање утиче на успех софтверског пројекта.

### 5.3.2.6. Сумарни приказ одговора у вези значаја примене и развоја, као и потребних карактеристика информационог система за подршку управљању софтверским пројектима

- **Питање 9 – Да ли сматрате да је потребно користити и развити информациони систем за праћење успешности софтверског пројекта?**



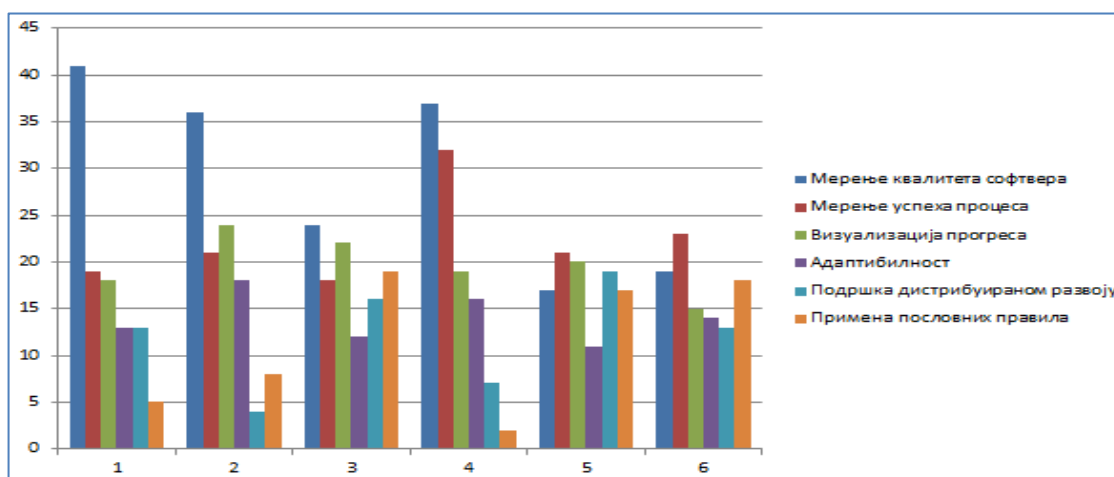


Слика 5.3.2.6.1. Графички приказ броја одговора у вези потребе за коришћењем и развојем информационих система за подршку управљању пројектом

➤ **Питање 10 – Заокружите особине (једну или више) које сматрате да такав информациони систем треба да има и рангирајте их по важности.**

Табела 5.3.2.6.1. Сумарни приказ броја одговора у рангирању карактеристика информационог система подршке управљању софтверским пројектима

Ранг	К1	К2	К3	К4	К5	К6
	Мерење квалитета софтвера	Мерење успешности процеса развоја	Визуализација података о прогресу	Адаптибилност на промене	Подршка дистрибуираном развоју	Примена пословних правила
1	41	36	24	37	17	19
2	19	21	18	32	21	23
3	18	24	22	19	20	15
4	13	18	12	16	11	14
5	13	4	16	7	19	13
6	5	8	19	2	17	18
<b>СУМ</b>	<b>280</b>	<b>290</b>	<b>368</b>	<b>269</b>	<b>360</b>	<b>339</b>
<b>РЕЛ</b>	<b>720</b>	<b>710</b>	<b>632</b>	<b>731</b>	<b>640</b>	<b>661</b>
<b>%</b>	<b>72</b>	<b>71</b>	<b>63.2</b>	<b>73.1</b>	<b>64</b>	<b>66.1</b>

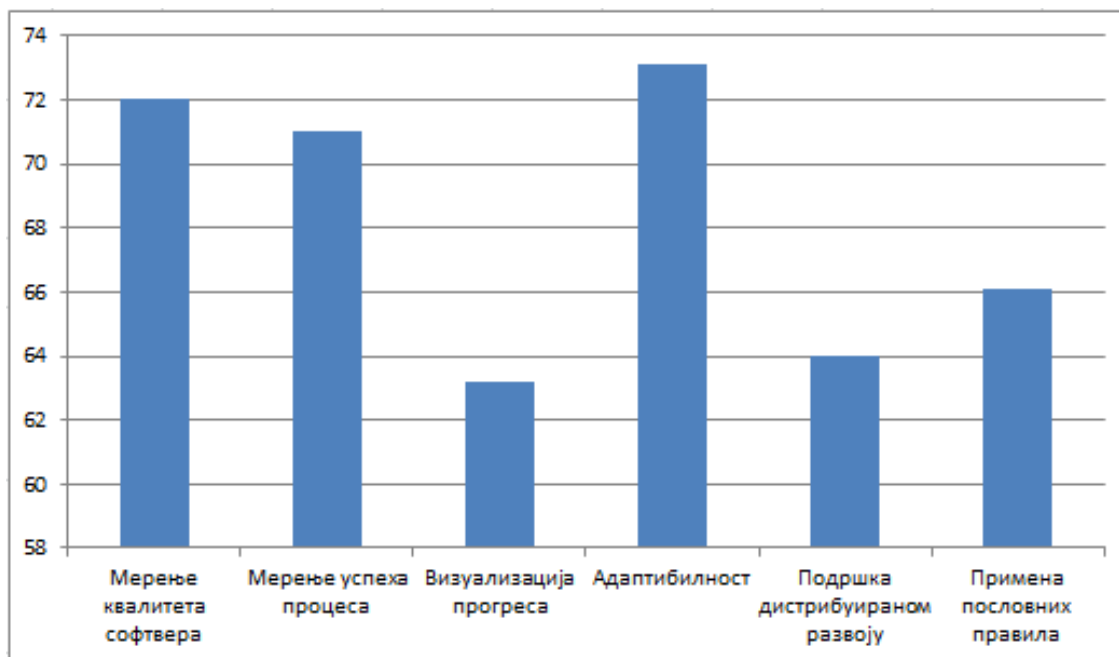


Слика 5.3.2.6.2. Графички приказ оцена значаја појединих карактеристика информационог система за подршку управљању софтверским пројектима

Сумарни подаци приказани аналитички и графички не дају довољно јасну слику о релевантности појединих карактеристика. Из наведеног разлога извршено је додатно сумирање, израчунавање релативне важности и процента.

$$\begin{aligned} \text{СУМ} &= K_i (1) * 1 + K_i (2) * 2 + K_i (3) * 3 + K_i (4) * 4 + K_i (5) * 5 + K_i (6) * 6 \\ \text{РЕЛ} &= 1000 - \text{СУМ} \\ \% &= \text{РЕЛ} * 100 / 1000 \end{aligned}$$

Слика 5.3.2.6.3. Формуле за израчунавање значаја појединих карактеристика информационог система за подршку управљању софтверским пројектима



Слика 5.3.2.6.4. Графички приказ сумарних података о значају појединих карактеристика информационог система за подршку управљању софтверским пројектима

На основу сумарног приказа одговора на питање 9, може се закључити да већина испитаника (укупно 104) сматра да је потребно развити и користити информациони систем за подршку управљању софтверским пројектима.

На основу сумарног приказа одговора на питање 10, може се закључити да је адаптивност на промене најважнија карактеристика која је потребно да поседује информациони систем за подршку управљању софтверским пројектима. Мање, али још увек веома значајна је подршка мерењу квалитета софтвера и мерењу успеху процеса.

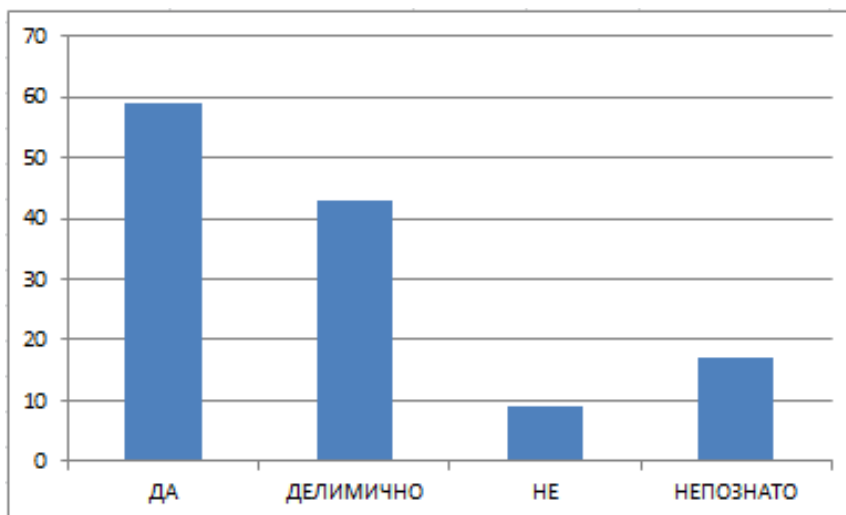
### 5.3.2.7. Сумарни приказ одговора у вези употребљивости развијеног прототипа, предности, недостатака и могућности унапређења

- **Питање 11 – Да ли сматрате да би web апликација за подршку управљању софтверским пројектима [www.it-project.rs](http://www.it-project.rs) могла да се користи за подршку управљању софтверским пројектима у реалној пракси?**

Табела 5.3.2.7.1. Сумарни приказ одговора о употребљивости развијеног решења

ОДГОВОР	УКУПНО АНКЕТА
ДА	59
НЕ	9
<b>ДЕЛИМИЧНО:</b>	<b>43</b>
• не знам (уз образложење)	38

• да, уз поправке	4
• можда	1
<b>НЕПОЗНАТ ОДГОВОР:</b>	<b>17</b>
• није успео да се пријави	5
• не знам (без образложења)	7
• није користио	1
• није одговорио	4



Слика 5.3.2.7.1. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа

На основу сумарног приказа података о ставовима испитаника у вези употребљивости развијеног прототипа, највећи број одговора (59) се односи на позитиван одговор (да), а такође испитаници сматрају да је делимично употребљив (укупно 43 испитаника).

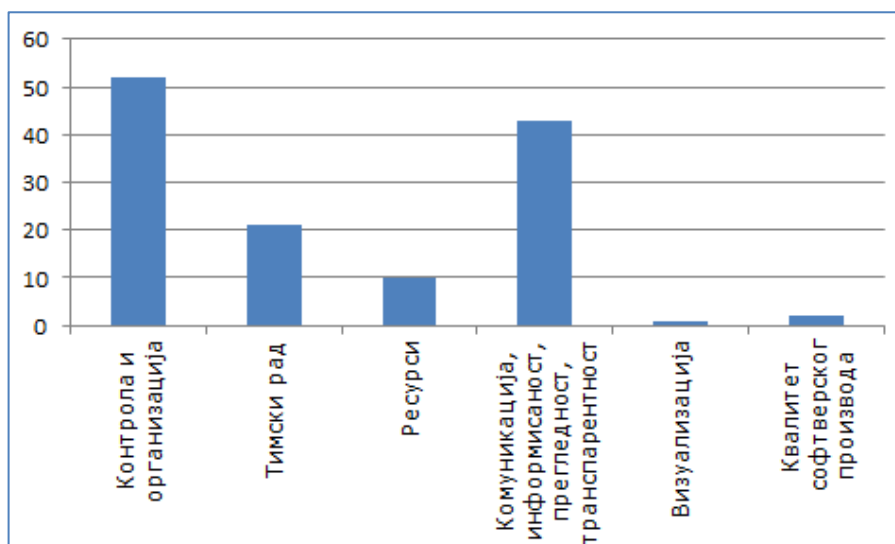
➤ **Питање 12 – Које су предности коришћења овакве апликације?**

Одговори отвореног типа у оквиру овог питања су груписани по категоријама, а затим додатно су категорије груписане. У наставку ће бити приказани сумарни подаци о одговорима који су груписани у оквиру 2 групе карактеристика.

1. група карактеристика – Утицај на успешност процеса у управљању пројектима, функционалне карактеристике

Табела 5.3.2.7.2. Сумарни приказ одговора о предностима коришћења развијеног решења – утицај на успешност процеса управљања пројектима и функционалне карактеристике

КАРАКТЕРИСТИКА	УКУПНО АНКЕТА
Контрола и организација	52
Тимски рад	21
Ресурси	10
Комуникација, информисаност, прегледност, транспарентност	43
Визуализација	1
Квалитет софтверског производа	2

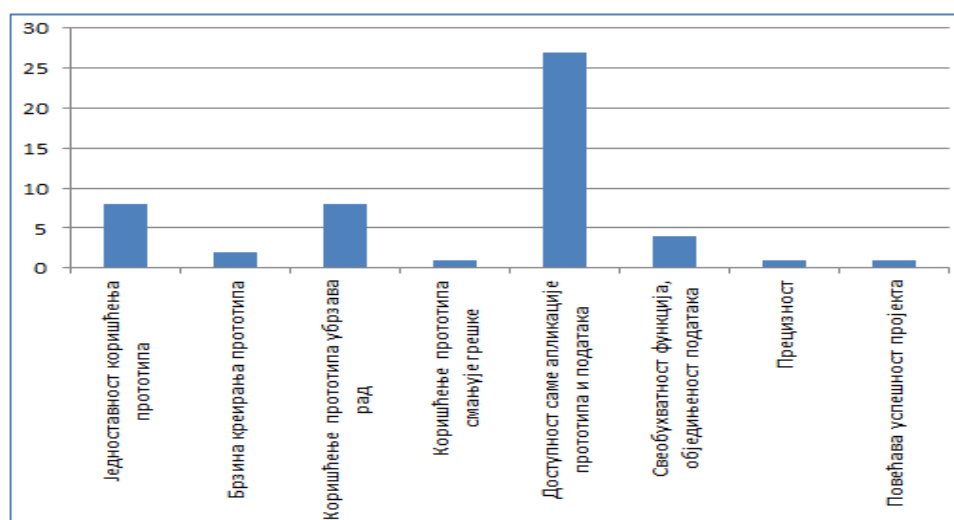


Слика 5.3.2.7.2. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа – утицај примене на процесе управљања пројектима и функционалне карактеристике

2. група карактеристика - Карактеристике прототипа [www.it-project.rs](http://www.it-project.rs), намена и утицај на успех пројекта

Табела 5.3.2.7.3. Сумарни приказ одговора о предностима коришћења развијеног решења – карактеристике прототипа и утицај на успех пројекта

КАРАКТЕРИСТИКА	УКУПНО АНКЕТА
Једноставност коришћења прототипа	8
Брзина креирања прототипа	2
Коришћење прототипа убрзава рад	8
Коришћење прототипа смањује грешке	1
Доступност саме апликације прототипа и података	27
Свеобухватност функција, обједињеност података	4
Прецизност	1
Повећава успешност пројекта	1



Слика 5.3.2.7.3. Графички приказ сумарних података о ставовима испитаника у вези употребљивости развијеног прототипа – карактеристике прототипа и утицај на успех пројекта

На основу сумарних података о предностима коришћења прототипа [www.it-project.rs](http://www.it-project.rs) може се закључити да се предности могу разматрати у два сегмента:

- а) у делу подршке процесима у управљању пројектима – највећи број испитаника сматра да је предност коришћења прототипа апликације у подршци организацији и контроли, као и комуникацијама у пројекту
- б) у делу карактеристика самог прототипа – највећи део испитаника сматра да прототип исказује особине доступности, а нешто мањи, али значајан број испитаника сматра да је позитивна карактеристика и једноставност коришћења, такође сматра да коришћење прототипа убрзава рад и да прототип исказује особине подршке свеобухватности функција и обједињености података.

У оквиру одговора на питање бр. 12 отвореног типа, неки од испитаника су наводили процене предности коришћења наведеног прототипа са аспекта намене:

- „За мање пројекте у оквиру научних организација“,
- „Организациони део може имати користи од овакве апликације у случају да се ради о систему са великим бројем пројеката“
- „У већем тимском окружењу оваква апликација доприноси флуидности развоја софтвера, праћење развоја, мерење времена израде, приоритет задатака.“
- „Уско специјализована апликација, за управљање искључиво софтверским пројектима. Већина сличних апликација су прављене за управљање билокојим пројектима, па је често потребно, а некада и немогуће, прилагодити их софтверским пројектима.“

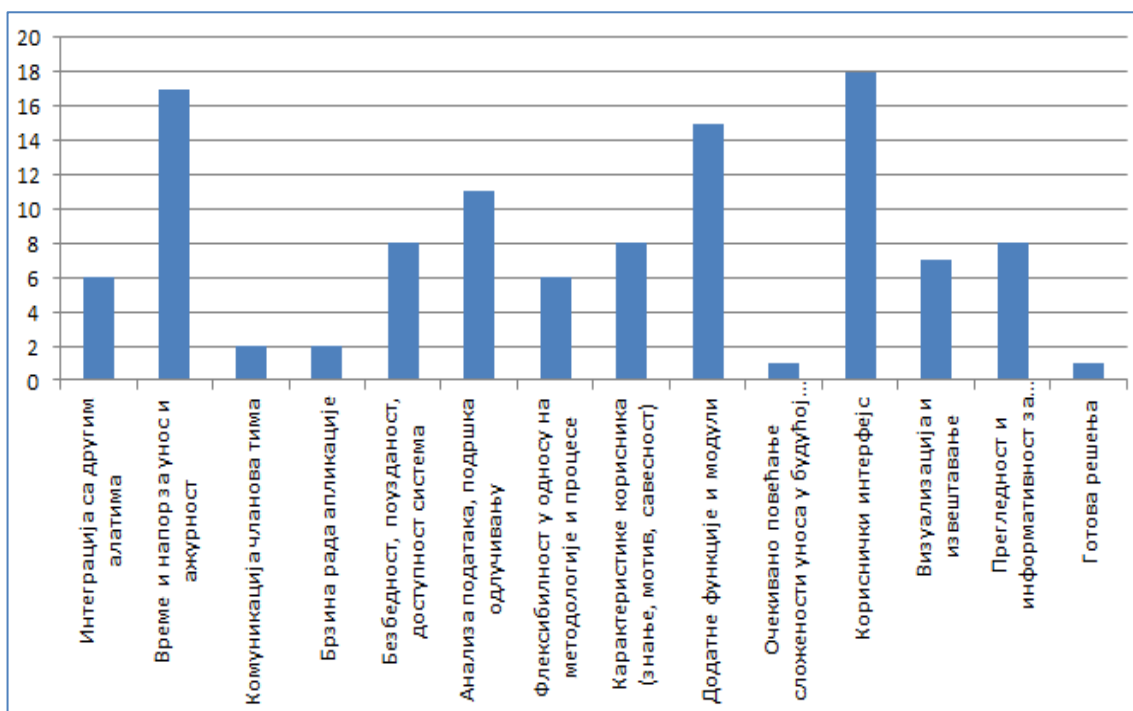
➤ **Питање 13 – Који су недостаци коришћења овакве апликације?**

Табела 5.3.2.7.4. Сумарни приказ група карактеристика са недостацима коришћења прототипа

ГРУПА КАРАКТЕРИСТИКА	БРОЈ АНКЕТА
Интеграција са другим алатима	6
Време и напор за унос и ажурност	17
Комуникација чланова тима	2
Брзина рада апликације	2
Безбедност, поузданост, доступност система	8
Анализа података, подршка одлучивању	11
Флексибилност у односу на методологије и процесе	6
Карактеристике корисника (знање, мотив, савесност)	8
Додатне функције и модули	15
Очекивано повећање сложености уноса у будућој верзији	1
Кориснички интерфејс	1
Визуализација и извештавање	18
Прегледност и информативност за коришћење	7
Готова решења	8
Интеграција са другим алатима	1

Детаљан приказ недостатака коришћења развијеног прототипа дат је у прилогу. У оквиру овог питања дати су и нетачни одговори (укупно 2):

- не постојање комуникације (писане и размене докумената) – размена докумената је коришћењем наведеног система уз опције upload i download.
- ако комуникација иде преко овакве апликације, она треба да буде што једноставнија за употребу, прегледна (субјективна констатација),



Слика 5.3.2.7.4. Графички приказ сумарних података о недостацима реализованог прототипа

На основу претходно приказаних сумарних података, може се закључити да се највећи недостаци развијеног прототипа односе на дизајн корисничког интерфејса, оптерећење (време и напор) потребни за ажурни унос података (посебно за мање пројекте), недовољна развијеност апликације (недостатак неких софтверских функција), недостатак подршке за анализу података и подршку одлучивању, безбедност и поузданост и прегледност и информативност о начину коришћења апликације.

Одговори на питање 14 – „Које софтверске функције сматрате да треба да има оваква апликација, а нису подржане у постојећем решењу?“ и питање 15 – „коментари и сугестије“ дати су у прилогу.

### 5.3.3. Анализа резултата са становишта истраживачких питања и хипотеза

У овом одељку биће приказана анализа резултата анкете са становишта истраживачких питања и хипотеза.

#### 5.3.3.1. Анализа одговора на истраживачка питања

1. Која су искуства у дистрибуираном развоју и управљању софтверским пројектима?

1.1. Који начин рада се примењује у развоју софтвера у смислу тимских и локацијских карактеристика?

ОДГОВОР: Према искуству 119 испитаника која су одговорила на ова питања, софтвер и софтверски пројекти у претходном периоду у већој мери су реализовани тимски (112 испитаника), док је подједнако реализација организована колокацијски (51 испитаник) и дистрибуирано (62 испитаника).

1.2. Који алати се користе у тимском раду и управљању софтверским пројектима?

ОДГОВОР:

а) Од укупно 148 испитаника који су одговорили на питање да ли су користили алате за подршку тимском раду, 87 је одговорило позитивно, а 61 негативно. Учесници анкете који су користили алате за подршку тимском раду навели су да су најчешће коришћени алати: Git, Jira, Skype, Team Viewer, Confluence, Microsoft Lync, Asana, CVS и други.

b) Од укупно 144 испитаника који су одговорили на питање да ли су користили алате за управљање пројектима, 71 је одговорило позитивно, док је 73 одговорило негативно. Најчешће коришћени алати за управљање пројектима су: Microsoft Project, JIRA, Microsoft Team Foundation Server, Active Collab, Redmine, dotProject, BaseCamp и други.

1.3. Према ставовима испитаника, који су фактори успеха и неуспеха, односно који је утицај људског, организационог и технолошког фактора на успех софтверског пројекта?

ОДГОВОР:

a) Према ставовима испитаника у рангирању људског, организационог и технолошког фактора, највише испитаника (69) је одговорило да је највише утиче организациони фактор, затим људски фактор (51 испитаника је одговорило да је најважнији људски фактор), а фактор који најмање утиче је технолошки фактор (14 испитаника је одговорило да је најважнији фактор).

б) Према ставовима испитаника, на основу анализе и класификације одговора отвореног типа, фактори који највише утичу на успех пројекта односе се на организацију и управљање пројектом, као и личне особине учесника пројекта итд.

Такође в) Према ставовима испитаника, на основу анализе и класификације одговора отвореног типа, фактори који највише утичу на неуспех пројекта су такође организација и управљање пројектом, као и личне особине учесника пројекта итд.

2. Развој информационог система за подршку управљању софтверским пројектима:

2.1. Да ли постоји потреба за коришћењем и развојем информационог система за подршку дистрибуираном развоју софтвера, односно управљању софтверским пројектима у дистрибуираном окружењу?

ОДГОВОР: Према ставовима испитаника, у највећој мери (104/122) испитаника сматра да треба да се користи или развије информациони систем за подршку дистрибуираном развоју софтвера, односно управљању софтверским пројектима у дистрибуираном окружењу.

2.2. Какве би биле карактеристике таквог система и које су најзначајније?

ОДГОВОР: На питање о рангирању функционалних карактеристика које су издвојене као значајне, највећи број испитаника (41) је издвојио мерење квалитета софтвера као најзначајнију потребну функционалност, затим адаптивност на промене (37) и мерење успешности процеса развоја (36). Средње релевантна, али не и безначајна је функција визуализације података о прогресу (24), док је подршка примени пословних правила (19) и подршке дистрибуираном развоју (17) најмање заступљена у одговорима анкета. Применом сумирања броја анкета у комбинацији са тежинским факторима и инверзијом, добијени резултати указују да је адаптивност најзначајнија особина софтвера и софтверског алата, док су мерење квалитета софтвера и мерење успеха процеса на другом месту (слика 5.3.2.6.4.).

### **5.3.3.2. Анализа одговора у вези употребљивости развијеног решења**

1. Да ли је реализовано решење употребљиво у реалној пракси?

ОДГОВОР: Да би разматрали могућности употребљивости реализованог прототипа, испитаници су имали могућност приступа и коришћења прототипа, а затим су у вези употребљивости, предности и недостатака описали своја искуства и ставове у анкети. На основу ставова испитаника, реализовано решење је делимично употребљиво у пракси. Укупно 59 испитаника је одговорило да је реализовани прототип употребљив, док је делимично прихватање (одговор „не знам“ уз образложење – 38 испитаника, одговор „да, уз поправке“ – 4 и одговор „можда“ – 1 испитаник) изразило укупно 43 испитаника. Реализовано решење је у анкетирању представљено као прототип који је у развоју, али и сами одговори и сугестије испитаника указују да је реализовано решење потребно унапредити, како би исказало пуну употребљивост.

2. Које су предности наведеног система?

ОДГОВОР: Предности реализованог прототипа се могу приказати кроз 2 групе предности:

1) предности коришћења прототипа у смислу функционалних карактеристика и подршке процесима управљања пројектима – најзначајнија предност је подршка контроли и организацији (52 учесника), комуникацији (43), тимском раду (21), а најмање предности су: ресурси (10), квалитет софтверског производа (2), визуализација (1).

2) предности коришћена наведеног система у односу на карактеристике прототипа и утицај на успех пројекта: - доступност апликације прототипа и података (27), једноставност коришћења, коришћење убрзава рад (8) итд

3. Који су недостаци наведеног система?

ОДГОВОР: Највише испитаника је одговорило да су недостаци реализованог прототипа: непостојање подршке визуализацији и извештавању (18), време и напор који би се улагао за ажурно вођење евиденције (17), непостојање појединих важних функција (15) и недостатак подршке за анализу података и подршци одлучивању (11) итд.

4. Које су могућности унапређења наведеног система?

ОДГОВОР: Сугестије за унапређење реализованог прототипа обједињене су у форми списка многобројних функција које нису реализоване (међу којима су најважније функције које се односе на визуализацију, обраду података и извештавање), али су потребне (дате су у прилогу), а сугерисано је и унапређење корисничког интерфејса, посебно у прилагођавању коришћењу система путем мобилних телефона.

### **5.3.3.3. Утврђивање статуса потврђености хипотезе и подхипотеза**

ХИПОТЕЗА - 4. ДЕО

„Имплементирани систем може да обезбеди подршку за управљање софтверским пројектима у реалној пракси.“

ОДГОВОР: Овај део главне хипотезе је потврђен, чињеницом да су компетентни испитаници утврдити да наведени систем, уз одређене допуне, може бити коришћен у реалној пракси. Испитаници су имали могућност да тестирају функционисање система и утврде његове функционалне и квалитативне карактеристике и највећи број одговора поводом употребљивости прототипа у реалној пракси био је позитиван.

1. ПОДХИПОТЕЗА

„Коришћење имплементираниог система смањује време реализације софтверског пројекта.“

ОДГОВОР: У питању отвореног типа које се односи на предности система, укупно 8 испитаника је дало одговор да систем убрзава рад у оквиру реализације софтверских пројеката. Међутим, у оквиру питања отвореног типа које се односи на недостатке апликације, укупно 17 испитаника сматра да би коришћење прототипа у реалној пракси, посебно за пројекте мањег обима и мањег тима, успорило рад на реализацији софтверских пројеката и из наведеног разлога се најчешће овакви софтвери ни не користе. Испитаници сматрају да се овакви софтверски алати за подршку управљању софтверским пројектима могу користити и имају значај код већих дуготрајнијих пројеката са већим тимом.

4. ПОДХИПОТЕЗА – 3. ДЕО

„Применом имплементираниог система повећава се квалитет реализованих софтверских пројеката у наставном процесу.“

ОДГОВОР: С обзиром да је утврђено на основу ставова компетентних испитаника да примена наведеног система омогућава ефикасну комуникацију у тиму, омогућава мерење квалитета и контролу, управљање пројектом и тимски рад, 3. део 4. подхипотезе је потврђен. Утицајем примене система на комуникације, ефикасност и тимски рад, као и омогућавањем мерења и контроле утиче се и на значајне елементе процеса реализације и управљања софтверским пројектом, чиме се у делу квалитета процеса утиче на укупан квалитет реализације софтверског пројекта.



## **5.4. Бенчмаркинг анализа најчешће коришћених алата издвојених анкетирањем**

### **5.4.1. Методологија истраживања**

Алати који се односе на предмет истраживања у овој докторској дисертацији, односно у оквиру овог одељка, категоришу се као алати за подршку тимском раду у реализацији софтвера и управљању софтверским пројектима у дистрибуираном окружењу. Узорак овог истраживања, односно избор алата који ће бити анализиран је изабран на основу претходно извршене анкете међу запосленима у ИТ струци, где су у оквиру анкете дати одговори о алатима које су учесници анкете најчешће користили. Посебан фокус анализе су заправо алати који дају подршку за оба аспекта: развоја софтвера у дистрибуираном тимском раду и управљање (софтверским) пројектима.

Анализа карактеристика најчешће коришћених алата може се извршити методом бенчмаркинга. У оквиру ове методе кључне активности се односе на дефинисање мерних карактеристика које ће представљати основ вредновања и компарације. Након дефинисања мерних карактеристика, сваки од алата ће бити анализиран или тестиран (и овом истраживању биће коришћени подаци о карактеристикама алата са одговарајућих web сајтова наведених алата) у складу са захтевима мерења. Након добијања резултата, биће извршена компарација резултата и утврђене предности и недостаци сваког алата, као и рангирање алата у односу на сумарне резултате.

#### **5.4.1.1. Појам, карактеристике и истраживања примене методе бенчмаркинга у ИТ и ИТ пројектном менаџменту**

У раду [Ajelabi&Tang, 2010] појам бенчмаркинга је дефинисан у контексту управљања квалитетом у организацијама који је дат од стране International Benchmarking Clearinghouse као: „систематичан и континуирани проце мерења и компарације организационих пословних процеса у односу на пословне процесе лидера у истој области у свету, како би се добиле информације које би помогле организацији да предузме активности унапређења својих перформанси.“ У наведеном раду наводе се типови бенчмаркинга у односу на мерење перформанси, стратешких аспеката, интерних аспеката, компетитивности, функционалних аспеката и генерички тип бенчмаркинга. У наведеном раду разматрају се могућности примене бенчмаркинг методе за вредновање квалитета пројектног менаџмента. У раду [Say JAL, 2008] приказано је истраживање критеријума успеха ИТ пројекта применом техника бенчмаркинга у оквиру фирми из различитих сектора, користећи упитнике који су садржали питања која се односе на критеријуме успеха ИТ пројекта, факторе неуспеха, методологије ИТ пројектног процесног менаџмента, алате који се користе у ИТ пројектном менаџменту и компетенције које су потребне за ИТ пројектног менаџера. У раду [Sim et al, 2003] разматран је бенчмаркинг као алат у истраживањима у области софтверског инжењерства. У раду [Dutta et al, 1998] описан је бенчмаркинг извршен у оквиру истраживања примене елемената добре праксе из области управљања софтверским пројектним менаџментом у оквиру европских земаља.

Према [Sim et al, 2003], бенчмаркинг се може дефинисати као „тест или скуп тестова који се користе за поређење перформанси алтернативних алата или техника.“ Бенчмаркинг [Sim et al, 2003] се састоји из 3 компоненте:

- мотивисана компарација – мотивација за истраживање и технички опис компарације карактеристика између алата/техника који се упоређују
- релевантан скуп тестова – тестови представљају део задатака које алати/технике треба да изврше или да реше у пракси. С обзиром да не могу сви задаци да се примене, важно је одабрати релевантне.
- мере перформанси – перформанса је веза технологије и начина како се користи, односно сврхе. Мерење перформанси изражава ниво усклађености са сврхом.

Према [Sim et al, 2003], најважније особине које добар бенчмарк треба да има су: приступачност (применљивост), исплативост (економска одрживост, тј. изведивост), јасност, релевантност, применљивост (решивост задатка), портабилност (на различите технолошке платформе), скалабилност (за различите нивое зрелости).

У раду [Gerogiannis et al, 2009] разматрана је улога заинтересованих страна (персонала организације у којој се примењује) у оквиру избора пројектног и портфолио менаџмент алата. У овом раду предложен је приступ аналитичког хијерархијског процеса (АНР). На основу упитника који је дат радницима Националног документационог центра Грчке утврђени су критеријуми за вредновање и избор алата за пројектни менаџмент.

#### 5.4.1.2. Креирање модела бенчмаркинг анализе

Кључни елемент бенчмаркинг анализе је заправо избор карактеристика које ће бити мерене и упоређиване. Избор карактеристика зависи од типа алата и његове намене, као и циљева који се желе постићи анализом. Избор карактеристика алата које ће бити анализирани треба првенствено да има особине свеобухватности и усклађености са стандардима, а такође и да омогући проверу усклађености могућности коришћења наведеног алата у оквиру ширег контекста организације и стратешких циљева који би се реализовали у дужем временском периоду кроз реализацију већег броја пројеката.

У циљу избора мерних карактеристика, у овом истраживању фокус је на функционалности, односно провери да ли алат поседује одговарајуће софтверске функције. Поред функционалности постоји и низ других квалитативних карактеристика софтверског алата, које би се могле проверавати у складу са стандардом ISO 9126 или важећим стандардом ISO 2010, међутим у овом истраживању неће бити разматране. У контексту утврђивања покривености потребним софтверским функцијама, као основ избора карактеристика изабрани су:

1. Стандарди пројектног менаџмента описани у оквиру PMBOK [PMBOK, 2013]
2. Стандарди софтверског инжењерства описани у оквиру SWEBOOK [SWEBOOK, 2014]
3. Елементи методологије стратегијског менаџмента Balanced Scorecard (BSC) [Kaplan&Norton, 1992].

На основу избора базичних методологија и стандарда, односно анализом њихових основних области, одређена су питања која ће бити основ квалитативне и квантитативне анализе у оквиру упоређивања алата, као што је дато у табели 5.4.1.2.1. Нека питања су у оквиру различитих група слична, али су ипак остављена како би се приказала одговарајућа група у потпуности.

Табела 5.4.1.2.1. Бенчмаркинг модел за вредновање алата за подршку софтверском пројектном менаџменту

ИЗВОР	ОБЛАСТ	ПИТАЊЕ - Да ли алат омогућује ...
PMBOK	интеграција	Интероперабилност са другим алатима, посебно развојним окружењима и различитим форматима фајлова? експорт података?
	обухват	Приказ дефинисаног обухвата потребних софтверских функција? Приказ измене обухвата потребних софтверских функција? Приказ реализованог обухвата потребних софтверских функција?
	време	Процену и планирање времена реализације? Приказ временских карактеристика тока реализације?
	квалитет	Мерење квалитета људи – учесника пројекта? Мерење квалитета тима у учешћу пројекта? Мерење квалитета резултата? Мерење квалитета тока резултата процеса?
	људски ресурси	Евиденцију учесника пројекта? Евиденцију реализованог рада? Персонализацију функција за различите профиле корисника у односу на различите улоге у реализацији пројекта?
	комуникације	Размену фајлова између учесника? Комуницирање – размену порука, идеја, питања?
	ризичи	Евидентирање ризика? Мерење ризика? Приказ ризика?
	набавке	Унапређење постојећег решења додатним модулима?
	заинтересоване стране	Комуникацију са заинтересованим странама?
SWEBOOK	захтеви	Евиденцију захтева? Евидентирање промене захтева? Праћење и

		приказ промена захтева? Евидентирање степена усклађености решења са захтевима?
	дизајн	Интеграцију са алатима за дизајн система?
	конструкција	Интеграцију са алатима за развој?
	тестирање	Интеграцију са алатима за тестирање?
	одржавање	Није покривено
	управљање конфигурацијом	Није покривено
	инжењерски менаџмент	Покривено другим питањима
	процес инжењерства	Приказ тока процеса реализације? Мерење успеха фазе или тока реализације?
	инжењерски модели и методе	Примену различитих приступа и методологија управљања пројектима? Примену различитих приступа и методологија развоја софтвера?
	квалитет	Мерење квалитета резултата? Мерење квалитета тока резултата процеса?
	инжењерска економија	Финансијски показатељи успеха фазе или процеса? Финансијски показатељи успеха производа?
Balanced Scorecard	финансије	Финансијски показатељи успеха фазе или процеса? Финансијски показатељи успеха производа?
	корисници	Комуникацију са корисницима? Креирање извештаја? Мерење задовољства корисника? Евидентирање потреба и захтева корисника?
	интерни процеси	Приказ тока процеса? Планирање тока процеса? Мерење успеха тока процеса?
	учење и раст	Евиденцију и приказ искустава? Евидентирање и приказ проблема и одговора?

#### 5.4.1.3. Припрема узорка истраживања

У оквиру попуњених анкета (сумарни резултати анкета приказани су у одељку 5.3.), учесници су издвојили алате које користе у тимском раду приликом реализације софтвера у дистрибуираном окружењу, као и алате које користе као подршку управљању пројектима. Поједини алати су наведени у оквиру обе категорије. Комплетан списак свих наведених алата у анкетама (алата за подршку тимском раду и алата за подршку управљању пројектима) дат је у прилогу.

Пресеком претходна два скупа алата издвајамо алате који припадају обема категоријама, односно подржавају тимски рад у дистрибуираном окружењу и дају подршку управљању пројектима.

Табела 5.4.1.3.1. Списак алата који припадају обема категоријама (за подршку тимском раду у развоју софтвера и управљању софтверским пројектима), који су наведени у анкетама

Назив алата	
Active Collab	Jira
Basecamp	RedMine
dotProject	Microsoft Team Foundation Server
gForge	TeamViewer

#### 5.4.2. Резултати анализе карактеристика и компарација алата

На основу квалитативне анализе функционалности (детали су дати у прилогу) која је подржана одговарајућим алатом, а приказана на одговарајућим web сајтовима тих алата, може се закључити следеће:

1. Алати који су издвојени на основу анализе одговора из претходне анкете могу се категорисати као:

а) општи алати за подршку тимском раду и управљању пројектима: ActiveCollab, Basecamp, dotProject

б) алати за подршку тимском раду и управљању софтверским пројектима: GForge, JIRA, Redmine, Microsoft Team Foundation Server / Visual Studio Online

ц) општи алат за удаљени приступ рачунарима и размену података: TeamViewer

У оквиру бенчмаркинг анализе, за свако од наведених питања дати су одговори „да“ уколико наведени алат подржава наведену функционалност. Од укупно 44 питања и могућих позитивних одговора, сваки од општих алата за подршку тимском раду и управљању пројектима има подједнако 12 поена, док алати за подршку тимском раду и управљању софтверским пројектима имају следећи број поена: Gforge (23), JIRA (27), Redmine (39) и Microsoft Team Foundation Server / Visual Studio Online (36). Алат TeamViewer подржава само основне функције размене података и не може се сматрати алатом за подршку управљању софтверским пројектима, већ само за подршку комуникацијама у процесу развоја и управљања софтверским пројектима.

Табела 5.4.2.1. Број поена за појединачне алате у бенчмаркинг анализи према категоријама PMBOK, SWEBOK и BSC

Број питања	Active Collab	Basecamp	dotProject	GForge	Jira	RedMine	Microsoft Team Foundation Server	TeamViewer
<b>PMBOK Max=21</b>	8	9	9	11	13	20	17	5
<b>SWEBOK Max=13</b>	1	1	1	8	9	10	12	0
<b>BSC Max=10</b>	3	2	2	4	5	9	7	1
<b>УКУПНО Max=44</b>	12	12	12	23	27	39	36	6

Табела 5.4.2.2. Процент заступљености одговарајућих функционалних карактеристика за појединачне алате у бенчмаркинг анализи према категоријама PMBOK, SWEBOK и BSC

%	Active Collab	Basecamp	dotProject	GForge	Jira	RedMine	Microsoft Team Foundation Server	TeamViewer
<b>PMBOK Max=21</b>	38	43	43	52	62	95	81	24
<b>SWEBOK Max=13</b>	8	8	8	62	69	77	92	0
<b>BSC Max=10</b>	30	20	20	40	50	90	70	10
<b>УКУПНО Max=44</b>	27	27	27	52	61	89	82	14

На основу наведене анализе можемо тврдити да је најпотпунији алат у смислу подршке потребним функцијама за подршку тимском раду и управљању пројектима алат Redmine.

Примена претходно наведеног бенчмаркинг модела исказала је предности у омогућавању нумеричког приказивања квалитативних особина и могућност компарације алата на основу дефинисаних питања, која су заснована на стандардима PMBOK, SWEBOK и BSC. Наравно, избор критеријума вредновања, односно методологија које представљају основ веома утиче на коначни резултат. Такође, у предложеном приступу треба нагласити да недостатак представља поједностављеност начина процене задовољавања критеријума (једноставним одговором „Да“, иако у оквиру појединих конкретних вредновања задовољавање неког критеријума би се могло прецизније изразити кроз више нивоа), као и делимична субјективност као основ процене. Наиме, иако је једноставно обележена као заступљена, код алата Redmine уопште подршка мерењима, као и проценама ризика није подржана у довољној мери, већ у овом алату је подршка на почетном нивоу. Из наведених разлога, у циљу повећања нивоа прецизности и објективности, предложени бенчмаркинг модел треба да буде унапређен могућностима степеноване евалуације, коришћењем бенчмаркинг модела од стране више учесника и увођењем аутоматизације у процес оцене карактеристика алата.

## 5.5. Анализа резултата емпиријских истраживања у односу на ефикасност примене прототипа

### 5.5.1. Циљ и методе истраживања

Циљ емпиријског истраживања који ће бити приказан у овом одељку је утврдити у којој мери реализовани прототип исказује ефикасност приликом примене у односу на анализу:

1. усклађености функционалних карактеристика са сврхом реализације прототипа
  2. како коришћење прототипа утиче на брзину реализације софтверских пројеката
  3. како коришћење прототипа утиче на квалитет реализованих софтверских пројеката.
- На овај начин би се емпиријски утврдила употребљивост реализованог прототипа, као и одговори на истраживачка питања која су формулисана на основу главне и помоћних хипотеза.

Методе истраживања у овом одељку односе се на детаљнију анализу појединих резултата из спроведеног анкетања ИТ кадра (описаног у одељку 5.3. и прилозима), као и анализу података који су прикупљени у току коришћења прототипа над софтверским пројектима који су реализовани за потребе реалног организационог система (описаног у одељку 5.2. и прилозима).

### 5.5.2. Испитивање усклађености функционалних карактеристика прототипа са сврхом и циљевима развоја апликације

У овом одељку можемо формулисати истраживачко питање: „Да ли је реализовани прототип по својим функционалним карактеристикама у складу са сврхом и циљевима развоја наведене апликације?“

Ради добијања одговора на наведено питање биће издвојени и анализирани резултати претходно описане анкете (најчешћи одговори отвореног типа о предностима и карактеристикама наведеног система) и упоређени са циљевима и планираном сврхом апликације.

Табела 5.5.2.1. Компарација развојних циљева, имплементираних карактеристика и утврђених особина анкетањем

ЦИЉ И СВРХА ПРОТОТИПА	ИМПЛЕМЕНТИРАНА КАРАКТЕРИСТИКА	ПРЕДНОСТ СИСТЕМА (најчешћи ставови ИТ кадра према резултатима анкете)
Једноставност коришћења	обједињени приказ свих активности и ресурса за све учеснике једног пројекта једноставан унос података о пројекту једноставно дељење ресурса	Карактеристика: Једноставност и практичност употребе
Прегледност	обједињени приказ свих активности и ресурса за све учеснике једног пројекта праћење тока извршавања	Карактеристика: Прегледност
Могућност мониторинга и планирања	праћење тока извршавања једноставно дељење ресурса	Предности: Омогућује мониторинг и контролу
Унапређење тимског рада у дистрибуираном окружењу	Web технологија је доступна свуда	Предности: Унапређује тимски рад Карактеристика: On-line доступност
Повећање квалитета комуникација у тиму	обједињени приказ свих активности и ресурса за све учеснике једног пројекта	Предности: Унапређује комуникације

На основу претходно приказане табеле, може се тврдити да су све циљне карактеристике прототипа имплементирани и искуствено потврђене ставовима ИТ кадрова у оквиру пробног коришћења прототипа и анкетања.

### **5.5.3. Ставови ИТ кадрова у контексту процене утицаја примене прототипа на укупно време реализације пројекта**

У оквиру претходно описаног анкетања у оквиру питања која се односе на предности и недостатке приликом коришћења реализованог прототипа, у питањима отвореног типа испитаници су навели и карактеристике које се односе на утицај коришћења прототипа на укупно време трајања (брзину рада) софтверског пројекта.

На питање бр. 12 из анкете укупно 8 испитаника је одговорило да се једна од група предности коришћења прототипа односи на подршку убрзавању рада на пројекту: „*ubrzava rad, povećava produktivnost, brži način upravljanja razvojem projekta, korišćenje softvera ne treba da zahteva mnogo dodatnog vremena, ušteda vremena, brže uočavanje problema na razvoju softvera, ubrzanje ispitivanja softvera, praćenje procesa razvoja, brža analiza rada i ispitivanja kvaliteta urađenog posla*“. Позитивне карактеристике прототипа су графички представљене на слици 5.3.2.7.3., где се види удео ове групе позитивних карактеристика у скупу свих позитивних карактеристика.

На питање број 13 из анкете укупно 17 испитаника је одговорило да се једна од категорија недостатака у коришћењу развијеног прототипа, као једног у скупу већ раније реализованих софтверских алата сличног типа, повећано време и труд потребни за унос података и настојање да подаци буду ажурни: „*Mislim da je u slučaju nekog manjeg projekta vreme potrebno za ažuriranje ovakve aplikacije znatno veće od koristi koje on donosi. Svakodnevno ažuriranje tokom izrade projekta. Jedini nedostak bi bilo utrošene vreme za unos podataka u aplikaciju, ali s obzirom na potrebu pitanje je da li je to utrošeno vreme nedostatak...Potencijalno velik dodatan napor u održavanju stanja ažurnim.Nedostatak bi mogao da bude nepotrebno komplikovanje procesa realizacije projekta sa dodatnim opterećenjem za inženjere. Jedino ako se radi veoma mali projekat od strane pojedinca u scenariju „one man show“ gde ne postoji interes ili potreba za informacijama ovog tipa, onda je evidentiranje podataka kroz ovakvu aplikaciju nepotrebno trošenje resursa, ali opet, i tada može da bude korisno, pogotovo gledano u širem vremenskom okviru. Odbojnost malih timova koji to shvataju kao dodatno opterećenje i nepotrebnu papirologiju (mora biti krajnje jednostavna i prosta za upotrebu, a ujedno kvalitetna i korisna što je svakako izazov implementatoru aplikacije), Na manjim projektima nije neophodna ovakva aplikacija. Vreme potrebno za administraciju Može zahtevati dodatnu administraciju samog sistema, jer se mogu javiti problemi tipa neažurnosti и naravno može zahtevati malo dodatnog vremena za sam unos progresа, већи utrošak vremena (Video sam da za manje projekte koriste I MS Excel), administracija (može da optereti proces proizvodnje), Unosenje podataka može da uspori projekat. Za prostije i kraće projekte može oduzeti previše vremena, a doprineti malo. Suviše dokumentacije može da bude prepreka za brzu izradu softvera. Potreba za dodatnim resursima koji će se baviti upravljanjem procesa i tumačenjem priključenih podataka.*“ На слици 5.3.2.7.4. дат је графички приказ сумарних података о недостацима реализованог прототипа, где се види удео овог сегмента у укупном скупу свих недостатака.

На основу претходно изнетих чињеница, може се закључити да један део испитаника сматра да коришћење наведеног прототипа може да убрза рад у оквиру пројекта, тиме што се олакшавају комуникације, транспарентност, доступност парцијалних резултата итд. Међутим, друга група испитаника сматра да уколико би се ажурно подаци уносили описивали сви елементи управљања софтверским пројектом, то може код мањих пројеката да представља непотребни терет и одузима време и ресурсе. Ипак, код друге групе испитаника која сматра да коришћење наведеног прототипа може да успори реализацију пројекта, наведени став не односи се само на конкретан тестирани прототип, већ и на читаву категорију сличних софтверских алата за подршку управљању реализацијом софтверских пројеката који захтевају ажурну администрацију података уношењем података од стране учесника пројекта. Ова запажања указују на потребу за унапређењем система прикупљања података, нпр. аутоматском аквизицијом података из развојног окружења.

Свакако, подаци о ставовима о утицају коришћења прототипа на ефикасност (брзину) рада на пројекту издвојени су из одговора на питања отвореног типа. Из наведеног разлога не могу се сматрати поузданим и стабилним основом закључивања о ефикасности примене прототипа, јер се заснивају на субјективној оцени дела испитаника који су у општијим питањима о предностима и недостацима заправо на основу сећања и искуства из коришћења сличних алата, као и кратког пробног коришћења наведеног прототипа, описали одговарајуће карактеристике.

Из разлога непрецизности текста анкете и субјективности у процени од стране испитаника, јавља се потреба за прецизнијим утврђивањем ефикасности коришћења прототипа.

#### **5.5.4. Испитивање фреквенције истовремених вишеструких активности и адекватности подршке реализованог решења**

На основу анализе сродних алата (одељак 5.4), као и на основу функционалних карактеристика и циљева реализованог прототипа, може се закључити да ни сродни алати ни реализовани прототип немају реализоване могућности подршке обради „истовремених вишеструких активности“ (ИВА), односно „композитних радних ставки“. Наиме, у оквиру емпиријског истраживања коришћења прототипа у наставном окружењу и професионалном усавравању студената, ситуација ИВА јављала се повремено у оквиру тока реализације софтверских активности када је у једном акту коришћења прототипа потребно унети више ставки које се реализују у једном тренутку. Такве ситуације се јављају најчешће у треницима верификације парцијалних резултата, када се дешава више активности – предаја резултата, верификација, задавање нових задатака итд. Доступна професионална решења софтверске подршке управљању софтверским пројектима, као и реализовани прототип имају могућност да се за сваку конкретну ставку посла мора уносити у истом дану свака ставка посебно - за сваки тип активности посебна ставка.

Убрзање коришћења алата овог типа био би у омогућавању бржег и једноставнијег уноса више радних ставки једним уносом. Поставља се питање: „Да ли је овакав аутоматизам потребно реализовати у овом случају?“ На ово питање можемо одговорити анализом фреквенције појава истовремених вишеструких активности, посебно у контексту уноса података коришћењем развијеног прототипа.

У оквиру овог истраживања, анализирани су радне ставке појединачно за сваки студентски пројекат (укупно 293 за све софтверске пројекте) које су у току коришћења прототипа биле унете у базу података развијеног прототипа у току реализације софтверских пројеката у 2013/14 години. Подаци о конкретним пројектима и мерењима броја радних ставки за сваки пројекат, уз анализу броја активности које припадају радним ставкама и категоризацију приказани су у прилогу.

Приликом анализе радних ставки (записа о радним активностима) утврђене су следеће ситуације:

1. Унос појединачне радне ставке за једну засебну радну активност која се једина реализовала у наведеном датуму. Овај начин је илустрован следећом сликом (слика 5.5.1).

May 13, 2014	Kazi Ljubica	Dalje za praksu - uputstva	task	information requirements	0
--------------------	-----------------	-------------------------------	------	--------------------------	---

Слика 5.5.5.1. Исечак из табеларног приказа радних ставки у оквиру евиденције рада на пројекту „Алумни“ – унос једне радне ставке за једну активност која се засебно реализовала једина у наведеном тренутку

2. „Правилан“ начин евиденције истовремених радних активности („вишеструки истовремени унос“) – свака активност из скупа истовремених активности (истог датума, у исто време се дешава) резултовала је засебним евидентирањем радне ставке за сваку засебну активност и одговарајућом категоризацијом. Овај начин је илустрован следећом сликом (слика 5.5.2). На слици је приказано да је дана 22.4.2014. године извршено 2 активности – дизајн базе података за динамички део и дизајн корисничког интерфејса за динамички део апликације, што је, правилно, забележено као 2 радне ставке у евиденцији пројектних активности.

April 22, 2014	Kazi Ljubica	Dizajn BP za dinamicki deo	verification	database	database	0
April 22, 2014	Kazi Ljubica	Dizajn UI za dinamicki deo	verification	source code	user interface	0

Слика 5.5.5.2. Исечак из табеларног приказа радних ставки у оквиру евиденције рада на пројекту „Алумни“ –унос више радних ставки, за сваку активност посебна ставка у оквиру једног датума

3. „Неправилан“ начин евиденције истовремених радних активности („композитни унос“) – у оквиру једне радне ставке у евиденцији бележено је више радних активности, а категоризација је извршена на основу доминантне једне активности. Овај начин је илустрован следећом сликом (слика 5.5.3). Наредна слика приказује детаље записа (радне ставке) из списка радних ставки за пројекат „Алумни“. На наведеној слици дана 25.3.2014. године под називом радне ставке „Личне консултације“ унет је опис за 2 радне активности – заједно је сређен use case дијаграм и shema базе података. Ово је неправилан начин бележења јер се цео запис радне ставке категорисао у оквиру сегмента RESULT, артефакта MODEL, фазе DATA MODEL, док део који се односи на use case дијаграм неће бити видљив у табеларном приказу и такође неће утицати на визуализацију приказа прогреса у оквиру реализације пројекта.

www.it-project.rs/ProjectRecordView.aspx

**PROJECT RECORD**

Project: Alumni record 1.0  
Participants: Simic Milos

**PROJECT RECORD - selected item view** [RECORDS LIST](#)

Date:

From:

Title:

Description:

Segment:

Artifact:

Phase:

Слика 5.5.5.3. Унос података о више активности у оквиру једног записа (једне радне ставке) на примеру пројекта “Алумни”



Сумарни приказ статистике у односу на евиденције појединачних, вишеструких и композитних уноса која се односи на софтверске пројекте реализоване у школској 2013/14 приказан је у наставку.

Табела 5.5.5.1. Статистика о броју записа и активности у оквиру студентских софтверских пројеката реализованих у шк. 2013/14

БРОЈ РАДНИХ АКТИВНОСТИ У ЈЕДНОМ ЗАПИСУ	УКУПНО ЗАПИСА	УКУПНО РАДНИХ АКТИВНОСТИ
1	121	121
2	160	320
3	11	33
4	1	4
<b>Укупно појединачних активности:</b>		<b>121</b>
<b>Укупно вишеструких и композитних активности:</b>		<b>357</b>

Графички приказ статистике о броју појединачних, вишеструких активности у оквиру једног записа и композитних активности у оквиру једног записа приказан је на наредној слици. На слици 5.5.5.4. је приказан проценат појединачних активности које су се евидентирале као засебни записи (25%) и проценат заступљености активности које су се евидентирале у оквиру вишеструких и композитних записа (75%).



Слика 5.5.5.4. Процентуални приказ појединачних активности и записа и активности које су бележене у оквиру вишеструких/композитних записа

Претходни резултати говоре о великој заступљености вишеструких и композитних уноса (75% активности су бележене у оквиру вишеструких или композитних записа). У оквиру претходно наведених примера једноставног уноса (ситуација 1), правилног вишеструког уноса (ситуација 2) и неправилног композитног уноса (ситуација 3) описан је проблем и потреба да се реализује подршка истовремено прецизном уносу, али и бржем уносу (ако је могуће да се у оквиру једног уноса разврстају и засебно аутоматски бележе подаци по категоријама). Слични покушаји су реализовани у савременим алатима за управљање пројектима користећи технике „mentions“ и „watchers“ које се користе у оквиру алата JIRA [JIRAwм] за јединствени унос текста и „спомињање“ запослених који су коришћењем ових техника повезивања, обавештени о новом или измењеном тексту, односно задацима, проблемима итд. У овом истраживању предлаже се идеја примене техника сличним „mentions“ којима би се у тексту описа активности користили специјални симболи којима би се разврставали подаци према категоријама аутоматски након снимања целине. Овај предложени приступ можемо назвати АСТ-М („mentions for activities“).

### 5.5.5. Утицај коришћења прототипа на квалитет реализованих софтверских пројеката

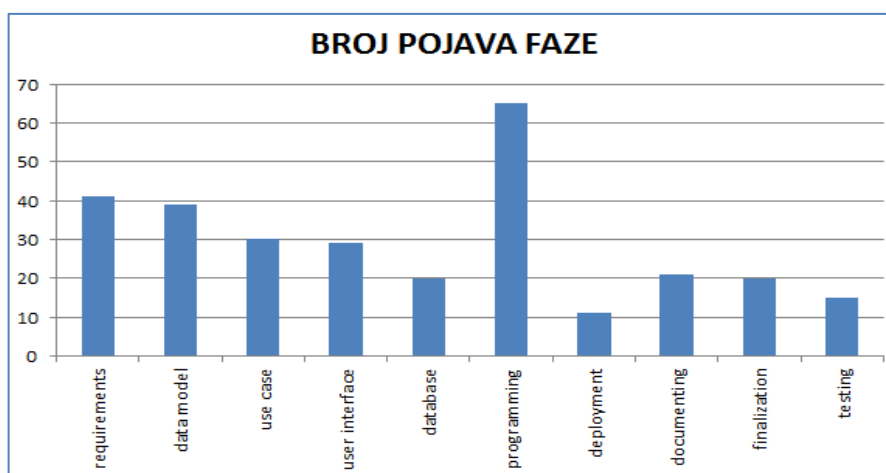
Доминантна карактеристика квалитета софтвера (узимајући у обзир стандард ISO 9216, односно актуелну верзију ISO 2010 који се односи на карактеристике квалитета софтвера) представља функционалност, тј. адекватност софтверских функција. С друге стране, по дефиницији успеха/квалитета софтверског пројекта, приликом изражавања и мерења успеха софтверског пројекта потребно је узети у обзир поред квалитета производа и квалитет процеса који је резултовао производом.

У наведеном контексту биће приказани сумарни резултати коришћења прототипа у оквиру подршке реализацији софтверских пројеката из школске 2013/14 године. Приказом наведених података могуће је извести закључке о резултатима емпиријског истраживања.

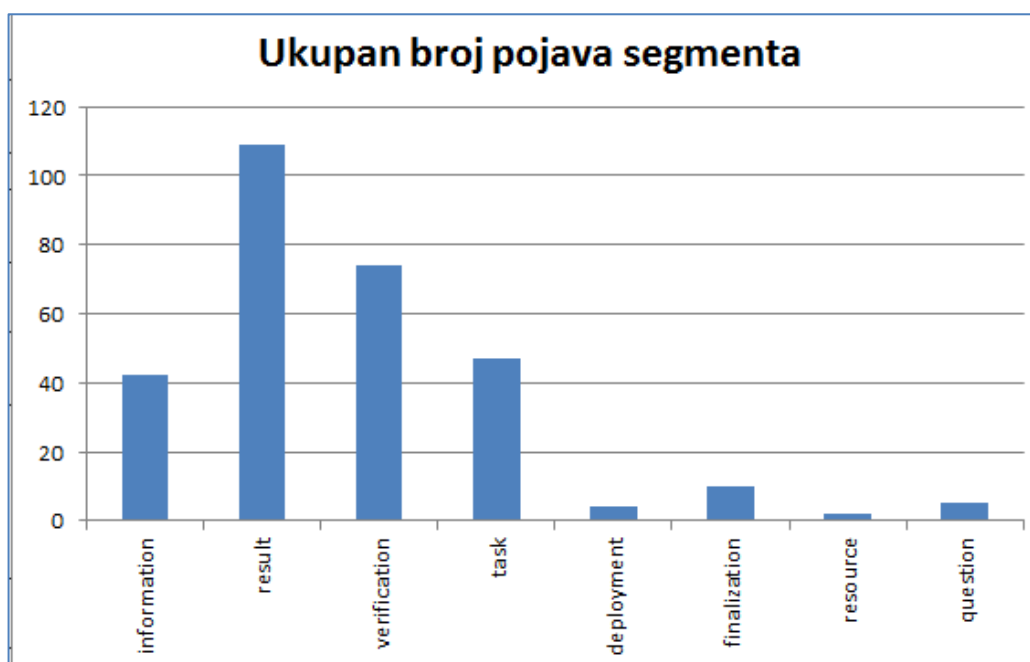
Табела 5.5.6.1. Сумарни приказ броја појава фаза, сегмената и артефакта у оквиру реализације софтверских пројеката у школској 2013/14 години

ФАЗА	БРОЈ ПОЈАВА ФАЗЕ	СЕГМЕНТ	БРОЈ ПОЈАВА СЕГМЕНТА	АРТЕФАКТ	БРОЈ ПОЈАВА АРТЕФАКТА
requirements	41	information	42	source code	127
data model	39	result	109	document	61
use case	30	verification	74	model	69
user interface	29	task	47	information	13
database	20	deployment	4	database	22
programming	65	finalization	10		
deployment	11	resource	2		
documenting	21	question	5		
finalization	20				
testing	15				

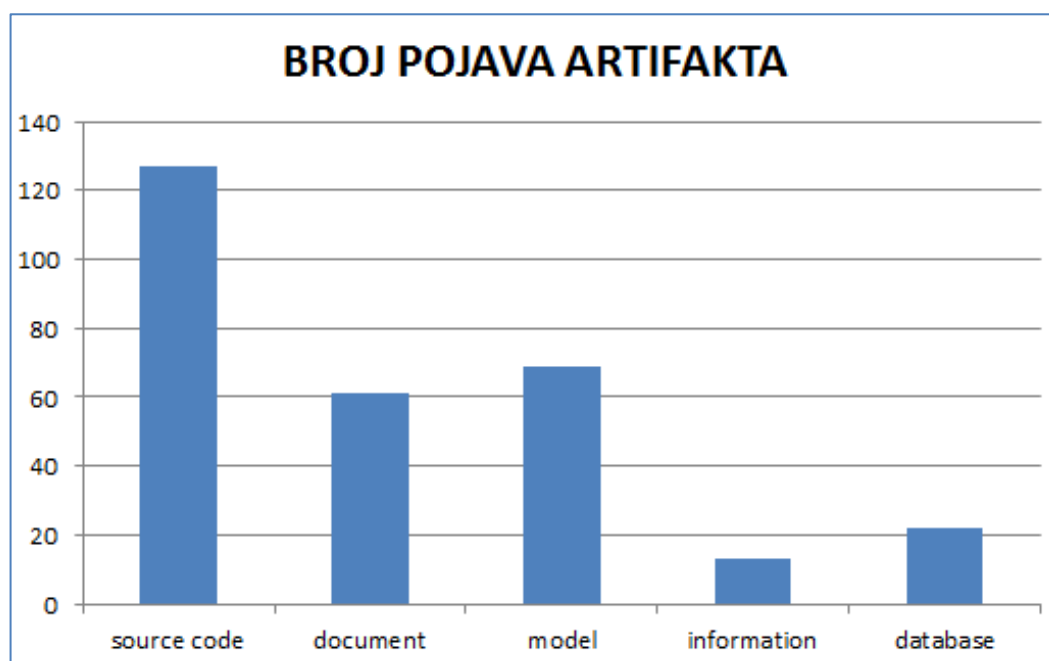
Претходно приказана табела представља основ за графички приказ резултата, који су дати у наставку.



Слика 5.5.6.1. Графички приказ броја појава фаза у оквиру свих реализованих софтверских пројеката у школској 2013/14 години



Слика 5.5.6.2. Графички приказ броја појава сегмената у оквиру свих реализованих софтверских пројеката у школској 2013/14 години



Слика 5.5.6.3. Графички приказ броја појава арטיפакта у оквиру свих реализованих софтверских пројеката у школској 2013/14 години

На основу претходних статистичких података и дијаграма, може се тврдити да је доминантна појава фазе програмирања, сегмента резултата и артефакта – изворног кода. Дакле, уколико бисмо требали да извршимо једноставно вредновање успеха софтверског пројекта на основу доминантних активности (фазе), сегмента и артефакта, свакако би требали да највећу пажњу посветимо квалитету програмског кода и квалитету процеса програмирања. Свакако, не треба занемарити и мање заступљене активности, сегменте и артефакте. Такође, можемо закључити да су прототипом покривене све релевантне фазе, сегменти и артефакти у смислу подршке евиденцији одговарајућих података.

### **5.5.6. Анализа резултата са становишта истраживачких питања и хипотеза**

*Истраживачко питање: „Да ли је реализовани прототип по својим функционалним карактеристикама у складу са сврхом и циљевима развоја наведене апликације?“*

ОДГОВОР: Реализовани прототип је у потпуности својим функционалним карактеристикама усклађен са сврхом и циљевима развоја наведене апликације. Наведено тврђење потврђено је компарацијом циљева и сврхе апликације са најчешћим ставовима утврђеним анкетирањем ИТ кадрова (у оквиру одговора отвореног типа) који су у оквиру анкете имали могућност пробног рада са наведеном апликацијом.

#### **1. ПОДХИПОТЕЗА**

*„Коришћење адаптивилног дистрибуираног информационог система за подршку управљању софтверским пројектима смањује време реализације софтверског пројекта.“*

ОДГОВОР:

А) На основу ставова испитаника у анкети у одговорима отвореног типа о предностима примене система, као једна од најважнијих предности коришћења система представља утицај на смањење времена трајања пројекта применом наведеног система. С друге стране, у одговорима о недостацима примене система, такође је најчешће присутан одговор да примена оваквог система може значајно да успори рад на пројекту. Уколико бисмо упоредили број одговора који сматра коришћење наведеног прототипа позитивним (8) у односу на број одговора који сматра коришћење прототипа негативним (17), могли бисмо закључити да већи број учесника сматра да би коришћење оваквог прототипа успорило рад на пројекту. На овај начин закључујемо да ова подхипотеза није потврђена.

Б) На основу анализе заступљености појединачних, вишеструких и композитних активности и подршке вишеструком уносу у оквиру развијеног прототипа, можемо тврдити да, у циљу прецизног уноса, реализовани прототип захтева унос појединачних записа за сваку активност која се евентуално извршила истовремено са другим активностима, чиме се успорава унос података у администрацији одговарајућих догађаја.

#### **4. ПОДХИПОТЕЗА – 2. ДЕО**

*„Применом система се повећава ефикасност наставе у области софтверског инжењерства и развоја информационог система.“*

#### **4. ПОДХИПОТЕЗА – 3. ДЕО**

*„Применом система повећава се квалитет реализованих софтверских пројеката у наставном процесу.“*

ОДГОВОР: Анализом броја активности-фаза, сегмената и артефакта који су се јавили у оквиру евиденција користећи развијени прототип, у току школске 2013/14 године, можемо закључити да употреба прототипа покрива све фазе и сегменте развоја, као и све типове артефакта. Евиденцијом и праћењем свих фаза, сегмената и артефакта развоја софтвера омогућавају се квалитетнија решења софтверског пројекта, а самим тим и ефикасност наставе у области софтверског инжењерства и развоја информационог система.

## 5.6. Анализа резултата примене метричких модела за евалуацију артефакта развоја софтвера

### 5.6.1. Узорак евалуације модела као артефакта развоја софтвера у области развоја информациониих система

Приликом избора узорка истраживања за проверу употребљивости метричких модела за евалуацију артефакта развоја софтвера разматране су могућности прикупљања података, величина узорка, хомогеност узорка и могућност детекције грешака у оквиру примене метричких модела. Из наведених разлога нису одабрани радови студената у оквиру софтверских пројеката унапређења информационог система факултета, већ радови студената са колоквијума у оквиру наставе у области информациониих система. Наиме, код студентских пројеката развоја софтвера за потребе унапређења информационог система (описаних у одељку 5.2. и прилогу 9.5.) факултета квалитет реализованих модела се мењао у току реализације и коначно је у задовољавајућој форми (квалитет коначних модела се не може вредновати, већ само „радне верзије у процесу“), број студентских пројеката је релативно мали (укупно 30), у студентским пројектима су учествовали студенти основних и мастер студија (нехомогеност узорка).

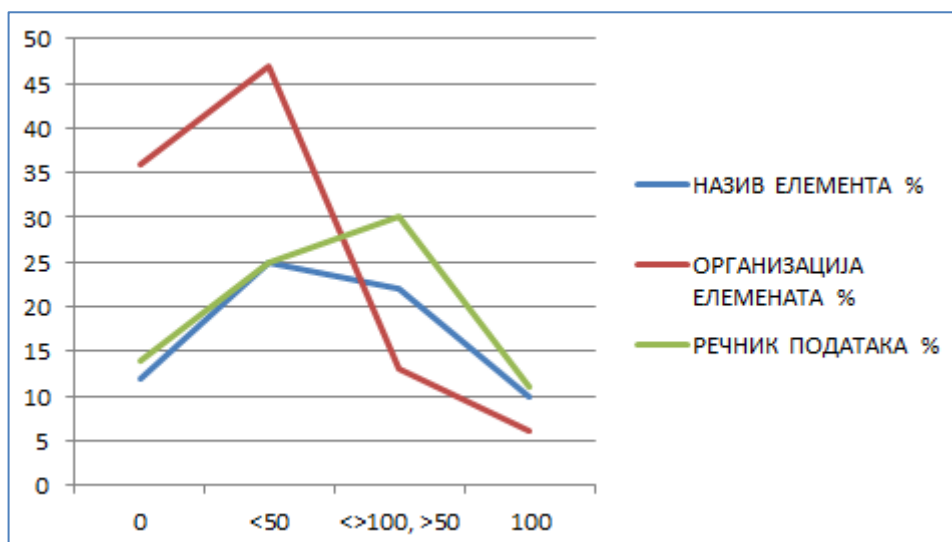
Узорак истраживања чине подаци о вредновању квалитета реализованих практичних лабораторијских колоквијума из предмета Информациони системи 1 на Техничком факултету „Михајло Пупин“ у Зрењанину, из школске 2013/14 године и школске 2014/15 године (детаљни подаци су расположиви на web сајту Техничког факултета „Михајло Пупин“ Зрењанин: <http://www.tfzr.uns.ac.rs/Predmet/informacioni-sistemi-1/oscene-vezbi-i-kolokvijuma>). Наведени подаци представљају илустрацију могућности примене предложених метричких модела (из одељка 3.3) за евалуацију артефакта развоја софтвера који се односе на фазу моделовања – моделовање пословних процеса, модел дизајна софтверских функција и концептуалног модела података.

### 5.6.2. Евалуација модела пословних процеса

У оквиру одељка прилози дати су подаци о примени метричког модела за евалуацију модела пословних процеса. У наставку је дат сумарни приказ резултата за школску 2013/14 и 2014/15 годину.

Табела 5.6.1.2.1. Табеларни сумарни приказ података о примени метричког модела за евалуацију модела пословних процеса – шк. 2013/14

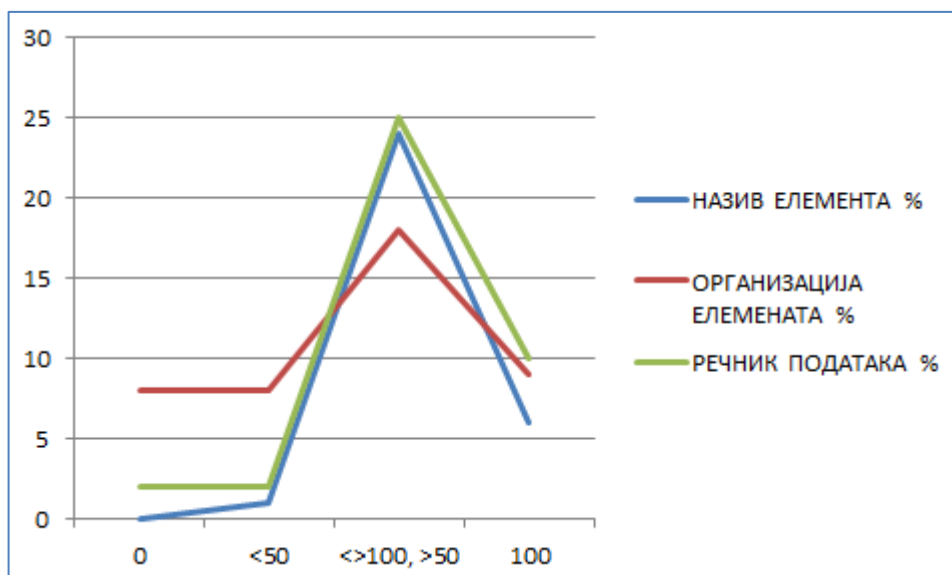
	2013/14		
УСЛОВИ	НАЗИВ ЕЛЕМЕНТА %	ОРГАНИЗАЦИЈА ЕЛЕМЕНАТА %	РЕЧНИК ПОДАТАКА %
0	12	36	14
<50	25	47	25
<>100, >50	22	13	30
100	10	6	11



Слика 5.6.1.2.1. Графички приказ сумарних података о примени метричког модела за евалуацију модела пословних процеса – шк. 2013/14

Табела 5.6.1.2.2. Табеларни сумарни приказ података о примени метричког модела за евалуацију модела пословних процеса – шк. 2014/15

УСЛОВИ	2014/15		
	НАЗИВ ЕЛЕМЕНТА %	ОРГАНИЗАЦИЈА ЕЛЕМЕНАТА %	РЕЧНИК ПОДАТАКА %
0	0	8	2
<50	1	8	2
<>100, >50	24	18	25
100	6	9	10



Слика 5.6.1.2.2. Графички приказ сумарних података о примени метричког модела за евалуацију модела пословних процеса – шк. 2014/15

На основу претходног сумарног приказа може се закључити да је у школској 2013/14 години највише радова студената исказало недовољно добре резултате у области организације елемената дијаграма пословних процеса и у области назива елемената, док је задовољавајуће резултате исказало у области речника података.

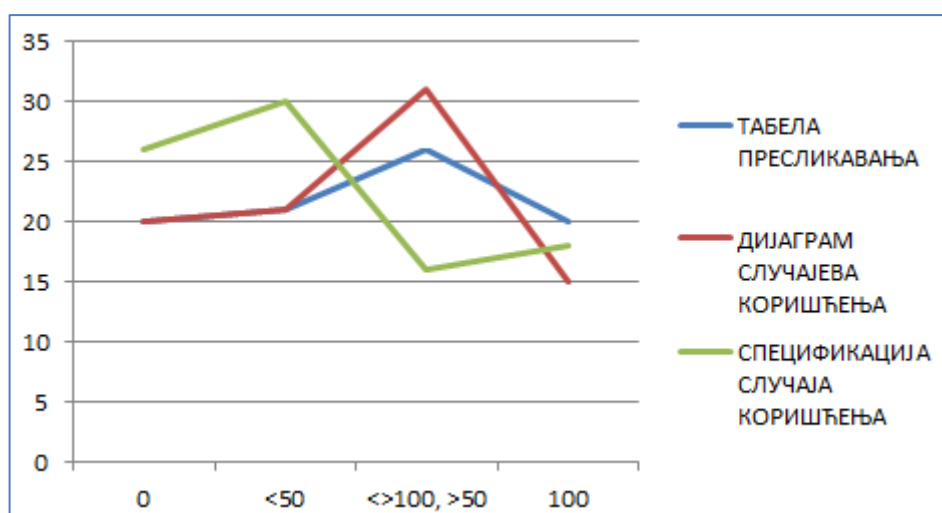
У школској 2014/15 години највише радова студената исказало је задовољавајући ниво у све три категорије: називи елемената, речник података и организација елемената.

### 5.6.3. Евалуација модела дизајна софтверских функција

У оквиру одељка прилози дати су подаци о примени метричког модела за евалуацију модела дизајна софтверских функција, односно табеле пресликавања пословних процеса у софтверске функције, дијаграма случајева коришћења и спецификације случаја коришћења. У наставку је дат сумарни приказ резултата за школску 2013/14 и 2014/15 годину.

Табела 5.6.1.3.1. Табеларни сумарни приказ података о о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2013/14

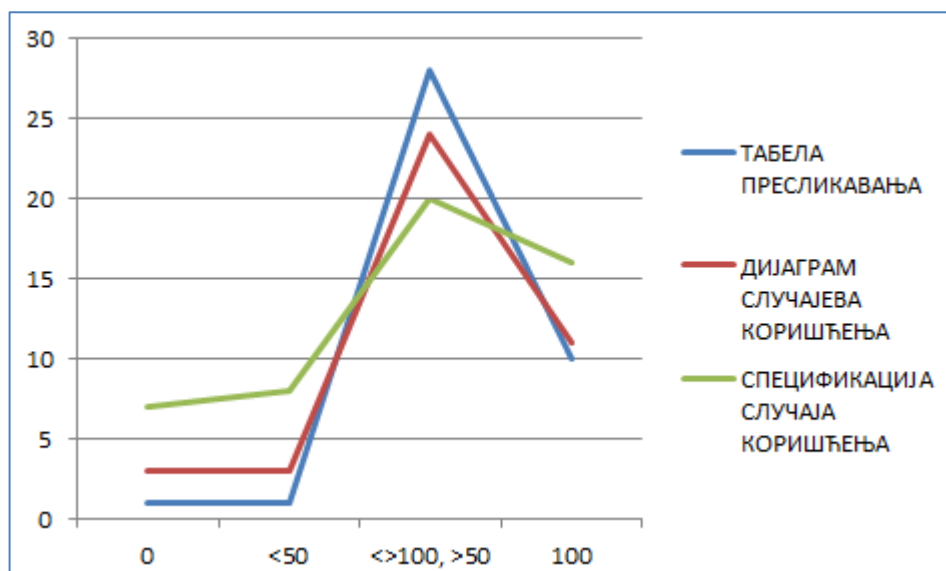
	2013/14		
УСЛОВИ	ТАБЕЛА ПРЕСЛИКАВАЊА	ДИЈАГРАМ СЛУЧАЈЕВА КОРИШЋЕЊА	СПЕЦИФИКАЦИЈА СЛУЧАЈА КОРИШЋЕЊА
0	20	20	26
<50	21	21	30
<>100, >50	26	31	16
100	20	15	18



Слика 5.6.1.3.1. Графички приказ сумарних података о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2013/14

Табела 5.6.1.3.2. Табеларни сумарни приказ података о о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2014/15

	2014/15		
УСЛОВИ	ТАБЕЛА ПРЕСЛИКАВАЊА	ДИЈАГРАМ СЛУЧАЈЕВА КОРИШЋЕЊА	СПЕЦИФИКАЦИЈА СЛУЧАЈА КОРИШЋЕЊА
0	1	3	7
<50	1	3	8
<>100, >50	28	24	20
100	10	11	16



Слика 5.6.1.3.2. Графички приказ сумарних података о примени метричког модела за евалуацију модела дизајна софтверских функција – шк. 2014/15

На основу сумарног приказа података о примени метричког модела за евалуацију модела дизајна софтверских функција за школску 2013/14 годину, може се утврдити да је за област спецификације случаја коришћења највише студената исказало недовољно познавање области, док су за област табеле пресликавања и дијаграма случајева коришћења највише резултата студената исказало задовољавајући ниво квалитета.

У школској 2014/15 години, највећи број студената у свакој од три области студенти исказали задовољавајући ниво квалитета радова.

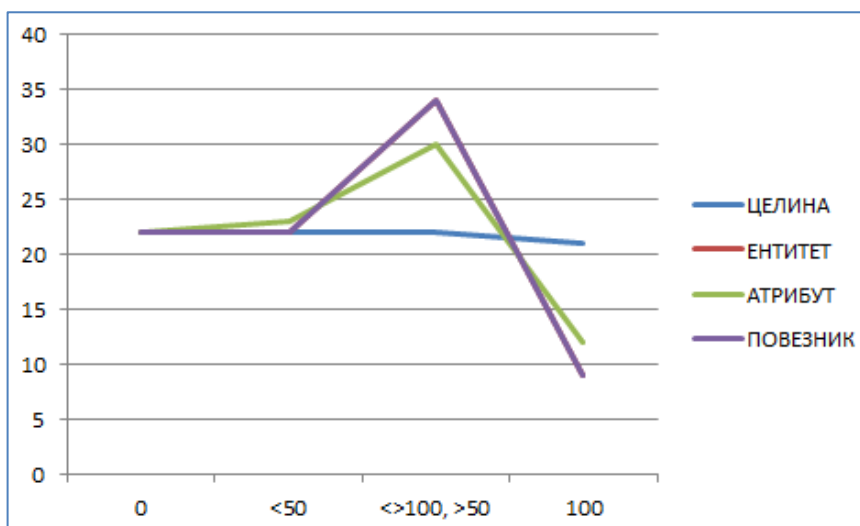
#### 5.6.4. Евалуација концептуалног модела података

У оквиру одељка прилози дати су подаци о примени метричког модела за евалуацију концептуалног модела података. У наставку је дат сумарни приказ резултата за школску 2013/14 и 2014/15 годину.

Табела 5.6.1.4.1. Табеларни сумарни приказ података о о примени метричког модела за евалуацију концептуалног модела података – шк. 2013/14

	2013/14				
УСЛОВИ	ЦЕЛИНА	ЕНТИТЕТ	АТРИБУТ	ПОВЕЗНИК	
0	22	22	22	22	
<50	22	22	23	22	
<>100, >50	22	34	30	34	
100	21	9	12	9	

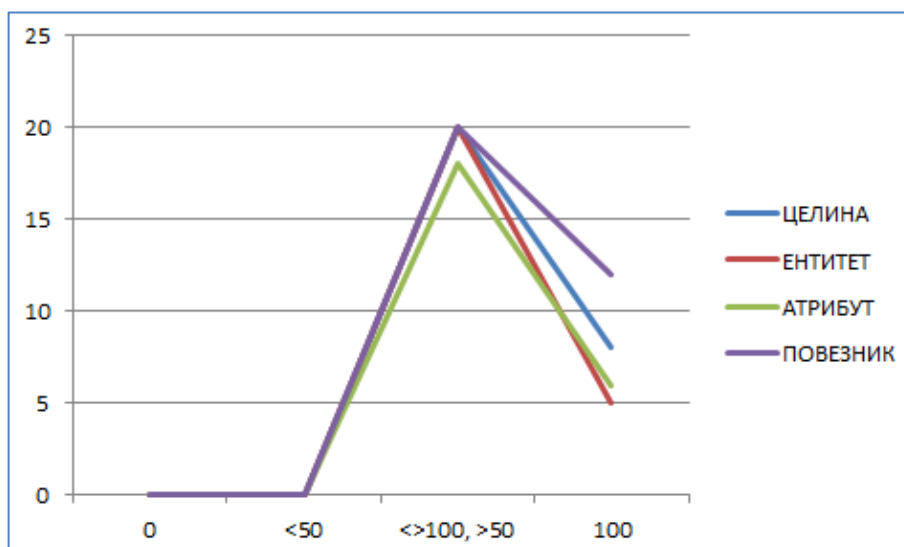




Слика 5.6.1.4.1. Графички приказ сумарних података о примени метричког модела за евалуацију концептуалног модела података – шк. 2013/14

Табела 5.6.1.4.2. Табеларни сумарни приказ података о о примени метричког модела за евалуацију концептуалног модела података – шк. 2014/15

	2014/15			
УСЛОВИ	ЦЕЛИНА	ЕНТИТЕТ	АТРИБУТ	ПОВЕЗНИК
0	0	0	0	0
<50	0	0	0	0
<>100, >50	20	20	18	20
100	8	5	6	12



Слика 5.6.1.4.2. Графички приказ сумарних података о примени метричког модела за евалуацију концептуалног модела података – шк. 2014/15

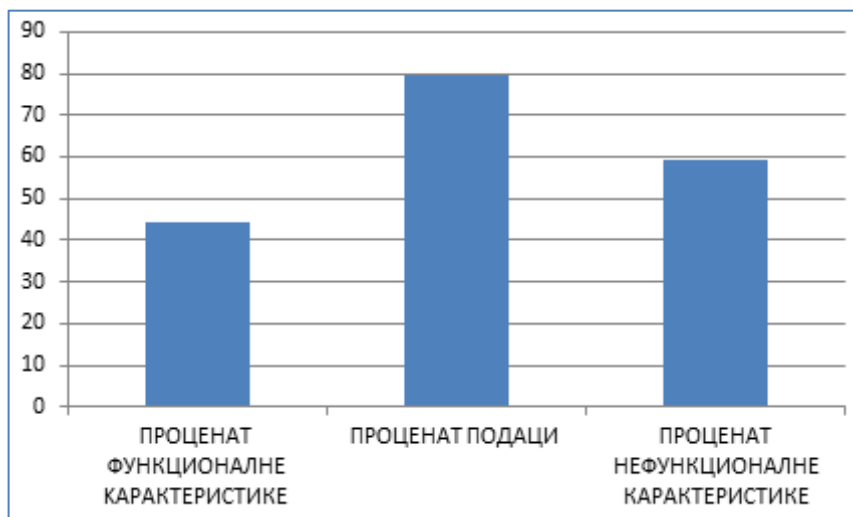
На основу сумарног приказа података о примени метричког модела за евалуацију концептуалног модела података, у школској 2013/14 години као и у школској 2014/15 години у свих четири категорије студенти су исказали задовољавајући ниво познавања области, при чему је у школској 2013/14 расподела вредности за категорије ентитет и повезник идентична, док је у школског 2014/15 години расподела за свих 4 категорија приближна.

### 5.6.5. Евалуација софтверске апликације

У наставку су приказани сумарни подаци о примени метричког модела за евалуацију софтверске апликације, који се односе на податке о евалуацији софтверских апликација реализованих у оквиру студентских софтверских пројеката у области развоја информационог система Техничког факултета „Михајло Пупин“ из Зрењанина реализованих у школској 2013/14 и 2014/15 години.

Табела 5.6.5.1. Сумарни приказ успешности студентских софтвера према областима

ОБЛАСТ	Процент успешности
<b>ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	44 %
<b>КВАЛИТЕТ ПОДАТАКА</b>	79 %
<b>НЕФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	59 %



Слика 5.6.5.1. Графички приказ успешности студентских софтвера према областима

Применом метричког модела за евалуацију софтверске апликације на наведеном узорку могу се извести следећи закључци:

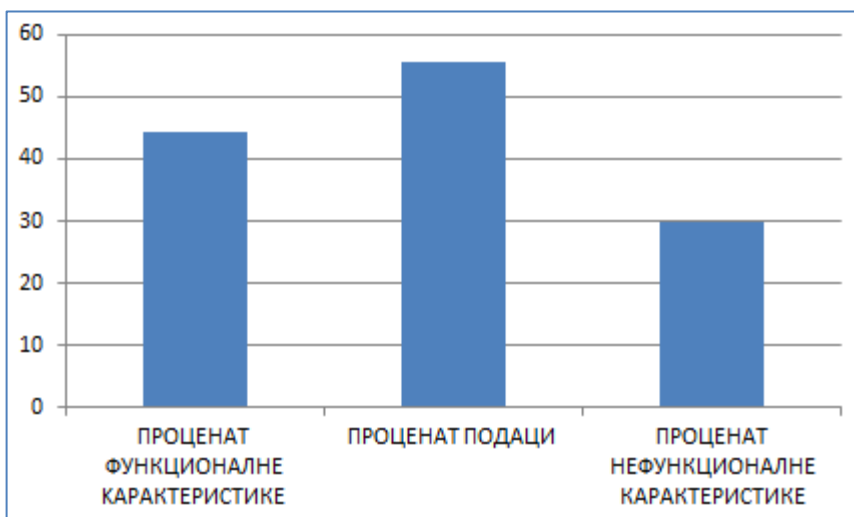
1. Постоји потреба за коришћењем тежинских фактора, којим би се појединим карактеристикама и групама карактеристика дао правилнији значај. Увођењем тежинских фактора и правилним подешавањем, резултати укупног броја поена, односно процента покривености карактеристика би се мењао и изражавао реалнију слику у складу са ставовима оцељивача о значају појединих карактеристика у вредновању. На пример, однос функционалних карактеристика и квалитета података према нефункционалним карактеристикама може бити одређен тежинским факторима.
2. Поједине карактеристике студентских софтвера нису заступљене (додељено им је 0 поена у примени метричког модела), јер нису дефинисане као потребне у спецификацији захтева и нису реализоване. С обзиром да су у примени метричког модела обухваћене све карактеристике, па чак и оне које нису биле захтеване, јавља се потреба за филтрирањем метричког модела ради прилагођавања спецификацији захтева.
3. За потребе евалуације софтвера који представља унапређење раније реализованог софтвера, може се размотрити могућност коришћења истог

метричког модела, при чему се мери „разлика поена“ почетног и унапређеног софтвера. Такође, може се користити и додатни метрички модел који је намењен ситуацијама када се ради о унапређењу софтвера, са одговарајућим метрикама које се односе на број и квалитет унапређених софтверских функција.

Уколико бисмо применили тежинске факторе, нпр. 1 за функционалне карактеристике, 0.7 за квалитет података, 0.5 за нефункционалне карактеристике, добили бисмо:

Табела 5.6.5.2. Сумарни приказ успешности студентских софтвера према областима уз примену тежинских фактора

<b>ОБЛАСТ</b>	<b>Процент успешности</b>
<b>ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	44
<b>КВАЛИТЕТ ПОДАТАКА</b>	56
<b>НЕФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>	30



Слика 5.6.5.2. Графички приказ успешности студентских софтвера према областима уз примену тежинских фактора

## 5.7. Анализа резултата примене метричких модела за мониторинг успеха и процену трајања процеса развоја софтвера

У циљу илустрације примене предложеног метричког модела за мониторинг успеха процеса реализације дистрибуираног агилног софтверског пројекта, као и модела за процену трајања процеса развоја софтвера на основу модела пословних процеса и модела дизајна укључујући технолошке могућности имплементације, у наставку ће бити приказани делимично сумарни подаци (приказ броја поена према групама карактеристика) на основу реализованих студентских софтверских пројеката у школској 2013/14 и 2014/15 години.

### 5.7.1. Примена модела за мониторинг успеха процеса развоја софтвера

Основне карактеристике студентских пројеката који су реализовани у школској 2013/14 и 2014/15 години су:

1. можемо сматрати да су пројекти мањег обима
2. нису реализовани у потпуности у складу са свим елементима управљања пројектима (нпр. трошкови, набавке... нису разматрани),
3. не можемо их сматрати стратегијски (за Технички факултет „Михајло Пупин“ у Зрењанину) релевантним (иако се односе на реализацију сегмената информационог система Техничког факултета „Михајло Пупин“ у Зрењанину).

Из наведених разлога, узорак истраживања може бити коришћен само за примену оперативног нивоа мониторинга (агилни ниво праћења успешности) и зато у овом одељку приказујемо анализу примене предложеног модела агилног мониторинга софтверског пројекта. Подаци о примени са конкретним софтверским пројектом.

С обзиром да нису сви студентски пројекти из узорка имали као резултат модел пословних процеса, у овом одељку биће приказана илустрација примене предложеног модела за процену трајања софтверског пројекта на примерима студентских пројеката који су као један од артефакта имали модел пословних процеса.

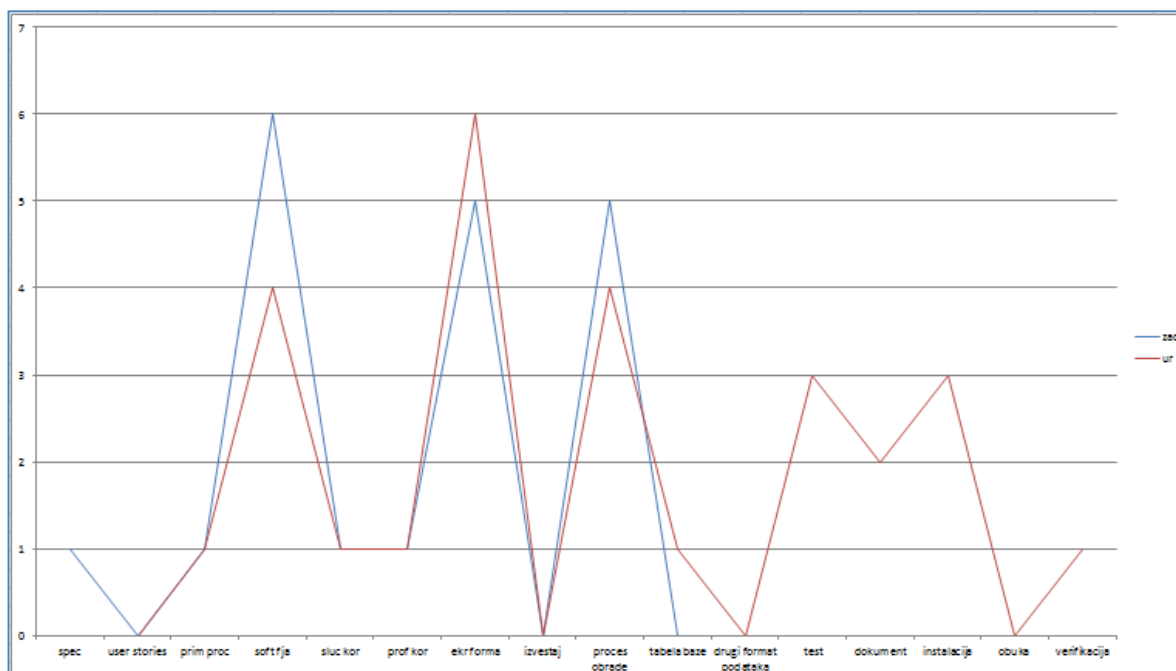
У наставку ће бити дат визуални приказ тока процеса реализације софтвера „Алумни“ (деталји су дати у прилогу).

Табела 5.7.1.1. Табеларни приказ тока процеса развоја софтвера „Алумни“

datum	zad /ur	spec	user stories	prim proc	soft tja	sluc kor	prof kor	ekr forma	izvestaj	proces obrade	tabela baze	drugi format podataka	test	dokument	instalacija	obuka	verifikacija
13.3.2014.	zad		1		1	1	1	1									
25.3.2014.	ur				1	1	1	1									
25.3.2014.	zad							1									
1.4.2014.	zad							1									
1.4.2014.	ur							1									
2.4.2014.	zad							1									
2.4.2014.	ur							1									1
3.4.2014.	zad							1									
3.4.2014.	ur							1									
8.4.2014.	zad																
8.4.2014.	ur															1	
16.4.2014.	zad							1									
16.4.2014.	ur							1									
18.4.2014.	zad																
18.4.2014.	ur							1								1	
22.4.2014.	zad					1					1						
22.4.2014.	ur							1			1						
6.5.2014.	zad					2					2						
6.5.2014.	ur					1					1						
8.5.2014.	zad																
8.5.2014.	ur					2					2						
13.5.2014.	zad					1					1						
13.5.2014.	ur													1			
24.5.2014.	zad																
24.5.2014.	ur															1	
27.5.2014.	zad					1					1						
27.5.2014.	ur															1	
28.5.2014.	zad																
28.5.2014.	ur										1			1			
4.7.2014.	zad																
4.7.2014.	ur													1		1	

Табела 5.7.1.2. Сумарни приказ активности у оквиру процеса развоја софтвера „Алумни“

	spec	user stories	prim proc	soft fja	sluc kor	prof kor
UKUPNO	1	0	2	10	2	2
zad	1	0	1	6	1	1
ur	0	0	1	4	1	1
	ekr forma	izvestaj	proces obrade	tabela baze	drugi format podataka	
UKUPNO	11	0	9	1	0	
zad	5	0	5	0	0	
ur	6	0	4	1	0	
	test	dokument	instalacija	obuka	verifikacija	
UKUPNO	3	2	3	0	1	
zad	0	0	0	0	0	
ur	3	2	3	0	1	



Слика 5.7.1.1. Графички упоредни приказ задатих и реализованих елемената процеса развоја софтвера у оквиру мониторинга пројекта „Алумни“

У оквиру графичког приказа можемо видети сумарни приказ односа задати и урађених активности у оквиру реализације софтверског пројекта „Алумни“, чиме је илустрована примена предложеног метричког модела за мониторинг тока и успешности процеса развоја софтвера.

### 5.7.2. Примена модела за процену трајања софтверског пројекта на основу модела пословних процеса

На основу формуле која је дата у оквиру дефиниције метричког модела за процену трајања процеса имплементације софтвера у области развоја информационих система, као о на основу усвојене структуре карактеристика и тежинских фактора који су предложени у оквиру одељка са дефиницијом метричког модела, у овом примеру можемо израчунати:

$$MPPT = SPT + PT + OJT$$

За конкретан пример модела пословних процеса „Алумни“ који је анализиран у оквиру прилога, у овом одељку можемо приказати резултате израчунавања трајања софтверског пројекта у делу имплементације софтвера, под претпоставком да су остали нефункционални услови и параметри идеални.

$SPT = 2$  складишта података сличне структуре и сложености \* сложеност по броју ел. Под (1) \* сложеност по структури података у складишту (1) =  $2 * 1 * 1 = 2$

$PT = 3$  процеса =

1 улазни (1) \* (улазни ток једноставне структуре (1) + интерни ток једноставне структуре (1) + складиште података једноставне структуре (1)) +  
 1 излазни (2) \* (интерни ток једноставне структуре (1) + излазни ток једноставне структуре (1) + складиште података једноставне структуре (1)) +  
 1 обрада (3) \* (улазни ток једноставне структуре (1) + излазни ток једноставне структуре (1) + улазно складиште једноставне структуре (1) + излазно складиште једноставне структуре (1)) =  $1 * 3 + 2 * 3 + 3 * (1 * 1 + 1 * 1) = 3 + 6 + 6 = 15$

$OJT = 3$  организационе јединице = 1 интерна \* (број процеса (1) \* (број складишта података (1) + број токова података (1))) + 2 екстерне \* (број процеса (1) \* (број складишта података (1) + број токова података (1))) =  $1 * (1 + 1) + 2 * (1 * (1 + 1)) = 2 + 4 = 6$

$$MPPT = 2 + 15 + 6 = 23 \text{ MPPTр}$$

Укупан износ поена у процени трајања имплементације софтвера у развоју информационог система на основу анализе карактеристика елемената модела пословних процеса (MPPTр) је 23.

Уколико бисмо усвојили (на основу статистике претходних искустава) да један MPPTр може трајати  $\frac{1}{2}$  човек-дан, тада бисмо могли израчунати да је трајање имплементације за наведени пример софтверског пројекта:

$$MPPT = 23 * \frac{1}{2} = 11.5 \text{ човек/дан.}$$

## 6. УНАПРЕЂЕЊЕ ПРОТОТИПА

### 6.1. Планирање унапређења прототипа

#### 6.1.1. Статистички приказ резултата анализе степена реализације захтеваних функционално-технолошких карактеристика у односу на предложени функционално-технолошки теоријски модел

У наставку ће бити приказани сумарни подаци покривености захтева према категоријама које представљају основ даље спецификације захтева.

Табела 6.1.1.1. Статистички приказ степена реализације захтеваних софтверских функција почетним прототипом

ОБЛАСТ	Да	Делимично	Не
Развој информационих система	2	1	3
Методологија развоја софтвера	3	2	2
Методологија пројектног менаџмента – РМВОК	1	7	5
Дистрибуирани информациони системи	0	1	2
Адаптивни и Адаптибилни системи	0	1	12
Системи за управљање пословним правилима, аутоматско резонување и онтологије	0	1	2
Управљање перформансама пословних процеса	0	2	1
Успех пројекта	0	2	4
УКУПНО	6	17	31



Слика 6.1.1.1. Графички приказ процента реализације захтеваних функција у оквиру почетног прототипа

Анализом резултата евалуације реализованог почетног прототипа можемо утврдити да је 43% захтеваних функција реализовано потпуно или делимично, док 57% захтеваних функција није реализовано.

#### 6.1.2. Издвајање непокривених функционално-технолошких области постојећим почетним прототипом

У оквиру одељка са прилозима издвојен је скуп функционалних карактеристика које нису обухваћене почетним прототипом, а затим је извршена анализа покривености одговарајућих карактеристика постојећим решењима из праксе. На основу наведене анализе издвојене су само функционалне карактеристике које су захтеване, а нису покривене почетним прототипом.

Унапређени прототип треба да има:

А) Функционалне карактеристике подршке метрички заснованом управљању софтверским пројектима

Б) Основне карактеристике технолошке имплементације:

- Дистрибуираности
- Метричке заснованости функционисања
- Процесирање применом екстерних правила

В) Додатне технолошке карактеристике:

- Адаптивност / Адаптибилност - Детекција промене у окружењу
- Адаптивност / Адаптибилност - Реаговање на промене
- Адаптивност / Адаптибилност - Самозаштита
- Адаптивност / Адаптибилност - Самоизлечење
- Адаптивност / Адаптибилност - Самоконфигурисање уз одлучивање о измени структуре или понашања
- Адаптивност / Адаптибилност - Самооптимизација
- Адаптивност / Адаптибилност - Свесност окружења - контекста
- Адаптивност / Адаптибилност - Свесност сопственог стања
- Адаптивност / Адаптибилност - Акција
- Адаптивност / Адаптибилност - Детекција промене
- Адаптивност / Адаптибилност - Одлучивање о акцији
- Адаптивност / Адаптибилност - Промене изменом понашања
- Адаптивност / Адаптибилност - Промене интерних стања и/или структуре

Г) Захтеви методолошког оквира истраживања:

- Адаптибилност у односу на промене типа софтвера
- Адаптибилност у односу на промене типа софтверског пројекта
- Адаптибилност у односу на технологију израде
- Адаптибилност у односу на промене технолошких развојних окружења

### **6.1.3. Анализа захтева непокривених области и план реализације унапређења**

У наставку ће бити приказана анализа захтева и план реализације даљег унапређења.

А) Функционалне карактеристике

С обзиром на велик број доступних готових техничких решења софтверских алата за подршку управљању пројектима и мањи број, али ипак доступност техничких решења софтверских алата за подршку управљању софтверским пројектима, у смислу унапређења почетног прототипа развијеног у оквиру ове докторске дисертације неће у наставку бити фокус на функционалном унапређењу почетног прототипа, већ на технолошко-имплементационом унапређењу, у складу са захтевима методолошког оквира истраживања.

Б) Основне карактеристике технолошке имплементације:

- Дистрибуираности – реализација путем web сервиса, односно реализација трослојне дистрибуиране архитектуре – слој података (база података, XML), процесни слој (web сервиси), презентациони слој (кориснички интерфејс web апликације)
- Метричке заснованости функционисања – примена web сервиса за мерење квалитета артефакта и производа, мерење карактеристика процеса и ресурса, подршку мониторингу успеха и процени трајања пројекта



- Процесирање применом екстерних правила – чување екстерних правила која одређују начин рада web сервиса у XML форми, динамичко повезивање и коришћење web сервиса

В) Додатне технолошке карактеристике – у односу на анализу теоријских основа и постојећих истраживања, потребно је јасно разграничити функционалне особине и извршити категоризацију у односу на адаптивност или адаптибилност. Ова категоризација ће бити извршена на основу [Salehie&Tahvildari, 2009]:

- Адаптивност - Детекција промене у окружењу
- Адаптивност - Самозаштита
- Адаптивност - Самоизлечење
- Адаптивност - Самоконфигурисање уз одлучивање о измени структуре или понашања
- Адаптивност - Самооптимизација
- Адаптивност - Свесност окружења - контекста
- Адаптивност - Свесност сопственог стања
- Адаптивност - Детекција промене
- Адаптивност - Одлучивање о акцији
- Адаптибилност - Реаговање на промене
- Адаптибилност - Акција
- Адаптибилност - Промене изменом понашања
- Адаптибилност - Промене интерних стања и/или структуре

Дакле, у функционалном смислу, с обзиром на обухват дефинисан насловом докторске дисертације и методолошким оквиром, језгро потребног унапређења софтверског прототипа представља подршка адаптибилности односно:

- Адаптибилност - Реаговање на промене
- Адаптибилност - Акција
- Адаптибилност - Промене изменом понашања
- Адаптибилност - Промене интерних стања и/или структуре

Реализација адаптибилности треба да се односи на сва 3 слоја апликације:

- Слој података (изменом структуре XML фајлова) – измена структуре
- Слој процесирања (композиција/позиви различитих web сервиса у складу са потребама измена) – измена структуре и понашања, реаговање на промене, акција
- Презентациони слој – измене изгледа корисничког интерфејса у односу на измене података који стижу из слоја процесирања

Г) Захтеви методолошког оквира истраживања:

На основу резултата анализе истраживања у области докторске дисертације, јавила се потреба за подршком адаптибилности у односу на промене методологије развоја софтвера, методологије управљања софтверским пројектом, као и методологије у домену примене метрика у процени квалитета артефакта, производа, мониторингу успеха пројекта и процени трајања пројекта. На овај начин добијамо комплетнији потребан скуп домена примене адаптибилности на промене које се односе на:

- тип софтвера
- тип софтверског пројекта
- технологију израде софтвера
- технолошка развојна окружења
- методологију развоја софтвера
- методологију управљања софтверским пројектом
- методологију примене метрика у процени квалитета артефакта

- методологију примене метрика у процени квалитета производа
- методологију примене метрика у мониторингу успеха пројекта
- методологију примене метрика у процени трајања пројекта

Потребе за адаптивност у односу на различите критеријуме можемо груписати:

а) Технолошки орјентисани захтеви – адаптивност у односу на:

- технологију израде софтвера
- технолошка развојна окружења

Технолошки орјентисани захтеви у односу на адаптивност подржани су чињеницом да web сервиси пружају подршку интеграцији у оквиру различитих технолошких окружења. Најчешће заступљене технолошке платформе које се јављају на тржишту рада истражене су у оквиру [Kazi et al, AIIT, 2012c] и обухватају: јава, PHP/MYSQL и Microsoft ASPX/C#/MSSQL. У делу унапређења почетног прототипа биће илустрована примена web сервиса у различитим технолошким платформама.

б) Семантички орјентисани захтеви – адаптивност у односу на:

- тип софтвера
- тип софтверског пројекта
- методологију развоја софтвера
- методологију управљања софтверским пројектом
- методологију примене метрика у процени квалитета артефакта
- методологију примене метрика у процени квалитета производа
- методологију примене метрика у мониторингу успеха пројекта
- методологију примене метрика у процени трајања пројекта

Подршка за семантички орјентисане захтеве у односу на адаптивност биће подржана кроз одговарајуће XML шифарнике којима се дефинишу елементи у односу на потребе различитих семантичких области примене.

У наставку је дата табела са сумарним приказом плана унапређења почетног прототипа.

Табела 6.1.3.1. Области технолошког унапређења прототипа и план имплементације

<b>КАТЕГОРИЈА ТЕХНОЛОШКОГ УНАПРЕЂЕЊА</b>	<b>ПЛАН ИМПЛЕМЕНТАЦИЈЕ</b>
Дистрибуираност	Трослојна архитектура: <ul style="list-style-type: none"> <li>• Кориснички интерфејс (web апликација)</li> <li>• web сервиси</li> <li>• база података и XML</li> </ul>
Метричка заснованост	Web сервиси за анализу артефакта, производа, процеса и ресурса у реализацији софтверског пројекта, подршку мониторингу успеха и процени трајања пројекта
Примена екстерних правила	<ul style="list-style-type: none"> <li>• XML запис правила која описују начин функционисања и примене web сервиса,</li> <li>• динамичко повезивање и коришћење web сервиса</li> </ul>
Адаптивност	Слојеви адаптивности: <ul style="list-style-type: none"> <li>• Презентациони слој – кориснички интерфејс динамички мења свој изгледа на основу промене података које добија</li> <li>• Слој процесирања – примена различитих web сервиса, динамичко позивање различитих web сервиса</li> <li>• Слој података – измене структуре XML записа који може да буде основ чувања података</li> </ul>
	Технолошки контекст адаптивности: Адаптивност у односу на технологију – примена web сервиса у различитим технолошким окружењима
	Семантички контекст адаптивности – параметри у XML: категоризација софтвера и пројекта, методологија развоја софтвера и управљања пројектом, различите категорије метрика

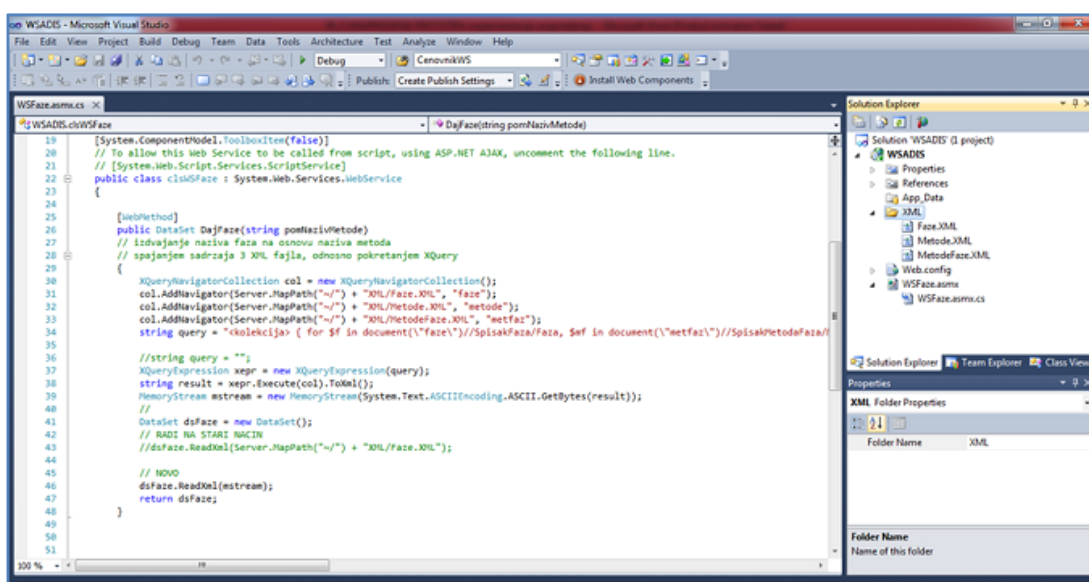
## 6.2. Опис елемената имплементације унапређења прототипа

Дистрибуираност вишеслојне апликације у оквиру овог одељка реализована је кроз развој апликације, где су најчешћи елементи три основна слоја имплементирани као:

- Слој података (база података, XML фајлови)
- Слој процесирања (web сервиси или динамичке библиотеке – \*.dll)
- Презентациони слој (кориснички интерфејс web апликације и сл.).

### 6.2.1. Подршка дистрибуираности и адаптивности на промене методологије развоја софтвера - креирање и тестирање ASP.NET Web сервиса уз коришћење XML шифарника

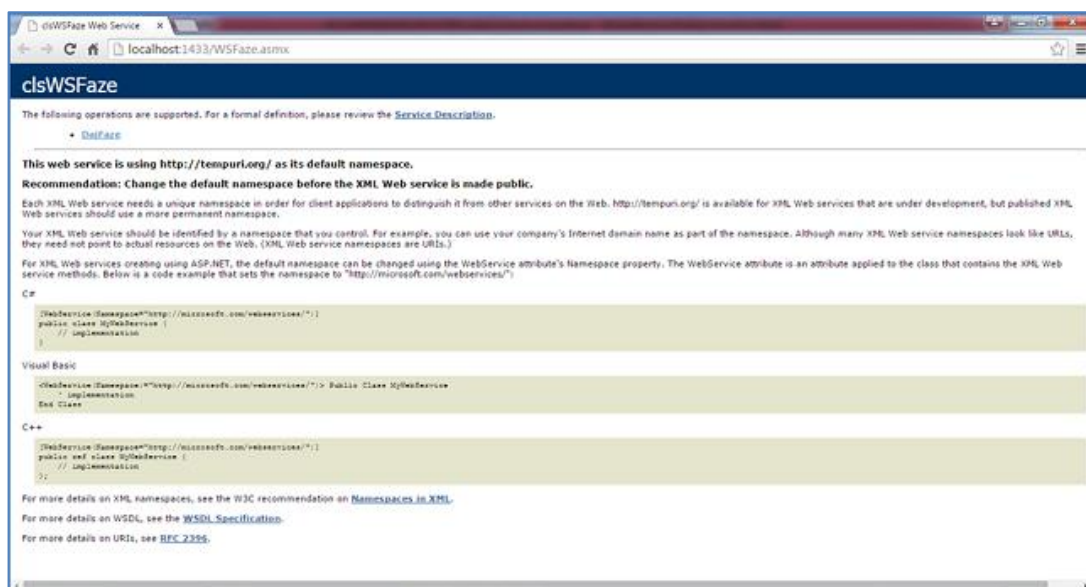
Ради подршке дистрибуиране обраде података, реализован је ASPX Web сервис као посебан пројекат. Слика 6.2.1.1. приказује развојно окружење Microsoft Visual Studio у оквиру ког је реализован ASPX Web сервис, списак фајлова пројекта ASPX Web сервиса и део изворног кода реализованог Web сервиса.



Слика 6.2.1.1. Изворни код и развојно окружење за креирање ASPX Web сервиса

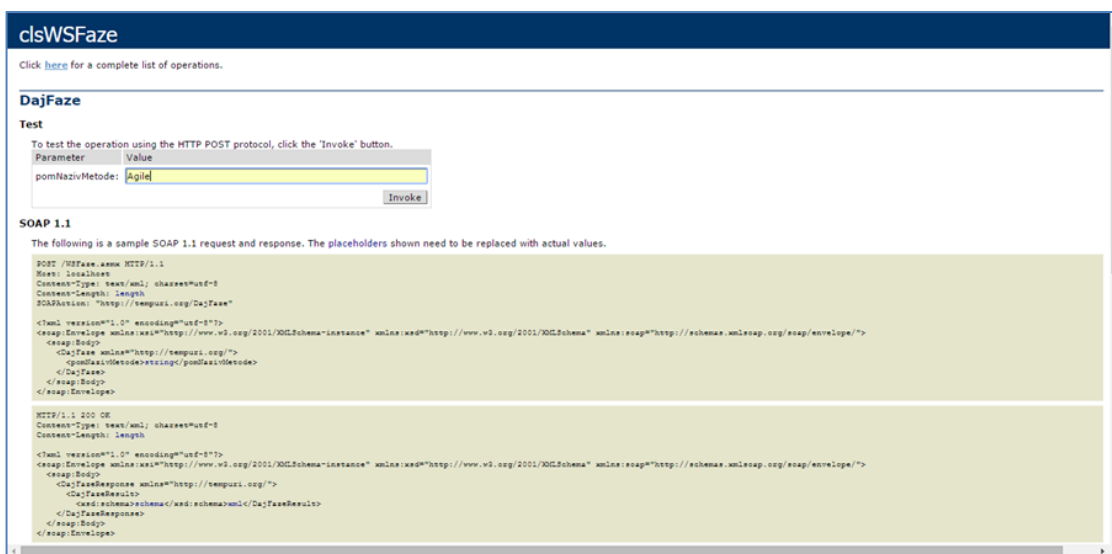
У оквиру наведеног решења омогућено је читање фаза реализације софтверског пројекта у складу са одабраном методологијом. Подаци о фазама и методама су дате у оквиру XML фајлова: Faze.XML, Metode.XML и MetodeFaze.XML, који се налазе у фолдеру у оквиру пројекта Web сервиса. Да би се могли повезати подаци из више XML фајлова и вршити филтрирање, користи се XQUERY. Да бисмо могли да радимо са XQUERY, додајемо у bin фолдер, односно у References везу ка Microsoft.XML.Xquery.dll фајлу, који је преузет са [MSXQueryDownload].

Након покретања Web сервиса у локалном развојном режиму, Web сервис се стартује у localhost-у на порту који је доступан, као што је приказано на слици 6.2.1.2. (<http://localhost:1433/WSFaze.asmx>). Након што је покренут Web сервис, добија се списак процедура које је могуће тестирати.



Слика 6.2.1.2. Web сервис након покретања у локалном режиму рада

Након избора назива процедуре „DajFaze“ добијамо могућност да се унесе параметар процедуре, који је потребан за извршавање процедуре, као што је приказано на слици 6.2.1.3.



Слика 6.2.1.1.3. Унос параметра ради извршавања процедуре из Web сервиса

Након уноса параметра избором тастера „Invoke“ добија се резултат у форми XML документа, слика 6.2.1.1.4.

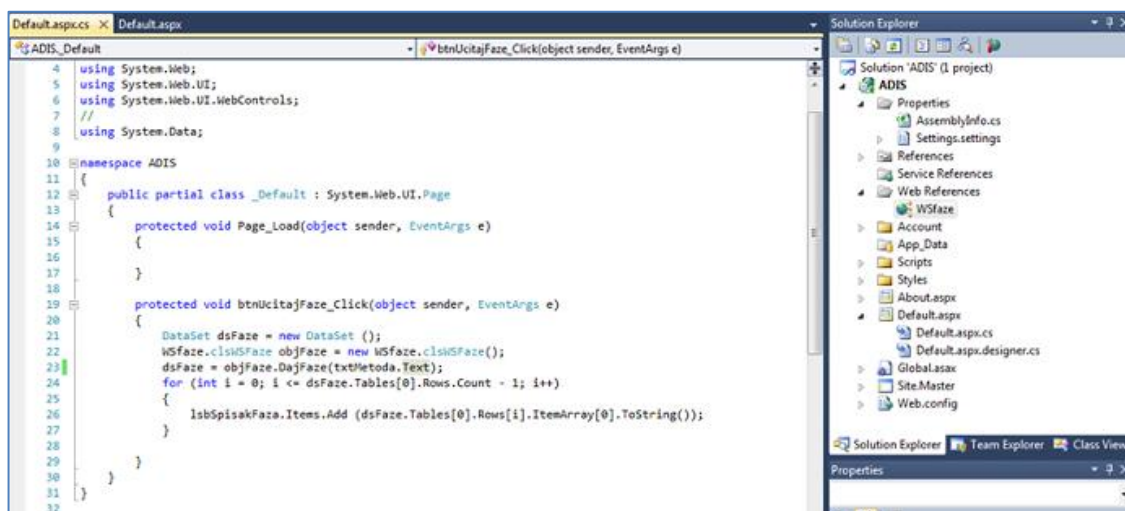


Слика 6.2.1.1.4. Резултат извршавања web сервиса - XML

## 6.2.2. Прилагођавање различитим технолошким окружењима - повезивање и коришћење Web сервиса у оквиру реализације различитих технологија корисничких интерфејса

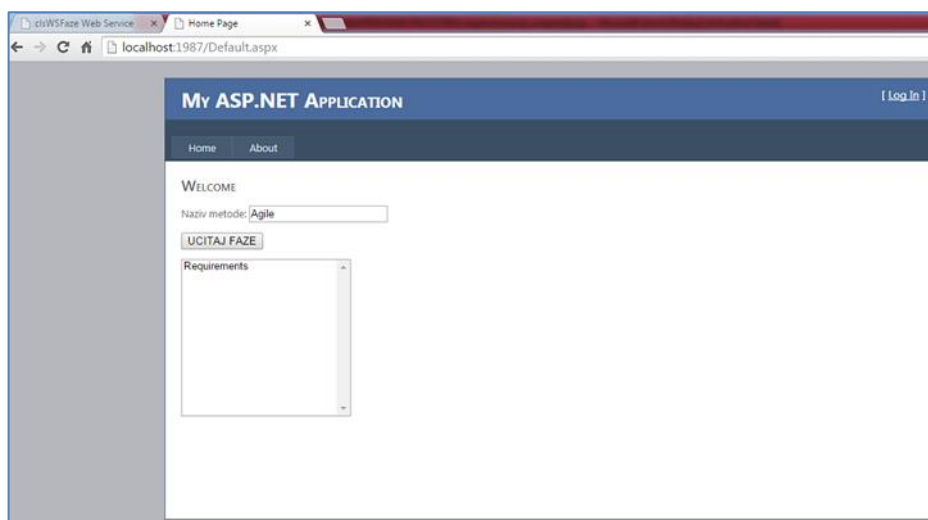
### 6.2.2.1. Кориснички интерфејс ASPX web апликација

Web сервис се може користити у оквиру реализације корисничких интерфејса различитих технологија имплементације. У наставку је приказано развојно окружење ASPX.NET web апликације и повезивање ASPX Web сервиса ради коришћења у корисничком интерфејсу ASPX.NET web апликације (Слика 6.2.2.1).



Слика 6.2.2.1. Развојно окружење ASPX web апликације са програмским кодом уз примену web сервиса

Примена web сервиса је приказана у оквиру извршног режима ASPX web апликације као корисничког интерфејса (Слика 6.2.2.2.). Приликом покретања ASPX web апликације, претходно је потребно покренути ASPX web сервис.



Слика 6.2.2.2. Извршавање пробне ASP.NET апликације за тестирање рада са ASP.NET web сервисом

### 6.2.2.2. Кориснички интерфејс PHP web апликација

У циљу демонстрирања могућности адаптивности реализованог софтверског алата на промене технолошких развојних окружења, креирана је PHP апликација. Изворни PHP код је приказан у оквиру Notepad++ окружења, слика 6.2.2.3. На наведеној слици, као и на наредној слици 6.2.2.4. представљено је коришћење, у оквиру PHP програмског кода, унапређене верзије web сервиса који је проширен додатним функцијама, како је описано у одељку 6.2.3.1.

```

C:\xampp\htdocs\adis\delovi\desnopocetna.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
desnopocetna.php
14 <b>FAZE RAZVOJA SOFTVERA</b>
15 <?php
16
17 //POVEZIVANJE PHP KODA SA ASP.NET WEB SERVISOM
18 require_once('lib/nusoap.php');
19 $wsdl='http://localhost:1433/WSFaze.asmx?wsdl';
20 $client = new nusoap_client($wsdl, 'WSDL');
21 $error = $client->getError();
22 if ($error) {
23     die("client construction error: {$error}\n");
24 }
25 // ----- UKUPAN BROJ FAZA
26 $param = array('pomNazivMetode' => 'Waterfall');
27 $answer =array();
28 $answer = $client->call('DajBrojFaza', array('parameters' => $param), '', '', false, true);
29 $brojfaza = $answer['DajBrojFazaResult'];
30 print_r('broj faza: '.$brojfaza);
31
32 // ISPIS VISESTRUKIH VREDNOSTI
33 echo "<ul>";
34 for ($row = 0; $row < $brojfaza; $row++) {
35     // ----- NTA FAZA
36     $param = array('pomNazivMetode' => 'Waterfall', 'rbFaze' => $row+1);
37     $answer =array();
38     $answer = $client->call('DajNtuFazu', array('parameters' => $param), '', '', false, true);
39     $nazivfaze = $answer['DajNtuFazuResult'];
40     echo "<li>". 'naziv faze: '.$nazivfaze."</li>";
41 }
42 echo "</ul>";
43

```

Слика 6.2.2.3. Изворни код пробне PHP апликације који укључује рад са ASP.NET web сервисом у оквиру Notepad ++ развојног окружења

Да би се из програмског кода PHP апликације могао повезати и извршити ASP.NET web сервис, потребно је креирати подфолдер, нпр. са називом lib у оквиру фолдера где је смештена PHP апликација и ту прекопирати библиотеку „nusoap.php“ (преузето са [NuSOAPDownload]), која садржи PHP класе које омогућавају рад са ASP.NET web сервисима. Главни фајл са класама који представља интерфејс ка другим класама из

наведене библиотеке се директно референцира у оквиру PHP кода апликације како би омогућио рад са web сервисима (класа „*\$client = new nusoap\_client...*“, позив методе: „*\$client->call...*“).

Коначно, демонстрирање коришћење „услуга“ ASP.NET web сервиса из PHP апликације (ASP.NET web сервис мора бити стартован пре покретања PHP апликације) приказано је на следећој слици (приказује се списак фаза које је „доставио“ web сервис „u“).



Слика 6.2.2.4. Извршавање пробне PHP апликације за тестирање рада са ASP.NET web сервисом

### **6.2.3. Графичка адаптивност и адаптивност слоја података – динамичко креирање компоненти корисничког интерфејса PHP апликације на основу ASP.NET Web сервиса и креирање структуре XML**

Адаптивност презентационог слоја реализована је кроз динамичко генерисање корисничког интерфејса на основу података из web сервиса. Адаптивност слоја података реализована је у делу структуре XML датотека које могу бити коришћене као носиоци записа података (програмски код у прилогу).

#### **6.2.3.1. Допуна функција ASPX web сервиса**

Како би се илустровала адаптивност корисничког интерфејса, реализована је допуна web сервиса додатним функцијама, тако да су реализоване методе: *DajBrojFaza*, *DajFaze*, *DajNtuFazu*, *DajPrvuFazu*. Ради реализације циклуса, било је важно реализовати додатну функцију „*DajBrojFaza*“ која враћа укупан број фаза и представља горњу границу бројача у циклусу.

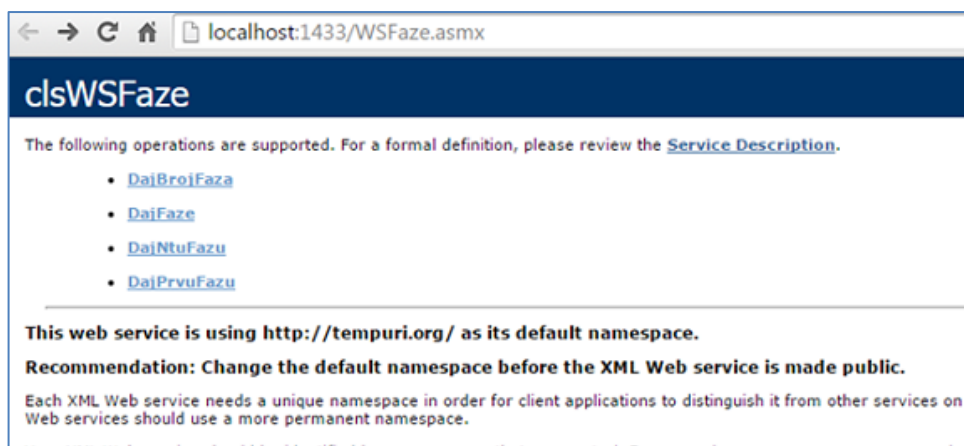
```

77 public string DajNtuFazu(string pomNazivMetode, int rbFaze)
78 // - kreće od 1, pa zato ide -1
79 // izdvajanje naziva faza na osnovu naziva metoda
80 // spajanjem sadržaja 3 XML fajla, odnosno pokretanjem XQuery
81 {
82     XQueryNavigatorCollection col = new XQueryNavigatorCollection();
83     col.AddNavigator(Server.MapPath("~/") + "XML/Faze.XML", "faze");
84     col.AddNavigator(Server.MapPath("~/") + "XML/Metode.XML", "metode");
85     col.AddNavigator(Server.MapPath("~/") + "XML/MetodeFaze.XML", "metfaze");
86     string query = "<kolekcija> { for $f in document(\"faze\")//SpisakFaza/Faza, $mf in document(\"metfaze\")//SpisakMetodaFaza/MetodaFaza [RBFaze=$
87
88     //string query = "";
89     XQueryExpression xexpr = new XQueryExpression(query);
90     string result = xexpr.Execute(col).ToXml();
91     MemoryStream mstream = new MemoryStream(System.Text.Encoding.ASCII.GetBytes(result));
92     //
93     DataSet dsFaze = new DataSet();
94     // RADI NA STARI NACIN
95     dsFaze.ReadXml(Server.MapPath("~/") + "XML/Faze.XML");
96
97     // NOVO
98     dsFaze.ReadXml(mstream);
99     return dsFaze.Tables[0].Rows[rbFaze-1].ItemArray[0].ToString();
100 }
101

```

Слика 6.2.3.1. Проширење ASP.NET web сервиса додатним функцијама

Приликом извршавања web сервиса, расположиве су методе (функције) као што је приказано на слици 6.2.3.2.



Слика 6.2.3.2. Списак расположивих метода/функција Web сервиса након покретања

### 6.2.3.2. Динамичко приказивање екранских контрола у PHP web апликацији

Реализована је могућност динамичког приказивања контрола (у овом случају екранских тастера) у корисничком интерфејсу PHP апликације која користи web сервис. У оквиру циклуса који прати број фаза, за сваку фазу се приказује тастер који је у коду представљен као *input type="submit"*.



```

C:\xampp\htdocs\adis\delov\desnopocetna.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
desnopocetna.php
30 print_r('broj faza:'. $brojfaza);
31
32 // ISPIS VISESTRUKIH VREDNOSTI
33 echo "<ul>";
34 for ($row = 0; $row < $brojfaza; $row++) {
35     echo "<p><b>Redni broj zapisa: $row</b></p>";
36
37     // ----- NTA FAZA
38     $param = array('pomNazivMetode' => 'Waterfall', 'rbFaze' => $row+1);
39     $answer = array();
40     $answer = $client->call('DajNtuFazu', array('parameters' => $param), '', '', false, true);
41     $nazivfaze =$answer['DajNtuFazuResult'];
42     echo "<li>". 'naziv faze:'. $nazivfaze. "</li>";
43     echo "<form ACTION=\"unosfazeform.php\" METHOD=\"POST\">";
44     echo "<input type=\"hidden\" name=\"nazivfaze\" value=\"$nazivfaze\">";
45     echo "<b><font face=\"Trebuchet MS\" color:#3F4534 size=\"4px\"><input TYPE=\"submit\" name=\"unosfaze\" value=\"$nazivfaze\"/></font></b>";
46     echo "</form>";
47 }
48 echo "</ul>";
49

```

Слика 6.2.3.3. Проширење PHP програмског кода ради омогућавања динамичког приказивања контрола на корисничком интерфејсу

Тестирање примене динамичког креирања елемената корисничког интерфејса приказан је на следећој слици.



Слика 6.2.3.4. Извршавање PHP апликације са динамичким креирањем елемената корисничког интерфејса на основу података из Web сервиса

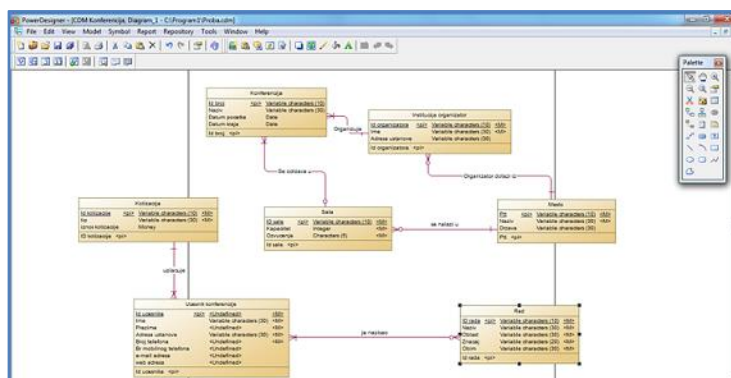
#### 6.2.4. Метричка заснованост управљања – мерење карактеристика концептуалног модела података помоћу ASP.NET web сервиса

Метричка заснованост управљања треба да се базира на објективном мерењу и примени формула којима се аутоматски може изразити процена квалитета артефакта, трајања пројекта или неки други аспект процене и подршке управљању софтверским пројектом. Мерења могу да се врше над подацима из базе података (која прикупља

податке о току извршавања процеса развоја софтвера) и над артефактима у току развоја софтвера (као што су разне врсте модела и сам програмски код).

У раду [Kazi et al, ARAB, 2016] и докторској дисертацији [Kazi, 2014] описана је примена система за парсирање концептуалног модела података (CDM) како би се упоређивањем са онтологијом извршило вредновање семантичког аспекта квалитета модела података. Изворни код наведеног решења (презентованог у [Kazi, 2014] и [Kazi et al, ARAB, 2016]), написан у Visual Basic програмском језику (укључен директно у оквиру Windows апликације). У овом раду део наведеног програмског кода трансформисан је у облик Web сервиса применом C# програмског језика у сврху креирања основе за процену трајања пројекта на основу броја ентитета концептуалног модела података који се односи на софтвер који се реализује.

Пример једног концептуалног модела података је приказан на следећој слици.



Слика 6.2.4.1. Пример концептуалног модела података

Како би се омогућило парсирање и мерење карактеристика CDM модела, креиран је посебан web сервис, којем се прослеђује String који садржи цео CDM модел (који је у суштини запис базиран на XML). Суштина парсирања је у „препознавању“ почетног и завршног „tag-a“, којим се специфично обележава одређена врста елемента. Између почетног и завршног тага конкретног елемента налази се назив одговарајућег елемента. У примеру парсирања концептуалног модела података у реализованом web сервису (начин парсирања се делимично разликује у односу на [Kazi, 2014]), издвајамо најпре део стринга целог CDM модела, који се односи на одељак са свим ентитетима на основу:

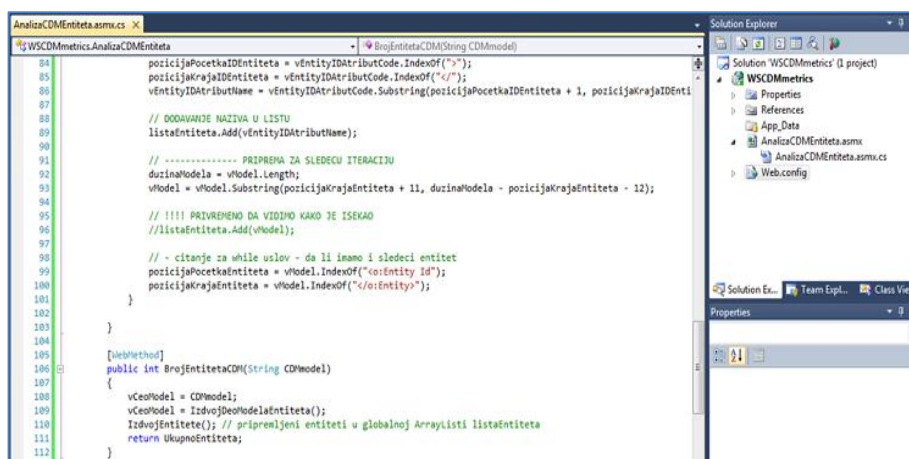
```
pozicijaPocetkaDelaEntiteti = vModel.IndexOf("<c:Entities>");
pozicijaKrajaDelaEntiteti = vModel.IndexOf("</c:Entities>");
```

Затим у оквиру тог стринга издвајамо део стринга који се односи само на назив конкретног ентитета:

```
pozicijaPocetkaEntiteta = vModel.IndexOf("<o:Entity Id");
pozicijaKrajaEntiteta = vModel.IndexOf("</o:Entity>");
```

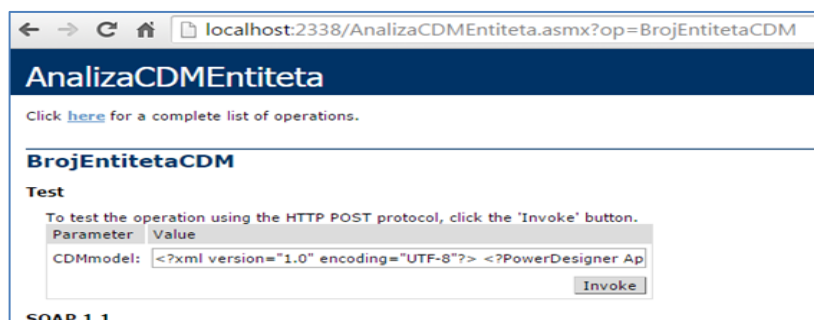
Друге врсте модела (нпр. модел пословних процеса, модел софтверских функција, односно дијаграм случајева коришћења), се могу парсирати на исти начин, с обзиром да су записи тих модела (ако су креирани коришћењем CASE алата Power Designer) слични као и код CDM модела, односно структура им је базирана на XML. Такође, на сличан начин могу се парсирати и онтологије (ако су креиране у алату нпр. Protege), као што је показано у [Kazi, 2014].

У наставку је приказан део изворног кода web сервиса у оквиру развојног окружења, који се односи на парсирање концептуалног модела података ради издвајања назива и броја ентитета.



Слика 6.2.4.2. Изворни код web сервиса за анализу концептуалног модела података

Након директног покретања web сервиса добијамо могућност уноса Stringa са XML записом концептуалног модела података, креиран из CASE алата Power Designer, као што је приказано на слици 6.2.4.3.



Слика 6.2.4.3. Покретање web сервиса за анализу концептуалног модела података уз унос XML stringa CDM модела

Након уноса Stringa са моделом и покретања опције „Invoke“ добијамо резултат у форми XML записа, са бројем ентитета у анализираном моделу:



Слика 6.2.4.4. Резултат извршавања web сервиса за анализу концептуалног модела података

## 6.2.5. Динамичко повезивање и извршавање web сервиса ради омогућавања измена у делу акције у IF-THEN правилу

Разматрање могућности за динамичко повезивање web сервиса описан је на примеру PHP апликације и ASP.NET апликације.

### 6.2.5.1. Могућности за динамичко повезивање и извршавање web сервиса у оквиру PHP web апликације

У оквиру реализације PHP web апликације (корисничког интерфејса), могуће је поставити URL адресу web сервиса, параметре и назив методе у оквиру параметризованог улазног фајла (нпр. у форми XML фајла са описом акције) и

наведени параметри се могу као стрингови учитати у PHP web апликацију и динамички извршити измена дела акције. Слика 6.2.5.1. приказује издвојене делове програмског кода PHP web апликације који би се могли динамички мењати, у зависности од параметара из екстерног XML параметарског фајла.

```
<?php

//POVEZIVANJE PHP KODA SA ASP.NET WEB SERVISOM
require_once('lib/nusoap.php');
$wsdl='http://localhost:1433/WSFaze.asmx?wsdl';
$client = new nusoap_client($wsdl, 'WSDL');
$error = $client->getError();
if ($error) {
    die("client construction error: {$error}\n");
}
// OVO RADI ZA PRVU FAZU
//$param = array('pomNazivMetode' => 'Waterfall');
//$answer = $client->call('DajPrvuFazu', array('parameters' => $param), '', '', false, true);

// ----- UKUPAN BROJ FAZA
$param = array('pomNazivMetode' => 'Waterfall');
$answer =array();
$answer = $client->call('DajBrojFaza', array('parameters' => $param), '', '', false, true);
$brojfaza = $answer['DajBrojFazaResult'];
print_r('broj faza:'. $brojfaza);
```

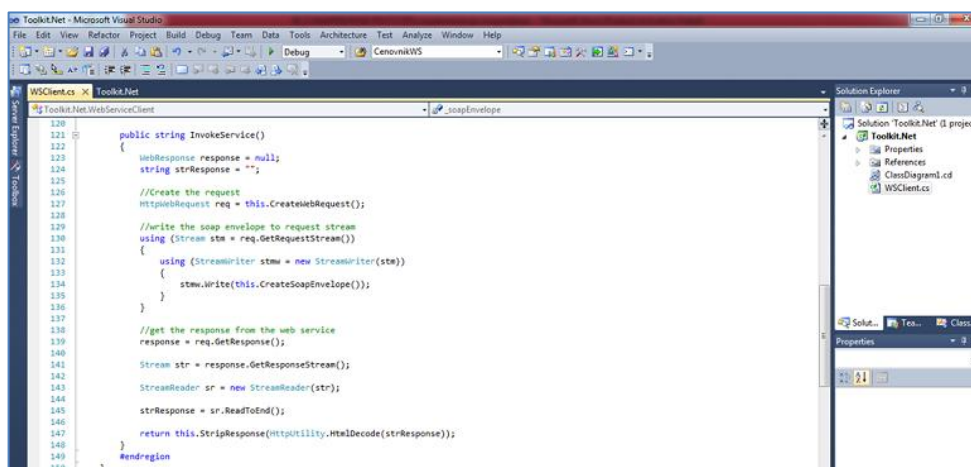
Слика 6.2.5.1. Делови кода PHP апликације који би могли да се параметризују и динамички мењају ради динамичког повезивања ASP.NET web сервиса

Учитавање екстерног XML фајла са параметрима је могуће у оквиру PHP програмског кода помоћу наредбе:

```
$xml=simplexml_load_file("pravila.xml");
```

### 6.2.5.2. Повезивање и извршавање једног web сервиса у оквиру извршавања другог web сервиса

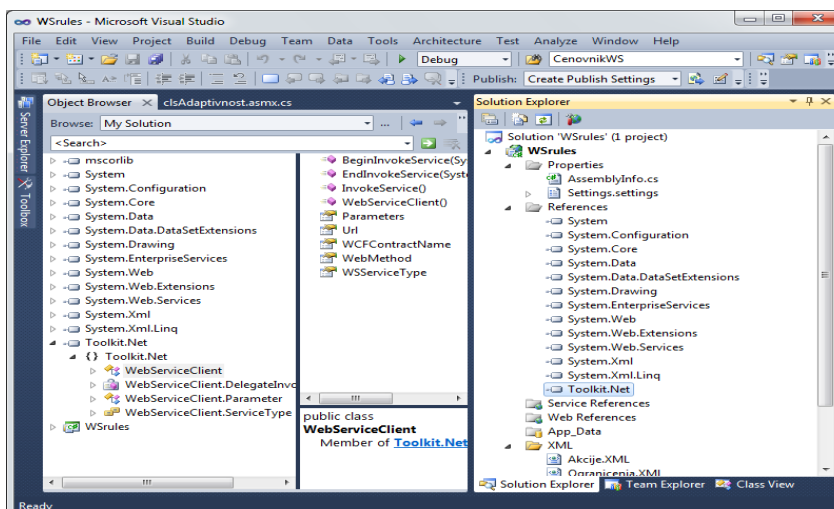
Ради омогућавања динамичког повезивања web сервиса у оквиру ASP.NET апликације, са web сајта [DinamASPNETWS] преузет је изворни код библиотеке класа "Toolkit.Net" која омогућава динамичко повезивање web сервиса у програмском коду и његово динамичко извршавање. Изворни код наведене библиотеке класа је преведен у оквиру Visual Studio .NET развојног окружења у верзију .Net Framework 3.5, како би се могао користити даље од стране других Web сервиса који су такође реализовани уз подршку .Net Framework 3.5. Након тога је креиран Toolkit.Net.dll.



```
120
121     public string InvokeService()
122     {
123         WebResponse response = null;
124         string strResponse = "";
125
126         //Create the request
127         HttpWebRequest req = this.CreateHttpRequest();
128
129         //write the soap envelope to request stream
130         using (Stream stm = req.GetRequestStream())
131         {
132             using (StreamWriter stmw = new StreamWriter(stm))
133             {
134                 stmw.Write(this.CreateSoapEnvelope());
135             }
136         }
137
138         //get the response from the web service
139         response = req.GetResponse();
140
141         Stream str = response.GetResponseStream();
142         StreamReader sr = new StreamReader(str);
143         strResponse = sr.ReadToEnd();
144
145         return this.StripResponse(HttpUtility.HtmlDecode(strResponse));
146     }
147
148     }
149 }
150
```

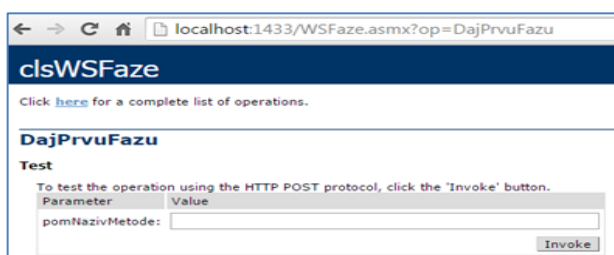
Слика 6.2.5.2. Изворни код библиотеке „Toolkit.Net“ у оквиру развојног окружења Visual Studio.NET

Креирана библиотека класа „Toolkit.Net.dll“ је повезана са пројектом другог web сервиса (Add Reference, using „Toolkit.Net“) који треба да омогући динамичко позивање web сервиса.



Слика 6.2.5.3. Укључивање библиотеке „Toolkit.Net.dll“ у оквиру референци пројекта Web сервиса

Тестирање динамичког рада web сервиса започињемо покретањем почетног web сервиса који ће бити коришћен динамички. Бирамо методу DajPrvuFazu из наведеног web сервиса, која враћа вредности које се односе на прву фазу одређене методологије развоја софтвера.



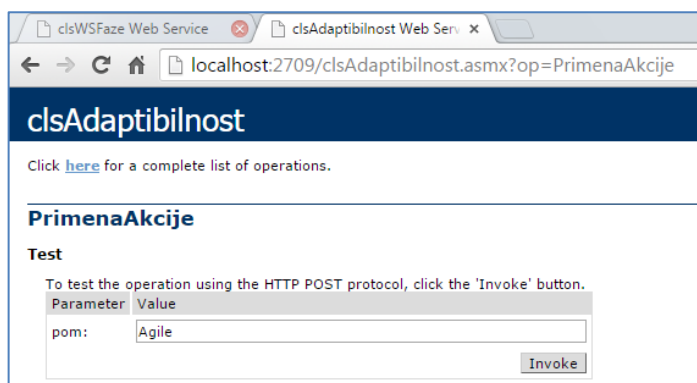
Слика 6.2.5.4. Покретање web сервиса који врши израчунавање и враћа вредност

Након тога покрећемо други web сервис који треба да омогући динамичко позивање почетног web сервиса (који је претходно требао бити покренут, активан – слика 6.2.5.4.). У оквиру тог другог web сервиса, метода је названа универзално „PrimenaAkcije“.



Слика 6.2.5.5. Покретање web сервиса који покреће други web сервис динамички

Након избора методе „PrimenaAkcije“ добијамо могућност уноса параметра позива методе, као што је приказано на слици 6.2.5.6.



Слика 6.2.5.6. Резултат покретања другог web сервиса од стране првог web сервиса и унос параметра

Коначно, након покретања опције „Invoke“, добија се резултат извршавања почетног web сервиса.



Слика 6.2.5.7. Резултат извршавања web сервиса који покреће други web сервис, у форми XML

### 6.2.5.3. Могућности примене екстерних правила у реализацији динамичких промена структуре и функционисања, применом Web сервиса и XML

У оквиру подршке адаптивности дистрибуираног web-базираног информационог система, основни приступ подршке адаптивности у структурно-функционалном смислу представља динамичка примена правила, која укључују web сервисе. Општи облик правила је:

Ако УСЛОВ онда АКЦИЈА.

У овом одељку предложен је приступ где се у делу услова могу користити позиви web сервиса који врше детекцију стања кроз мерења вредности из базе података или праћења наступања одређених догађаја. Такође у делу акције, може се иницирати покретање web сервиса који би омогућили снимање података којима би се реализовала реакција на почетни услов. На овај начин било би могуће реализовати аутоматско управљање, односно подршку управљању пројектом. Имплементација правила овог типа треба да је заснована на екстерном чувању (због могућности лакшег одржавања) у форми датотека, нпр. XML.

Имплементација XML датотеке са правилима може бити реализована кроз следеће елементе структуре XML:

- УСЛОВ: ID број Web сервиса за детекцију услова (који садржи позиве web сервиса за мерење одређених параметара и код за упоређивање са граничним вредностима параметара мерења, који се такође могу чувати у параметарским XML датотекама)
- АКЦИЈА: ID број Web сервиса за имплементацију акције.

У оквиру предложеног решења, у посебном пратећем XML фајлу били би сачувани подаци о URL адресама расположивих web сервиса, регистровани под одређеним ID бројем, који би омогућио позив одговарајућег web сервиса из XM датотеке правила.

## 7. ЗАКЉУЧНА РАЗМАТРАЊА

### 7.1. Осврт на резултате у односу на методолошки оквир истраживања

Предмет истраживања ове докторске дисертације представља *процес, методологија и технологија развоја адаптивних дистрибуираних информационих система за подршку управљању софтверским пројектима*. Докторска дисертација обрађује проблем и предмет истраживања кроз теоријску анализу, преглед постојећих истраживања, технолошких решења и искустава из праксе, моделовање, имплементацију софтверског решења и емпиријско истраживање и анализу искустава и ставова који се односе на предмет истраживања, као и искустава и ставова у вези практичне употребе реализованог прототипа.

Остварени су основни циљеви истраживања кроз креирање теоријски заснованог модела адаптивних дистрибуираних информационих система за подршку управљању софтверским пројектима, његову практичну имплементацију у форми прототипа, чија је употребљивост, позитивне и негативне карактеристике утврђена у оквиру емпиријског истраживања пробном применом од стране ИТ кадрова и применом у наставним условима у реализацији софтверских пројеката за потребе реалне праксе.

Реализовани су задаци истраживања уводним емпиријским истраживањем карактеристика претходно реализованог софтвера и софтверских пројеката (у периоду који је претходио изради докторске дисертације) са становишта управљања пројектом. Узорак су представљали реализовани софтверски пројекти и софтвери у области образовања, здравства, културе и јавне управе. Извршена је анализа резултата емпиријског истраживања, уз дефинисање проблема и недостатака, као и синтеза закључака. Извршена је анализа теоријских и истраживачких резултата и креиран теоријски модел адаптивних дистрибуираних информационих система за подршку управљању софтверским пројектима, уз пројектовање вишеслојне архитектуре система у складу са савременим објектно-оријентисаним приступом моделовању и језиком UML (Unified Modelling Language). Имплементиран је прототип софтверске подршке адаптивних дистрибуираних информационих система за подршку управљању софтверским пројектима, који садржи основне софтверске функције за подршку управљању софтверским пројектима у дистрибуираном окружењу. Извршено је емпиријско истраживање ефикасности реализованог прототипа система. Предложен је метрички модел за евалуацију артефакта, процену трајања и мониторинг процеса реализације софтвера, уз емпиријску евалуацију метричких модела, где узорак представљају софтверски пројекти, модели софтверског решења и софтвер у оквиру наставног процеса у раду студената Техничког факултета „Михајло Пупин“ у Зрењанину у оквиру наставних предмета Информациони системи 2, Дистрибуирани информациони системи, Стручна пракса, Електронско пословање и Управљање ИТ пројектима. У оквиру емпиријских истраживања пробним коришћењем од стране ИТ кадрова и анкетањем прикупљени су подаци о ставовима поводом употребљивости, предности и недостатака развијеног прототипа. Извршена је бенчмаркинг анализа најчешће коришћених алата за подршку тимском дистрибуираном раду у развоју софтвера и управљању софтверским пројектима. Извршена је анализа резултата емпиријског истраживања и синтеза закључака, који се односе на позитивне и негативне карактеристике, односно предности и недостатке имплементираног система. Извршена је синтеза свих закључака у вези предности и недостатака развијеног прототипа, као и сугестије и могућности унапређења карактеристика прототипа. Дефинисан је план унапређења прототипа и реализовано је унапређење прототипа. Коначно, у оквиру завршних разматрања представљен је сумарни приказ резултата, доприноса и могућности даљег развоја прототипа и наставка истраживања.

### **7.1.1. Резултати у односу на статус потврђености хипотезе и подхипотеза**

Основна хипотеза: „Могуће је креирати теоријски модел адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима, који би омогућио имплементацију наведеног система и на тај начин обезбедио подршку за управљање софтверским пројектима у реалној пракси развоја софтвера, као и у оквиру наставног процеса високошколске установе у области развоја софтвера и управљања софтверским пројектима.“ Основна хипотеза потврђена је креирањем функционално-технолошког теоријског модела софтверске подршке адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима. На основу теоријског модела система, који представља основ спецификације захтева, креиран је почетни прототип система који садржи подршку основним функцијама. Применљивост почетног прототипа у оквиру наставног процеса доказана је емпиријским истраживањем у оквиру редовне наставе у области информационог система на Техничком факултету „Михајло Пупин“ у Зрењанину у школској 2013/14 години, као и у оквиру професионалног усавршавања студената на Техничком факултету „Михајло Пупин“ у Зрењанину у оквиру пројекта „Професионално усавршавање студената у оквиру унапређења информационог система факултета“ у школској 2014/15 години. Применљивост реализованог прототипа у оквиру реалне праксе доказана је реализацијом два софтвера (уз административну подршку примене реализованог прототипа) у оквиру наведених активности наставе и професионалног усавршавања студената, а који су инсталирани априла 2014 („Алумни“) и септембра 2014. године („On-line скриптарница“) и од тада су у употреби на Техничком факултету „Михајло Пупин“ у Зрењанину. Применљивост реализованог прототипа у оквиру реалне праксе такође је доказана емпиријским истраживањем пробног коришћења прототипа од стране ИТ кадрова који су запослени у оквиру ИТ фирми и других организација, уз пратеће анкетирање. У одговарајућој анкети већина анкетираних испитаника је изразила став да је прототип употребљив у реалној пракси. Како би се ускладиле карактеристике реализованог прототипа са теоријским моделом, извршено је унапређење прототипа имплементацијом додатних технолошко-функционалних карактеристика.

Подхипотеза 1 – „Коришћење адаптивбилног дистрибуираног информационог система за подршку управљању софтверским пројектима смањује време реализације софтверског пројекта“. Ова подхипотеза није потврђена у општем смислу, већ само делимично. У оквиру анкетирања ИТ кадрова у вези пробног коришћења развијеног прототипа, мањи број испитаника је у оквиру одговора на питање отвореног типа одговорио да примена наведеног прототипа може убрзати и олакшати рад у оквиру управљања софтверским пројектом, док је нешто већи број испитаника одговорио да примена оваквог система у реалној пракси може представљати додатно успоравање рада на пројекту, посебно уколико се ради о мањим пројектима. Већина испитаника, који су изразили негативан став, су свој став изразили у погледу коришћења софтверских алата ове категорије уопште, а не специфично у односу на наведени конкретан прототип. Из разлога амбивалентности одговора испитаника, реализована је додатна анализа ефикасности примене прототипа. У оквиру додатне анализе ефикасности примене прототипа јавио се проблем уноса „вишеструких истовремених вредности“, који је указао на додатно успоравање коришћења у оквиру уноса података. Ипак, у оквиру анкете, највећи број испитаника сматра да је развијени прототип генерално употребљив и да примена оваквог система испитаници може бити корисна у оквиру реализације већих пројеката. У оквиру одељка анализе проблема уноса вишеструких истовремених вредности, описан је предлог могућности решења проблема коришћењем заједничког уноса више радних ставки које би биле одвојене ознакама сличним „mentions“ (које се користе код неких професионалних решења за разврставање радних задужења учесника) и каснијим аутоматским разврставањем радних ставки по врстама, односно типовима активности.



Подхипотеза 2 – „Адаптибилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим типовима софтвера и софтверских пројеката“. Ова подхипотеза је потврђена, с обзиром да су у емпиријском истраживању коришћења прототипа реализовани софтверски пројекти различитог типа софтвера и различитог типа софтверског пројекта, према претходно дефинисаним класификацијама. Такође, унапређењем прототипа реализовани су XML шифарници и web сервиси који омогућавају адаптибилност у односу на различите методологије развоја и управљања софтверским пројектима, које могу бити прилагођене различитим типовима софтвера и софтверских пројеката.

Подхипотеза 3 – „Адаптибилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се прилагоди различитим технологијама имплементације софтвера и реализације софтверских пројеката“. Ова подхипотеза је потврђена, с обзиром да су у емпиријском истраживању коришћења прототипа реализовани софтверски пројекти различитих технологија имплементације. Такође, у оквиру имплементације унапређења прототипа, повезивањем web сервиса са корисничким интерфејсом реализованим у различитим технолошким окружењима, доказано је да се реализовани прототип може прилагодити различитим технологијама имплементације.

Подхипотеза 4 – „Адаптибилни дистрибуирани информациони систем за подршку управљању софтверским пројектима може да се користи у оквиру наставног процеса у области образовања развоја софтвера и управљања софтверским пројектима, чиме се повећава ефикасност наставе у области софтверског инжењерства и развоја информационих система, као и квалитет реализованих софтверских пројеката у наставном процесу.“ Ова подхипотеза је потврђена емпиријским истраживањем примене имплементираних прототипа у оквиру наставног окружења.

## **7.2. Анализа доприноса истраживања**

### **7.2.1. Научни, практични и друштвени допринос истраживања**

#### НАУЧНИ ДОПРИНОС

1. област – методологија научно-истраживачког рада:

- Метода примене SE-PM матрице за анализу истраживања у комбинованим областима пројектног менаџмента и софтверског инжењерства, која је у овој докторској дисертацији примењена на анализу постојећих истраживања проблема управљања софтверским пројектима у дистрибуираном окружењу,
- Метода ELIM којом се елиминишу из узорка анкете које не задовољавају скуп строгих критеријума погодности узорка проширена је методом ELIG-RANK за вредновање и рангирање свих анкета у емпиријском истраживању путем анкетирања испитаника, уместо одбацавања анкета које су попунили испитаници који по строгим критеријумима нису „подобни“. Вредновање свих анкета се врши додељивањем нумеричке вредности свакој карактеристици испитаника и сумарним израчунавањем степена елигибилности (подобности) испитаника. На основу тога, одговарајућа анкета се вреднује у даљој статистичкој обради са наведеним степеном (процентом или бројем поена) утицајности и вредности наведене анкете. Коначно, предложена је и метода ELIM-F која омогућује примену строгих критеријума елиминације анкете из методе ELIM, али не одбацује анкете које имају мање од 100% поена, већ узима у обзир све анкете које имају број поена > 50%.

2. област – методологија развоја информационих система:

- Интеграција метода за креирање концептуалних модела података у оквиру развоја софтвера информационог система у оквиру методологије интегралног приступа креирања концептуалних модела података,

- Метода декомпозиције (нормализације) складишта података за креирање концептуалног модела података,
  - Метода пресликавања примитивних пословних процеса у софтверске функције коришћењем табеле пресликавања
3. област – метричка заснованост управљања реализацијом софтверских пројеката:
- Евалуација артефакта софтверског пројекта: метрички модел за евалуацију квалитета модела пословних процеса, концептуалног модела података и модела софтверских функција, заснован на синтаксном, семантичком и прагматичком аспекту анализе (примена и проширење за специфичну врсту модела у односу на Van Belle-ов приступ), као и метрички модел за евалуацију софтверске апликације у области развоја информационих система, проширен и допуњен у односу на оквир ISO 2010 стандарда.
  - Евалуација процеса и других елемената софтверског пројекта - метрички модел за мониторинг успеха реализације софтверског пројекта на три нивоа: стратегијски ниво (базиран на Balanced Scorecard методологији стратегијског менаџмента), тактички ниво пројекта (заснован на PMBOK методологији управљања пројектима) и оперативни ниво (заснован на агилном приступу).
  - Процена трајања процеса имплементације софтвера у оквиру софтверског пројекта, заснован на анализи елемената модела пословних процеса.
4. област – модел новог система и анализа постојећих система:
- Модел адаптивбилног дистрибуираног информационог система за подршку управљању реализацијом софтверских пројеката исказан кроз теоријски функционално-технолошки модел, модел пословних процеса, модел података, модел случајева коришћења, модел архитектуре (компоненти и размештаја).
  - Бенчмаркинг модел за анализу алата подршке тимском раду у дистрибуираном развоју софтвера и софтверском пројектном менаџменту, заснован на интеграцији метрика које се односе на Balanced Scorecard методологију, стандарде пројектног менаџмента PMBOK и стандарде софтверског инжењерства SWEBOOK.

#### ПРАКТИЧАН ДОПРИНОС

- Унапређење ефикасности примене софтверских алата за подршку управљању софтверским пројектима - Метода АСТ-М за формирање и анализу записа о вишеструким активностима у оквиру јединственог записа који описује све активности једног тренутка интеракције корисника и софтверског алата. Посебним бележењем специјалних симбола, користећи „mentions“ примењен над елементима текста описа активности, систем треба да аутоматски изврши разврставање података према типу активности и снимити као посебан запис податке о свакој појединачној активности која се реализовала истовремено.
- Имплементација прототипа ([www.it-project.rs](http://www.it-project.rs)) и унапређење прототипа, који омогућава коришћење у оквиру наставе и реалне праксе развоја софтвера и реализације софтверских пројеката

#### ДРУШТВЕНИ ДОПРИНОС

Унапређење наставног процеса и професионалне праксе омогућавањем коришћења:

- метричке заснованости евалуације артефакта софтверског процеса,
- метричке заснованости у мониторингу софтверског пројекта и процени трајања софтверског пројекта
- имплементираног система којим се омогућава праћење реализације софтверских пројеката у дистрибуираном окружењу.

## 7.2.2. Анализа доприноса истраживања у оквиру објављених радова

У оквиру овог одељка дат је кратак преглед садржаја радова који су објављени из области истраживања, а који су наведени у одељку 8.2. Анализа радова је извршена са становишта доприноса истраживању у оквиру ове докторске дисертације. Радови који су наведени у одељку 8.2. представљају радове који су објављени у оквиру истраживања која су претходила периоду израде докторске дисертације, а односе се на област истраживања докторске дисертације, као и радови који су објављени у току процеса реализације докторске дисертације.

Доприноси у области методологије развоја информационих система и софтверског инжењерства:

- Моделовање представља један од важних фаза у развоју информационог система. У оквиру рада [Kazi et al, ICDQM, 2011] извршена је анализа приступа евалуацији различитих типова модела у развоју информационих система. У оквиру рада [Kazi et al, АИТ, 2012a] описан је приступ мапирању елемената модела пословних процеса у елементе дизајна софтверског решења тако што се пословни процеси из модела процеса пресликавају у софтверске функције користећи табелу пресликавања, а складишта података се пресликавају, коришћењем нормализације складишта података у ентитете концептуалног модела података. Приступ интеграцији различитих метода развоја информационог система кроз стратешке приступе, различите фазе и међусобну интеграцију резултата појединачних фаза дат је у овом раду.
- Наставак истраживања, у односу на рад [Kazi et al, АИТ, 2012a], у контексту интеграције стратешких приступа, метода и артефакта у развоју информационог система реализован је и приказан у раду [Kazi et al, Metalurgia, 2012]. С обзиром да је концептуално моделовање података један од најважнијих корака развоја информационог система, у раду [Kazi et al, Metalurgia, 2012] описан је приступ интеграцији најчешће коришћених метода концептуалног моделовања података, уз допуну новим методама, где се наведене методе користе у оквиру развоја информационог система и интегришу са фазама развоја информационог система. Наведени приступ је у овом раду илустрован студијом случаја. Предложена је метода за евалуацију концептуалних модела података, заснован на Van Belle-ovom општем приступу вредновању модела. Користећи предложену методу за евалуацију концептуалних модела података, извршена је компарација појединачних метода за концептуално моделовање података. У овом раду научни допринос представља увођење нове методе синтаксне анализе складишта података и предлог методологије интеграције различитих метода концептуалног моделовања података у оквиру развоја информационог система, са становишта интеграције различитих фаза развоја информационог система. Допринос представља и предложена и емпиријски верификована метода за евалуацију концептуалних модела података.
- У раду [Kazi et al, ICIST, 2015] анализирана је улога моделовања пословних процеса у контексту развоја информационог система применом приступа дисциплиноване агилне испоруке.

Доприноси у области методологије управљања софтверским пројектима у дистрибуираном окружењу:

- У оквиру рада [Kazi et al, UrbanEco, 2011] разматране су могућности примене Green IT приступа у оквиру управљања дистрибуираним софтверским пројектима
- У оквиру рада [Kazi et al, e-Society, 2011] разматране су могућности и проблеми комуникација у оквиру дистрибуираног пројектног менаџмента у развоју софтвера.
- У раду [Kazi et al, SYMOPIS, 2011] описане су могућности и дат је модел за подршку одлучивању у оквиру управљања софтверским пројектима.
- У раду [Kazi et al, АИТ, 2014] анализирани су прегледни радови који разматрају проблеме дистрибуираног развоја софтвера и предложена је SE-PM матрица за категоризацију радова ради омогућавања систематизације покривених области и припреме основе за даља истраживања у непокривеним областима. У раду [Kazi et al, COMENG, 2014] примена SE-PM матрице је проширена у методолошком смислу увођењем прецизније категоризације појединачних аспеката проблема дистрибуираног развоја софтвера у матричном запису (ознака колоне и реда), приказивањем броја радова у којим су изучавани наведени проблеми и графичким приказом покривености области софтверског инжењерства или управљања пројектима према стандардима SWEBOOK и PMBOOK.

Доприноси у области подршке метричкој заснованости управљања софтверским пројектима вредновањем квалитета артефакта:

- У оквиру рада [Kazi et al, TTEM, 2012] предложен је систем за подршку аутоматском вредновању квалитета концептуалних модела података са семантичког аспекта, упоређивањем елемената концептуалног модела података са елементима онтологије која описује доменску семантику. Концепт система интегрише концептуални модел података (креиран у CASE алату) са онтологијом (креираном у онтолошком алату) и правилима аутоматског резоновања. Приликом интеграције, концептуални модел и онтологија се аутоматски трансформишу у облик погодан за аутоматско резоновање. У оквиру рада [Kazi et al, ARAB, 2016] предложени приступ из [Kazi et al, TTEM, 2012] је детаљно елабориран кроз примењена екстерно чувана правила за поређење елемената концептуалног модела и онтологије, односно правила за утврђивање синонима. У раду [Kazi et al, ARAB, 2016] описани су и експериментални резултати којима се илуструје могућност коришћења наведеног система.
- У оквиру рада [Kazi et al, YUINFO, 2011] описан је модел система за вредновање перформанси општинских ИТ одељења применом интеграције различитих Balanced Scorecard модела. У оквиру рада [Kazi et al, JEMC, 2011] дефинисан је метрички модел за мерење перформанси софтверског пројекта који се заснива на оквиру стратегијског менаџмента Balanced Scorecard. У раду [Kazi, Almanah, 2011] разматране су могућности примене претходно дефинисаног метричког модела у контексту подршке одлучивању. У оквиру рада [Kazi et al, ITRO, 2011] разматране су могућности примене предложеног метричког система као подршка у управљању софтверским пројектима који се реализују у оквиру студентског тимског рада. У раду [Kazi&Radulovic, MIPRO, 2011] описана је основа креирања информационог система за евиденцију података о студентским софтверским пројектима, који би био основ примене наведеног метричког система базираног на Balanced Scorecard моделу. У раду [Kazi et al, SISY, 2012] проширен је метрички модел заснован на Balanced scorecard моделу и описан у контексту индикатора перформанси у мониторингу софтверског пројекта.

Доприноси у области аутоматизоване примене пословних правила у оквиру дистрибуираних информационих система:

- У раду [Kazi et al, Tibiscus, 2008] анализирани су могућности примене BASELOG система да буде коришћен у функцији система за управљање пословним правилима.
- У раду [Kazi et al, YUPMA, 2011] предложен је модел чијом имплементацијом би се омогућила примена и дистрибуција пословних правила у оквиру управљања софтверским пројектима који се реализују у дистрибуираном окружењу.
- У раду [Kazi et al, ICTSME, 2011] приказана је анализа ризика примене управљања пословним правилима у дистрибуираним информационим системима.

Доприноси у области архитектуре дистрибуираних информационих система:

- У раду [Kazi et al, eComm, 2006] разматране су архитектуре дистрибуираних информационих система и могућности примене у оквиру информационог система плућне болнице из Зрењанина. Предложен је дизајн дистрибуираног информационог система који би омогућио размену података између централне болнице у Зрењанину и болничког одељења у месту Јаша Томић. Наведени рад представља један од резултата у оквиру пројекта „е-Рентгенологија Специјалне болнице за плућне болести Др Ваца Савић из Зрењанина“, финансиран од стране Министарства за науку Републике Србије.
- У раду [Kazi et al, MIPRO, 2007] описан је систем примене аутоматског резоновања над подацима који се налазе у удаљеним базама података, чиме се може реализовати сегмент дистрибуираног информационог система.
- У раду [Sljarić et al, AIIT, 2014] описан је софтверски систем који представља симулацију рада дистрибуираног информационог система који би могао бити примењен у Нафтној индустрији Србије.

Доприноси у области софтверске подршке настави у области информационих система у контексту управљања софтверским пројектима:

- У раду [Kazi et al, AIIT, 2012b] описан је први прототип апликације за подршку управљању пројектима у едукационом окружењу, кроз UML моделе и корисничко упутство уз опис имплементације. Наведени прототип имплементиран је користећи Java програмски језик – реализован као windows и web апликација.
- У раду [Kazi et al, ITRO, 2014] описана је web апликација која је као прототип имплементирана у оквиру реализације ове докторске дисертације, а која је реализована с циљем подршке управљању софтверским пројектима који се реализују у

оквиру наставе из области информационих система на Техничком факултету „Михајло Пупин“ у Зрењанину.

- У раду [Kazi et al, АИТ, 2014b] описане су активности на унапређењу информационог система Техничког факултета „Михајло Пупин“ у организационом, кадровском, технолошком и софтверском слоју. Софтверски аспект унапређења односи се на реализацију појединачних пројеката у сарадњи са студентима.

Доприноси унапређења наставе у области информационих система и софтверског инжењерства:

- У раду [Kazi et al, MIPRO, 2012] описана су резултати рада у практичној настави у области информационих система у току 10 година обрађени коришћењем DataWarehouse технологије, где су утврђени трендови, међу којима је најзначајнији тренд ка повећању заинтересованости студената за практичним самосталним радом у оквиру самосталних или тимских пројеката ради реализације предиспитних обавеза у форми семинарских радова, наспрам увођења практичне провере знања путем колоквијума.
- У раду [Bhatt et al, АИТ, 2012] анализирана су истраживања у области одржавања софтвера и поставке да је релевантно укључити елементе планирања и погодности одржавања софтвера у оквиру процеса развоја софтвера.
- У раду [Kazi et al, АИТ, 2012c] анализирано је тржиште рада у Србији у контексту послова из области информационих технологија. У оквиру наведене прелиминарне студије која се односи на град Нови Сад, анализирано је 6 web сајтова који служе за приказ понуде и потражње за запошљавање. У области развоја софтвера, утврђено је да су најтраженија занимања програмера за технологије: Java, PHP и C#.

### **7.3. Анализа предности, недостатака и могућности даљег унапређења имплементираних прототипа система**

У оквиру емпиријског истраживања пробног коришћења развијеног прототипа од стране ИТ кадрова и анкетања о ставовима у вези реализованог решења, издвојене су предности, недостаци и могућности даљег унапређења реализованог прототипа.

На основу ставова испитаника из анкете са питањима отвореног типа, предности реализованог прототипа се односе на омогућавање ефикасне комуникације у тиму, омогућавање мерења квалитета и контроле, омогућавање управљања пројектом и омогућавање тимског рада. Предности реализованог система огледају се у on-line доступности web апликације која омогућава реализацију софтверског пројекта у дистрибуираном окружењу, једноставност коришћења, подршке свим фазама развоја, могућности примене са различитим типовима пројеката и методологија управљања софтверским пројектима. Као кључне позитивне квалитативне карактеристике извојене су следеће: реализовани софтвер убрзава рад на пројекту, предност је стална on-line доступност, прегледност, једноставност коришћења и практичност употребе.

На основу ставова испитаника из анкете са питањима отвореног типа, недостаци реализованог система се односе на утицај коришћења система на укупно време трајања пројекта, због непотребног додатног уношења података у систем, посебно код мањих пројеката где нема потребе за оваквим евиденцијама. Остали недостаци односе се на недостатак неких функција, недовољно разрађену комуникацијску компоненту између чланова тима, а такође сугерисани су и проблеми са безбедношћу података на интернету, као и сугестије које се односе на графички дизајн и обухват података.

Могућности даљег унапређења система формулисане су на основу сугестија из питања отвореног типа (недостаци, функције које недостају, коментари и сугестије). Највише сугестија односи се на недостатак неких важних софтверских функција, визуализацију и организацију података. Остале сугестије односе се на обухват података, графички дизајн, могућности аутоматизације у мерењу и вредновању решења, подршци персонализацији, бољој подршци комуникацијама између чланова тима, поузданости система, безбедности података и додавање функција које се односе на креирање извештаја.

Додатне могућности унапређења реализованог прототипа односе се на реализацију скупа свих потребних софтверских функција у складу са функционално-технолошким теоријским моделом система, имплементацијом web сервиса који би покривали све области метричке заснованости управљања (за мерење квалитета артефакта и производа, мерење карактеристика процеса и ресурса, подршку мониторингу успеха и процени трајања пројекта), као и детаљнију имплементацију правила којима би се усмеравало аутоматско извршавање реализације web сервиса. Наравно, сврха web сервиса, као и концепција екстерних правила која одређују њихово коришћење зависи од усвојене методологије развоја софтвера, управљања софтверским пројектима, типа софтвера и софтверског пројекта, технологије имплементације и слично, тако да се рад на развоју одговарајућих web сервиса и правила никада не може сматрати завршеним, с обзиром на динамичан развој у овој области и сталну појаву нових методологија и приступа.

#### **7.4. Анализа ограничења реализованих истраживања**

У оквиру истраживања у овој докторској дисертацији, ограничења се односе на следеће:

1. Анкета која је формулисана за потребе емпиријског истраживања исказује недостатке: непрецизност и нејасност питања, недостају нека питања и слично. Издвојене су само одређене карактеристике као релевантне у питањима затвореног типа и на тај начин усмеравале начин и фокус размишљања у питањима отвореног типа која су следила након њих. На овај начин неке важне карактеристике и питања нису обрађени, односно резултати истраживања су адекватни фокусу истраживања.

2. Избор учесника анкете ограничен је могућностима контактирања и мотивације потенцијалних испитаника. Анкетом су обухваћени у највећој мери алумни Техничког факултета „Михајло Пупин“ из Зрењанина, односно дипломирани студенти наведеног факултета који су студирали у области ИТ и дужи низ година су запослени у ИТ сектору. Другу, мању групу, чине учесници који су студирали и раде у другим земљама света, а припадају категорији ИТ кадрова према образовању и радном месту. Опредељење за овај начин избора учесника анкетирања било је из разлога омогућавања успешности спровођења анкетирања.

3. Резултати бенчмаркинга алата засновани су на одабраним методологијама које су дале основ избора мерних карактеристика ради упоредне анализе. Такође, избор алата за бенчмаркинг реализован је на основу претходне анкете и условљен је избором учесника анкете.

4. Емпиријска истраживања ефикасности примене прототипа у односу утицај који примена система има на време трајања пројекта нису извршена експериментално користећи контролну и експерименталну групу, услед немогућности формирања узорка (контролна и експериментална група) довољне величине са уједначеним карактеристикама. Ипак, ефикасност у контексту времена потребног за коришћење прототипа анализирана је на основу података о коришћењу појединачних и вишеструких активности у оквиру радних записа.

5. Анализа потврђености хипотезе и подхипотеза реализована је у контексту применљивости у настави и реалној пракси користећи реализоване функционалности почетног прототипа са ужим скупом функција.

#### **7.5. Правци даљих истраживања**

Реализована истраживања би могла бити настављена првенствено кроз превазилажење ограничења истраживања представљеног у овој докторској дисертацији:

1. Анкетом треба обухватити шири скуп демографских података (нпр. назив земље у којој испитаник живи и ради, детаљније карактеристике предзнања – у форми процене нивоа познавања области, формалног образовања и професионалних обука и сертификата у области од интереса за истраживање, односно пројектног менаџмента), као и детаљније карактеристике које описују професионална искуства, како би се прецизније могли вредновати испитаници у контексту вредновања компетенција наведеног испитаника и погодности узимања у обзир одговора таквих испитаника.

Анкета треба да омогући усклађеност одговора отвореног и затвореног типа и минимални утицај одговора затвореног типа на усмеравање одговора отвореног типа или чак одвајање у посебне анкете са питањима отвореног и затвореног типа.

2. Повећање вредности резултата анкете било би омогућено спровођењем анкетирања међу непознатим и заиста случајно одабраним учесницима анкетирања из земље и иностранства.

3. Бенчмаркинг алата би требало засновати на детаљније анализираним могућим моделима за избор релевантних карактеристика анализе, као и спровођење анализе над већим бројем алата који припадају наведеним областима од интереса.

4. Спровођење експерименталног истраживања којим би се утврдио утицаја примене развијеног алата/протописа за подршку управљању софтверским пројектима на трајање пројекта требало би бити реализовано формирањем контролне и експерименталне групе које би по демографским карактеристикама (пол, године старости), знању (ниво студија, тип студија, просек студија, оцене релевантних предмета) и искуству (предзнање технологије, године искуства технологије) биле уједначене. Компарација резултата била би извршена на основу времена израде софтверских пројеката исте тежине (сложености – изражене бројем и врстом случајева коришћења, бројем табела и поља у табелама у бази података), при чему би контролна група реализовала пројекте без употребе наведене апликације/протописа, док би експериментална група реализовала пројекте уз употребу наведене апликације/протописа.

5. Анализа потврђености хипотезе и подхипотеза треба да буде извршена над подацима добијеним из коришћења унапређеног протописа, који поседује све захтеване функционалне и техничко-имплементационе карактеристике у складу са целовитим функционално-технолошким теоријским моделом система.

Даља истраживања, која се односе на области ове докторске дисертације, могла би бити усмерена ка следећим областима и темама:

1. Истраживања у оквиру четири димензије: управљање софтверским пројектима у дистрибуираном окружењу, метричку заснованост управљања софтверским пројектима, адаптивност у управљању агилним софтверским пројектима и адаптивност дистрибуираних информационих система.
2. Статистички заснована анализа утицаја појединих фактора (карактеристика испитаника) на резултате анкете, ради утврђивања да ли одговарајућа карактеристика битно утиче на резултате анкетирања. Испитивање улоге искуства и профила образовања испитаника на ставове о факторима успеха и неуспеха софтверског пројекта. Испитивање улоге искуства и профила образовања испитаника на перцепције о функционалним карактеристикама софтверског алата за подршку управљању софтверским пројектима и усклађеност са стандардним и теоријским основама.
3. Примена ELIG-RANG методе и компарација у односу на примењену ELIM-F методу евалуације анкетираних испитаника у контексту погодности примене.
4. Реализација софтвера за подршку управљању пројектима као репозиторијума у којем би се прикупљали подаци о реализованим пројектима, а који би представљали основ за процену (могуће чак и у форми експертног система или аутоматска процена) трајања и трошкова нових сличних пројеката.
5. Испитивање утицаја избора технологије имплементације софтверског пројекта на успех пројекта. Испитивање односа избора технологије имплементације према методологији реализације софтвера и управљања софтверским пројектом. Испитивање односа типа софтвера или типа софтверског пројекта према методологији реализације софтвера и управљања софтверским пројектом.
6. Истраживања у областима: аутоматско управљање пројектима, web резоновање, мониторинг перформанси дистрибуираних система, софтверска телеметрија, са посебним освртом на правила резоновања у аутоматској подршци управљању софтверским пројектима.
7. Истраживања метричких модела успеха у процесу реинжењеринга ИТ решења, односно реинжењеринга информационих система, рефакторисању или реинжењерингу софтвера.

## 8. РЕФЕРЕНЦЕ

### 8.1. Литература

#### 8.1.1. Радови објављени у часописима и зборницима конференција

1. [Abdel-Hamid& Madnick, 1987] Abdel-Hamid T.K, Madnick S.E: On the Portability of Quantitative Software Estimation Models, Elsevier, Information & Management 13 (1987) I-10
2. [Abdel-Hamid&Madnick, 1990] Abdel-Hamid, T, Madnick, S (1990). The elusive silver lining: How we fail to learn from software development failures, Sloan Management Review, 32 (1), pp. 39-47.
3. [Abrahamsson et al, 2003] Abrahamsson P, Warsta J, Siponen M.T, Ronkainen J: "New Directions on Agile Methods: A Comparative Analysis", IEEE International Conference on Software Engineering, May 3-5, 2003, Portland,Oregon, USA
4. [Abrahamsson et al, 2009] Abrahamsson P, Conboy K, Wang X: "Lots Done, More to do, The Current State of Agile Systems Development Research", European Journal of Information Systems (2009) 18, 281-284
5. [Ågerfalk et al, 2008] Ågerfalk P, Fitzgerald B, Holmstrom H, Ó, Conchúir E (2008). Benefits of Global Software Development: The Known and Unknown, in Q. Wang, D Pfahl, and D.M. Raffo (Eds.): Making Globally Distributed Software a Success Story, ICSP 2008, LNCS 5007, pp. 1-9, Springer-Verlag Berlin Heidelberg.
6. [Ajelabi&Tang, 2010] Ajelabi I, Tang Y: "The Adoption of Benchmarking Principles for Project Management Performance Improvement", International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT), Vol 1, No 2, December 2010
7. [Albrecht& Gaffney, 1983] Albrecht A. J., Gaffney J. A., "Software function, source lines of codes, and development effort prediction: a software science validation," IEEE Trans Software Eng. SE, vol. 9, pp. 639-648, 1983
8. [Alleman 4.0] Alleman G.B.: Is There an Underlying Theory of Software Project Management? (A critique of the transformational and normative views of project management), A Work in Progress, Version 4.0, 1.30.04
9. [Alleman, 2003] Alleman G.B: "Using Balanced Scorecard to Build a Project Focused IT Organization", Balanced Scorecard Conference IQPC, San Francisco, Oct 28-30, 2003.
10. [Allen et al, 1998] Allen R., Douence R., Garlan D., "Specifying and analyzing dynamic software architectures," in Proceedings of the 1998 Conference on Fundamental Approaches to Software Engineering (FASE'98), (Lisbon, Portugal), March 1998.
11. [Alshawi&Ingirige, 2003] Alshawi M, Ingirige B: "Web-enabled project management: an emerging paradigm in construction", Automation in Construction 12 (2003), 349-364
12. [Alqahtani et al, 2013] Alqahtani A.S, Moore J.D, Harrison D.K, Wood B.M: "The Challenges of applying distributed agile software development: A systematic review", International Journal of Advances in Engineering & Technology, Jan. 2013.
13. [Ambler&Lines, 2013] Ambler S. W., Lines M.: "Disciplined Agile Delivery, The Foundation for Scaling Agile", CrossTalk, November/December 2013
14. [Amrit& Hillegersberg, 2008] Amrit, C., Van Hillegersberg, J. (2008). Detecting coordination problems in collaborative software development environments. Information Systems Management, 25(1), 57-70.
15. [Andres, 2002] Andres H.P: "A comparison of face-to-face and virtual software development teams", Team Performance Management: An International Journal, Volume 8 . Number 1/2, 2002 . pp. 39±48
16. [Apers, 1988] Apers, P.: "Data allocation in distributed database systems", ACM Trans. Database Systems 31 (1988) 263-304
17. [Arhippainen&Tahti, 2003] Arhippainen L, Tahti M: "Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes", *Proceedings of the 2nd International Conference on Mobile Idots (January 2003)*
18. [Arvinder et al, 2007] Arvinder K, Bharti S, Abhilasha S,"Software Testing Product Metrics - A Survey", Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007) RIMT-IET, Mandi Gobindgarh. March 23, 2007.
19. [Asosheh et al, 2010] Asosheh A, Nalchigar S, Jamporzmay M: "Information technology project evaluation: An integrated data envelopment analysis and balanced scorecard approach", Expert Systems with Applications 37 (2010), 5931-5938



20. [Atak&Gencil, 2006] Atak O, Gencil C: "An Experimental Study on Conceptual Data Model Based Software Code Size Estimation", IWSM/MetriKon, 2006
21. [Atkinson, 1999] Atkinson R: "Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria", International Journal of Project Management Vol. 17, No. 6, pp. 337±342, 1999, Elsevier Science Ltd and IPMA.
22. [Augustin et al, 2002] Augustin I, Barbosa J.L.V, Yamin A.C, Geyer C.F.R: "ISAM, a software architecture for adaptive and distributed mobile applications", Proceedings of Seventh International Symposium on Computers and Communications, 2002
23. [Augustine et al, 2005] Augustine S, Payne B, Sencindiver F, Woodcock S: "Agile Project Management: Steering from the Edges", Communications of the ACM, December 2005, Vol. 48, No 12, pp. 85-89
24. [Baader et al, 2005] Baader F, Lutz C, Miličić M, Satter U, Wolter F: „A description logic based approach to reasoning about web services", WWW 2005, May 10-14, 2005, Chiba, Japan
25. [Baccarini, 1999] Baccarini, D. "The Logical Framework Method for Defining Project Success", Project Management Journal, 30, 4 (Dec. 1999), 25-32.
26. [Basili& Weiss, 1984] Basili V.R., Weiss D.M., "A Methodology for Collecting Valid Software Engineering Data", IEEE Transactions on Software Engineering, IEEE, 1984, pp. 728-738.
27. [Badica et al, 2011] Badica C, Budimac Z, Burkhard H-D, Ivanovic M: "Software Agents: Languages, Tools, Platforms", ComSIS Vol. 8, No. 2, Special Issue, May 2011
28. [Badica et al, 2014] Badica, C., Bassiliades, N., Ilie, S., Kravari, K.: Agent Reasoning on the Web using Web Services. Computer Science and Information Systems, Vol. 11, No. 2, 697–721. (2014)
29. [Balasubramaniam et al, 2006] Balasubramaniam R, Cao L, Mohan K, Xu P: "Can Distributed Software Development be Agile?", Communications of the ACM, October 2006/Vol. 49, No. 10
30. [Baldauf et al, 2007] Baldauf, M., Dustdar, S. and Rosenberg, F. (2007) 'A survey on context-aware systems', Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, pp.263–277.
31. [Banker et al, 1991] Banker R.D, Kauffman R, J, Kumar R: "An Empirical test of object-based output measurement metrics in a computer aided software engineering (CASE) Environment", Journal of Management Information Systems archive, Volume 8 Issue 3, Winter 1991–92, Pages 127 – 150
32. [Banker et al, 1994] Banker R. D, Kauffman, R. J, Wright C, Zweig, D: "Automating Output Size and Reuse Metrics in an Repository Based Computer Based Software Engineering (CASE) Environment" (1994). *IEEE Transactions on Software Engineering*, 20(3), 169.
33. [Barcus& Montibeller, 2008] Barcus A, Montibeller G: "Supporting the allocation of software development work in distributed teams with multi-criteria decision analysis", The International Journal of Management Science OMEGA, 36 (2008) 464 – 475
34. [Basili&Rombach, 1988] Basili, V.R., Rombach, H.D., "The TAME Project: Towards Improvement-Oriented Software Environments", IEEE Transactions on Software Engineering , 14(6), 758-773, 1988.
35. [Battistin&Rettore, 2008] Battistin E, Rettore E: "Ineligibles and eligible non-participants as a double comparison group in regression-discontinuity designs", Journal of Econometrics 142 (2008), 715-730
36. [Bendeck, 1998] Bendeck F, Goldmann S, Holz H, Kötting B: "Coordinating Management Activities in Distributed Software Development Projects", Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998.(WET ICE'98) Proceedings., Seventh IEEE International Workshops on. IEEE, 1998.
37. [Beise, 2004] Beise C.M: "IT Project Management and Virtual Teams", SIGMIS'04, April 22–24, 2004, Tucson, Arizona, USA.
38. [Bell& Kozlowski, 2002] Bell, B. S., Kozlowski, S. W. J. (2002). A typology of virtual teams: Implications for effective leadership. *Group and Organization Management*, 27(1), 14-49.
39. [Belle J-P, 2006] Van Belle J-P: A Framework for the Evaluation of Business Models and its Empirical Validation, The Electronic Journal Information Systems Evaluation 2006, Volume 9 Issue 1, pp 31-44
40. [Ben-Shaul et al, 2000] Ben-Shaul I, Gazit H, Holder O, Lavva B: "Dynamic Self Adaptation in Distributed Systems", In: P. Robertson, H. Shrobe, and R. Laddaga (Eds.): IWSAS 2000, LNCS 1936, pp. 134–142, Springer-Verlag Berlin Heidelberg 2000
41. [Berg et al, 1995] Berg J.P, Xiao C, Eberherr K.L.L: "Efficient implementation of Adaptive Software", ACM Transactions on Programming Languages and Systems, V.1 17 No 2, March 1995 PiIgs 26'4-292
42. [Berntsson-Svensson&Aurum, 2006] Berntsson-Svensson R, Aurum A: Successful Software Project and Products: An Empirical Investigation, ISESE'06, September 21–22, 2006, Rio de Janeiro, Brazil.
43. [Beynon-Davies et al, 2000] Beynon-Davies P, Owens I, Lloyd-Williams M: "Melding Information Systems Evaluation with the Information Systems Development Life-Cycle", European Conference on Information Systems ECIS, 2000

44. [Bianchi, 2001] Bianchi A.J., "Management Indicators Model to Evaluate Performance of IT Organizations", Proceedings of International Conference "Management of Engineering and Technology", vol. 2, IEEE Press, 2001, pp. 217-229.
45. [Bird et al, 2009] Bird C, Nagappan N, Devanbu P, Gall H, Murphy B: "Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista", Communications of the ACM, August 2009, vol. 52, no. 8
46. [Boden et al, 2007] Boden A, Nett B, Wulf V (2007). Coordination Practices in Distributed Software Development of Small Enterprises. IEEE 2nd International Conference on Global Software Engineering (ICGSE'07), 27-30 August, 2007, in Munich, Germany, IEEE Press, pp. 235-244
47. [Boehm, 1979] Boehm B.W: "Software Engineering – as it is", ICSE '79 Proceedings of the 4th International conference on Software engineering, Pages 11-21, IEEE Press Piscataway, NJ, USA, 1979
48. [Boehm, 1986] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes 11(4), 14–24 (1986).
49. [Boehm&Ross, 1989] Boehm B, Ross R: Theory-W Software Project Management: Principles and Examples, IEEE Transactions on Software Engineering, Vol. 15, No. 7, July 1989
50. [Boehm, 1991] Boehm, B.W., 1991. Software risk management principles and practices. IEEE Software 8 (1), 32–41
51. [Boehm et al, 1995] Boehm B, Clark B, Horowitz E, Westland C, Madachy R, Selby R: Cost models for future software life cycle process: COCOMO II, Annals of Software Engineering, Vol 1, 1995, pp. 57-94.
52. [Boehm et al, 2000] Boehm B, Chris Abts C, Chulani S: Software development cost estimation approaches – A survey, Annals of Software Engineering 10 (2000) 177–205
53. [Boehm, 2006] Boehm B.W : "A View of 20th and 21st Century Software Engineering", ICSE'06, May 20–28, 2006, Shanghai, China.
54. [Boehm, 2002] Boehm B.: "Get Ready for Agile Methods, with Care", IEEE Computer, January 2002, pp 64-69.
55. [Boden et al, 2008] Boden A, Nett B, Wulf V: "Articulation Work in Small-Scale Offshore Software Development Projects", Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), Leipzig, Germany: ACM , 2008, pp. 21-24.
56. [Boden et al, 2009] Boden A, Avram G, Bannon L, Wulf V: „Knowledge Management in Distributed Software Development Teams –Does Culture Matter?“, 4<sup>th</sup> IEEE International conference on global software engineering (ICGSE'09), 13-16 July, 2009, Limerick, Ireland, IEEE press, pp 18-27.
57. [Boden et al, 2010] Boden A, Nett B, Wulf V. Operational and Strategic Learning in Global Software Development - Implications from two Offshoring Case Studies in Small Enterprises. IEEE Software. 2010, Vol 27, Issue 6, pp. 58-65
58. [Bose, 2008] Bose I: "Lessons Learned from Distributed Agile Software Projects: A Case-Based Analysis", Communications of the Association for Information Systems, Vol 23, No 1, 2008.
59. [Bourgault et al, 2002] Bourgault M, Lefebvre É, Lefebvre L.A, Pellerin R, Elia E: "Discussion of Metrics for Distributed Project Management: Preliminary Findings", Proceedings of the 35th Hawaii International Conference on System Sciences, 2002
60. [Bourque, 2002] Bourque P, Dupuis R, A. Abran A, Moore J. W., Tripp L., Wolffe S., Fundamental principles of software engineering - A journey, Journal of Systems and Software 62(1), 59–70 (2002).
61. [Bowen& Maurer, 2002] Bowen, S., Maurer, F. (2002, August). „Process support and knowledge management for virtual teams doing agile software development“ Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International
62. [Braubach& Pokahr, 2011] Braubach L, Pokahr A: Addressing Challenges of Distributed Systems using Active Components, Proceedings of the 5th International Symposium on Intelligent Distributed Computing – IDC 2011, Delft, The Netherlands – October 2011, also in: Chapter, Intelligent Distributed Computing V, Volume 382 of the series Studies in Computational Intelligence, 2012, pp 141-151
63. [Brock et al, 2003] Brock S, Hendricks D, Linnel S, Smith D: "A Balanced Approach to IT Project Management", Proceedings of SAICSIT 2003, pp. 2-10
64. [Brown, 1996] Brown, P.J. (1996) 'The stick-e document: a framework for creating context-aware applications', Proceedings of the Electronic Publishing, Palo Alto, pp.259–272.
65. [Brynjolfsson& Hitt, 2003] Brynjolfsson E, Hitt L.M: Computing Productivity: Firm-Level Evidence, Review of Economics and Statistics, November, 2003.
66. [Cao&Balasubramaniam, 2008] Cao L, Balasubramaniam R: "Agile Requirements Engineering Practices: An Empirical Study", IEEE Software, 2008.

67. [Cappiello et al, 2006] Cappiello C, Comuzzi M, Mussi E, Pernici B: "Context Management for Adaptive Information Systems", *Electronic Notes in Theoretical Computer Science* 146 (2006) 69–84
68. [Carberry, 2001] Carberry, S.: 2001, "Techniques for Plan Recognition", *Journal User Modeling and User-Adapted Interaction*, Vol.11 (1-2), pp. 31-48
69. [Carmel&Agarwal, 2001] Carmel E, Agarwal R (2001). Tactical approaches in Alleviating Distance in Global Software Development. *IEEE Software* March/April 2001, pp. 22-29.
70. [Casey&Richardson, 2006] Casey V, Richardson I, "Project management within virtual software teams." *Global Software Engineering, 2006. ICGSE'06. International Conference*
71. on. *IEEE*, 2006.
72. [Casey, 2010] Casey V: "Developing Trust In Virtual Software Development Teams", *Journal of Theoretical and Applied Electronic Commerce Research*, ISSN 0718-1876 *Electronic Version*, VOL 5 / ISSUE 2 / AUGUST 2010 / 41-58
73. [Cataldo& Herbsleb, 2008] Cataldo M, Herbsleb J.D: "Communication Networks in Geographically Distributed Software Development", *CSCW'08*, November 8–12, 2008, San Diego, California, USA
74. [Cavano&McCall, 1978] Cavano J.P, McCall J.A: "A framework for the Measurement of Software Quality", *Proceedings of the software quality assurance workshop on Functional and Performance Issues*, pp 133-139, 1978
75. [Chan&Chung, 2002] Chan K.C.C, Chung L.M.I: "Integrating Process and Project Management for Multi-Site Software Development", *Annals of Software Engineering*, 14, 115-143, 2002
76. [Charfi&Mezini, 2004] Charfi A, Mezini M: "Aspect-Oriented Web Service Composition with AO4BPEL", In L.-J. Zhang and M. Jeckle (Eds.): *ECOWS 2004*, LNCS 3250, pp. 168–182, 2004., Springer-Verlag Berlin Heidelberg 2004
77. [Cheng, 2004] Cheng L-T, Cleidson R., de Souza B., Hupfer S, Patterson J, Ross S, "Building Collaboration into IDEs", *Distributed Development*, Vol. 1, No. 9 - December/January 2003-2004
78. [Chu, 1969] Chu, W.: "Optimal file allocation in a multiple computer system", *ACM Trans. on Database Systems* C-18 (1969) 885-889
79. [Chow&Dac-Buu, 2008] Chow T, Dac-Buu C: "A survey study of critical success factors in agile software projects", *The Journal of Systems and Software* 81 (2008)
80. [Cicmil et al, 2006] Cicmil S, Williams T, Thomas J, Hodgson D: Rethinking Project Management: Researching the actuality of projects, *International Journal of Project Management* 24 (2006) 675–686
81. [Clemmons, 2006] Clemmons R.K: "Project Estimation with Use Case Points", *CrossTalk, The Journal of Defense Software Engineering*, February 2006
82. [Cohen et al, 2004] Cohen D., Lindvall M., Costa P., "An introduction to agile methods", *Advances in Computers* 62, 1–66 (2004).
83. [Coleman, 2009] Coleman, G: "Code is speech: Legal Tinkering, Expertize and Protest among Free and Open Source Software Developers", *Cultural Anthropology*, Vol. 24, Issue 3, pp. 420–454. ISSN 0886-7356, online ISSN 1548-1360, 2009 by the American Anthropological Association
84. [Colman&Han, 2005] Colman A, Han J: "Coordination Systems in Role-Based Adaptive Software", In: Jacquet J.-M., Picco G.P. (Eds.): *COORDINATION 2005*, LNCS 3454, pp. 63 – 78, Springer-Verlag Berlin Heidelberg 2005
85. [Conboy&Fitzgerald, 2004] Conboy K, Fitzgerald B: "Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines", *Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, pp. 37-44
86. [Coram&Bohner, 2005] Coram M, Bohner S, "The Impact of Agile Methods on Software Project Management", *Proceedings of the 12<sup>th</sup> IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS) 2005*
87. [Crowston et al, 2003] Crowston, K., Annabi, H., & Howison, J. Defining Open Source Software project success. In *Proceedings of the International Conference on Information Systems (ICIS 2003)*, Seattle, WA, USA, December.
88. [Crowston&Howison, 2003] Crowston, K, Howison, J, "The Social Structure of Open Source Software Development Teams" (2003). *iSchool Faculty Scholarship*. Paper 123.
89. [Curbera et al, 2002] Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, Weerawarana S: "Unraveling the Web Services Web – An Introduction to SOAP, WSD: and UDDI", *Journal IEEE Internet Computing*, 2002
90. [Dafoulas& Macaulay, 2001] Dafoulas G, Macaulay L, „Investigating Cultural Differences in Virtual Software Teams“, 2001, *The Electronic Journal on Information Systems in Developing Countries*, 7,4, 1-14
91. [Dalkey& Helmer, 1963] Dalkey N., Helmer O., "An experimental application of the Delphi method to the use of experts," *Management Science*, vol. 9, pp. 458-467, 1963.

92. [Daskalantonakis, 1992] Daskalantonakis M.K: A Practical View of Software Measurement and Implementation Experiences Within Motorola, IEEE Transactions on Software Engineering, Vol 18, No 11, 1992.
93. [David&Ledoux, 2006] David P-C, Ledoux T: "An Aspect-Oriented Approach for Developing Self-Adaptive Fractal Components", W. Löwe and M. Südholt (Eds.): SC 2006, LNCS 4089, pp. 82-97, Springer-Verlag Berlin Heidelberg 2006
94. [Davies et al, 1996] Davies N, Friday A, Blair G.S, Cheverst K: "Distributed systems support for adaptive mobile applications." J. Mobile Networks and Applications 1.4 (1996): 399-408.
95. [Davies et al, 1997] Davies N, Wade S.P, Friday A, Blair G.S: „Limbo: A tupl space based platform for adaptive mobile applications", In: Proceedings of The International Conference on Open Distributed processing /Distributed Platforms 1997, pp. 291-302
96. [Davison, 2007] Davison R, "Offshoring information technology: sourcing and outsourcing to a global workforce," Inf. Technol. Dev., vol. 13, no. 1, pp. 101-102, 2007.
97. [Decker&Sycara, 1997] Decker K, Sycara K: "Intelligent Adaptive Information Agents", Journal of Intelligent Information Systems, 9, pp. 239-260, Kluwer Academic Publishers, 1997
98. [DeLone& McLean, 1992] DeLone W. H., McLean E. R., "Information Systems Success: The Quest for the Dependent Variable," Information Systems Research (3:1), 1992, p. 87. Copyright, The Institute of Management Sciences, now the Institute for Operations Research and the Management Sciences (INFORMS), 901 Elkridge Landing Road, Suite 400, Linthicum, MD 21090 USA.
99. [Delgado et al, 2004] Delgado, N. Gates, A.Q. ; Roach, S. A: "Raxonomy and catalog of runtime software-fault monitoring tools", Software Engineering, IEEE Transactions on , Volume:30, Issue: 12, pp 859 – 872, 2004
100. [Delone& McLean, 2003] Delone W.H, McLean E.R (2003), 'The DeLone and McLean Model of Information Systems Success: A Ten-Year Update', Journal of Management Information Systems, vol. 19, no. 4, pp. 9-30.
101. [Demir et al, 2009] Demir A.K, Michael J.B, Osmundson, J.S: "Approaches for Measuring Software Project Management Effectiveness.", Journal Software Engineering Research and Practice. 2009.
102. [Dey& Abowd, 2000] Dey, A.K. Abowd, G.D. (2000) 'Towards a better understanding of context and context-awareness', Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, ACM Press, New York.
103. [Dhar&Balakrishnan, 2006] Dhar S, Balakrishnan B (2006). Risks, Benefits, and Challenges in Global IT Outsourcing: Perspectives and Practices. Journal of Global Information Management, vol. 14, issue 3, Idea Publishing Group
104. [Dingsøyr et al, 2012] Dingsøyr T, Nerur S, Balijepally V, Moe N.B, A decade of agile methodologies: Towards explaining agile software development, The Journal of Systems and Software, 85 (2012), 1213-1221
105. [Dobrović et al, 2008] Dobrović Ž, Tomičić M, Vrček N: Towards an effective e-government: implementation of a balanced scorecard in the public sector, Intellectual Economics Journal, 2008, No. 1(3), p. 7-17
106. [Dove, 1999] Dove R, "Knowledge management, response ability, and the agile enterprise," Journal of Knowledge Management, vol. 3, no. 1, pp. 18- 35, 1999.
107. [Dowdy& Foster, 1982] Dowdy, L., Foster, D.: "Comparative models of the file assignment problem", ACM Compt. Surveys 14 (1982) 287-313
108. [Dustdar, 2004] Dustdar S: "Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams", Distributed and Parallel Databases, 15, 45-66, 2004, Kluwer Academic Publishers
109. [Dustdar&Schreiner, 2005] Dustdar S, Schreiner W: „A survey on web services composition", Int. J. Web and Grid Services, Vol. 1, No. 1, 2005
110. [Dutta et al, 1998] Dutta S, Kulandiswamy S, Van Wassenhove I.N: "Benchmarking European Software Management Best Practices", Communications of ACM, 1998, 41 (6): 77-86.
111. [Dutoit&Bruegge, 1998] Dutoit A.H, Bruegge B: "Communication Metrics for Software Development", IEEE Transactions on Software Engineering, VOL. 24, NO. 8, AUGUST 1998
112. [Dutoit et al, 2001] Dutoit A.H, Johnstone J, Bruegge B: "Knowledge scouts: Reducing communication barriers in a distributed software development project", 8th Asia-Pacific Software Engineering Conference (APSEC 2001), Macau, December 4-7, 2001.
113. [Dvir et al, 1998] Dvir D., Lipovetsky S., Shenhar A. J. , Tishler A.: "In search of project classification: a non-universal approach to project success factors", Research Policy 27, 915-935 (1998).
114. [Dvir et al, 2003] Dvir D, Raz T, Shenhar A.J: An empirical analysis of the relationship between project planning and project success, International Journal of Project Management 21 (2003) 89-95
115. [Dyba& Dingsøyr, 2008] Dyba T, Dingsøyr T: "Empirical studies of agile software development: A systematic review", Information and Software Technology, 2008.

116. [Dzitac&Barbat, 2009] Dzitac I, B̂arbat B.E: „Artificial Intelligence + Distributed Systems = Agents“, *Int. J. of Computers, Communications & Control*, ISSN 1841-9836, E-ISSN 1841-9844, Vol. IV (2009), No. 1, pp. 17-26
117. [Ebert&De Neve, 2001] Ebert C, De Neve P (2001). *Surviving Global Software Development*. IEEE Software March/April 2001
118. [Ebrahim et al, 2009] Ebrahim N.A, Ahmed S, Taha Z: “Virtual Teams: a Literature Review”, *Australian Journal of Basic and Applied Sciences* , 3(3): 2653-2669, 2009, ISSN 1991-8178, 2009, INSInet Publication
119. [Ebrahim et al, 2010] Ebrahim N.A, Ahmed S, Taha Z: “SMEs; Virtual research and development (R&D) teams and new product development: A literature review”, *International Journal of the Physical Sciences* Vol. 5(7), pp. 916-930, July 2010.
120. [Edwards& Pengelly, 1997] Edwards D.J, Pengelly A.D: “Using System Dynamics (SD) simulation to explore and derive self-adaptive techniques for information systems”, *The 15th International Conference of the System Dynamics Society 1997*, Istanbul, Turkey
121. [Edwards& Sridhar, 2003] Edwards H.K, Sridhar V: „Analysis of the Effectiveness of Global Virtual Teams in Software Engineering Projects“, *Proceedings of the 36th Hawaii International Conference on System Sciences*, 2003
122. [Efstratiou et al, 2000] Efstratiou C, Cheverst K, Davies N, Friday A: “Architectural requirements for the effective support of adaptive mobile applications”, *Conference: Middleware 2000*, New York, USA
123. [El-Ghalayini et al., 2005] El-Ghalayini H., Odeh M., McClatchey R., Solomonides T. *Reverse Engineering Ontology to Conceptual Data Models*, *IASTED International Conference on Databases and Applications, DBA 2005*.
124. [Emanuel et al, 2011] Emanuel A. W. R., Wardoyo R, Istiyanto J.E, Mustofa K (2011), “Statistical Analysis of Software Metrics Affecting Modularity in Open Source Software”, *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol 3, No 3, June 2011
125. [Espinosa et al, 2001] Espinosa J.A., Kraut R.E, Lerch J.F, Slaughter S.A, James D. Herbsleb, Mockus A: “Shared Mental Models and Coordination in Large-Scale, Distributed Software Development”, 2001, *Twenty-Second International Conference on Information Systems*
126. [Espinosa et al, 2002] Espinosa J.A, Kraut R. E, Slaughter S.A, Lerch J.F, Herbsleb J.D, Mockus A: “Shared Mental Models, Familiarity and Coordination: A Multi-Method Study of Distributed Software Teams”, *International Conference for Information Systems, ICIS 2002*, December 2002, Barcelona, Spain
127. [Espinosa&Carmel, 2004] Espinosa J.A, Carmel E: “The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation”, *Journal Software Process Improvement and Practice*, 2004; 8
128. [Espinosa et al, 2007a] Espinosa A, Slaughter S.A, Kraut R.E, Herbsleb J.D.: “Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development”, *Organization Science*, Vol. 18, No. 4, July–August 2007, pp. 613–630
129. [Espinosa et al, 2007b] Espinosa J.A, Slaughter S.A, Kraut R.E, Herbsleb J.D: “Team Knowledge and Coordination in Geographically Distributed Software Development”, *Journal of Management Information Systems / Summer 2007*, Vol. 24, No. 1, pp. 135–169., 2007 M.E. Sharpe, Inc.
130. [Estler et al, 2012] Estler, H. C., Nordio, M., Furia, C. A., Meyer, B., & Schneider, J. (2012, August). Agile vs. structured distributed software development: A case study. In *Global Software Engineering (ICGSE)*, 2012 IEEE Seventh International Conference on (pp. 11-20). IEEE.
131. [Eveleens& Verhoef, 2010] Eveleens J.L., Verhoef C, “The rise and fall of the Chaos Report Figures”, *IEEE Software*, IEEE Computer Society, ISBN 0740-7459, 2010.
132. [Erdogmus, 2008 ] Erdogmus H., “Essentials of Software Process”, *IEEE Software* 25(4), 4–7, 2008
133. [Faraj& Sproull, 2000] Faraj S, Sproull L: „Coordinating Expertise in Software Development Teams“, *Management Science*, Vol. 46, No. 12 (Dec., 2000), pp. 1554-1568, INFORMS
134. [Farooq et al, 2011] Farooq S.U, Quadri S. M. K., Ahmad N: “Software Measurements and Metrics: Role in Effective Software Testing”, *International Journal of Engineering Science and Technology (IJEST)*, ISSN : 0975-5462 Vol. 3 No. 1 Jan 2011 671 - 680
135. [Ferreira&Otlely, 2009] Ferreira A, Otlely D, “The design and use of performance management systems: An extended framework for analysis,” *Management Accounting Research*, 2009, 20, pp. 263–282.
136. [Fenton& Neil, 1999] Fenton N.F, Neil M: *A Critique of Software Defect Prediction Models*, *IEEE Transactions on Software Engineering*, Vol. 25, NO. 3, May/June 1999

137. [Fendandes&Almeida, 2010] Fernandes J.M, Almeida M: "Classification and Comparison of Agile Methods", 2010 Seventh International Conference on the Quality of Information and Communications Technology
138. [Fernandez&Fernandez, 2008] Fernandez D.J, Fernandez J.D: "Agile Project Management – Agilism versus Traditional Approaches", The Journal of Computer Information Systems: Winter 2008/2009, 49, 2, ABI/INFORM Global, pg. 10.
139. [Fleming&Koppelman, 1998] Fleming Q.W, Koppelman J.M: "Earned Value Project Management A Powerful Tool for Software Projects", CROSSTALK The Journal of Defense Software Engineering, July 1998, pp. 19-23
140. [Fortino&Russo, 2012] Fortino G, Russo W: "ELDAMeth: A Methodology for Simulation-based Prototyping of Distributed Agent Systems", Information and Software Technology 54.6 (2012): 608-624.
141. [Fox et al, 2008] Fox D, Sillito J, Maurer F: "User-Centered Design: How These Two Methodologies Are Being Successfully Integrated In Industry", Conference Agile 2008.
142. [Fox& Spence, 1998] Fox T. L., Spence, J. W.: „Tools of the trade: A survey of project management tools.“, Project Management Journal, Sep98, Vol. 29 Issue 3, p20
143. [Frakes&Kang, 2005] Frakes W.B, Kang K: "Software Reuse Research: Status and Future", IEEE Transactions on Software Engineering, Vol. 31, No. 7, July 2005
144. [Froehlich& Dourish, 2004] Froehlich J, Dourish P: "Unifying artifacts and activities in a Visual Tool for Distributed Software Development Teams", Proceedings of the 26th International conference on Software Engineering (ICSE'04), 2004, IEEE
145. [Frolick&Ariyachandra, 2006] Frolick M.N, Ariyachandra T.R: "Business Performance Management: One Truth", Journal Information Systems Management, Winter 2006
146. [Fujita et al, 1998] Fujita S, Hara H, Sugaware K: "Agent-Based Design Model of Adaptive Distributed Systems", Applied Intelligence 9, 57-70 (1998)
147. [Gallivan, 2001] Gallivan M. J: „Striking a balance between trust and control in a virtual organization: a content analysis of open source software case studies“, Info Systems J (2001) 11, 277-304
148. [Ganguly et al, 2009] Ganguly A., Nilchiani R., Farr J. V., "Evaluating agility in corporate enterprises," International Journal of Production Economics, vol. 118, no. 2, pp. 410-423, Apr. 2009.
149. [Gao et al, 2004] Gao J, Chandra S. Ravi C.S, Espinoza R: "Measuring Component Performance Using A Systematic Approach and Environment", SOQUA/TECOS 2004: 77-92
150. [Ghada&Fitzgerald, 2013] Ghada A, Fitzgerald, G: "Re-Conceptualizing Agile Information Systems Development Using Complex Adaptive Systems Theory", Emergence: Complexity & Organization . 2013, Vol. 15 Issue 3, p1-23.
151. [Garcia et al, 2006] Garcia F, Bertoa M.F, Calero C, Vallecillo A,Ruiz F, Piattini M, Genero M: Towards a consistent terminology for software measurement, Elsevier, Information and Software Technology 48 (2006), 631-644
152. [Gasser&Ripoche, 2003] Gasser L, Ripoche G (2003): "Distributed Collective Practices and Free/Open-Source Software Problem Management: Perspectives and Methods", CITE'2003.
153. [German&Mockus, 2003] German D, Mockus A: "Automating the Measurement of Open Source Projects", Proceedings of the 3rd workshop on open source software engineering. 2003.
154. [Gerogiannis et al, 2009] Georgiannis V.C, Fitsilis P, Kakarontzas G, Tzikas A: "Involving Stakeholders in the Selection of Project and Portfolio Management Tool", (21ο Εθνικό Συνέδριο Ελληνικής Εταιρίας Επιχειρησιακών Ερευνών (ΕΕΕΕ), Αθήνα, 28-30 Μαΐου 2009) 21. National conference of Greek Operational Research (EUSR), Atina, 28-30 maj 2009
155. [Gilbert& Karahalios, 2007] Gilbert E, Karahalios K: "CodeSaw: A Social Visualization of Distributed Software Development", Proceedings from conference INTERACT, 2007.
156. [Gorton&Motwani, 1996] Gorton I, Motwani S (1996). Issues in co-operative software engineering using globally distributed teams. Information and Software Technology vol 38, pp. 647-655. Elsevier.
157. [Graves et al, 2000] Graves T.L, Karr A.F, Marron J.S., Siy H: Predicting Fault Incidence Using Software Change History, IEEE Transactions on Software Engineering, Vol. 26, No. 7, July 2000
158. [Grechanik et al, 2010] Grechanik M, Jones J.A, Orso A, van der Hoek A: „Bridging Gaps between Developers and Testers in Globally distributed Software Development“, FoSER 2010, November 7-8, 2010, Santa Fe, New Mexico, USA
159. [Greenwood&Calisti, 2004] Greenwood D, Calisti M: "Engineering Web Service – Agent Integration", In: IEEE Systems, Cybernetics and Man Conference, SMC 2004, Hague, Netherlands, vol 2, pp 1918-1925, IEEE Computer Society
160. [Grembergen, 2000] Grembergen V. W: "The Balanced Scorecard and IT Governance", IRMA Conference, 2000

161. [Grembergen&Amelinckx, 2002] Grembergen Van W, Amelinckx I: "Measuring and Managing e-Business Projects through the Balanced Scorecard", Proceedings of the 35<sup>th</sup> Hawaii International Conference on System Sciences, 2002
162. [Gruber, 1995] Gruber T.R., Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *Human Computer Studies*, 43 (5-6), 1995, 907-928.
163. [Gumm, 2006] Gumm, D.C. (2006): "Distribution Dimensions in Software Development Projects: A Taxonomy", *IEEE Software* (september/october 2006), pp. 45-51
164. [Gutwin et al, 2004] Gutwin C, Penner R, Schneider K: "Group Awareness in Distributed Software Development", *CSCW'04*, November 6-10, 2004, Chicago, Illinois, USA.
165. [Guzman et al, 2010] Guzman J. G, Ramos J.S, Seco A.A., Esteban A.S: "How to get mature global virtual teams: a framework to improve team process management in distributed software teams", *Software Qual J* (2010) 18:409-435
166. [Hall et al, 2005] Hall W.A, Long B, Bermbach N, Jordan S, Patterson K, "Qualitative Teamwork Issues and Strategies: Coordination Through Mutual Adjustment", *Qual Health Res* 2005 15: 394
167. [Hande et al, 2012] Hande A, Suralkar S, Chawan P.M.: Distributed Software Project Development, *International Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.998-1003, 998
168. [Hartman&Ashrafi, 2002] Francis Hartman; Rafi A Ashrafi: Project management in the information systems and information technologies Industries, *Project Management Journal*; Sep 2002; 33, 3; ABI/INFORM Global
169. [Heeks et al, 2001] Heeks R, Krishna S, Nicholson B, Sahay S (2001), Synching or Sinking: Global Software Outsourcing Relationships, *IEEE Software* March/April 2001
170. [Hendriks et al, 2000] Hendriks R, van Vonderen R, van Veenendaal E: "Measuring software product quality during testing", *Conference proceedings European Software Quality Week*, October 2000
171. [Herbsleb et al, 1995] Herbsleb J.D, Klein H, Olson G.M, Brunner H, Olson J. S, Harding J: "Object-Oriented Analysis and Design in Software Project Teams", *Human-Computer Interaction*, 1995, Volume 10, pp. 249-292., Lawrence Erlbaum Associates, Inc.
172. [Herbsleb et al, 1997] Herbsleb J, Zubrow D, Goldenson D, Hayes W, Paulk M, "Software Quality and Capability Maturity Model", *Communications of the ACM*, June 1997/Vol. 40, No. 6
173. [Herbsleb et al, 2001] Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2001, July). "An empirical study of global software development: distance and speed", In *Proceedings of the 23rd international conference on software engineering* (pp. 81-90). IEEE Computer Society.
174. [Herbsleb&Moitra, 2001] Herbsleb J. D, Moitra D. (2001). "Global Software Development", *IEEE Software* (March/April 2001), pp. 16-20.
175. [Herbsleb&Mockus, 2003] Herbsleb J, Mockus A (2003), An Empirical Study of Speed and Communication in Globally Distributed Software Development, *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, VOL. 29, NO. 6, JUNE 2003
176. [Herbsleb et al, 2005] Herbsleb J, Paulish D, Bass M (2005). *Global Software Development at Siemens: Experience from Nine Projects*, ICSE'05, May 15-21, 2005, St. Louis, Missouri, USA.
177. [Herbsleb, 2007] Herbsleb J.D: "Global Software Engineering: The Future of Socio-technical Coordination", *Proceedings of Conference "Future of Software Engineering"(FOSE'07)*
178. [Hildenbrand et al, 2008] Hildenbrand T, Rothlauf F, Geisser M, Heinzl A, Kude T (2008), Approaches to Collaborative Software Development, *IEEE International Conference on Complex, Intelligent and Software Intensive Systems*
179. [Hippel&Krogh, 2003] von Hippel E, von Krogh G: Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science, *Organization Science* (2003) 14 (2)208-223.
180. [Hofer et al, 2003] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G. and Altmann, J. (2003) 'Context-awareness on mobile devices - the hydrogen approach', *Proceedings of the 36<sup>th</sup> Annual Hawaii International Conference on System Sciences*, pp.292-302.
181. [Holmstrom et al, 2006] Holmstrom H, Fitzgerald B, Agerfalk M, Conchuir E:"Agile practices reduce distance in global software development", *Information Systems Management*, Vol 23, No. 3, pp. 7-18.
182. [Houben, 2005] Houben, G.J.: "Challenges in Adaptive Web Information Systems: Do Not Forget the Link", *Engineering Advanced Web Applications*, *Proceedings of Workshops in Connection with 4th International Conference on Web Engineering (ICWE2004)*, pp. 3-11, Publ. Rinton Press (2005)
183. [Huang et al, 2001] Huang L, Thiel U, Hemmje M, Neuhold E.J: Distributed Information Search with Adaptive Meta-Search Engines, K.R. Dittrich, A. Geppert, M.C. Norrie (Eds.): *CAISE 2001*, LNCS 2068, pp. 315-329, 2001. Springer-Verlag Berlin Heidelberg 2001

- 184.[Hudson&King, 1989] Hudson S.E., King R: "Cactis: A Self-Adaptive, Concurrent Implementation of an Object/Oriented Database Management System", ACM Transactions on Database Systems, Vol. 14, No 3, September 1989, Pages 291.321.
- 185.[Humphrey, 1988] Humphrey, W. S. (March 1988). "Characterizing the software process: A maturity framework" IEEE Software 5 (2): 73-79.
- 186.[Huo et al, 2004] Huo M, Verner J, Zhu L, Babar M.A: "Software Quality and Agile Methods", Proceedings of the 28<sup>th</sup> Annual International Computer Software and Applications Conference, IEEE Computer Society.
- 187.[Irani, 2002] Irani Z: "Information systems evaluation: navigating through the problem domain", Information & Management, 40, 2002, 11-24
- 188.[Isaev, 2010] Isaev D., "Information support of corporate governance and strategic management using analytical software," In: Proceedings of 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2010). October 29-31, 2010, Xiamen, China. Edited by Wen Chen, Shaozi Li. Beijing: IEEE, 2010, vol. 3, pp. 44-48.
- 189.[Isaev, 2011] Isaev D: "Performance Management Systems: Conceptual Modeling", 2011 International Conference on Economics and Business Information, IPEDR vol.9 (2011) IACSIT Press, Bangkok, Thailand
- 190.[Issa&Rub, 2007] Issa A.A, Rub F.A.A: "Performing Early Feasibility Studies of Software Development Projects Using Business Process Models", Proceedings of the World Congress on Engineering 2007, Vol I, WCE 2007, July 2-4, 2007, London, U.K.
- 191.[Jadhav&Sonar, 2009] Jadhav A.S, Sonar R.M: "Evaluating and selecting software packages: A review", Information and Software Technology, 51, (2009), 555-563
- 192.[Janus, 2012] Janus A: "Towards a common Agile Software Development Model", ACM SIGSOFT Software Engineering Notes, July 2012, Volume 37, Number 4
- 193.[Jarvenpaa et al, 1998] Jarvenpaa S.L, Knoll K, Leidner D.E: "Is Anybody Out There? Antecedents of Trust in Global Virtual Teams", Journal of Management Information Systems; Spring 1998; 14, 4; ABI/INFORM Global pg. 29
- 194.[Jarvenpaa& Leidner, 2006] Jarvenpaa S. L, Leidner D.E: "Communication and Trust in Global Virtual Teams", Journal of computer mediated communication, 23 JUN 2006.
- 195.[Javanmard&Alian, 2015] JavanMard M, Alian M: "Comparison between Agile and Traditional software development methodologies", The Second National Conference on Applied Research in Science and Technology, Faculty of Science, Cumhuriyet University, 2015.
- 196.[Jensen& Bjørn, 2012] Jensen, R. E., Bjørn, P. (2012): „Divergence and convergence in global software development: Cultural complexities as social worlds“, In From research to practice in the design of cooperative systems: Results and open challenges (pp. 123-136). Springer London.
- 197.[Jevtić&Letić, 2011] Jevtić V, Letić D: "Postojeće metode i prikaz razvijenog modela za određivanje trajanja projekta", Konferencija YUINFO, 2011.
- 198.[Jiang et al, 2004] Jiang J.J, Klein G, Hwang H-G, Huang J, Hung S-Y: An exploration of the relationship between software development process maturity and project performance, Information & Management 41 (2004) 279-288
- 199.[Jianga et al, 2002] Jianga J. J, Kleinb G, Discenzab R: "Perception differences of software success: provider and user views of system metrics", Journal of Systems and Software, Volume 63, Issue 1, 15 July 2002, Pages 17-27
- 200.[Jiménez et al, 2009] Jiménez M, Piattini M, Vizcaíno A (2009). Challenges and Improvements in Distributed Software Development: A Systematic Review, Advances in Software Engineering. Hindawi Publishing Corporation, Volume 2009
- 201.[Jiménez et al, 2010] Jiménez M, Vizcaíno A, Piattini M (2010). Improving Distributed Software Development in Small and Medium Enterprises, The Open Software Engineering Journal, 2010, 4, 26-37
- 202.[Jugdev&Müller, 2005] Jugdev K, Müller R: A RETROSPECTIVE LOOK AT OUR EVOLVING UNDERSTANDING OF PROJECT SUCCESS, Project Management Journal; Dec 2005; 36, 4; ABI/INFORM Global pg. 19
- 203.[Jung et al, 2004] Jung H-W, Kim S-G, Chung C-S: Measuring Software Product Quality: A Survey of ISO/IEC 9126, IEEE Software, 2004.
- 204.[Jurison, 1999] Jurison J: "Software Project Management: The Manager's view", Communications of the Association for Information Systems, Volume 2, Article 17, September 1999
- 205.[Joosten et al, 2011] Joosten D., Basten D., Mellis W: "Measurement of Information System Project Success in Organizations – What Researchers Can Learn from Practice", 19th European Conference on Information Systems, ECIS 2011, Helsinki, Finland, June 9-11, 2011
- 206.[Kaner&Bond,2004] Kaner, C., Bond, W.: "Software Engineering Metrics: What Do They Measure and How Do we Know?", 10th International Software Metrics Symposium, 2004
- 207.[Kaplan&Norton, 1992] Kaplan, R. S., Norton D. P., The balanced scorecard - Measures that drive performance, Harvard Business Review (January-February 1992.): 71-79.



208. [Kayworth& Leidner, 2001] Kayworth T.R, Leidner D.E: "Leadership Effectiveness in Global Virtual Teams, *Journal of Management Information Systems* / Winter 2001–2002, Vol. 18, No. 3, pp. 7–40., M.E. Sharpe, Inc.
209. [Kesh, 1995] Kesh S, Evaluating the Quality of Entity Relationship Models, *Information and Software Technology*. 37(12), 1995, 681-689.
210. [Keshlaf& Riddle, 2010] Keshlaf A.A, Riddle S: "Risk management for web and distributed software development projects." *Internet Monitoring and Protection (ICIMP)*, 2010 Fifth International Conference on. IEEE, 2010.
211. [Khalid et al, 2014] Khalid A, Zahra S, Khan M.F: "Suitability and Contribution of Agile Methods in Mobile Software Development", *I.J. Modern Education and Computer Science*, 2014, 2, 56-62
212. [Khan et al, 2007] Khan R.A., Mustafa K, Ahson S.I: An Empirical Validation of Object Oriented Design Quality Metrics, *Journal of King Saud University, Computer and Information Sciences*, Vol. 19, 2007, pp. 1-16
213. [Khan&Ahmad, 2006] Khan S.U, Ahmad I: "Data Replication in Large Distributed Computing Systems using Supergames," in *International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, Las Vegas, NV, USA, June 2006, pp. 38-44.
214. [Khan et al, 2014] Khan H.H, bin Mahrin M.N, bt Chuprat S: „Factors Generating Risks during Requirement Engineering Process in Global Software Development Environment“, *International Journal of Digital Information and Wireless Communications (IJDIWC)* 4(1): 63-78 The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2225-658X)
215. [Khatibi&Jawawi, 2011] Khatibi V. B, Jawawi D.N.A., "An Analytical Model for Software Development Effort Estimation Based on Investigation of Project Characteristics", *The 3rd Software Engineering Postgraduate Workshop (SEPoW)*, 2011, pp. 34-40.
216. [Khatibi et al, 2012] Khatibi B. V, Dayang N. A. Jawawi D.N.An, Khatibi E: " Investigating the Effect of Using Methodology on Development Effort in Software Projects", *International Journal of Software Engineering and Its Applications* Vol. 6, No. 2, April, 2012
217. [Khriss et al, 2008] Khriss I, Tremblay G, Lévesque E, Jacques A: "Towards Adaptability Support in Collaborative Business Processes", *2008 International MCETECH Conference on e-Technologies*
218. [Kimble et al, 2000] Kimble C, Li F, Barlow A: „Effective Virtual Teams Through Communities of Practice, *Management Science, Theory, Method and Practice*, Research paper no 2000/9, Strathclyde Business School, Glasgow, Scotland
219. [Kitchenham et al, 1997] Kitchenham B, Linkman S, Law S: „DESMET: A methodology for evaluating Software Engineering methods and tools“, *COMPUTING 82 CONTROL ENGINEERING JOURNAL* JUNE 1997
220. [Klemm et al, 2005] Klemm F, Datta A, Aberer K: „A Query-Adaptive Partial Distributed Hash Table for Peer-to-Peer Systems“, *Current Trends in Database Technology-EDBT 2004 Workshops*. Springer Berlin Heidelberg, 2005.
221. [Kocaguneli et al, 2013] Kocaguneli E, Zimmermann T, Bird C, Nagappan N, Menzies T (2013). Distributed Development Considered Harmful? In *Proceedings of Software Engineering in Practice, ICSE 2013*, San Francisco, CA, USA
222. [Kogut&Metiu, 2001] Kogut B, Metiu A (2001): "Open-source software development and distributed innovation", *Oxford review of economic policy*, Vol 17, No 2, Oxford university press
223. [Kommeren& Parviainen, 2007] Kommeren R, Parviainen P (2007). Philips experiences in global distributed software development. *Empirical Software Engineering*, vol. 12:647–660, Springer
224. [Komi-Sirvio&Tihinen, 2005] Komi-Sirvio S, Tihinen M (2005), *Lessons Learned by Participants of Distributed Software Development, Knowledge and Process Management* Volume 12 Number 2 pp 108–122 (2005), Wiley InterScience
225. [Kontio et al, 2004] Kontio J, Höglund M, Rydén J, Abrahamsson P, "Managing Commitments and Risks: Challenges in Distributed Agile Development:", *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*
226. [Korkala& Abrahamsson, 2007] Korkala M, Abrahamsson P: "Communication in Distributed Agile Development: A Case Study", *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2007)*
227. [Kotlarsky& Oshri, 2005] Kotlarsky J, Oshri I: "Social ties, knowledge sharing and successful collaboration in globally distributed system development projects", *European Journal of Information Systems* (2005) 14, 37–48
228. [Kunz&Black, 1999] Kunz T, Black J.P: "An Architecture for Adaptive Mobile Applications", *Proceedings of Wireless 99, the 11th International Conference on Wireless Communications*, Calgary, Alberta, Canada, pp. 27-38, July 1999.
229. [Kvan, 2000] Kvan T: "Collaborative design: what is it?", *Automation in Construction* 9, 2000, pp.409–415

230. [Kwak&Ibbs, 2002] Young Hoon Kwak Y.H, Ibbs C.W: "Project Management Process Maturity (PM<sup>2</sup>) Model", *Journal of Management in Engineering*, July 2002, pp 150-155
231. [Laanti et al, 2011] Laanti M, Salo O, Abrahamsson P: "Agile Methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation", *Information and Software Technology*, 53 (2011) 276-290
232. [Lamersdorf et al, 2009] Lamersdorf A, Jürgen Münch J, and Dieter Rombach D: "A survey on the state of the practice in distributed software development: Criteria for task allocation." Fourth IEEE International Conference on Global Software Engineering (ICGSE 2009). IEEE, 2009.
233. [Lamersdorf& Münch, 2010] Lamersdorf A, Münch J: A Multi-Criteria Distribution Model for Global Software Development Projects, *Journal of the Brazilian Computer Society*, August 2010, Volume 16, Issue 2, pp 97-115
234. [Lavazza, 2000] Lavazza L., Providing Automated Support for the GQM Measurement Process, *IEEE Software*, 2000, pp. 56-62.
235. [Larman& Basili, 2003] Larman C., Basili V. R. "Iterative and Incremental Development: A Brief History", *IEEE Computer Archive* 36(6), 47-56 (2003)
236. [Lee et al, 2006] Lee G., Delone W., Espinosa J. A.: "Ambidextrous Coping Strategies in Globally Distributed Software Development Projects," *Communications of the ACM* (49)10, pp. 35-40, October 2006
237. [Lee et al, 2009] Lee S-Y T, Kim H-W, Gupta S: "Measuring open source software success", Elsevier, *The International Journal of Management Science*, Omega 37 (2009) 426 - 438
238. [Lee& Yong, 2009] Lee S, Yong H-S: "Distributed agile: project management in a global environment", *Empir Software Eng*, Springer Science + Business Media, LLC 2009
239. [Lehman, 1996] Lehman, M. M. "Laws of software evolution revisited." *Software process technology*. Springer Berlin Heidelberg, 1996. 108-124.
240. [Lehman, 1978b] Lehman M.M: "Laws of Program Evolution - Rules and Tools for Programming Management", *Proc. Infotech State of the Art Conf., Why Software Projects Fail*, - Apr. 1978, pp. 11/1 11/25
241. [Lehman, 1980a] Lehman, M. M. "On understanding laws, evolution, and conservation in the large-program life cycle." *Journal of Systems and Software* 1 (1980): 213-221.
242. [Lehman, 1980b] Lehman M.M, "Programs, Life Cycles and Laws of Software Evolution", *Proc. IEEE Special Issue on Software Engineering*, v. 68, n. 9, Sept. 1980, pp. 1060 - 1076
243. [Lehman, 1991] Lehman MM: "Software Engineering, the Software Process and Their Support", *IEE Softw. Eng. Journal*, Spec. Issue on Software Environments and Factories, Sept. 1991, v. 6, n. 5, pp. 243 - 258
244. [Lemos& Fiadeiro, 2002] de Lemos R, Fiadeiro J.L: "An Architectural Support for Self-Adaptive Software for Treating Faults", *WOSS '02*, Nov 18-19, 2002, Charleston, SC, USA.
245. [Leau et al, 2012] Leau Y.B, Loo W.K, Tham W.Y, Tan S.F: *Software Development Life Cycle AGILE vs Traditional Approaches*, 2012 International Conference on Information and Network Technology (ICINT 2012), *IPCSIT vol. 37* (2012), IACSIT Press, Singapore
246. [Lenz& Kuhn, 2004] Lenz R., Kuhn K.A.: "Towards a continuous evolution and adaptation of information systems in healthcare", *International Journal of Medical Informatics* (2004) 73, 75-89
247. [Levesque et al, 2001] Levesque L.L, Wilson J.M, Wholey D.R: "Cognitive divergence and shared mental models in software development project teams", *Journal of Organizational Behavior*, 22, 135±144 (2001)
248. [Leymann et al, 2002] Leymann F, Roller D, Schmidt M-T: "Web services and business process management", *IBM SYSTEMS JOURNAL*, VOL 41, NO 2, 2002
249. [Lgerfalk et al, 2009] Lgerfalk P.J., Fitzgerald B, Slaughter S.A: *Flexible and Distributed Information Systems Development: State of the Art and Research Challenges*, *Information Systems Research*, Vol. 20, No. 3, September 2009, pp. 317-328
250. [Liang et al, 2007] Liang T-P, Liu C-C, Lin T-M, Lin B: "Effect of team diversity on software project performance", *Industrial Management & Data Systems*, Vol. 107 No. 5, 2007, pp. 636-653, Emerald Group Publishing Limited
251. [Lilburne et al, 2004] Lilburne B, Devkota P, Khan K.M: *Measuring Quality Metrics for Web Applications*, IRMA International Conference, New Orleans, USA, 2004.
252. [Liu, 2009] Liu G: *Tracking Software Development Progress with Earned Value and Use Case Point*, *Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009)*, Qingdao, China, November 21-22, 2009.
253. [Liu et al, 2005] Liu B, Zhu Y, Jbantova M, Momberger B, Rundensteiner E.A: "A Dynamically Adaptive Distributed System for Processing Complex Continuous Queries", *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, 2005
254. [Linberg, 1999] Linberg K.R.: "Software developer perceptions about software project failure: a case study", *The Journal of Systems and Software* 49 (1999) 177±192

255. [Lockemann, 2003] Lockemann P.: "Information system Architectures: From Art to Science", Conference BTW2003 (Business, Technologie and Web) Proceedings 2003.
256. [Maalej& Happel, 2008] Maalej W, Happel H-J: A Lightweight Approach for Knowledge Sharing in Distributed Software Teams, T. Yamaguchi (Ed.): PAKM 2008, LNAI 5345, pp. 14-25, 2008. Springer-Verlag Berlin Heidelberg 2008
257. [Madey et al, 2002] Madey G, Freeh V, Tynan R (2002): "The Open Source Software Development Phenomenon: An Analysis Based on Social Network Theory", 2002, Eighth Americas Conference on Information Systems
258. [Marinescu et al, 1990] Marinescu D.C, Lumpp J.E, JR ., Casavant T.L, Siegel H.J: "Models for Monitoring and Debugging Tools for Parallel and Distributed Software", Journal of Parallel and Distributed Computing , 17 - 184 ( 1990)
259. [Martinsons et al, 1999] Martinsons M, Davison R, Tse D, The balanced scorecard: a foundation for the strategic management of information systems, Decision support systems journal, Elsevier, 1999
260. [Mahapatra& Das, 2012] Mahapatra D.K, Das T.K.: Prioritizing SCM for Managing Inconsistency in Distributed Software Project Development, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-1, December 2012
261. [Mahmoud&Riordon, 1976] Mahmoud, S., Riordon, J.: Optimal allocation of resources in distributed information networks. ACM Trans. on Database Systems 1 (1976) 66-78
262. [Mahnich&Zabkar, 2008] Mahnic V, Zabkar N: Using COBIT Indicators for Measuring Scrum-based Software Development, WSEAS Transactions on Computers, Issue 10, Volume 7, October 2008
263. [Mair& Shepperd, 2011] Mair, C., Shepperd, M. (2011) Human Judgement and Software Metrics: Vision for the Future. WETSOM'11 (May 24, 2011), Waikiki, Honolulu, HI USA.
264. [Martins et al, 2004] Martins L.L, Gilson L.L, Maynard M.T: "Virtual Teams: What Do We Know and Where Do We Go From Here?", Journal of Management 2004; 30; 805
265. [Matson& DeLoach, 2005] Matson E, DeLoach S: "Autonomous Organization-Based Adaptive Information Systems", Integration of Knowledge Intensive Multi-Agent Systems, 2005. International Conference on 18-21 April, 2005
266. [Maznevski& Chudoba, 2000] Maznevski M.L, Chudoba K.M: „Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness", Journal Organization Science, 2000 INFORMS, Vol. 11, No. 5, September-October 2000, pp. 473-492
267. [Maurino et al, 2003] Maurino A, Modafferi S, Pernici B: "Reflective architectures for adaptive information systems", Service-Oriented Computing - ICSOC 2003, and also in: Chapter, Volume 2910 of the series Lecture Notes in Computer Science pp 115-131
268. [Marinescu et al, 1990] Marinescu D.C, Lumpp J.E, Casavant T.L., Siegel H. J: Models for Monitoring and Debugging Tools for Parallel and Distributed Software, JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 17 - 184 (1990)
269. [Maxmilien&Singh, 2004] Maximilien E.M, Singh M.P: "A framework and ontology for dynamic web services selection", IEEE Internet Computing, 2004
270. [McKinley et al, 2004] McKinley P. K., Sadjadi S. M., Kasten E. P., Cheng B. H. C., "Composing adaptive software," IEEE Computer, vol. 37, no. 7, pp. 56-64, 2004.
271. [Meso&Jain, 2006] Meso P, Jain R: "Agile Software Development: Adaptive Systems Principles and Best Practices", Journal Information Systems Management, 2006.
272. [Mekonnen& Lamenuw, 2013] Mekonnen A, Lamenuw W: "The Role of Personality of Software Developers in Selecting Software Development Methodology", HiLCoE Journal of Computer Science and Technology, Volume 1, Number 2, June 2013
273. [Milošević, 2010] Milošević B: "Обрада подataka у дистрибуираним базама", YUINFO 2010.
274. [Milosevic&Patanakul, 2005] Milosevic D, Patanakul P, "Standardized project management may increase development projects success", International Journal of Project Management 23 (2005) 181-192
275. [Mishra et al, 2010] Mishra S, Tripathy K.C, Mishra M.K: "Effort Estimation Based on Complexity and Size of Relational Database System", International Journal of Computer Science and Communication, Vol. 1, No. 2, July-December 2010, pp. 419-422
276. [Misirli et al, 2011] Misirli, A., Caglayan, B., Miransky, A., Bener, A., Ruffolo, N. (2011). Different Strokes for Different Folks: A Case Study on Software Metrics for Different Defect Categories. WETSOM'11 (May 24, 2011), Waikiki, Honolulu, HI USA.
277. [Mockus et al, 2000] Mockus A, Fielding T.R, Herbsleb J, A Case Study of Open Source Software Development: The Apache Server, Proceedings of ICSE'2000
278. [Mockus& Herbsleb, 2002] Mockus, A., Herbsleb, J. D. (2002). Why not improve coordination in distributed software development by stealing good ideas from open source. In Meeting Challenges and Surviving Success: The 2nd Workshop on Open Source Software Engineering, pp. 19-25.

279. [Moe & Šmite, 2008] Moe N.B, Šmite D: "Understanding a Lack of Trust in Global Software Teams: A Multiple-case Study", *Journal Software Process Improvement and Practice*, 2008; 13: 217-231, Published online 12 June 2008 in Wiley InterScience
280. [Monsalve et al, 2011] Monsalve C, Abran A, April A: "Measuring Software Functional Size from Business Process Models", *International Journal of Software Engineering and Knowledge Engineering*, World Scientific Publishing Company, June 10, 2011
281. [Montoya-Weiss et al, 2001] Montoya-Weiss, M. M., Massey, A. P., & Song, M. (2001). Getting it together: Temporal coordination and conflict management in global virtual teams. *Academy of management Journal*, 44(6), 1251-1262
282. [Moody, 1998] Moody L: Metrics For Evaluating the Quality of Entity Relationship Models, *Proceedings of the Seventeenth International Conference on Conceptual Modelling (ER '98)*, Singapore, November 1998, 213-225.
283. [Munns & Bjeirmi, 1996] Munns A.K, Bjeirmi B.F: The role of project management in achieving project success, *International Journal of Project Management* Vol. 14, No. 2, pp. 81-87, 1996.
284. [Nagappan & Ball, 2007] Nagappan, N., Ball, T. (2007) Using Software Dependencies and Churn Metrics to Predict Field Failures: An Empirical Case Study. *IEEE First International Symposium on Empirical Software Engineering Measurement*.
285. [Narayanan & McIlraith, 2002] Narayanan S, McIlraith S: "Simulation, Verification and Automated Composition of Web services", *WWW 2002*, May 8, 2002
286. [Narayan & Balan, 2007] Narayan R, Balan, R. K: "Globally Distributed Software Development Project Performance: An Empirical Analysis", In *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE '07)*, 2007, 125-134. New York: ACM
287. [Narayan & Balan, 2012] Narayan R, Balan R.K: "Overcoming the challenges in cost estimation for distributed software projects." *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press, 2012.
288. [Nelson, 2007] Nelson R: IT Project Management: Infamous Failures, Classic Mistakes, and Best practices, *MIS Quarterly Executive* Vol. 6 No. 2 / June 2007
289. [Nerur et al, 2005] Nerur S, Manapatra R.K, Mangalaraj G: "Challenges of Migrating to Agile Methodologies", *Communications of the ACM*, May 2005, Vol 48. No. 5
290. [Nidiffer & Dolan, 2005] Nidiffer K.E, Dolan D: *Evolving Distributed Project Management*, September / October 2005 IEEE Software
291. [Niebecker et al, 2008] Niebecker, K., Eager, D., Kubitza, K. (2008) Improving cross-company project management performance with a collaborative project scorecard, *International Journal of Managing Projects in Business*, Emerald Group
292. [Nordio et al, 2011] Nordio M, Estler H-C, Meyer B, Tschannen J, Ghezzi C, Di Nitto E: "How do Distribution and Time Zones affect Software Development? A Case Study on Communication", *Global Software Engineering (ICGSE)*, 2011 6th IEEE International Conference on. IEEE, 2011.
293. [Noll et al, 2010] Noll J, Beecham S, Richardson I (2010). *Global Software Development and Collaboration: Barriers and Solutions*. ACM Inroads, 2010 September, Vol. 1, No. 3
294. [Nuevo et al, 2011] del Nuevo E, Mario Piattini M, Francisco J. Pino F.J: "Scrum-based Methodology for Distributed Software Development", 2011 Sixth IEEE International Conference on Global Software Engineering
295. [Oktaba et al, 2007] Oktaba H, Garcia F, Piattini M, Ruiz F, Pino F, Alquicira C: "Software Process Improvement: The Competisoft Project", *IEEE Computer*, October 2007.
296. [Omoroniya, 2010] Omoroniya I, Ferguson J, Roper M, Wood M: "A review of awareness in distributed collaborative software engineering", *Journal Software: Practice and Experience*, Volume 40, Issue 12, pp.1107-1133, November-December 2010.
297. [Oreizy et al, 1999] Oreizy, P., Gorlick, M. M., Taylor, R. N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D. S., and Wolf, A. L. 1999. "An architecture-based approach to self-adaptive software", *IEEE Intelligent Systems and their applications*, Vol. 14, 3, 54-62, 1999.
298. [Ojala & Tyrväinen, 2007] Ojala, A, Tyrväinen, P. (2007). Market Entry and Priority of Small and Medium- Sized Enterprises in the Software Industry: An Empirical Analysis of Cultural Distance, Geographical Distance, and Market Size. *Journal of International Marketing*, 15 (3), 123-149.
299. [Olson & Olson, 2003] Olson J, Olson G: "Culture Surprises in Remote Software Development Teams", December/January 2003-2004 ACM QUEUE
300. [Oshri et al, 2008] Oshri I, Kotlarsky J, Willcocks L: "Missing Links: Collaborative Building Social Ties for Global Teamwork", *Communications of the ACM*, April 2008/Vol. 51, No. 4
301. [Paasivaara et al, 2009] Paasivaara M, Durasiewicz S, Lassenius C (2009), Using Scrum in Distributed Agile Development: A Multiple Case Study, *Fourth IEEE International Conference on Global Software Engineering*, 2009

302. [Paetsch et al, 2003] Paetsch F, Eberlein A, Maurer F: "Requirements Engineering and Agile Software Development", Proceedings of Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE '03
303. [Palmius, 2007] Palmius J: "Criteria for measuring and comparing information systems", Proceedings of the 30<sup>th</sup> Information Systems Research Seminars in Scandinavia IRIS 2007
304. [Pandey et al, 2014] Pandey G, Jelschen J, Kuryazov J.D., Winter A: Quality Measurement Scenarios in Software Migration, *Softwaretechnik-Trends* 34(2) (2014)
305. [Papazoglou&Georgakopoulos, 2003] Papazoglou M.P, Georgakopoulos D: „Service-Oriented Computing“, *Communications of the ACM*, October 2003, Vol. 46, No. 10
306. [Parkinson et al, 2008] Parkinson, S. T.Hierons, R. M., Lycett, M. and Norman M. "Practitioner-Based Measurement: A Collaborative Software Measurement Approach", *Communications of the ACM*, 2008
307. [Pathak et al, 2007] Pathak S.D, Day J.M, Nair A, Sawaya W.J, Kristal M.M.: "Complexity and Adaptivity in Supply Networks: Building Supply Network Theory Using a Complex Adaptive Systems Perspective", *Decision Sciences*, Volume 38 Number 4, November 2007
308. [Pautasso, 2008] Pautasso C: "REST vs. SOAP: Making the Right Architectural Decision", SOA Symposium 2008, Amsterdam
309. [Pena-Mora et al, 2000] Pena-Mora F., Hussein K., Vadhavkar S., Benjamin K.: „CAIRO: a concurrent engineering meeting environment for virtual design teams“, *Artificial Intelligence in Engineering* 14 (2000) 203-219.
310. [Pereira et al, 2008] Pereira J, Cerpa N, Verner J, Rivas M, Procaccino J.D.: "What do software practitioners really think about project success: A cross-cultural comparison", *Elsevier, The Journal of Systems and Software*, 81 (2008) 897-907
311. [Perkowitz&Etzioni, 2000] Perkowitz, M., Etzioni, O.: "Towards Adaptive Web Sites: Conceptual Framework and Case Study", *Journal Artificial Intelligence*, 2000, 118(1-2), 245-275.
312. [Pestic&Aalst 2006] Pestic M., van der Aalst W.M.P.: "A Declarative Approach for Flexible Business Processes Management", In: J. Eder, S. Dustdar et al. (Eds.): *BPM 2006 Workshops, LNCS 4103*, pp. 169-180, 2006. Springer-Verlag Berlin Heidelberg 2006
313. [Petersen&Wohlin, 2009] Petersen K, Wohlin P: "A comparison of issues and advantages in agile and incremental development between state of the art and industrial case", *Journal of Systems and Software*, 2009.
314. [Phalnikar et al, 2008] Phalnikar R, Deshpande V.S., Joshi S.D.: „Applying Agile Principles for Distributed Software Development“, *IEEE International Conference on Advanced Computer Control*, 2008.
315. [Phung-Khac et al, 2007] Phung-Khac A, Beugnard A, Gilliot J-M, Segarra M-T: "A Model of Self-Adaptive Distributed Components", *Practical Approaches for Software Adaptation (2007)*: 9.
316. [Piattini et al, 2000] Piattini M, Genero M, Calero C: Data model metrics, *UK Academy of Information Systems Annual Conference 2000*, Cardiff, 26-28 April 2000
317. [Piprani, 2008] Piprani B, Ernst D: A Model for Data Quality Assessment, *Proceedings of OTM Workshops 2008*. pp.750~759
318. [Porter&Selby, 1990] Porter A.A, Selby R.W: "Empirically-Guided Software Development Using Metric-Based Classification Trees", *IEEE Software*, April 1990
319. [Prikladnicki et al, 2004] Prikladnicki R, Yamaguti M.H, Antunes D.C.: "Risk Management in Distributed Software Development: A Process Integration Proposal." *Virtual Enterprises and Collaborative Networks*. Springer US, 2004. Pp. 517-526.
320. [Prikladnicki et al, 2008] Prikladnicki R., Damian D., Audy J. L. N., "Patterns of evolution in the practice of distributed software development: quantitative results from a systematic review," In: 12<sup>th</sup> International Conference on Evaluation and Assessment in Software Engineering (EASE) University of Bari, Italy, 2008.
321. [Procaccino& Verner, 2006] Procaccino J.D, Verner J.M: Software project managers and project success: An exploratory study, *The Journal of Systems and Software* 79 (2006) 1541-1551
322. [Putnam, 1987] Putnam L. H., "A general empirical solution to the macrossoftware sizing and estimating problem," *IEEE Transactions on Software Engineering*, vol. 4, pp. 345-361, 1987.
323. [Qumer&Henderson-Sellers, 2006] Qumer A, Henderson-Sellers B: „Measuring Agility and Adoptability of Agile Methods: A 4-Dimensional Analytical Tool“, *IADIS International conference Applied Computing*, 2006
324. [Qumer&Henderson-Sellers, 2008] Qumer A, Henderson-Sellers B: „A framework to support evaluation, adoption and improvement of agile methods in practice“, *The Journal of Systems and Software* 81 (2008), 1899-1919
325. [Rabinovich, 1998] Rabinovich, M.: Issues in web content replication. *Data Engineering Bull.* 21 (1998) 21-29

- 326.[Ramesh& Dennis, 2002] Ramesh V, Dennis A.R: „The Object-Oriented Team: Lessons for Virtual Teams from Global Software Development“, Proceedings of the 35th Hawaii International Conference on System Sciences, 2002
- 327.[Rasli&Mohd, 2008] Rasli A.M, Mohd W.M.W: Project Performance Framework: The Role of Knowledge Management and Information Technology Infrastructure, Asian Journal of Business and Accounting, 1(2), 2008, 39-64 ISSN-1985-4064
- 328.[Raut, 2008] Raut S: "Management Framework for Globally Distributed Teams in IT Industry", Communications of the IBIMA, Volume 2, 2008
- 329.[Raymond& Bergeron, 2008] Raymond L, Bergeron F: Project management information systems: An empirical study of their impact on project managers and project success, International Journal of Project Management 26 (2008) 213-220
- 330.[Reichert et al, 2005] Reichert M, Rinderle S, Kreher U, Dadam P: "Adaptive Process Management with ADEPT2", Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)
- 331.[Reifer, 2000] Reifer D.J: Web Development: Estimating Quick-to-Market Software, IEEE SOFTWARE, November/December 2000.
- 332.[Rocha et al, 2011] Rocha R.G.C, Costa C, Rodrigues C, Ryan Ribeiro de Azevedo R.R, Junior I.H, Meira S, Prikladnicki R: "Collaboration Models in Distributed Software Development: a Systematic Review", CLEI Electronic Journal, Vol 14, No 2, Paper 1, August 2011
- 333.[Royce, 1970] Royce W. W., Managing the Development of Large Software Systems, in Proceedings of IEEE WESCON, pages 1-9, 1970.
- 334.[Rosemann& Wiese, 1999] Rosemann, M. Wiese, J., Measuring the Performance of ERP Software - a Balanced Scorecard Approach, Conference Proceeding, The 10<sup>th</sup> Australasian Conference on Information Systems, 1999.
- 335.[Rubin&Rubin, 2010] Rubin E, Rubin H: "Supporting agile software development through active documentation", Journal Requirements Engineering, Springer-Verlag, London, 2010
- 336.[Salehie&Tahvildari, 2009] Salehie M, Tahvildari L: Self-Adaptive Software: Landscape and Research Challenges, ACM Transactions on Autonomous and Adaptive Systems, March 2009
- 337.[Sandusky&Gasser, 2005] Sandusky R.J, Gasser L: „Negotiation and the Coordination of Information and Activity in Distributed Software Problem Management“, GROUP'05, November 6-9, 2005, Sanibel Island, Florida, USA.
- 338.[Sanz et al, 2005] Sanz L.F.: "Software quality: concepts and evidences", First Workshop on Ontology, Conceptualizations and Epistemology for Software and Systems Engineering, ONTOSE 2005
- 339.[Salaun et al, 2004] Salaun G, Bordeaux L, Schaerf M: "Describing and Reasoning on Web Services using Process Algebra", Proceedings of the 2<sup>nd</sup> International Conference on Web Services, 2004
- 340.[Sarker& Sahay, 2002] Sarker S, Sahay S: „Information Systems Development by US-Norwegian Virtual Teams: Implications of Time and Space“, Proceedings of the 35th Hawaii International Conference on System Sciences, 2002
- 341.[Satoh, 2000] Satoh I: "MobileSpaces: A Framework for Building Adaptive Distributed Applications Using Hierarchical Mobile Agent System" In: ICDCS 2000, pp. 161-168 (2000)
- 342.[Schilit&Theimer, 1994] Schilit, B., Theimer, M. (1994) 'Disseminating active map information to mobile hosts', IEEE Network, Vol. 8, No. 5, pp. 22-32.
- 343.[Sengupta et al, 2006] Sengupta B, Chandra S, Sinha V (2006). A Research Agenda for Distributed Software Development, ICSE'06, May 20-28, 2006, Shanghai, China.
- 344.[Sharif et al, 2012] Sharif B, Khan S.A, Bhatti M.W.: Measuring the Impact of Changing Requirements on Software Project Cost: An Empirical Investigation, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012
- 345.[Shenhar&Dvir, 1996] Shenhar J, Dvir D: "Toward a typological theory of project management", Research Policy 25, 607-632 (1996).
- 346.[Shenhar et al, 2000] Shenhar J., Poli M., Lechler T., "A new framework for strategic project management", in T. Khalil (ed.), Management of Technology VIII, University of Miami, Miami, FL (2000).
- 347.[Shenhar et al, 2001] Shenhar A.J, Dvir D, Levy O, Maltz A.C.: "Project Success: A Multidimensional Strategic Concept", Pergamon, Long Range Planning 34 (2001) 699-725
- 348.[Shepperd& Schofield, 1997] Shepperd M., Schofield C., "Estimating Software Project Effort Using Analogies," IEEE Transaction on software engineering, vol. 23, pp. 736-743, 1997.
- 349.[Shrivastava&Date, 2010] Shrivastava S.V, Date H: "Distributed Agile Software Development: A Review", JOURNAL OF COMPUTER SCIENCE AND ENGINEERING, VOLUME 1, ISSUE 1, MAY 2010
- 350.[Shirani, 2008] Shirani A: "Performance of virtual teams in global software development: the mediating role of leadership", Proceedings of the 39th Annual meeting - Decision Sciences, 2008, pp.5221-5226.

- 351.[Sim et al, 2003] Sim S.E, Easterbrook S, Holt R.C: "Using Benchmarking to Advance Research: A Challenge to Software Engineering", Software Engineering, 2003. Proceedings. 25th International Conference on, 3-10 May, 2003
- 352.[Sowe et al, 2009] Sowe, S.K., Ghosh R, Soete L: "Annals of Knowledge Sharing in Distributed Software Development Environments: Experience from Open Source Software Projects." in: Jurgen Munch, Peter Liggesmeyer (Eds.), Lecture Notes in Informatics, Social Aspects in Software Engineering, (SENSE, SofTEAM), 347-359
- 353.[Spinellis, 2006] Spinellis D (2006). Global software development in the freebsd project. In Proceedings of the 2006 international workshop on Global software development for the practitioner, GSD '06, pages 73–79, New York, NY, USA, 2006. ACM.
- 354.[Stewart& Gosain, 2006] Stewart K.J, Gosain S: "The Impact of Ideology of Effectiveness in open source software development teams", MIS Quarterly, 2006.
- 355.[Stojanovic et al, 2003] Stojanovic Z, Dahanayake A, Sol H: "Architectural Design in Agile Development Methodologies", International conference EMMSAD, 2003.
- 356.[Stotts& Williams, 2002] Stotts D, Williams L: "Distributed Pair Programming", In: (Wells D, Williams L eds.) Extreme Programming and Agile Methods – XP/Agile Universe 2002, Second XP Universe and First Agile Universe Conference Chicago, IL, USA, August 4-7, 2002 Proceedings, Lecture Notes in Computer Science 2418
- 357.[Suchan& Hayzak, 2001] Suchan J, Hayzak G: "The Communication Characteristics of Virtual Teams: A Case Study", IEEE Transactions on professional communication, Vol. 44, No. 3, September 2001
- 358.[Sukhoo et al, 2005] Sukhoo A, Barnard A, Eloff M.M, Van der Poll J.A., Motah M: Accommodating Soft Skills in Software Project Management, Issues in Informing Science and Information Technology, Informing Science, Volume 2, 2005, pp.691-703.
- 359.[Suri et al, 2009] Suri P.K., Bhushan B, Jolly A: Simulator for Risk Assessment of Software Project Performance Measurement, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009
- 360.[Sutherland et al, 2007] Sutherland J, Viktorov D.A, Blount J, Puntikov N: Distributed Scrum: Agile Project Management with Outsourced Development Teams, 40<sup>th</sup> Hawai'i International conference on system sciences, 2007.
- 361.[Swanson, 1976] Swanson E.B, "The dimensions of maintenance", Proceedings of the 2nd international conference on Software engineering, San Francisco, 1976, pp 492. — 497.
- 362.[Sycara et al, 2003] Sycara K, Paolucci M, Ankolekar A, Srinivasan N: "Automated discovery, interaction and composition of semantic web services", Journal of Web semantic: Science, Services and Agents on the World Wide Web 1 (2003), 27-46
- 363.[Šmite& Borzovs, 2008] Šmite D, Borzovs J: "Managing uncertainty in globally distributed software development projects." University of Latvia, Computer Science and Information Technologies 733 (2008): 9-23.
- 364.[Tan et al, 2009] Tan H.B.K, Zhao Y, Zhang H: "Conceptual data model-based software size estimation for information systems", Journal ACM Transactions on Software Engineering and Methodology, Vol 19, Issue 2, October 2009
- 365.[Taxen, 2006] Taxen L: "An integration centric approach for the coordination of distributed software development projects", Information and Software Technology 48 (2006) 767–780
- 366.[Theußl& Zeileis, 2009] Theußl S, Zeileis A: "Collaborative Software Development Using R-Forge", The R Journal Vol. 1/1, May 2009 ISSN 2073-4859
- 367.[Thirumaran, 2010] Thirumaran M, Ilavarasan E, Thanigaivel K, Abarna S: "Business Rule Management Framework for Enterprise Web Services", International Journal on Web Service Computing (IJWSC), Vol.1, No.2, December 2010.
- 368.[Thomas, 2008] Thomas J: Researching the value of project management, PMI research conference, Warshaw, Poland, 2008.
- 369.[Tjørnehøj et al, 2012] Tjørnehøj G, Fransgård M, Skalkam S: "Trust in agile teams in distributed software Development", Iris, 35 (2012)
- 370.[Vanderfeesten et al, 2008] Vanderfeesten I, Cardoso J, Mendling J, Reijers H.A., Van der Aalst W: Quality Metrics for Business Process Models, Computers in Industry Journal, Volume 59, Issue 5, May 2008.
- 371.[Vasconcelos et al, 2007] Vasconcelos A, Soursa P, Tribolet J: "Information system Architecture Metrics: an Enterprise Engineering Evaluation Approach", The Electronic Journal of Information Systems Evaluation, Vol 10, Issue 1, pp 91-122, ISSN 1566-6379, 2007.
- 372.[Vasilescu et al, 2011] Vasilescu, B., Serebrenik, A., van den Brand, M. (2011). By No Means: A Study on Aggregating Software Metrics. WETSoM'11 (May 24, 2011), Waikiki, Honolulu, HI USA.
- 373.[Vogelzang, 2006] Vogelzang, F: "Using COSMIC-FFP for sizing, estimating and planning in an ERP environment", Software measurement conference, IWSM/MetriKon 2006, pp. 327-342

- 374.[Vollman, 1994] Vollman T: Standards Support for Software Tool Quality Assessment, Assessment of Quality Software Development Tools, 1994, Proceedings., IEEE Third Symposium on, 7-9 June 1994
- 375.[Vick,1998] Vick J.R.M: "Perspectives On And Problems With Computer-Mediated Teamwork: Current Groupware Issues And Assumptions", Journal of Computer Documentation May 1998/Vol. 22, No. 2
- 376.[Vreede et al, 2006] de Vreede G-J, Kolfshoten G.L, Briggs R.O: "ThinkLets: a collaboration engineering pattern Language", Int. J. Computer Applications in Technology, Vol. 25, Nos. 2/3, 2006, pp.140-154.
- 377.[Wahyudin &Tjoa, 2007] Wahyudin D, Tjoa A.M.: "Event-based monitoring of open source software projects."Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on. IEEE, 2007.
- 378.[Wallace et al, 2004] Wallace L, Keil M, Rai A: How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and An Exploratory Model, Decision Sciences, Vol 35, No 2, 2004.
- 379.[Wang, 2005] Wang Q: „Towards a Rule Model for Self-adaptive Software”, ACM SIGSOFT Software Engineering Notes Page 1 January 2005 Volume 30 Number 1
- 380.[Warkentin et al, 1997] Warkentin M.E, Sayeed L, Hightower R: "Virtual Teams versus Face-to-Face Teams: An Exploratory Study of a Web-based Conference System", Decision Sciences, Volume 28 Number 4, Fall 1997
- 381.[Watson-Manheim et al, 2002] Watson-Manheim, M.B., Chudoba K., Crowston K, "Discontinuities and Continuities: A New Way to Understand Virtual Work," Information, Technology, and People, Vol. 15 (3), 2002, p. 191-209.
- 382.[Wendler, 2013] Wendler R.: "The Structure of Agility from Different Perspectives", Proceedings of the Federated Conference on Computer Science and Information Systems, 2013, pp. 1165-1172
- 383.[Whittaker, 1999] Whittaker B: "What went wrong? Unsuccessful information technology projects", Information Management & Computer Security 7/1 [1999] 23±29
- 384.[Whitworth&Zaic, 2003] Whitworth B, Zaic M: "The Wosp Model: Balanced Information System Design And Evaluation", Communication of the Association for Information Systems, Volume 12, 2003,pp. 258-282.
- 385.[Wohlin, 2000] Wohlin C., von Mayrhauser A., Höst M., Regnell B., "Subjective Evaluation as a Tool for Learning from Software Project Success", Information and Software Technology, Vol. 42, No. 14, pp. 983-992, 2000.
- 386.[Westhuizen& Fitzgerald, 2005] Van Der Westhuizen D, Fitzgerald E. P. (2005) Defining and measuring project success. In: European Conference on IS Management, Leadership and Governance 2005, 7-8 Jul 2005, Reading, United Kingdom.
- 387.[Wu et al, 2003] Wu D, Parsia B, Sirin E, Hendler J, Nau D: "Automating DAML-S Web services composition using SHOP2",The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003. Proceedings
- 388.[Wu et al, 2012] Wu L, White S, Helm J, Feng Y: Distributed Software Development Modelling and Control Framework, International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.5, September 2012
- 389.[Yoder et al, 2001] Yoder J.W, Balaguer F, Johnson R: "Architecture and Design of Adaptive Object-Models", ACM Sigplan Notices 36.12 (2001): 50-60.
- 390.[Yoo& Kanawattanachai, 2001] Yoo Y, Kanawattanachai P: "Developments of transactive memory systems and collective mind in virtual teams", The International Journal of Organizational Analysis, 2001, Vol 9, No 2, pp. 187-208.
- 391.[Yu et al, 2011] Yu L, Ramaswamy S, Nair A: "Using Bug Reports As Software Quality Measure", Proceedings of the 16<sup>th</sup> International conference on Information Quality, 2011.
- 392.[Zachman, 1987] Zachman, J:"A Framework for Information System Architecture", IBM system journal Vol.26 № 3, 1987, p.276 - 292.
- 393.[Zanoni&Audy, 2003] Zanoni R, Audy J.L.N: "Project Management Model for a Physically Distributed Software Development Environment", Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003.
- 394.[Zeng et al, 2004] Zeng L, Benatallah B, Ngu A.H,H, Dumas M, Kalagnanam J, Chang H: "QoS-Aware Middleware for Web Services Composition", IEEE Transactions on Software Engineering, Vol. 30, No. 5, May 2004
- 395.[Zhang&Cheng, 2005] Zhang J., Cheng B. H., "Specifying adaptation semantics," in Proceedings of ICSE 2005 Workshop on Architecting Dependable Systems, (St. Louis, Missouri), May 2005.
- 396.[Zhang&Cheng, 2006] Zhang J, Cheng B.H.C: "Model-Based Development of Dynamically Adaptive Software", ICSE'06, May 20-28, 2006, Shanghai, China



397. [Zhang et al, 2008] Zhang S, Tremaine M, Egan R, Milewski A, O'Sullivan P, Fjermestad J (2008). Occurrence and Effects of Leader Delegation in Virtual Software Teams, *International Journal of e-Collaboration*, Volume 5, Issue 1, IGI Global Publishing
398. [Zhuge, 2002] Zhuge H, "Knowledge flow management for distributed team software development", *Knowledge-Based Systems* 15 (2002) 465-471
399. [Zukerman & Albrecht, 2001] Zukerman I, Albrecht D. W: "Predictive Statistical Models for User Modeling", *Journal User Modeling and User-Adapted Interaction*, March 2011, Volume 11, Issue 1, pp 5-18, Kluwer Academic Publishers
400. [Živković et al, 2005] Živković A, Heričko M, Brumen B, Beloglavec S, Rozman I: "The Impact of Details in the class diagram on software size estimation", *Informatika*, 2005, Vol. 16, No 2, 295-312, Institute of Mathematics and Informatics, Vilnius

### 8.1.2. Књиге и поглавља у зборницима радова

1. [Alonso et al, 2004] Alonso G, Casati F, Kuno H, Macairaju V: "Web Services – Concepts, Architectures and Applications", Springer Verlag, 2004
2. [Andrews, 2000] Andrews G. R. (2000), *Foundations of Multithreaded, Parallel, and Distributed Programming*, Addison-Wesley, ISBN 0-201-35752-6.
3. [Avison & Fitzgerald, 2003] Avison D, Fitzgerald G, „Information systems development: Methodologies, Techniques and Tools“, McGraw-Hill, 2003.
4. [Balaban et al, 1996] Balaban N, Ristić Ž, Đurković J: „Principi informatike“, Savremena administracija, Beograd, 1996.
5. [Beck, 2004] Beck K., „Extreme Programming Explained: Embrace Change“, second ed., Addison-Wesley, 2004, ISBN 978-0321278654.
7. [Black, 2002] Black R., *Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*, John Wiley & Sons, Inc., 2002.
8. [Blanchard & Fabrycky, 2003] Blanchard B. S., Fabrycky W. J., "Systems Engineering and Analysis", John Wiley & Sons, Inc., 3rd edition, 2003.
9. [Blum, 1992] Blum B: *Software engineering, A Holistic View*, Oxford University Press, 1992.
10. [Boehm, 1981] Boehm B. W., "Software Engineering Economics", Englewood Cliffs, NJ, Prentice-Hall, 1981.
11. [Booch et al, 1999] Booch G, Rumbaugh J, Jacobson I: *The unified modelling language user guide*, Addison-Wesley, 1999.
12. [Breiman et al, 1984] Breiman L, Friedman L, Friedman J, Stone C.J, Olshen R.A: "Classification and Regression Trees", Pacific Grove, CA: Wadsworth, 1984.
13. [Brooks, 1975] Brooks F., *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley, 1975.
14. [Bruegge & Dutoit, 1996] Bruegge B, Dutoit A.H: "Software Metrics for Distributed Development", Carnegie-Mellon University. Department of Computer Science, 1996.
15. [Bruegge & Dutoit, 2004] Bruegge B, Dutoit A: "Object-oriented software engineering using UML, Patterns and Java", Pearson Education, 2004.
16. [Cadle & Yeates, 2008] Cadle J, Yeates D: "Project Management for Information Systems", Pearson Prentice Hall, 5<sup>th</sup> Edition, 2008.
17. [Cockburn, 2004] Cockburn A., „Crystal Clear: A Human-Powered Methodology for Small Teams“, Addison-Wesley, 2004, ISBN 0-201-69947-8.
19. [Coulouris et al, 2011] Coulouris, G., Dollimore J, Kindberg T, Blair G (2011). „Distributed Systems: Concepts and Design (5th Edition)“, Addison-Wesley ISBN 0-132-14301-1.
20. [Devedzic, 2002] V. Devedzic, "Software Project Management", in: Chang, S.K. (ed.), "Handbook of Software Engineering and Knowledge Engineering Vol.2 – Emerging Technologies", World Scientific Publishing Co., Singapore, 2002, pp. 419-446.
21. [Dulović, 1991] Dulović Lj: „Osnovi informaciologije i informacione tehnologije“, Institut za nuklearne nauke Vinča, Beograd, 1991.
22. [Elmasri & Navathe, 2000] Elmasri R, Navathe S. B. (2000), *Fundamentals of Database Systems (3rd ed.)*, Addison-Wesley, ISBN 0-201-54263-3.
23. [Elmasri & Navathe, 2007] Elmasri R, Navathe S: "Fundamentals of database systems", Addison Wesley, 2007.
24. [Fairley, 2009] Fairley R: "Managing and Leading Software Projects", Wiley, 2009
25. [Fenton & Pfleeger, 1997] Fenton N. E., Pfleeger S. L., "Software Metrics: A Rigorous and Practical Approach," 2nd Edition Revised ed. Boston: PWS Publishing, 1997.
26. [Ghosh, 2007] Ghosh, S (2007), „Distributed Systems – An Algorithmic Approach“, Chapman & Hall/CRC, ISBN 978-1-58488-564-1.
27. [Gibson & Cohen, 2003] Gibson C.B, Cohen S.G.(Eds.): *Virtual Teams That Work, Creating Conditions for Virtual Team Effectiveness*, Jossey-Bass, Willey, 2003.

28. [Goodchild et al, 2004] Goodchild M.F., Johnston D.M, Maguire D.J, Noronha V.T.: "Distributed and Mobile Computing", Book Chapter In: McMaster, R.B. & Usery, E.L. (Eds.) A Research Agenda for Geographic Information Science, Boca Raton: CRC Press, pp. 257–286.
29. [Hatley et al, 2000] Hatley D., Hruschka P., and Pirbhai I. A., "Process for system architecture and requirements engineering", Dorset House Publishing Co., Inc., 2000.
30. [Hauc, 1975] Хауц А: Управљање пројектима, Информатор, Загреб, 1975.
31. [Hawryszkiewicz, 2001] Hawryszkiewicz, I: "Introduction to systems analysis and design", Prentice Hall, Pearson Education, 2001.
32. [Jacic et al] Jacić Lj. A., Nikolić G., Rančić M.V., Debeljković D. Lj.: „Osnovi automatskog upravljanja i regulisanja", GIP "Kultura", Beograd
33. [Jacobson et al, 1999] Jacobson I., Booch G., Rumbaugh J., "The Unified Software Development Process", Addison-Wesley, 1999.
34. [Jauković, 1992] Јауковић М: „Увод у информационе системе", Техничка књига, Београд, 1992.
35. [Jones, 2007] Jones C., "Estimating software costs: Bringing realism to estimating", 2nd ed. New York: NY: McGraw-Hill, (2007).
36. [Jovanović, 2006] Јовановић П: „Управљање пројектом", Факултет организационих наука, Београд, 2006.
37. [Kamran&Chignell, 1993] Kamran P, Chignell M. Intelligent Database Tools and Applications: Hyperinformation access, data quality, visualization, automatic discovery. John Wiley & Sons, Inc., 1993.
38. [Kerzner, 1989] Kerzner, H. "Project Management", Third Edition, New York: Van Nostrand Reinhold, 1989
39. [Keyes, 2005] Keyes J: "Implementing the IT balanced scorecard: Aligning IT with organizational strategy", Auerbach Publications, 2005
40. [Keyes, 2011] Keyes J: "Implementing the Project Management Balanced Scorecard", Taylor and Francis Group LLC, 2011.
41. [Krsmanović& Mandić, 1995] Krsmanović S, Mandić D.P.: „Menadžment informacionih sistema", Univerzitet Braća Karić, Beograd, 1995.
42. [Lanubile, 2009] Lanubile F: "Collaboration in Distributed Software Development", In (A. De Lucia and F. Ferrucci Eds.): ISSSE 2006–2008, Lecture Notes in Computer Science LNCS 5413, pp. 174–193, 2009, Springer-Verlag Berlin Heidelberg 2009
43. [Larson, 1969] Larson, H.J.: Introduction to Probability Theory and Statistical Inference. Wiley International Edition, 1969.
44. [Lazarević et al, 1988] Лазаревић Б, Јовановић В, Вучковић М: "Пројектовање информационих система I део", Научна књига, Београд 1988.
45. [Lehman, 1978a] Lehman M.M: „Programs, Cities, Students, Limits to Growth?", Book Chapter in Gries D (ed): "Programming Methodology – A Collection of Articles by Members of IFIP WG2.3", Series Texts and Monographs in Computer Science, Springer, Verlag, 1978, pp. 42 – 62
46. [Lynch, 1996] Lynch N. A. (1996), Distributed Algorithms, Morgan Kaufmann, ISBN 1-55860-348-4.
47. [Marchewka, 2006] Marchewka J.T, Information Technology Project Management, John Wiley & Sons, 2006.
48. [Martin, 1991] Martin J, "Rapid Application Development", MacMillan Publishing Co., Inc., 1991.
49. [Martin, 2002] Martin R.C.: "Agile Software Development: Principles, Patterns and Practices." Pearson Education. (2002)
50. [Mogin et al, 2000] Mogin P., Lukovic I., Govedarica M.: "Principi projektovanja baza podataka", Fakultet tehničkih nauka, Novi Sad, 2000.
51. [Nimmer, 1997] Nimmer R.T, "Legal Issues in Open Source and Free Software Distribution", a book chapter from The Law of Computer Technology (1997, 2005 Supp.)
52. [Palmer& Felsing, 2002] Palmer S.R., Felsing J.M., „A Practical Guide to Feature-driven Development", Prentice Hall, Upper Saddle River, NJ, 2002, ISBN 0-13-067615-2.
54. [Peleg, 2000] Peleg D (2000), Distributed Computing: A Locality-Sensitive Approach, SIAM, ISBN 0-89871-464-8.
55. [Pfleeger& Atlee, 2003] Pfleeger S. L., Atlee J. M.: "Software Engineering – Theory and Practice", Prentice Hall, 3rd edition, 2003.
56. [Pigoski, 1996] Pigoski T. M., "Practical Software Maintenance: Best Practices for Managing Your Software Investment", John Wiley & Sons, Inc., 1996.
57. [Pilato et al, 2004] Pilato C.M, Ben Collins-Sussman B, Fitzpatrick B.W. "Version Control with Subversion", O'Reilly, 2004.
58. [Poppendieck&Poppendieck, 2003] M. Poppendieck, T. Poppendieck, „Lean Software Development – An Agile Toolkit for Software Development Managers", Addison-Wesley, Boston, 2003, ISBN 0-321-15078-3.

60. [Pressman, 2007] Pressman R. S., „Software Engineering: A Practitioner’s Approach“, McGraw-Hill, 6th edition, 2007.
61. [Radovan, 1989] Radovan M: “Projektiranje informacijskih sistema“, Informator, Zagreb, 1989.
62. [Raymond, 1999] Raymond E.: “The Cathedral & the Bazaar: Musings on linux and open source by an accidental revolutionary" O'Reilly Media, Inc.", 2001.
63. [Riel, 1996] Riel A.J: Object-oriented Design Heuristics, Addison-Wesley Longman, 1996.
64. [Royce, 1998] Royce W: Software Project Management, a Unified Framework, Addison Wesley, 1998.
65. [Schwaber& Beedle, 2001] Schwaber K., Beedle M., „Agile Software Development with Scrum“, Prentice Hall, Upper Saddle River, 2001.
66. [Siegelaub, 2004] Siegelaub J. M: How PRINCE2 can complement PMBOK and your PMP, PMI/Westchester chapter, 2004.
67. [Solingen& Berghout, 1999] van Solingen R, Berghout E., “The Goal/Question/Metric Method: a practical guide for quality improvement of software development“, McGraw-Hill Publishing Company, Berkshire, UK, 1999.
68. [Sommerville& Kotonya, 1998] Sommerville I., Kotonya G: “Requirements Engineering: Processes and Techniques“, John Wiley & Sons, Inc., 1998.
69. [Štankić, 2005] Станкић Р: “Пословна информатика“, Економски факултет, Београд, 2005.
70. [Šoti, 1978] Шоти, Ф: „Увод у кибернетику“, Раднички универзитет „Радивој Ћирпанов“, Нови Сад, 1978.
71. [Stanojević, 1986] Станојевић, М: „Основи пројектовања информационих система“, Научна књига Београд, 1986.
72. [Stapleton, 2003] Stapleton J., „DSDM: Business Focused Development“, second ed., Pearson Education, 2003, ISBN 978-0321112248.
73. [Stellman& Green, 2005] Stellman A, Green J: “Applied Software Project Management“, O’Reilly Media, 2005.
74. [Turban et al, 2013] Turban E, Volonino L, Wood G.R.: “Information Technology for Management: Advancing Sustainable, Profitable Business Growth“, Willey, 2013.
75. [Veljović, 2003] Вељовић, А : „Пројектовање информационих система“, Компјутер библиотека, Чачак, 2003.
76. [Wysocki, 2006] Wysocki, R „Effective Software Project Management“, Willey Publishing, Inc, 2006

### **8.1.3. Техничке публикације, технички и научни извештаји**

1. [Agile PrepCast, 2013] “Comparison of Agile Methods by The Agile PrepCast“, OSP International LLC (PMI Registered Education Provider), 2013
2. [Alleman, 2002] Alleman G.B: Is There an Underlying Theory of Software Project Management? (A critique of the transformational and normative views of project management), Niwot, Colorado, 2002
3. [Ambler&Lines, 2012] Ambler S.W., Lines M., “Disciplined Agile Delivery: A Practitioner’s Guide to Agile Software Development in the Enterprise“, IBM Press, 2012.
4. [Ambler, 2013] Ambler S. W.: “Going Beyond Scrum, Disciplined Agile Delivery“, Disciplined Agile Consortium, White Paper Series, October 2013.
5. [Atkinson, 1997] Atkinson RW. „Effective Organisations, Re-framing the Thinking for Information Systems Projects Success“, 13±16. Cassell, London, 1997.
6. [Bertelsmann, 2002] Bertlesman Foundation (Schmidt O., Friedrichs S., Hart T Eds.): “Balanced e-Government, E-Government – Connecting Efficient Administration and Responsive Democracy“, Bertlesman Foundation, 2002
7. [Booch&Brown, 2002] Booch G, Brown A: Collaborative Development Environments, Rational Software Corporation, 2002
8. [CCTA, 1999] CCTA: Managing Successful Projects with PRINCE2, Key Skills Limited, 1999.
9. [Cederqvist, 2006] Cederqvist P et al. „Version Management with CVS“, Network Theory Limited, Bristol, 2006.
10. [CHAOS, 1995] CHAOS Report, Standish Group, 1995.
11. [Craig& Jaskiel, 2002] Craig R. D., Jaskiel S. P., „Systematic Software Testing“, Artech House, Inc., 2002.
12. [Demir& Osmundson, 2008] Demir K. A., Osmundson J.S., “A Theory of Software Project Management and PROMOL: A Project Management Modeling Language“, Technical Report, NPS-IS-08-006, Naval Postgraduate School, Monterey, CA, USA, March 2008.
13. [Eckerson, 2009] Eckerson W.W.: “Performance management strategies - How to Create and Deploy Effective Metrics“, TDWI (The Data Warehousing Institute), 2009

14. [Economist, 2009] The Economist Intelligence Unit Limited , "Managing virtual teams", A report from the Economist Intelligence Unit Taking a more strategic approach, 2009
15. [El Emam, 2000] El Emam K: "A methodology for validating software product metrics", National Research Council Canada, Jun 2000.
16. [Fan&Gauch, 1999] Fan Y, Gauch S: "Adaptive Agents for Information Gathering from Multiple, Distributed Information Sources", AAAI Technical Report SS-99-03, Association for the advancement of artificial intelligence, 1999
17. [Green et al, 1997] Green S, Hurst L, Nangle B, Cunningham P: "Software Agents: A review", Dublin, Trinity College Dublin, Department of Computer Science, TCD-CS-1997-06, 1997, pp 51.
18. [Haban& Wybraniec, 1989] Haban D, Wybraniec D: „Behaviour and Performance Analysis of Distributed Systems Using a Hybrid Monitor“, International Computer Science Institute, Tech. Rep. TR-89-029, Berkeley, CA (1989)
19. [Hill& Thomson, 2001] Hill J, Thomson K: „System Performance Management Moving from Chaos to Value“, SUN Microsystems, 2001.
20. [ITPM, 2001] Information Technology Performance Management, Measuring IT's Contribution to Mission Results, A Case study of the applied information economics methodology for an infrastructure it investment, September 2001, IT Performance Management Subcommittee for the Capital Planning and IT management Committee of the Federal Chief Information Officers Council
21. [ITPM, 2002] "Lessons Learned on Information Technology Performance Management: Applying the Balanced Scorecard and Applied Information Economics to Federal Information Technology Initiatives", Federal Chief Information Officers Council, Community of Practice for IT Performance Management, 2002
22. [Judson et al, 2013] Judson D.H, Parker J.D, Larsen M.D: "Adjusting sample weights for linkage-eligibility using SUDAAN", National Center for Health Statistics, Hyattsville Maryland. May 2013
23. [Laddaga, 1997] Laddaga, R. 1997. "Self-adaptive software". Tech. Rep. 98-12, DARPA BAA.
24. [Lehman, 1969] Lehman M M, "The Programming Process", IBM Res. Rep. RC 2722, IBM Res. Centre, Yorktown Heights, NY 10594, Sept. 1969
25. [Kreger, 2011] Kreger H: "Web Services Conceptual Architecture", IBM Software Group, May 2001
26. [Monperrus et al, 2008] Monperrus M, Jezequel J-M, Champeau J, Hoeltzener B: "Measuring models", In Rech J, Bunse C (eds.) Model-Driven Software Development: Integrating Quality Assurance, IDEA Group, 2008
27. [Moore&Barnett, 2004] Moore, S, Barnett L.: "Offshore Outsourcing and Agile Development," Forrester Research, September 2004
28. [Naab, 2012] Naab C: Business Rules Management with SAP, Active Global Support, SAP America, 2012.
29. [Roper&Millar 1999] Roper M; Millar L (Eds.): "Understanding Computers: An overview for Records and Archives Staff", International Records Management Trust, 1999.
30. [Say JAL, 2008] Say JAL International Consulting: "IT Project Management Benchmarking", November 12, 2008
31. [Schmidt, 1982] Schmidt E.E: "Controlling Large Software Development in a Distributed Environment", Xerox Corporation, Palo Alto Research Centers, December 1982
32. [Siles, 2004] Siles R: "Project Management Information Systems – Guidelines for Planning, Implementing and Managing a DME Project Information System", CARE, 2004
33. [Tsui&Jeng, 2009] Tsui T-T, Jeng A.B: Challenges and Solutions of Distributed Systems Composition, Telecom Technology Center, 2009.
34. [Wolfson& Kalpakis, 1997] Wolfson O, Kalpakis K: „Adaptive Data Replication in Distributed Systems“, Project Summary, 1997-2001.
35. [Yap, 2010] Yap M: "Successful Distributed Agile Team Working Patterns", White paper, SolutionsIQ, 2010.
36. [Yeong, 2007] Yeong A: The marriage proposal of PRINCE2 and PMBOK, 2007.

#### **8.1.4. Наставни материјали**

1. [Kaidi, 2000] Kaidi Z: „Data visualization“, School of Computing, National University of Singapore, 2000.
2. [Krzyzanowski, 2000] Krzyzanowski P: A taxonomy of distributed systems, Rutgers University, 2000/2003.
3. [Magnusson et al, 2003] Magnusson K., Kroslid D., Bergman B., Six Sigma – The Pragmatic Approach, Studentlitteratur, Lund, Sweden, 2003.

- [Redmiles et al, 2007] Redmiles D, van der Hoek A, Al-Ani B, Hildenbrand T, Quirk S, Sarma A, Filho R.S.S., de Souza C, Trainer E: Continuous Coordination: A New Paradigm to Support Globally Distributed Software Development Projects, Lehrstuhl für ABWL und Wirtschaftsinformatik, Universität Mannheim, D-68131, Mannheim, Germany, 2007
- [Steen, 2012] van Steen M: "Distributed Systems Principles and Paradigms", Vrije University Amsterdam, Dept. Computer Science, 2012.

### 8.1.5. Докторске дисертације

- [Bay, 2008] Bay T.G.B, "Hosting distributed software projects: concepts, framework and the Origo experience", PhD Dissertation, ETH Zurich, 2008
- [Bani-Salameh, 2011] Bani-Salameh H.A., "A Social Collaborative Distributed Software Development Environment", Phd Thesis, College of Graduate Studies, University of Idaho, 2011.
- [Castro Souto, 2010] Castro Souto L.M: "On the Development Life Cycle of Distributed Functional Applications: A Case Study", PhD Thesis, Universidade Da Coruna, 2010.
- [Cataldo,2007] Cataldo M: Dependencies in Geographically Distributed Software development: Overcoming the Limits of Modularity, PhD Thesis, School of Computer Science, Institute for Software Research, Carnegie Mellon University, Pittsburgh, PA, 2007.
- [Damm, 2002] Damm C.H: Collaborative Software Development - Design and Implementation of Tools for Object-Oriented Modelling, PhD Dissertation, Department of Computer Science, University of Aarhus, Denmark, 2002.
- [Grinter, 1996] Grinter R.E: "Understanding Dependencies: A Study of the Coordination Challenges in Software Development", PhD Thesis, University of California, Irvine, 1996.
- [Jevtic, 2010] Јевтић В: „Развој модела за одређивање трајања пројекта на основу Кларкових једначина“, Технички факултет „Михајло Пупин“, Зрењанин, 2010.
- [Kazi, 2014] Кази З: „Онтолошки заснована анализа семантичке коректности модела података применом система аутоматског резоновања“, Докторска дисертација, Универзитет у Новом Саду, Технички факултет „Михајло Пупин“, Зрењанин, 2014.
- [Kotlarsky, 2005] Kotlarsky J: Management of Globally Distributed Component-Based Software Development Projects, PhD Thesis, Erasmus University Rotterdam, 2005.
- [Nassif, 2012] Nassif A.B: "Software Size and Effort Estimation from Use Case Diagrams Using Regression and Soft Computing Models" (2012). University of Western Ontario, PhD Thesis
- [Lal, 2011] Lal R: "Strategic factors in agile software development method adaptation: A study of Market-Driven Organisations", Massey University, Albany campus, New Zealand, PhD Thesis, 2011.
- [Nguyen, 2006] Nguyen Q.M: Planning in Software Project Management – an Empirical Research of Software Companies in Vietnam, PhD Thesis, Faculty of Economics and Social Sciences, Fribourg, Switzerland
- [Moser, 1996] Moser S: "Measurement and Estimation of Software and Software Processes", PhD Thesis, Institut für Informatik und angewandte Mathematik, Bern, 1996.
- [Ressmann, 2003] Ressmann D: „Performance Monitoring in Service-Based Computing“, PhD Thesis, De Montfort University, 2003.
- [Zafar, 2008] Zafar B: "Conceptual Modelling of Adaptive Web Services based on High-level Petri Nets", PhD Thesis, De Montfort University, 2008.

### 8.1.6. Пројекти

- [DAISY] Azuma M, Nagasaki H, Nonaka M: "DAISY: Distributed and Adaptive Information Systems A Framework and a Mechanism for 21's IS and Software Engineering" ([http://www.researchgate.net/publication/253109693\\_DAISY\\_Distributed\\_and\\_Adaptive\\_Information\\_Systems\\_A\\_Framework\\_and\\_a\\_Mechanism\\_for\\_21%27s\\_IS\\_and\\_Software\\_Engineering](http://www.researchgate.net/publication/253109693_DAISY_Distributed_and_Adaptive_Information_Systems_A_Framework_and_a_Mechanism_for_21%27s_IS_and_Software_Engineering))
- [OPHELIA] Dewar, R. G., MacKinnon, L. M., Pooley, R. J., Smith, A. D., Smith, M. J., Wilcox, P. A. (2002). The OPHELIA Project: Supporting Software Development in a Distributed Environment. ICWI, pp. 568-571.
- [U-QASAR, 2014] Пројекат U-QASAR „Measuring, monitoring and visualizing software quality“, FP7-ICT-2011-8.Project #: 318082, План конференције пројекта, Aalto University(Finland), February11th, 2014.

### 8.1.7. Стандарди, манифесто, збирке стандарда

1. [AgileManifesto, 2001] Agile Manifesto (2001) <http://www.agilemanifesto.org/> (accessed June 2009)
2. [ANSI/IEEE 1471-2000] ANSI/IEEE Std 1471-2000, Recommended Practice for architectural description of Software-Intensive systems, 2000.
3. [BPM, 2004] BPM Standards Group (2004). Business Performance Management: Industry Framework Document. BPM Framework Document, 27p (retrieved on April 14, 2005, from <http://www.bpmstandardsgroup.org/documents/BPMFrameworkDraft-V4.0092904.pdf>).
4. [CHAOS, 2013] CHAOS Manifesto 2013, Standish Group, 2013.
5. [CMM-S, 1993] Paulk M.C, Curtis B, Mary Beth Chrissis M.B, Weber C.V: Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, February 1993
6. [COSMIC] ISO/IEC 19761:2003 „Software Engineering – COSMIC – FFP – A functional size measurement method“
7. [IEEE SE Glossary 729-1983] IEEE, „IEEE Standard Glossary of Software Engineering Terminology“, 1983.
8. [IEEE SE Glossary, 1990] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," IEEE Computer Society, IEEE Std 610.12-1990 (Revision and redesignation of IEEE Std 729-1983), 1990.
9. [IEEE QMetrics 1061-1998] IEEE, "IEEE Std. 1061-1998, Standard for a Software Quality Metrics Methodology, revision." Piscataway, NJ,: IEEE Standards Dept., 1998.
10. [ISO/IEC 15504] ISO/IEC 15504 Information Technology –Process Assessment
11. [ISO/IEC 15939:2007] ISO/IEC 15939:2007 Systems and Software Engineering – Measurement Process.
12. [ISO/IEC 9126] International Organization for Standardization and the International Electrotechnical Commission:"ISO/IEC 9126: Information Technology - Software Product Evaluation – Quality Characteristics and Guidelines for their Use," 1991.
13. [PMBOK, 2004] Project Management Body of Knowledge, Project Management Institute, Upper Darby, 2004.
14. [PMBOK, 2013] A Guide to the Project Management Body of Knowledge PMBOK 5th edition (2013). International Project Management Association IPMA
15. [SWEBOK, 2014] A Guide to the Software Engineering Body of Knowledge SWEBOK v.3 (2014). IEEE Computer Society
16. [UNSPSC] United Nations Standard Products and Services Code, <http://www.unspsc.org/>

### 8.1.8. Web садржаји

Сви web садржаји су доступни. Приступ сваком од садржаја је рађен више пута, али је верификација доступности сваке од наведених адреса реализована 2.9.2015. године.

1. [UNSPSC-cc] United Nations Standard Products and Services Code, Commodity Codes [http://www.doa.la.gov/osp/vendorcenter/docs/unspsc\\_commoditycodes.pdf](http://www.doa.la.gov/osp/vendorcenter/docs/unspsc_commoditycodes.pdf)
2. [UNSPSC-sw] United Nations Standard Products and Services Code, категорије и позиција софтвера, <http://www.dfa.cornell.edu/procurement/buying/unspsc.cfm>
3. [w1] Дефиниција софтвера, <http://www.merriam-webster.com/dictionary/software>
4. [w2] Дефиниција рачунарског софтвера, <http://www.thefreedictionary.com/computer+software>
5. [Google Code] <https://code.google.com>
6. [Git Hub] <https://github.com>
7. [Launch Pad] <https://launchpad.net>
8. [Agitar] Agitar Technologies: Measuring Risk to Improve Java Software Quality, White Paper, <http://www.agitar.com/pdf/Paper-Agitar-Measuring-Risk-to-Improve-Java-Quality.pdf>
9. [Agile SD Alliance] Agile Software Development Alliance, [www.agileAlliance.org](http://www.agileAlliance.org)
10. [Highsmith, 2006] Highsmith J: "An Adaptive Performance Management System", Cutter Consortium, Agile Project Management, Executive Summary, Vol. 7, No. 6, <http://www.infoq.com/articles/Adaptive-Performance-Management>
11. [Intaver APM] Intaver Institute Inc.: Adaptive Project Management, [http://www.intaver.com/Articles/Article\\_AdaptiveProjectManagement.pdf](http://www.intaver.com/Articles/Article_AdaptiveProjectManagement.pdf)
12. [DeBloch] Stefan DeBloch: Distributed Information System Architecture, Technische Universität Kaiserslautern, <http://www.uni-kl.de/>
13. [Ward, 2009] Ward M: "Overview of Data Visualization." URL <http://web.cs.wpi.edu/~matt/courses/cs563/talks/datavis.html> (2009).

14. [wGershon] Gershon M: Choosing Which Process Improvement Methodology to Implement, <http://t.www.na-businesspress.com/JABE/Jabe105/GershonWeb.pdf>
15. [Kan, 2009] Stephen H. Kan: Measuring and Managing In-process Software Quality, <http://rube.asq.org/software/2009/07/continuous-improvement/measuring-and-managing-in-process-software-quality.pdf>
16. [w Desharnais] Desharnais J-M: "Software Measurement - Analysis of ISO/IEC 9126 and 25010", <http://www.testingnotes.eu/wp-content/uploads/9126-vs-25010.pdf>
17. [PRINCE 2 templates] Department for Business innovation and skills: *PRINCE 2 templates*, <http://www.berr.gov.uk/aboutus/corporate/projectcentre/pm-templates/page12526.html>
18. [wThomsen] Thomsen C: "Project Management Information Systems", Construction Management Association of America, <http://www.e-builder.net/white-papers/project-management-information-systems>
19. [activecollab] <https://www.activecollab.com/>
20. [basecamp] <https://basecamp.com/>
21. [dotproject] [www.dotproject.net/](http://www.dotproject.net/)
22. [gforge] <https://gforge.com/>
23. [gforge inria] <https://gforge.inria.fr/>
24. [gforge group] <https://gforgegroup.com/>
25. [jira] <https://www.atlassian.com/software/jira>
26. [redmine] [www.redmine.org/](http://www.redmine.org/)
27. [easyredmine] [www.easyredmine.com](http://www.easyredmine.com)
28. [tfs vstudio] <https://msdn.microsoft.com/en-us/vstudio/ff637362.aspx>
29. [tfs overview] <https://www.visualstudio.com/en-us/products/tfs-overview-vs.aspx>
30. [teamviewer] <https://www.teamviewer.com/sr/index.aspx>
31. [projectmanager] [www.projectmanager.com](http://www.projectmanager.com)
32. [JIRAwM] <http://blogs.atlassian.com/2013/06/using-watchers-and-mentions-effectively/>
33. [MSXQueryDownload] <http://thedownloadtohavefun.us/best-apps-for-mac/microsoftxmlquerydll-download.html>
34. [NuSOAPDownload] <http://sourceforge.net/projects/nusoap/>
35. [DinamASPNETWS] <http://www.c-sharpcorner.com/UploadFile/f9935e/invoking-a-web-service-dynamically-using-system-net-and-soap/>

## 8.2. СПИСАК ОБЈАВЉЕНИХ РАДОВА КАНДИДАТА ИЗ ОБЛАСТИ ИСТРАЖИВАЊА И ЦИТИРАНОСТ РАДОВА

У наставку су приказани радови који су цитирани у тексту докторске дисертације, а односе се на радове у часописима и саопштења са конференција, као и уџбенике у којима је представљен део резултата истраживања кандидата Мр Љубице Кази из области истраживања ове докторске дисертације. Такође, у овом одељку дат је и списак радова других аутора, који су цитирали објављене радове кандидата, а који представљају резултате ове докторске дисертације.

### РАД У МЕЂУНАРОДНОМ НАУЧНОМ ЧАСОПИСУ (М23)

1. [Kazi et al, Metalurgia, 2012] Љубица Кази, Золтан Кази, Биљана Радуловић, Душко Летић, Madhusudan Bhatt, „Applying integration of conceptual data modelling methods within information system development: A case study”, Metalurgia International Journal 2012, ISSN: 1582-2214, Vol. XVII, No. 6, pp. 67-75 (IF 2012 = 0.134)
2. [Kazi et al, ТТЕМ, 2012] Золтан Кази, Љубица Кази, Радуловић Биљана, „Analysis of data model correctness by using automated reasoning system”, Technics Technologies Education Management-ТТЕМ, (2012), ISSN 1840-1503, vol. 7 no. 3, pp. 1090-1100, (IF 2012 = 0.414)
3. [Kazi et al, АРАВ, 2016] Золтан Кази, Љубица Кази, Биљана Радуловић, Madhusudan Bhatt: „Ontology-based system for Conceptual Data Model Evaluation”, International Arab Journal of Information Technology, Vol 13, No 5, September 2016. (IF 2014 = 0.582)

### РАД У НАУЧНОМ ЧАСОПИСУ (М53)

1. [Kazi et al, Tibiscus, 2008] Љубица Кази, Радуловић Биљана, Кази Золтан: »Predicate Logic Baselog - A Business Rules Management System”, Annals. Economic Science Series, Faculty of Economics, Tibiscus University, Timisoara, Romania, Editura Eurostampa, Romania, I.S.S.N. 1582 - 6333, Vol.XIV/2008, pp. 147-157, напомена: рад је у целости публикован као изабран са међународне конференције»Sustainable development of Romania and its Convergence to the EU« 2008.
2. [Kazi et al, YUPMA, 2011] Љубица Кази, Ofelia Stanciu, Madhusudan Bhatt: “Distribution of Business Rules in Software project management”, Serbian project management Journal, ISSN 2217- 7256 (Online), Vol 1, Issue 2, pp. 54-59, Decembar 2011, Serbian Project Management association - YUPMA, Beograd, напомена: рад је у целости публикован као изабран са међународне конференције „Пројектни менаџмент у Србији – Успеси и Могућности» 2011.
3. [Kazi et al, e-Society, 2011] Љубица Кази, Биљана Радуловић, Драгица Ивин, Madhusudan Bhatt, Shiv Gheeya: “Communications In Distributed Software Project Management”, Journal e-Society Research and Applications, ISSN 2217-3269, vol.2 No 2, pp: 122-130, November 2011, Technical faculty "Mihajlo Pupin" Zrenjanin, напомена: рад је у целости публикован као изабран са међународне конференције ICTSME 2011.
4. [Kazi et al, ЈЕМС, 2011] Љубица Кази, Драгица Радосав, Милан Николић, Narendra Chotaliya: “Balanced Scorecard Framework In Software Project Monitoring”, Journal of Engineering Management and competitiveness, Technical faculty "Mihajlo Pupin" Zrenjanin, Online ISSN: 2217-8147, Vol. 1, No. 1/2, 2011, pp. 51-56, напомена: рад је у целости публикован као изабран са међународне конференције ЕМС 2011.
5. [Kazi, Almanah, 2011] Љубица Кази: «Мерење перформанси софтверског пројекта – балансиране карте успеха и подршка одлучивању», Алманах Техничког факултета «Михајло Пупин» у Зрењанину, ISSN 2217-6063, Vol 2, No 1, 2011, pp. 172-179
6. [Kazi et al, ITRO, 2014] Љубица Кази, Биљана Радуловић, Миодраг Ивковић, Весна Макитан, Бранко Маркоски: “Web Application for Project Management Support in Information Systems Higher Education”, Journal “Information technologies and development of education” ITRO, ISSN 2217-7949, Vol 4, No 1, 2014, pp.43-49, напомена: рад представља проширену верзија рада са ITRO 2014 конференције

### САОПШТЕЊЕ СА МЕЂУНАРОДНОГ СКУПА ШТАМПАНО У ЦЕЛИНИ (М33)

1. [Kazi et al, eComm, 2006] Љубица Кази, Добриловић Далибор, Кази Золтан: “Design of distributed information systems for lung diseases hospital”, 7 – th European conference eCommLine 2006, Bucharest, Romania, September 18-19, 2006, ISBN: 973-88046-0-4, 978-973-88046-0-9
2. [Kazi et al, MIPRO, 2007] Золтан Кази, Хотомски Петар, Радуловић Биљана, Љубица Кази: "Automated reasoning systems and remote databases in distributed information systems", International convention on Information and Communication Technology,



- Electronics and Microelectronics, MIPRO 2007, секција CIS (computer intelligence systems), Опатија, Хрватска, мај 2007., ISBN , pp. 151-156
3. **[Kazi et al, ICDQM, 2011]** Љубица Кази, Золтан Кази, Биљана Радуловић, Драгица Радосав: «Евалуација модела у развоју информационих система», International Conference Dependability and quality management ICDQM 2011, Београд, ISBN 978-86-86355-05-8, pp.589-595
  4. **[Kazi et al, UrbanEco, 2011]** Љубица Кази, Amar Kansara, Shiv Charan Gheeya: "Green IT approach to distributed software project management", International conference Ecology of Urban Areas UrbanECO 2011, Ечка, ISBN 978-86-7672-145-0, pp. 535-544
  5. **[Kazi et al, ICTSME, 2011]** Љубица Кази, Биљана Радуловић, Ubedullah Phulphoto: «Business Rules Management in Distributed information system - risk analysis», Information and communication technologies for small and medium enterprises ICTSME 2011, Аранђеловац, ISBN 978-86-7672-140-5, pp. 30-35
  6. **[Kazi et al, ITRO, 2011]** Љубица Кази, Биљана Радуловић, Золтан Кази, Madhusudan Bhatt: «Balanced Scorecard for Student Teamwork Software Project Management», International conference Information technologies and development of education ITRO 2011, Зрењанин, ISBN 978-86-7672-134-4, pp.105-111
  7. **[Kazi&Radulovic, MIPRO, 2011]** Љубица Кази, Биљана Радуловић: «Information system based on balanced scorecard for Student Teamwork Software Project Management», International convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2011, 23-27.5.2011, Опатија, Хрватска, ISBN 978-1-4577-0996-8, pp. 1549 - 1554
  8. **[Kazi et al, SISY, 2012]** Љубица Кази, Биљана Радуловић, Золтан Кази: "Performance indicators in Software Project Monitoring: Balanced Scorecard Approach", IEEE International Symposium on Intelligent Systems and Informatics, SISY 2012, September 20-22, 2012, Subotica, Serbia, ISBN 978-1-4673-4750-1, pp. 19-25
  9. **[Kazi et al, MIPRO, 2012]** Љубица Кази, Золтан Кази, Биљана Радуловић: «Data Warehouse based evaluation of students' achievements in information systems education», International convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2012, Опатија, Croatia
  10. **[Bhatt et al, AIIT, 2012]** Madhusudan Bhatt, Жељко Стојанов, Љубица Кази: «Thinking of Maintenance During Software Development: A Preliminary Review», International conference on Applied Internet and Information Technologies ICAIIT2012, Zrenjanin, ISBN 978-86-7672-166-5, pp. 355-358
  11. **[Kazi et al, AIIT, 2012a]** Љубица Кази, Биљана Радуловић, Драгица Радосав, Madhusudan Bhatt, Наташа Грмуша, Немања Штиклица: "Business Process Model and Elements Of Software Design: The Mapping Approach", "International conference on Applied Internet and Information Technologies" ICAIIT2012, Zrenjanin, ISBN 978-86-7672-166-5, pp.17-20.
  12. **[Kazi et al, AIIT, 2012b]** Љубица Кази, Биљана Радуловић, Миодраг Ивковић, Бранко Маркоски, Дејан Лацмановић, Амар Кансара: "Development of java application for project management support in educational information system", "International conference on Applied Internet and Information Technologies" ICAIIT2012, Zrenjanin, ISBN 978-86-7672-166-5, pp.367-371
  13. **[Kazi et al, AIIT, 2012c]** Љубица Кази, Биљана Радуловић, Миодраг Ивковић, Madhusudan Bhatt, Ofelia Stanciu: „IT jobs market in Serbia – a preliminary analysis“, International conference on Applied Internet and Information Technologies" ICAIIT2012, Zrenjanin, ISBN 978-86-7672-166-5, pp.300-304.
  14. **[Kazi et al, MIPRO, 2014]** Љубица Кази, Биљана Радуловић, Ивана Берковић, Золтан Кази: „Integration of conceptual data Modeling Methods: Higher Education Experiences“, International convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2014, Опатија, Хрватска
  15. **[Kazi et al, AIIT, 2014]** Љубица Кази, Биљана Радуловић, Миодраг Ивковић, Весна Макитан, Бранко Маркоски: "Software Project Management in Distributed Environment: a Review" – Applied Internet and Information Technologies, AIIT2014, October 24, 2014, Zrenjanin
  16. **[Kazi et al, AIIT, 2014b]** Љубица Кази, Биљана Радуловић, Миодраг Ивковић, Бранко Маркоски, Драгана Глушац, Милан Павловић, Далибор Добриловић, Дејан Лацмановић, Владимир Каруовић: „Improving Information System of Higher Education Institution: a Case Study“, Applied Internet and Information Technologies, AIIT2014, October 24, 2014, Zrenjanin
  17. **[Sljapic et al, AIIT, 2014]** Шљапић Немања, Марковић Андреа, Љубица Кази, Биљана Радуловић: "Simulation of distributed information system in oil industry", International Conference Applied Internet and Information Technologies, AIIT2014, October 24, 2014, Zrenjanin

18. [Kazi et al, COMENG, 2014] Љубица Кази, Биљана Радуловић, Душко Летић, Драгица Радосав, Ивана Берковић: „Applying SE-PM matrix in analysis of research in distributed software development“, Global conference on Computer science, Software, Networks and Engineering, COMENG 2014, Antalya, Turska
19. [Kazi et al, ICIST, 2015] Љубица Кази, Миодраг Ивковић, Биљана Радуловић, Madhusudan Bhatt, Narendra Chotaliya: „The Role of Business Process Modeling in Information System Development with Disciplined Agile Delivery Approach“, ICIST 2015 5th International Conference on Information Society and Technology, Proceedings, pp. 455- 493.

#### **САОПШТЕЊЕ СА СКУПА НАЦИОНАЛНОГ ЗНАЧАЈА ШТАМПАНО У ЦЕЛИНИ (М63)**

1. [Kazi et al, YUINFO, 2011] Љубица Кази, Биљана Радуловић, Миодраг Богуновић, Татјана Орел: "Integration of balanced scorecard models for performance evaluation of municipality ICT department", YUINFO 2011, Кораоник
2. [Kazi et al, SYMOPIS, 2011] Љубица Кази, Биљана Радуловић, Душко Летић: «Подршка одлучивању у управљању софтверским пројектом», Симпозијум о операционим истраживањима SYMOPIS 2011, Златибор, ISBN 978- 86-403-1168-7, pp. 60-63

#### **УЏБЕНИЦИ**

1. [Radulovic et al, 2011] Биљана Радуловић, Љубица Кази, Золтан Кази: „Информациони системи – одабрана поглавља“, 2. поновљено издање, Технички факултет „Михајло Пупин“, Зрењанин, 2011, ISBN 978-86-7672-149-8
2. [Kazi et al, UZD, 2013] Љубица Кази, Золтан Кази, Биљана Радуловић: „Информациони системи 1 и Информациони системи 2 – практикум за вежбе“, Технички факултет „Михајло Пупин“, Зрењанин, 2013, ISBN 978-86-7672-215-0

#### **ЦИТИРАНОСТ ОБЈАВЉЕНИХ РАДОВА**

1. Putri D. P: „Analysis SIM Card Provisioning Process at Telecommunication Operator using Balance Scorecard“, 2014, International Journal of Engineering Research & Technology (IJERT), IJERT/IJERT, ISSN: 2278-0181, Vol. 3 Issue 3, March – 2014,  
**Цитиран рад: [Kazi et al, JEMC, 2011]**
2. Doraisamy, M, bin Ibrahim, S. , Mahrin, M.N: „Formulation of metric based software project performance monitoring model: A roadmap“, Open Systems (ICOS), 2014 IEEE Conference on, Date of Conference: 26-28 Oct. 2014,  
**Цитиран рад: [Kazi et al, SISY, 2012]**
3. Atiwithayaporn S, Rivepiboon W: „A Development of the Effectiveness Evaluation Model for Agile Software Development using the Balanced Scorecard“, Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong, ISBN: 978-988-19251-8-3,  
**Цитиран рад: [Kazi et al, SISY, 2012]**

## 9. ПРИЛОЗИ

### 9.1. Списак анализираних софтверских пројеката у уводном емпиријском истраживању

Табела 9.1.1. Списак софтверских пројеката и софтвера у емпиријском истраживању

РБ	Реализовани софтверски модул, софтвер, информациони систем	Софтверски пројекат
1	Клијент-сервер Windows апликација за евиденцију медицинских услуга Специјалне болнице за плућне болести »Др Васа Савић« Зрењанин Web апликација за размену и приказ медицинских слика	»е-Рентгенологија Специјалне болнице за плућне болести «Др Васа Савић Зрењанин», Министарство науке и заштите животне средине Републике Србије, број уговора: 62336, руководилац пројекта: проф. др Велимир Сотировић, проф. др Ивана Берковић 2004-2008. године
2	Образовни web ауторски систем са елементима учења на даљину Модул за тестирање у оквиру љуске система за учење на даљину «dLearn» SQL e-learning system	„Систем учења на даљину базиран на Internet технологијама уз коришћење мултимедијалних образовних софтвера“, Министарство за развој, науку и животну средину Републике Србије, шифра: IT.3212, руководилац пројекта: проф. др Велимир Сотировић, 2002 – 2004.
3	Програм за вођење евиденције инвентара и промета слика у Савременој галерији УК Ечка, Зрењанин	Истоимени стручни пројекат Техничког факултета «Михајло Пупин» Зрењанин у сарадњи са Савременом галеријом УК Ечка Зрењанин, руководилац: проф. др Драгица Радосав, 2001. - 2004. године
4	Званични Web site Општине Зрењанин (www.zrenjanin.org.yu)	Истоимени стручни пројекат Градске управе општине Зрењанин, са учесницима са Техничког факултета „Михајло Пупин“ у Зрењанину, руководилац пројекта: Миодраг Вогуновић, рачунски центар Општине Зрењанин, мај-октобар 2001.
5	Информациони систем Специјалне хируршке болнице „Свети Јован“ Зрењанин	Истоимени стручни пројекат у оквиру Специјалне хируршке болнице Свети Јован из Зрењанина, 2003-2005. године
6	Информациони систем Опште болнице „Свети Јован“ Зрењанин,	Истоимени стручни пројекат у оквиру Опште болнице „Свети Јован“ из Зрењанина, 2008-2009. године
7	Информациони систем гинеколошке ординације Др „Љиљана Бурсић“	Истоимени стручни пројекат у оквиру Гинеколошке ординације „Др Љиљана Бурсић“ из Зрењанина, 2003-2004. Године
8	Информациони систем гинеколошке ординације Др „Нада Хрњак Еремић“	Истоимени стручни пројекат у оквиру Гинеколошке ординације „Др Нада Хрњак Еремић“ из Зрењанина, 2004-2005. године
9	Прототип web апликације за пријављивање испита студената	Интерни пројекат Техничког факултета Михајло Пупин Зрењанин, 2009-2010. године под називом „Унапређење информационог система студентске

		службе”
10	Web сајт предшколске установе Зрењанин www.predskolskazr.edu.rs	Истоимени стручни пројекат у сарадњи са Предшколском установом Зрењанин, децембар 2014-јануар 2015.
11	Windows апликација за припрему података за потребе штампања диплома	Интерни пројекат Техничког факултета Михајло Пупин Зрењанин за припрему података за штампање диплома, октобар 2014-април 2015.

Табела 9.1.2. Радне улоге кандидата (Мр Љубице Кази) у оквиру реализације софтверских пројеката из емпиријске анализе

<b>Рб. Пр.</b>	<b>УЛОГА КАНДИДАТА У ОКВИРУ СОФТВЕРСКОГ ПРОЈЕКТА</b>
1	Секретар пројекта, Пројектант, Програмер
2	Програмер, Тестер, Докуменер (Technical Writer)
3	Пројектант, програмер
4	Програмер, Докуменер (Technical Writer)
5	Руководилац пројекта, пројектант, програмер, тестер
6	Руководилац пројекта, пројектант, програмер, тестер
7	Руководилац пројекта, пројектант, програмер, тестер
8	Руководилац пројекта, пројектант, програмер, тестер
9	Секретар пројекта, Пројектант, Програмер, Докуменер (Technical Writer)
10	Пројектант, програмер, тестер, докуменер (Technical Writer)
11	Пројектант, програмер, тестер, докуменер (Technical Writer)

## 9.2. Категоризација радова који анализирају појединачне проблеме дистрибуираног развоја софтвера

Табела 9.2.1. Приказ појединачних проблема у дистрибуираном развоју софтвера и одговарајућих референци радова који реализују истраживање наведеног проблема

КЛАСА ПРОБЛЕМА	ПРОБЛЕМ	ДСД референца	ВИРТУАЛНИ ТИМОВИ референца
Управљање софтверским пројектима	Дистрибуирано управљање софтверским пројектима	[Hande et al, 2012], [Sutherland et al, 2007], [Nidiffer & Dolan, 2005], [Raut, 2008], [Zanoni&Audi, 2003], [Lamersdorf & Munch, 2010]	[Casey & Richardson, 2006], [Beise, 2004]
	Перформансе пројекта	[Narayan & Balan, 2007]	
Управљање	Стратешки приступи		[Economist, 2009]
	Делегација лидерства, ефикасност лидерства		[Kayworth & Leidner, 2001] [Zhang et al, 2008]
	Управљање конфликтима		[Montoya-Weiss et al, 2001]
	Процена трошкова	[Narayan & Balan, 2012]	
	Управљање ризицима	[Prikladnicki et al, 2004] [Keshlaf & Riddle, 2010]	
	Артикулација рада	[Boden et al, 2008]	
	Алокација задатака	[Lamersdorf et al, 2009], [Lamersdorf& Münch, 2010], [Barcus & Montibeller, 2008]	
	Контрола	[Schmidt, 1982]	
	Управљање непрецизношћу (неизвесношћу)	[Smite & Borzovs, 2008]	
	Ефикасност и ефективност		[Maznevski & Chudoba, 2000], [Kimble et al, 2000], [Edwards & Sridhar, 2003], [Guzman et al, 2010]
Организациони облици	Програмирање у пару	[Stotts & Williams, 2002]	
	Виртуални тимови	[Gorton & Motwani, 1996]	[Warkentin et al, 1997], [Bell & Kozlowski, 2002], [Watson-Manheim et al 2002], [Andres, 2002], [Gibson & Cohen 2003], [Martins et al, 2004], [Ebrahim et al, 2009]
	Мала и средња предузећа	[Boden et al, 2007], [Boden et al, 2008], [Ojala& Tyrväinen, 2007], [Boden et al, 2010]	[Ebrahim et al, 2010]
	Велике мултинационалне компаније	Alcatel [Ebert&De Neve, 2001], IBM [Sengupta, 2006], Philips [Kommeren and Parviainen, 2007]	
Психолошки аспекти и тимски	Групна свесност	[Gutwin et al, 2004], [Omoroniya et al, 2010]	

Љубица Кази - Докторска дисертација

рад	Друштвене везе	[Kotlarsky& Oshri, 2005], [Cataldo& Herbsleb, 2008], [Oshri et al, 2008]	
	Фамилијарност	[Espinosa et al, 2002], [Espinosa et al, 2007a]	
	Компјутерски подржан тимски рад	[Vick, 1998]	
	Ментални модели	[Espinosa et al, 2001], [Levesque et al, 2001], [Espinosa et al, 2002]	
	Когнитивна дивергенција	[Levesque et al, 2001]	
	Поверење	[Tjørnehøj et al, 2012]	[Jarvenpaa et al, 1998], [Gallivan, 2001], [Jarvenpaa & Leider, 2006], [Moe & Šmite, 2008], [Casey, 2010]
	Прилагођавање тимском раду	[Hall et all, 2005]	
	Колаборација	[Lanubile, 2009], [Rocha et al, 2011]	
	Улога вође	[Shirani, 2008]	
	Различитост индивида		[Beise, 2004]
Управљање знањем	Управљање знањем	[Boden et al, 2009]	[Bowen &Maurer, 2002]
	Дељење знања	[Sowe et al, 2009], [Maalej& Happel, 2008], [Espinosa et al, 2007b]	[Yoo& Kanawattanachai, 2001]
	Координација експертизе	[Faraj& Sproull, 2000]	
	Ток размене знања	[Zhuge, 2002]	
	Путујући тимови знања	[Dutoit et al, 2001]	
	Оперативно и стратегијско учење	[Boden et al, 2010]	
Комуникације	Комуникације	[Korkala&Abrahamsson, 2007]	[Suchan& Hayzak, 2001]
	Комуникационе везе	[Cataldo& Herbsleb, 2008]	
	Удаљеност	[Herbsleb et al, 2001]	
	Брзина	[Herbsleb et al, 2001]	
	Размена задатака	[Sandusky & Gasser, 2005]	
	Језик комуникације	[Vreede et al, 2006]	
	Координација	[Espinosa et al, 2001], [Espinosa et al, 2002], [Amrit & Hillegersberg, 2008], [Boden et al, 2007], [Espinosa et al, 2007b], [Mockus& Herbsleb, 2002], [Sandusky & Gasser, 2005], [Taxen, 2006], [Bendeck et al, 1998]	[Espinosa&Carmel, 2004]
	Преговарање	[Sandusky & Gasser, 2005]	
	Временске зоне	[Nordio et al, 2011], [Espinosa&Carmel, 2004]	
	Координација развојног и тест	[Grechanik et al, 2010]	

	тима		
Култура	Конвергенција	[Jensen & Bjørn, 2012]	
	Култура и знање	[Boden et al, 2009]	
	Културне разлике		[Dafoulas&Macaulay, 2001], [Olson&Olson, 2003]
	Утицај идеологије		[Stewart & Gosain, 2006]
Квалитет	Квалитет софтвера	[Bird et al, 2009]	
	Перформансе успеха	[Narayan & Balan 2007]	
	Метрике успеха	[Bourgault et al, 2002]	
	Метрике комуникације	[Dutoit & Bruegge, 1998]	
Методологија развоја софтвера	Спецификација захтева	[Khan et al, 2014]	
	Агилни приступ	[Balasubramaniam et al, 2006], [Phalnikar et al, 2008], [Lee & Yong, 2009], [Shrivastava & Date, 2010], [Yap, 2010], [Alqahtani et al 2013], [Kontio et al, 2004]	
	SCRUM	[Sutherland et al, 2007], [Nuevo et al, 2011]	
	Агилни vs. структурни приступи	[Estler et al, 2012]	
	Покривеност фаза развоја софтвера	[Rocha et al, 2011]	
Правни аспекти	Open source & free software	[Coleman, 2009] [Nimmer, 1997]	
Области примене	Колаборативни дизајн	[Herbsleb et al, 1995], [Kvan 2000]	
	Развој инф. система		[Sarker & Sahay 2002]
	Објектно оријентисани развој		[Ramesh & Dennis, 2002]

### 9.3. ТЕКСТ АНКЕТЕ КОРИШЋЕН У ЕМПИРИЈСКОМ ИСТРАЖИВАЊУ НА СРПСКОМ ЈЕЗИКУ

#### АНКЕТА

**Molim Vas da popunite anonimnu anketu, gde bi se rezultati anketiranja analizirali u okviru izrade doktorske disertacije Mr Ljubice Kazi pod nazivom: „Razvoj adaptibilnog distribuiranog informacionog sistema za podršku upravljanju realizacijom softverskih projekata“. Lični podaci neće biti prikazani u doktoratu, već samo statistički obrađeni podaci.**

**Zahvaljujem Vam na saradnji i kolegijalnosti.**

**Mr Ljubica Kazi**

Datum: \_\_\_\_\_ Godina rođenja: \_\_\_\_\_ Pol: M Ž

Naziv zanimanja (završen smer studija): \_\_\_\_\_

Nivo završenog školovanja:  
(srednja škola) (fakultet) (master) (magistar) (doktor nauka)

Status zaposlenja: (STUDENT) (NEZAPOSLEN) (ZAPOSLEN)

Radno mesto: \_\_\_\_\_

Tip institucije gde ste zaposleni (zaokružiti):  
(obrazovna institucija) (javno preduzeće) (IT firma) (Privredni subjekat)

1.

a) Da li ste učestvovali u realizaciji softvera? DA NE

b) Ukupan broj realizovanih softvera: \_\_\_\_\_ broj godina iskustva programiranja: \_\_\_\_\_

c) Tip realizovanih softvera (broj): informacioni sistem \_\_\_\_\_ web sajt \_\_\_\_\_  
multimedija \_\_\_\_\_ drugo \_\_\_\_\_

d) Drugo (opis):

e) Broj profesionalnih softvera (u upotrebi): \_\_\_\_\_ Broj ostalih softvera: \_\_\_\_\_

2. a) Da li ste učestvovali u softverskom projektu? DA NE

b) Broj softverskih projekata na kojima ste učestvovali: \_\_\_\_\_

-----  
3. Najčešći način realizacije softverskih projekata (zaokružiti):

a) (timski) (pojedinačno)  
b) (kolokacijski - na jednoj zajedničkoj lokaciji) (distribuirano - na odvojenim lokacijama)

4. a) Da li ste u softverskim projektima koristili neke alate za podršku timskom radu?  
DA NE

b) Koje alate ste koristili?

5. a) Da li ste u softverskim projektima koristili neke alate za upravljanje projektom?  
DA NE

b) Koje alate ste koristili?



6. Šta, po Vašem mišljenju i iskustvu, najviše utiče na uspeh softverskog projekta?
7. Šta, po Vašem mišljenju i iskustvu, najviše utiče na neuspeh softverskog projekta?
8. Rangirati značaj faktora uticaja na uspeh projekta (1 – najvažniji, 2- srednje važan 3- najmanje važan)?
- a) ljudski \_\_\_\_\_
- b) organizacioni \_\_\_\_\_
- c) tehnoloski \_\_\_\_\_
9. Da li smatrate da je potrebno koristiti ili razviti informacioni sistem za praćenje uspešnosti softverskog projekta? (DA) (NE) (NE ZNAM)
10. Zaokružite osobine (jednu ili više) koje smatrate da takav informacioni sistem treba da ima i rangirajte ih po važnosti (1 – najvažnije, itd.):
- a) Merenje kvaliteta softvera \_\_\_\_\_
- b) Merenje uspešnosti procesa razvoja \_\_\_\_\_
- c) Vizualizacija podataka o progresu \_\_\_\_\_
- d) Adaptivnost na promene \_\_\_\_\_
- e) Podrška distribuiranom razvoju \_\_\_\_\_
- f) Primena poslovnih pravila \_\_\_\_\_
11. Da li smatrate da bi web aplikacija za podršku upravljanju softverskim projektima [www.it-project.rs](http://www.it-project.rs) (корисничко име: *dissoft@it-project.rs*, шифра: ABC) могла да се користи за подршку управљању софтверским пројектима у реалној пракси? (DA) (NE) (NE ZNAM)
12. Koje su prednosti korišćenja ovakve aplikacije?
13. Koji su nedostaci korišćenja ovakve aplikacije?
14. Koje softverske funkcije smatrate da treba da ima ovakva aplikacija, a nisu podržane u postojećem rešenju?
15. Komentari i sugestije:

## 9.4. ТЕКСТ АНКЕТЕ КОРИШЋЕН У ЕМПИРИЈСКОМ ИСТРАЖИВАЊУ НА ЕНГЛЕСКОМ ЈЕЗИКУ

### SURVEY

**Please fill this form for an anonymous survey, whose results will be analyzed within the PhD thesis of author PhD candidate Ljubica Kazi, title: „Development of adaptive distributed information system for software project management support“. Personal data will not be presented in the PhD Thesis document, but only statistical data.**

**Thank you for your cooperation and collegiality!**

**Ljubica Kazi**

Survey Date: \_\_\_\_\_ Year born: \_\_\_\_\_ Gender: M F

Finished school graduation title: \_\_\_\_\_

Level of finished school:  
(high school) (college) (University school) (master) (PhD)

Employment status: (student) (not employed) (employed)

Work position name: \_\_\_\_\_

Type of institution of employment:  
(Educational institution) (Public company) (IT company) (Business enterprise)

1.

a) Were you engaged in software creation? YES NO

b) Total number of software created: \_\_\_\_\_ Number of years of programming experience: \_\_\_\_\_

c) Type of software (number): information system \_\_\_ web site \_\_\_  
multimedia \_\_\_ other \_\_\_

d) Other (short description):

e) Number of professional software created that are in use: \_\_\_\_\_ Other soft. num: \_\_\_\_\_

2. a) Were you engaged in software projects? YES NO

b) Number of software projects you were engaged in: \_\_\_\_\_

-----  
3. The most frequent type of software project implementation:

a) (Teamwork) (single work)

b) (Collocated – at the common location) (distributed – at separate locations)

4. a) Did you use any tool for teamwork support, during your work on software projects?  
YES NO

b) Which tools did you use?

5. a) Did you use any project management tool, during your work on software projects?  
YES NO

b) Which tools did you use?

---

6. What, upon your opinion and experience, influences software project success the most?

7. What, upon your opinion and experience, influences software project failure the most?

8. Rank the software project success factors (1 – most important, 2- medium 3- the least important)?

- a) Human factor \_\_\_\_\_
- b) Organizational factor \_\_\_\_\_
- c) Technological factor \_\_\_\_\_

9. Upon your opinion, is it needed to use or develop an information system for software project monitoring and control support?

(YES) (NO) (I DON'T KNOW)

10. Select one or more features that such an information system should have and rank them according to the importance (1 – most important... etc.):

- a) Software quality measurement \_\_\_\_\_
- b) Development process success measurement \_\_\_\_\_
- c) Visualization of data about the progress \_\_\_\_\_
- d) Adaptivity to changes \_\_\_\_\_
- e) Support to distributed development \_\_\_\_\_
- f) Application of business rules \_\_\_\_\_

11. Upon your opinion, is web application for software project support: *www.it-project.rs* (user name: *dissoft@it-project.rs*, pass: *ABC* or user name: *proba@it-project.rs*, pass: *proba*) could be used in software project management in real professional practice?

(YES) (NO) (I DON'T KNOW)

12. What are the advantages of using this web application?

13. What are the disadvantages of using this web application?

14. Upon your opinion, which software functions are missing and should be added, since they are needed in this type of application?

15. Comments and suggestions:

## 9.5. Подаци из емпиријског истраживања коришћења прототипа у настави и професионалном усавршавању студената

У наставку је дат списак софтверских пројеката, који су реализовани у оквиру проблематике унапређења информационог система Техничког факултета „Мохајло Пупин“ у Зрењанину, под менторством Мр Љубице Кази. Реализовани софтверски пројекти имплементирани су уз административну подршку управљању софтверским пројектима коришћењем развијеног прототипа апликације, односно web сајта [www.it.project.rs](http://www.it.project.rs) у току школске 2013/14 (ознака 2014\_редни број пројекта) и у току школске 2014/15 (ознака 2015\_редни број пројекта).

Табела 9.5.1. Списак реализованих софтверских пројеката у школској 2013/14 години

Ознака прој.	Пројекат
2014_1	ASPX апликација за подршку евиденције Alumni (дипломираних студената)
2014_2	ASPX апликација за подршку евиденцији опреме факултета
2014_3	ASPX апликација за подршку организацији међународне конференције
2014_4	PHP апликација за групно слање е-mail обавештења
2014_5	ASPX апликација за праћење евиденције наставног и научног рада на факултету
2014_6	ASPX апликација за праћење евиденције конкурса и запослених на факултету
2014_7	ASPX апликација за подршку организацији одбрана завршних радова
2014_8	PHP апликација за Web форум
2014_9	ASPX апликација за евиденцију правних докумената факултета
2014_10	PHP апликација за евиденцију рада библиотеке факултета
2014_11	ASPX апликација за подршку раду скриптарнице факултета
2014_12	ASPX апликација за on-line CIP евиденцију
2014_13	Windows C# апликација за израду профактура
2014_14	ASPX апликација за евиденцију предиспитних обавеза
2014_15	ASPX апликација за евиденцију научног финансијског фонда факултета
2014_16	ASPX апликација за евиденцију записника са седница
2014_17	ASPX апликација за евиденцију оцењивања наставника
2014_18	ASPX апликација за подршку организацији рада на факултету
2014_19	ASPX апликација за евиденцију података о понашању студената
<b>Укупно</b>	<b>19</b>

Табела 9.5.2. Списак реализованих софтверских пројеката у школској 2014/15 години

Рб пројекта	Пројекат
2015_1	PHP апликација за евиденцију пријема екстерних докумената и прописа
2015_2	ASPX апликација за подршку раду скриптарнице факултета – унапређење софтвера
2015_3	PHP апликација за евиденцију опреме
2015_4	PHP апликација за подршку оцењивању рада наставника
2015_5	PHP апликација за подршку евиденцији проблема у оквиру менторског рада
2015_6	PHP апликација за подршку евиденцији радних задужења и квалитета реализованих активности
2015_7	PHP апликација за праћење тока докумената
2015_8	PHP апликација за евиденцију набавке потрошног материјала и рециклаже тонера
2015_9	PHP апликација за евиденцију издавачке делатности
2015_10	PHP апликација за евиденцију ИТ пројеката
2015_11	ASPX апликација за евиденцију тендера
<b>Укупно</b>	<b>11</b>

### 9.5.1. Приказ основних карактеристика реализованих софтверских пројеката

Број учесника: 1\* - један учесник на теми, али тимски рад у реализацији 2 повезана пројекта  
Технологије:

- ASPX - ASPX web апликација + MS SQL Server база података
- ASPX MVC - ASPX MVC web апликација + MS SQL Server база података
- PHP - PHP web апликација + MY SQL база података
- PHP Symp - PHP (Symphony framework) web апликација + MY SQL база података
- Win C# - Windows Forms C# + MS SQL Server

Могућност интеграције:

- Модул – Могућност укључивања као проширење постојећег web сајта факултета
- ПОС - Засебна апликација

Табела 9.5.1.1. Основне карактеристике реализованих софтверских пројеката

Ознака прој.	Датум почетка	Датум завршетка	Број месеци	Број учесника	Технологија израде	Могућност интеграције
2014_1	13.3.2014.	4.7.2014.	4	1	ASPX	Модул
2014_2	11.3.2014.	13.5.2014.	3	2	ASPX MVC	Модул
2014_3	29.3.2014.	5.6.2014.	3	1	ASPX	Модул
2014_4	2.4.2014.	2.6.2014.	3	1	PHP	ПОС
2014_5	13.3.2014.	1.7.2014.	4	1*	ASPX	Модул
2014_6	13.3.2014.	31.5.2014.	3	1*	ASPX	Модул
2014_7	13.3.2014.	22.5.2014.	3	1	ASPX	Модул
2014_8	13.3.2014.	21.5.2014.	3	2	PHP	ПОС
2014_9	13.3.2014.	28.5.2014.	3	1	ASPX	Модул
2014_10	19.3.2014.	29.4.2014.	2	1	PHP Symp	Модул
2014_11	13.3.2014.	17.9.2014.	7	1	ASPX	Модул
2014_12	13.3.2014.	2.6.2014.	3	1	ASPX	Модул
2014_13	1.4.2014.	27.5.2014.	2	1	Win C#	ПОС
2014_14	30.3.2014.	25.9.2014.	6	1	ASPX	Модул
2014_15	13.3.2014.	10.6.2014.	3	1	ASPX	Модул
2014_16	13.3.2014.	4.9.2014.	6	1	ASPX	Модул
2014_17	13.3.2014.	26.9.2014.	7	1	ASPX	Модул
2014_18	13.3.2014.	28.5.2014.	3	3	ASPX	Модул
2014_19	29.5.2014.	25.9.2014.	4	1	ASPX	Модул
2015_1	18.3.2015.	25.6.2015.	3	1	PHP	Модул
2015_2	24.3.2015.	11.6.2015.	3	1	ASPX	Модул
2015_3	18.3.2015.	10.6.2015.	3	1	PHP	Модул
2015_4	20.2.2015.	30.5.2015.	3	3	PHP	Модул
2015_5	18.3.2015.	7.6.2015.	3	1	PHP	Модул
2015_6	5.3.2015.	28.5.2015.	3	1	PHP	Модул
2015_7	15.7.2015.	26.8.2015.	1	1	PHP	Модул
2015_8	7.3.2015.	4.7.2015.	4	1	PHP	Модул
2015_9	19.2.2015.	8.9.2015.	7	1	PHP	Модул
2015_10	19.2.2015.	6.9.2015.	7	1	PHP	Модул
2015_11	19.2.2015.	2.9.2015.	7	1	ASPX	Модул

### 9.5.2. Приказ класификације реализованих софтвера и софтверских пројеката

UNSPSC класификација софтвера

- 432315 – Business Function Specific Software
- 432316 – Enterprise Resource Planning ERP software
- 432335 – Information Exchange software

Turban класификација софтвера

- Према намени (N): AON -Апликативни софтвер опште намене, ASN - Апликативни софтвер специфичне намене
- Према корисницима (K): ОК - опште, КК - категорија корисника, К - конкретан корисник

*Nguyen класификација софтвера*

- КОМ - Комерцијални (општа намена, шири круг корисника)
- NAR - По наруџби (конкретан корисник)

*Shenhar&Dvir класификација софтверских пројеката*

- LT - „Low-Tech Projects“, раније познате технологије, понављање постојећих решења
- MT - „Medium-Tech Projects“ – већином познате технологије уз мање сегменте увођења новијих, раније непознатих технологија

*Berntsson-Svensson&Aurum класификација софтверских пројеката*

- према типу активности (ТА) – нови софтвер (ТАН), унапређење постојећег (ТАУ)
- у односу на клијента (К) – клијент орјентисани (КО), тржишно орјентисани (ТО), комбиновани (К), интерни (И)

*Nassif&Boehm класификација софтверских пројеката*

- О - Organic (мањи тим, висок ниво знања, флексибилни захтеви)
- SD - Semi-detached (мешовити тим, мешовити ниво знања, мешовити захтеви)

*Reifer класификација софтверских пројеката*

- Т- Традиционални (Desktop)
- W - web

*Cadle&Yeates (IS) класификација софтверских пројеката*

- RNS - Развој новог софтвера
- UPS - Унапређење постојећег софтвера

Табела 9.5.2.1. Класификација пројеката према типовима софтвера и софтверских пројеката

Озн. прој.	Класификација софтвера			Класификација софтверских пројеката				
	UNSPSC	Turban	Nguyen	Shenhar & Dvir	Berntsson-Svensson & Aurum	Nassif & Boehm	Reifer	Cadle & Yeates (IS)
2014_1	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_2	432316	ASN, KK	NAR	MT	TAN, KO	SD	W	RNS
2014_3	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_4	432335	AON, KK	KOM	LT	TAN, TO	O	W	RNS
2014_5	432315	ASN, KK	NAR	LT	TAN, KO	SD	W	RNS
2014_6	432315	ASN, KK	NAR	LT	TAN, KO	SD	W	RNS
2014_7	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_8	432335	AON, OK	KOM	LT	TAN, TO	SD	W	RNS
2014_9	432316	ASN, KK	KOM	LT	TAN, TO	O	W	RNS
2014_10	432316	ASN, KK	NAR	MT	TAN, TO	O	W	RNS
2014_11	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_12	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_13	432316	ASN, KK	KOM	LT	TAN, TO	O	T	RNS
2014_14	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_15	432315	ASN, K	NAR	LT	TAN, KO	O	W	RNS
2014_16	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_17	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2014_18	432316	ASN, KK	KOM	LT	TAN, TO	SD	W	RNS
2014_19	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2015_1	432316	ASN, KK	KOM	LT	TAN, TO	O	W	RNS
2015_2	432315	ASN, KK	NAR	LT	TAU, KO	O	W	UPS
2015_3	432316	ASN, KK	NAR	LT	TAN, TO	O	W	RNS
2015_4	432315	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2015_5	432315	ASN, K	NAR	LT	TAN, KO	O	W	RNS
2015_6	432316	ASN, KK	KOM	LT	TAN, TO	O	W	RNS
2015_7	432316	ASN, KK	KOM	LT	TAN, TO	O	W	RNS
2015_8	432316	ASN, KK	NAR	LT	TAN, TO	O	W	RNS
2015_9	432316	ASN, KK	NAR	LT	TAN, KO	O	W	RNS
2015_10	432316	ASN, KK	KOM	LT	TAN, TO	O	W	RNS
2015_11	432316	ASN, KK	NAR	LT	TAN, KO	O	W	RNS

### 9.5.3. Приказ реализације наставних обавеза у оквиру реализације софтверских пројеката

Ниво студија:

- О – Основне студије
- М – Мастер студије

Наставни предмет:

- IS2 - Информациони системи 2
- SP - Стручна пракса
- DIS - Дистрибуирани информациони системи
- EP – Електронско пословање
- UITP – Управљање ИТ пројектима
- 

Табела 9.5.3.1. Реализација наставних обавеза у оквиру реализације софтверских пројеката

Ознака прој.	Рб. студ.	Рб. студ. у прој.	Ниво студија	Наставни предмет	Завршни рад
2014_1	1	1	О	IS2, SP	DA
2014_2	2	1	М	DIS	DA
2014_2	3	2	М	DIS	NE
2014_3	4	1	М	DIS	DA
2014_4	5	1	М	DIS	NE
2014_5	6	1	М	DIS	DA
2014_6	7	1	М	DIS	DA
2014_7	8	1	М	DIS	DA
2014_8	9	1	М	DIS, SP	NE
2014_8	10	2	М	DIS	NE
2014_9	11	1	М	DIS	NE
2014_10	12	1	М	DIS	DA
2014_11	13	1	О	IS2	DA-PLAN
2014_12	14	1	М	DIS	DA-PLAN
2014_13	15	1	О	IS2, SP	DA
2014_14	16	1	О	IS2, SP	DA
2014_15	17	1	О	IS2	DA-PLAN
2014_16	18	1	М	DIS	DA
2014_17	19	1	О	IS2, SP	DA
2014_18	20	1	О	IS2, SP	NE
2014_18	21	2	О	IS2, SP	NE
2014_18	22	3	О	IS2, SP	NE
2014_19	23	1	О	IS2, SP	DA
<b>Укупно студената 2014:</b>				<b>23</b>	
2015_1	1	1	О	SP	DA - PLAN
2015_2	2	1	О	SP	NE
2015_3	3	1	О	SP	DA
2015_4	4	1	О	IS2	DA
2015_4	5	2	О	IS2	DA
2015_4	6	3	О	IS2	DA
2015_5	7	1	О	SP	DA
2015_6	8	1	О	SP	DA
2015_7	9	1	О	SP	DA- PLAN
2015_8	10	1	О	SP	DA
2015_9	11	1	О	IS2	DA-PLAN
2015_10	12	1	М	UITP	DA-PLAN
2015_11	13	1	М	EP	DA-PLAN
<b>Укупно студената 2015:</b>				<b>13</b>	
<b>Укупно студената:</b>				<b>36</b>	

## 9.6. Два примера софтверских пројеката реализованих уз административну подршку управљању пројектом коришћењем прототипа система

У наставку ће бити приказани подаци који су у току реализације софтверских пројеката евидентирани у оквиру коришћења прототипа ([www.it-project.rs](http://www.it-project.rs)). Одабрани су подаци за два софтверска пројекта, чија је реализација резултовала успешном интеграцијом са постојећим информационим системом факултета, односно званичним web сајтом Техничког факултета „Михајло Пупин“ у Зрењанину и применом наведених софтвера у реалној пракси.

### 9.6.1. „Alumni web“

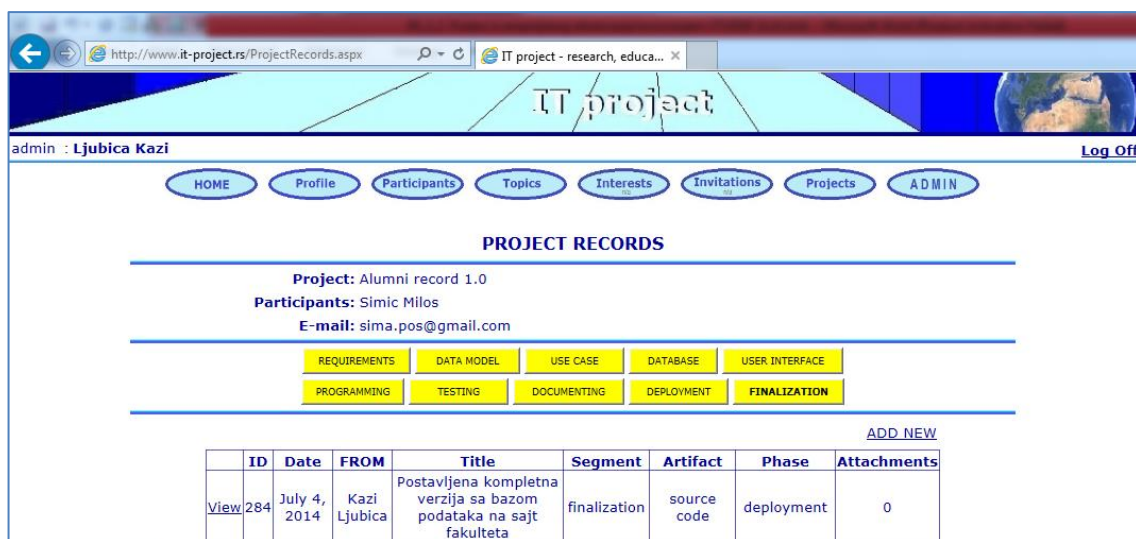
Сврха „Alumni web“ апликације је евиденција и приказ података о дипломираним студентима Техничког факултета „Михајло Пупин“ у Зрењанину, њиховим радним биографијама и утисцима о Факултету за време студирања. На овај начин будући студенти факултета могу да се информишу о успеху дипломираних факултета у реалној професионалној пракси. Слика 9.6.1.1. приказује реализовану апликацију која је постављена и користи се са web сајта Техничког факултета „Михајло Пупин“ у Зрењанину.



Слика 9.6.1.1. Приказ апликације „Alumni Web“ са web сајта Техничког факултета „Михајло Пупин“ у Зрењанину

Слика 9.6.1.2. приказује коришћење прототипа апликације ([www.it-project.rs](http://www.it-project.rs)) за потребе евиденције података у току реализације и праћења реализације пројекта „Alumni Web“.





Слика 9.6.1.2. Приказ коришћења прототипа апликације [www.it-project.rs](http://www.it-project.rs) за евиденцију података о току реализације пројекта "Alumni Web"

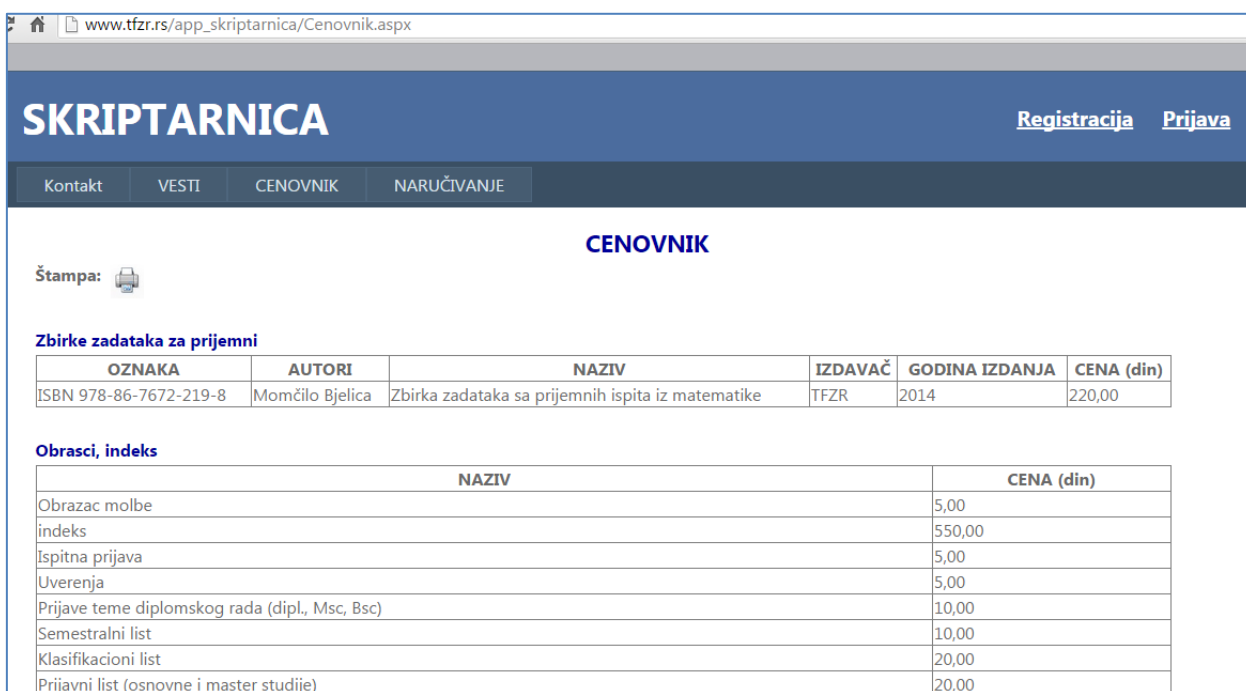
Табела 9.6.1.1. Листа записа у оквиру реализације пројекта „Алумни“

Date	Title	Segment	Artifact	Phase
13-Mar-14	Specifikacija zahteva i obuka za korisnicki interfejs	information	document	requirements
25-Mar-14	Use case - Licne konsultacije	result	model	use case
25-Mar-14	Licne konsultacije	result	model	data model
1-Apr-14	Korisnicki interfejs - licno	result	source code	user interface
2-Apr-14	Verifikacija – продекан за наставу	verification	information	requirements
3-Apr-14	Dogovor - staticki html	task	information	programming
9-Apr-14	Postavljen staticki na sajt	result	source code	user interface
16-Apr-14	Izmena statickog sajta	result	source code	programming
18-Apr-14	Moja dopuna statickog sajta - poslato za postavljanje na sajt fakulteta	result	source code	user interface
19-Apr-14	Postavljen staticki sajt alumni	deployment	source code	deployment
22-Apr-14	Dizajn BP za dinamicki deo	verification	database	database
22-Apr-14	Dizajn UI za dinamicki deo	verification	source code	user interface
6-May-14	Konekcija i forma za unos	verification	database	programming
8-May-14	Forma za unos - file upload i tabelarni	information	source code	programming

13-May-14	Dalje za praksu - uputstva	task	information	requirements
13-May-14	Testiranje - probni	result	source code	testing
24-May-14	Dokument za IS2	result	document	documenting
27-May-14	Dokument za praksu	task	document	documenting
27-May-14	Verzija za deployment - sredjivanje snimanja potvrde	result	source code	finalization
28-May-14	Putanje do baze zbog deployment	deployment	source code	finalization
4-Jul-14	Postavljena kompletna verzija sa bazom podataka na sajt fakulteta	finalization	source code	deployment

### 9.6.2. „On-line скриптарница“

Сврха апликације „Он-лине скриптарница“ је евиденција комплетне понуде скриптарнице Техничког факултета „Михајло Пупин“ у Зрењанину у односу на уџбенике, збирке задатака и обрасце који се користе у администрацији у току студирања. Слика 9.6.2.1. приказује апликацију која је постављена на web сајт Техничког факултета „Михајло Пупин“ у Зрењанину и која се користи у оквиру редовног рада скриптарнице и у оквиру пописа.



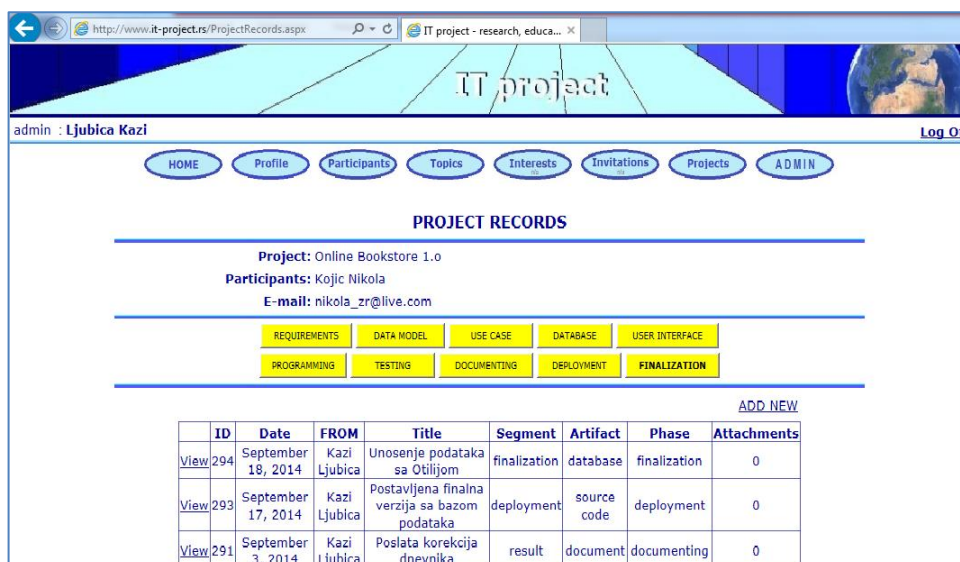
OZNAKA	AUTORI	NAZIV	IZDAVAČ	GODINA IZDANJA	CENA (din)
ISBN 978-86-7672-219-8	Momčilo Bjelica	Zbirka zadataka sa prijemnih ispita iz matematike	TFZR	2014	220,00

NAZIV	CENA (din)
Obrazac molbe	5,00
indeks	550,00
Ispitna prijava	5,00
Uverenja	5,00
Prijave teme diplomskog rada (dipl, Msc, Bsc)	10,00
Semestralni list	10,00
Klasifikacioni list	20,00
Prijavni list (osnovne i master studije)	20,00

Слика 9.6.2.1. Приказ апликације Он-лине скриптарница са web сајта Техничког факултета „Михајло Пупин“ у Зрењанину

Слика 9.6.2.2. приказује коришћење прототипа апликације (www.it-project.rs) за потребе евиденције података у току реализације и праћења реализације пројекта Он-лине скриптарница.



Слика 9.6.2.2. Приказ коришћења прототипа апликације [www.it-project.rs](http://www.it-project.rs) за евиденцију података о току реализације пројекта “On-line скриптарница”

Табела 9.6.2.1. Листа записа у оквиру реализације пројекта „On-line скриптарница”

Date	Title	Segment	Artifact	Phase
13-Mar-14	Specifikacija i obuka za ASPX	information	document	requirements
3-Apr-14	Use case	verification	model	use case
3-Apr-14	use CASE I cdm - POCETAK	verification	model	data model
29-Apr-14	Nista novo - uraditi dizajn KI	task	source code	user interface
27-May-14	Baza radi	result	database	database
27-May-14	Uradio unos i tabelarni prikaz - sredjivali greske	verification	source code	programming
29-May-14	Dogovor za praksu i diplomski	task	document	requirements
29-May-14	Program za evidenciju robe radi - IS2	verification	source code	testing
29-May-14	Odobrena IS2 dokumentacija	verification	document	documenting
10-Jul-14	Finalizacija programa	result	source code	testing
15-Jul-14	Poslat dnevnik e-mailom, treba popravke	result	document	documenting
3-Sep-14	Poslata korekcija dnevnika	result	document	documenting
17-Sep-14	Postavljena finalna verzija sa bazom podataka	deployment	source code	deployment
18-Sep-14	Unosenje podataka sa Otilijom	finalization	database	finalization

## 9.7. Подаци из емпиријског истраживања пробног коришћења прототипа од стране ИТ кадрова уз пратеће анкетирање

Категоризација радних места испитаника у оквиру анкетирања је урађена на следећи начин (Табела 9.7.1.):

Табела 9.7.1. Категоризација радних места испитаника у анкетирању

Категорија радног места	Назив радног места
Општи информатичар	informatičar, strucni saradnik za informatiku, inženjer informacionog sistema, organizator informacionog sistema, referent systemske logistike, informatički laborant, samostalni IT inženjer, samostalni referent IT, IT engineer, koordinator za IT servise
Систем администратор	IT administrator, IT system administrator, administrator, administrator mreže, sistem administrator
Консултант	Tehnicki konsultant za SAP, Senior SAP konsultant, Stariji konsultant, IT support consultant
Бизнис аналитичар	Business intelligence analyst, saradnik za statistiku i izvestavanje, Senior Business intelligence Developer
Пројектант	Senior system analyst, software architect, Senior Premier Field Engineer, projektant informacionog sistema, Lead Software engineer / Architect, software architect, Customer Processes Expert
Софтвер девелопер	Software developer, senior software engineer, developer, medior software engineer, Software Developer Java, .NET Software Engineer, senior software developer, senior PHP developer, software engineer, Qlickview Administrator / Developer, .NET developer, Junior Software developer, java developer, web developer
Програмер	Programer, PHP programer, junior programer
Администратор базе података	Administrator baze podataka, Senior inženjer baza podataka, Inženjer baza podataka – Senior, RW/DBA North America Task Manager
Графичар	GIS developer, Front-end developer, graficki dizajner, CG^VFX Artist, web dizajner
Тестер	Verification Expert, Interpreter Supervisor, Security Software Engineer, Saradnik za IT obezbedjenje kvaliteta.
Неинформатичар	Nastavnik engleskog jezika, nastavnik matematike, nastavnik fizike, savetnik za zaposljavanje, bankarski sluzbenik, marketing menadzer, profesor matematike, sluzbenik, professional shop advisor, Saradnik za obezbedenje kvaliteta
Менаџер	Direktor, načelnik, šef odseka, vlasnik, Deputy Team Leader, Customer Care Specialist, Rukovodilac inženjering sektora, rukovodilac odeljenja, zamenik nacelnika, sef IT sektora, pomocnik gradonacelnika, deputy team lead/technical lead, sef odseka IT, rukovodilac odeljenja, Head of Customer Support, Rukovodilac za odrzavanje, Head of Computed Science Department, Country technical manager, PD infrastructure and services team lead deputy
Пројектни менаџер	Project manager, Team Lead / Scrum Master, asistent na projektima, Project manager for IT infrastructure, PMP, IT project manager
Наставник	Nastavnik informatike i računarstva, saradnik u nastavi, profesor informatike, profesor, nastavnik informatike, asistent, docent, predavac u skoli racunara
Истраживач	Senior researcher, istrazivac, Senior research and development engineer, Computer Scientist and Programmer

**Питање 4 б) - Које алате за подршку тимском раду сте користили?**

Табела 9.7.2. Списак алата за подршку тимском раду, на основу анкете

<b>Алат</b>	<b>Број одговора</b>	<b>Алат</b>	<b>Број одговора</b>
GIT, Github, Git Extensions, Gitorius, Gitlab	24	Amazon Internal Tools	1
SVN, TortoiseSVN	19	Artifactory	1
Jira	16	BITTORRENT SYNC	1
Microsoft Team Foundation Server	16	ChiliProject	1
Skype	15	Crucible	1
Google Docs, Google Drive	8	Documentum	1
Team Viewer	7	Email	1
Bugzilla	6	eRoom	1
Confluence	6	file sharing	1
Microsoft Sharepoint	6	FootPrints	1
Microsoft Lync	5	FoxPro	1
RedMine	5	G-forge	1
Asana	4	Gerrit	1
Basecamp	4	GroupWise	1
CVS	4	HipChat	1
Microsoft Visio	4	HG	1
Slack	4	IBM TeamConcert	1
BitBucket	3	IRC	1
case alat	3	Jabber	1
DropBox	3	Microsoft Excel	1
IceScrum	3	Microsoft Exchange	1
dotProject	2	Microsoft Office	1
FishEye	2	Microsoft Outlook	1
Goto Meeting	2	Microsoft Project	1
HP Quality Center	2	Microsoft SQL	1
Jenkins	2	Microsoft Visual Studio Online	1
Kanbanize	2	NEX	1
Knowledge Tree	2	RDC	1
Mantis	2	Oracle	1
mercurial	2	PivotalTracker	1
Source Tree	2	power designer	1
Subversion for common repositories	2	PTC Integrity	1
TestLink	2	PVCS	1
TortoiseCVS	2	Rally	1
Trello	2	Remedy	1
Wrike	2	remote desktop sharing	1
Yammer	2	Review Board	1
Active Collab	1	Roundup	1
ALM	1	SAP SolMan	1
Spark	1	Viber	1

Source Control in Visual Studio	1	VPN	1
Ankh SVN	1	xWiki	1
Team Site	1	web calendar	1
Teleconferencing apps	1	WebEx	1
TGS	1	Wiki	1
TortoiseHG	1	Zimbra	1
Vault	1		

**Питање 5 б) - Које алате за подршку управљању пројектом сте користили?**

Табела 9.7.3. Списак алата за подршку управљању пројектом, на основу анкете

Алат	Број одговора	Алат	Број одговора
Microsoft Project	25	Mercurial	1
JIRA	23	Microsoft Word	1
Microsoft Team Foundation Server	10	Microsoft Excel	1
Active Collab	8	Microsoft Project Manager	1
Redmine	8	Microsoft Project Planing	1
dotProject	6	Microsoft OneNote	1
Base Camp	3	MS Project Server	1
FogBugz	3	Mingle	1
Thrello	3	gForge	1
Mantis	2	ISVV	1
Microsoft Excel	2	Kanbanize	1
Pivotal tracker	2	LeanKit	1
Team Gantt	2	NEX	1
wrike	2	PowerDesigner	1
2_Plane	1	projektaL MyNIS Siemens SAPS	1
Amazon Internal Tools	1	Radar	1
ARIS	1	Rally	1
Asana	1	SAP Solution Manager	1
BitBucket	1	SAP modul za upravljanje projektima	1
Confluence	1	Scrumwise	1
CA Clarity	1	Service Desk Manage Engine	1
GIT	1	SVN	1
DaPulse	1	SQC	1
Daylite	1	SW milestone reviews	1
dotProject	1	TeamProject	1
Houston project and program management	1	TeamViewer	1
IBM BPC	1	TargetProcess	1
LP	1	Zoho CRM	1

**Евалуација елемената почетног узорка на основу ELIM - F методе**

У наставку је дата табела са подацима који се односе на примену ELIM - F методе вредновањем сваке анкете, односно испитаника коришћењем скупа од 17 критеријума. Критеријуми су описани у одељку који се односи на емпиријска истраживања анкетирањем у делу описа метода за евалуацију елемената узорка. Подаци у овој табели су сортирани према проценту задовољавања ELIM критеријума, а затим груписани према ELIM-F критеријуму (ELIM % > 50%).

Табела 9.7.4. Резултати примене методе ELIM-F над подацима о учесницима анкете ради селекције узорка истраживања

РБ. Ан.	Критеријуми ELIM методе																	По.	%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
125	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17	100
126	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17	100
140	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17	100
148	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17	100
156	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	17	100
33	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	16	94
46	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	16	94
80	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	94
94	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	16	94
105	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	94
114	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	16	94
135	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	16	94
142	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	16	94
160	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	94
161	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16	94
116	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	16	94
47	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	15	88
57	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	15	88
60	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	15	88
74	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	15	88
87	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	15	88
88	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	15	88
96	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	15	88
100	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1	15	88
104	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	15	88
130	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	15	88
132	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	15	88
137	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	15	88
139	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	15	88
2	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	15	88
19	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	15	88
26	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	15	88
69	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	15	88
111	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	15	88
117	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	15	88

Љубица Кази - Докторска дисертација

131	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	15	88
162	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	15	88
21	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	14	82
54	1	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	1	14	82
97	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	14	82
99	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	14	82
106	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	14	82
112	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	14	82
127	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	14	82
153	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	14	82
154	1	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	14	82
5	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	14	82
44	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	14	82
66	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	14	82
77	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	14	82
90	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	14	82
107	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	14	82
124	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	14	82
147	1	1	1	1	1	1	1	0	1	0	1	0	1	1	1	1	1	14	82
150	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	14	82
27	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	0	1	13	76
31	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	0	1	13	76
36	1	1	1	0	0	1	1	1	1	1	1	1	0	1	0	1	1	13	76
39	1	1	1	1	0	1	1	0	1	0	1	1	1	1	1	0	1	13	76
45	1	1	0	1	0	1	1	1	1	0	1	1	0	1	1	1	1	13	76
72	1	0	0	1	0	1	1	1	1	1	1	1	1	0	1	1	1	13	76
73	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	0	0	13	76
89	1	1	0	1	0	1	1	1	1	1	1	1	1	0	1	1	0	13	76
119	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	13	76
134	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	13	76
136	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	13	76
158	1	1	1	1	0	1	1	1	1	1	1	1	0	1	0	1	0	13	76
159	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	1	13	76
3	1	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	13	76
4	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	1	0	13	76
6	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	13	76
35	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	1	1	13	76
67	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	1	13	76
76	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1	0	13	76
110	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0	1	1	13	76
25	1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	1	12	71
30	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	12	71
38	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	1	1	12	71
63	1	1	0	1	0	1	1	0	1	1	1	0	1	1	0	1	1	12	71
95	1	1	1	1	0	1	1	0	0	0	1	1	1	1	1	0	1	12	71
113	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0	1	1	12	71



Љубица Кази - Докторска дисертација

115	1	0	1	1	0	1	1	1	1	1	1	1	0	1	1	0	0	12	71
163	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	12	71
165	1	0	1	1	0	1	1	0	1	1	0	0	1	1	1	1	1	12	71
14	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	12	71
59	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0	12	71
102	1	0	0	1	1	1	1	0	1	1	1	0	0	1	1	1	1	12	71
118	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	1	12	71
122	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	12	71
145	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	0	12	71
164	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1	0	12	71
1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	0	11	65
20	1	0	1	1	0	1	0	1	1	1	1	1	1	1	0	0	0	11	65
41	1	1	1	1	0	1	1	0	1	0	0	1	1	1	1	0	0	11	65
52	1	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	11	65
75	1	1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	0	11	65
85	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	1	1	11	65
86	1	1	0	1	0	1	1	0	1	1	1	1	0	1	0	1	0	11	65
123	1	0	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	11	65
152	1	0	1	1	0	1	1	1	1	1	0	1	1	0	0	0	1	11	65
7	1	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	0	11	65
16	1	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	11	65
62	1	1	1	1	1	1	1	0	0	1	0	1	0	0	1	1	0	11	65
70	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	11	65
71	1	0	1	1	1	1	1	0	1	1	0	1	0	1	1	0	0	11	65
82	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	11	65
98	1	0	1	1	1	1	1	0	1	0	0	0	1	1	0	1	1	11	65
101	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	11	65
138	1	1	1	1	1	1	1	1	0	1	0	0	0	1	0	0	1	11	65
8	1	1	1	1	0	1	1	0	0	1	0	1	0	1	0	1	0	10	59
15	1	1	1	1	0	1	0	1	0	1	1	1	1	0	0	0	0	10	59
17	1	1	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0	10	59
78	1	1	1	1	0	1	1	1	1	0	0	1	0	1	0	0	0	10	59
81	1	0	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0	10	59
13	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	10	59
18	1	1	0	1	1	0	1	0	1	0	0	0	1	1	0	1	1	10	59
24	1	1	1	1	1	1	1	1	0	0	0	1	0	1	0	0	0	10	59
28	1	0	1	1	1	1	1	1	0	1	0	1	0	1	0	0	0	10	59
37	1	0	0	1	1	1	1	1	1	1	0	0	1	1	0	0	0	10	59
53	1	1	1	1	1	1	0	1	0	1	0	1	1	0	0	0	0	10	59
83	1	1	0	1	1	1	0	0	0	1	1	1	1	0	1	0	0	10	59
93	1	1	0	1	1	1	0	1	0	1	1	0	1	0	0	0	1	10	59
108	1	1	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1	10	59
121	1	0	0	1	1	1	1	1	1	0	1	0	1	1	0	0	0	10	59
23	1	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0	0	9	53
43	1	1	1	1	0	1	1	0	0	0	0	0	0	1	0	1	1	9	53
51	1	1	0	1	1	0	0	1	0	0	1	0	1	1	1	0	0	9	53

Љубица Кази - Докторска дисертација

155	1	0	0	1	1	1	1	0	0	1	1	1	0	1	0	0	0	9	53
<b>UKUPNO (ELIM &gt; 50%)</b>																		<b>128</b>	
129	1	0	0	1	0	1	1	0	1	0	1	0	0	1	1	0	0	8	47
133	1	1	1	1	0	0	1	0	1	0	0	0	0	1	1	0	0	8	47
29	1	0	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	8	47
58	1	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	0	8	47
91	1	0	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	8	47
141	1	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	8	47
84	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	0	7	41
92	1	0	0	1	0	1	0	1	0	0	1	0	0	1	1	0	0	7	41
32	1	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	7	41
56	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	7	41
151	1	0	1	1	1	0	1	0	0	0	0	0	0	1	1	0	0	7	41
22	1	0	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	6	35
42	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	0	0	6	35
55	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	6	35
128	1	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	6	35
157	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	6	35
103	1	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	5	29
12	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	5	29
65	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	5	29
143	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	5	29
144	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	5	29
146	1	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	5	29
9	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	4	24
11	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	24
49	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	4	24
120	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	24
34	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	4	24
40	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	4	24
61	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	24
64	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	4	24
68	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	4	24
149	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	4	24
10	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3	18
48	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3	18
50	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3	18
109	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3	18
79	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	12
<b>UKUPNO (ELIM &lt; 50%)</b>																		<b>37</b>	

**Питање 6 - Шта по вашем мишљењу и искуству највише утиче на успех софтверског пројекта?**

Табела 9.7.5. Аналитички приказ одговора о факторима успеха софтверског пројекта, класификованих према категоријама

КАТЕГОРИЈА	СТАВКЕ ОПИСА
Организација, управљање пројектом	Темељно планирање, процене, координација, Организација послова, Добра организација тима, тимски рад, добра raspodela poslova, postovanje rokova - завршавање на време, добра raspodela vremena, analiza problema I zahtevanih ciljeva, dobro definisan plan upravljanja projektom, kontrola izmena na projektu, balans izmedju ocekivanja, vremena I budzeta, Duže vreme analiziranja zahteva i planiranja od vremena implementiranja softvera, Jasno definisane projektne faze i ishodi svake od njih, Dobro definisani ciljevi i faze projekta, istrazivanje trzista, Pravilno upravljanje svim fazama razvoja projekta, Dobro definisani zahtevi, jasna vizija, korektno izabran tim za realizaciju, Detaljna funkcionalna specifikacija, Jasna definicija zahteva i opis softvera, Pre početka projekta da su jasno definisani zadaci projekta, Cilj projekta, pravilno definisan projektни zadatak, dobro organizovan projektни тим, Ideja (upotrebljivost), brzina isporuke resenja klijentu, potreba da klijent konstantno bude ukljucen, dobro poznavanje domena problema, kontrola izmena na projektu, dobro definisan plan upravljanja projektom, prodaja, Što bolje definisanje prvih zahteva korisnika, svakodnevni daily scrumovi, Ideja sa dobrom upotrebom vrednošću, Efikasno rukovođenje. Jasno definisani zahtevi. Organizovanost i prisutnost svih uloga u jednom timu (biznis analitičar, software architect, menadžment, kontrola kvaliteta i razvoj). Jasno definisani korisnički zahtjevi, kvalitetna raspodijela zadataka I vremena. marketing, Jasni korisnički zahtevi, Unapred osmisljen koncept, struktura tima, jasno definisane potrebe i zelje klijenta, Kontrola projekta kroz sve faze, od zakljucivanja ugovora (u smislu da je projekat izvodljiv u zadatim okvirima finansijskim, vremenskim, tehnoloskim itd, pa sve do isporuke, instalacije i kasnije održavanja aplikacije).Balans realnosti zahteva od strane korisnika i uticaj na postojeći kod. Procenje napretka, marketing – u smislu prodora na tržištu, uredne i jasne specifikacije, precizna business analiza.Sagledavanje potreba korisnika, dobra organizacija (vođenje) rada tima, zacrtani ciljevi, fokus, time to market, Organizacija je ključna stvar, definisanje inicijalni zahteva sto je tacnije moguće na konto kojih ce se formirati znanje o potrebama sistema koje su potrebne za formulisanje dobre arhitekture kao osnove za dalji rad. Odredjivanje tima tj ljudi koji ce ucestvovati u realizaciji software-a (manager, architect, team lead, IT, development, testing...). ispravno odredjivanje opsega projekta i fokus na isti.Dobro poznavanje poslovnih procesa, Planiranje projekta, Detaljna funkcionalna specifikacija, Vreme utronseno na projekat (ukljucujuci planiranje i duskutovanje). Jasno definisani ciljevi, detaljna priprema (dokumentacija, organizacija i podela posla na etape i segmente), dobra specifikacija softvera, Zaokružen i realan scope, dobro razrađeni requirements, realan project schedule, dovoljno vremena za QA.
Активности у развоју софтвера	Максимално тестирање софтвера, пројектовање, оцјенивање квалитета софтвера, организација и планирање самог софтвера, Начин реализације, развјање из перспективе корисника, коришћење агилних метода развоја I пројектовања, Kvalitetno i sveobuhvatno testiranje softvera (ručni testovi, unit testovi, auto testovi). Korišćenje source control-a radi lakšeg snalaženja u kodu, Jasno deljenje projekata u okviru softverskog rešenja (npr. logičko uprošćavanje softverskog rešenja na više međusobno nezavisnih dll-ova). Postojanje bar osnovnog modela softvera (UML dijagrama npr.) pre samog razvoja, upotreba novih tehnologija koja omogućavaju brži, bolji i kasnije nadograđiv projekat, tehnologije koje ce se koristiti, dobra arhitektura I tehnologija izrade, dobar Q&A.
Карактеристике решења	Мogućност касније надоградње, dobar dizajn, Adaptivnost resenja na konstantne promene, jednostavnost, Dobro osmišljena baza podataka i jasan kod, upotrebljivost I lakoca rada, funkcionalan kod, Funkcionalnost, kvalitetan programski kod, Korisnički interfejs, tehnologije koje ce se koristiti, Kvalitet softvera i funkcionalnost u skladu sa zahtevima korisnika.
Документација	stabilna dokumentacija, dobra dokumentovanost projekta, dobro dokumentovani i precizni projektни zadaci.

Личне особине учесника пројекта	strucnost, motivacija, znanje, iskustvo, saradnja, odgovornost, transparentnost, dobar project manager I jaka podrška viseg menadžmenta, Vodja tima, kreativnost, kvalitetan razvojni tim, Dobar tim i ljudi koji realno znaju da procene svoje mogućnosti/znanje, Tim mora imati određeno tehnološko predznanje, dobar projektni voditelj, međuljudski odnosi (svi u timu podjednako vrede), predanost pojedinca, obavezivanje timova na dogovorene rokove, dobro sastavljen tim, dobar timski rad, predanost projektu, sklonost ka inovacijama i efikasnijim rešenjima, dobar menadžment, dobri odnosi u timu, edukacija, odličan tim, timska saradnja, timski duh, motivacija programera, dobra saradnja, stručnost i iskustvo ljudi u njemu i profesionalnost PM (Project manager), vestina pojedinca ili grupe, kvalitetna saradnja, ljudi (zainteresovanost), Znanje učesnika, strucnost vodeceg clana razvojnog tima, Motivisani učesnici. Strucan tim, kvalitet drugih učesnika na tom projektu, znanje svake individue koji učestvuje u izradi projekta, zalaganje project managera, Menadžer projekta, obučenost tima, saradnja programera u timu, i saradnja klijenata i programera. znanje, timski rad, Entuzijizam tima i odlučnost vođe tima, kvalitet ljudi u timu, Stručnost, timski rad, pozitivan duh tima, sposobnost motivacije I upravljanja projektom tim lidera. Strpljenje, entuzijizam, upornost, oko za detalje, podrška I zainteresovanost menadžmenta, kompetencija, posvećen Project Core Team.
Ресурси	Ljudski pecypси, novcани resursи, budžet, definisani hardverski interfejs, dobro isplanirani resursи, pravovremena I iskrena komunikacija izmedju klijenta, menadžmenta I samog tima koji izradjuje softver.
Комуникација	komunikacija izmedju clanova tima, dobra informisanost sta softver treba da radi, dobra komunikacija sa klijentom, (Konstantna komunikacija sa korisnicima), Jasni korisnički zahtjevi, Efektivna komunikacija, dostupnost svih informacija, komunikacija u timu, međusobna, uspešna komunikacija, unapred poznate sve informacije vezane za projekat, potreba da klijent prati razvoj softvera, stalna komunikacija sa klijentima, definicije taskova kako bi se taskovi jednoznačajno, prcizno i tačno definisali, dobra komunikacija sa menadžerom projekta, stalna i kvalitetna korespodencija sa korisnikom, kvalitetan i pravovremeni odziv (feedback) korisnika. dobra komunikacija i razumevanje klijenta, Dobra/otvorena komunikacija.
Алат за управљање пројектом	Dobar softverski alat koji omogućava stalno i kontinuirano dobijanje informacija o rasporedu resursa, dodeli zadataka, protoku dokumentacije, kao i upravljanje finansijama, vremenom i kvalitetom na projektu, dobar alat za upraljanje projektom.

### Приказ појединачних одговора:

If it does really automate workflow of all the participant roles, but not only supervisors.
Ako kao 'uspeh' mislite – da bude uradjeno ono sto je napocetku zacrtano onda Posvecenost projektu tima. Dobar project leader jako ponu znaci isto. Ali motivisan tim je presudan. Ako pod 'uspeh' mislite – da projekat napravi novac/vrednost onda mislim da je sve drugo vaznije od kodiranja – Design, Marketing, Sales, Copyright...
1. Needs (or Requirement) of S/W project should be understood at initial stage; 2. The enitre S/W Project has been split up in a no. of phases, though at the end all these phases are to be these parts are to be linked / coordinate as per the requirement. 3. Dedication in the work play one of the important role in success of any project.
* Proper ways of requirement gathering (TOBE Documentation) * Time to time positive reviews of project pieces at end user level * Selection of right development tool.
Team work, collaboration.
Knowing the detailed requirements for the project and the target deadlines. Hence, Communication is the key.
Comfortable environment, good organizing, the usage of modern program development tools.
Experience, good planning, team work.
Good planning and good collaboration between team members and between team leads.
Experience, skills, used CASE-tools.

- funkcionalne specifikacije:
- krajnji korisnik treba da ucestvuje u izradi i validaciji,
  - dobro izbalansirati izmedju funkcionalnih zahteva i resursa za realizaciju
  - dobra dokumentacija
- tehnicka realizacija:
- brojnost i iskustvo realizacionog tima
  - odabir tehnologija (framework) – nova tehnologija trazi vreme za učenje
  - komunikacija/jezik (par projekata moje firme je radjeno eksterno sa inostranim partnerima. Kada nije isti maternji jezik to otezava razumevanje)
  - testiranje
  - redovno pracenje stanja projekta od strane sefa (sefova) projekta.

Use of appropriate tools that encourages collaboration , assist in managing project epic values and deliver products, services or results within stipulated budget and timeframe. The tools helps in ensuring quality assurance standards are adhered to, risks are managed proactively and resources used optimally.

### Питање 7 - Шта по вашем мишљењу и искуству највише утиче на неуспех софтверског пројекта?

Табела 9.7.6. Аналитички приказ одговора о факторима неуспеха софтверског пројекта, класификованих према категоријама

КАТЕГОРИЈА	СТАВКЕ ОПИСА
Организација и управљање пројектом	<p>losa organizacija, nepostovanje rokova, neorganizacija, nerealni rokovi, proizvodavanje rokova zavrsetka, neorganizacija, probijanje rokova, losa priprema, nepreciznost (rasplinutost) postavljenih ciljeva, propust kod planiranja obima projekta, losa procena vremena potrebnog za odredjene aktivnosti, Lose uradjen plan I nedovoljno podrške od viseg menadzmenta, projekat, Nedovoljna definisanosti zadataka svake projektne faze, Los marketing, lose definisan i/ili odradjen projekat, Loše definisan projektni zahtev, loša upravljanje životnim ciklusima projekta, nije utrošio dovoljno vremena za definisanje posla, Pogresan izbor metodologije razvoja, dugi stintovi razvoja bez kontrole kvaliteta, Lose vodjenje tima, loše sastavljen tim, Ignorisanje perspektive korisnika jer korisnik je taj koji ce da koristi finalni softver a ne ljudi koji su ga razvili, Menjanje zahteva, loše isplanirani projekti, Ogranicavanje, konkurencija, planirnje svih faza i blagovremeno reagovanje, los menadzment, Kratki rokovi za zavrsetak projekta, menjanje zahteva od strane klijenta, Loša organizacija poslova, Loše definisani zahtjevi, Nejasna funkcionalna specifikacija, Promena zahteva tokom realizacije projekta. Loše postavlljen projektni zadatak, loše definisan plan projekta, Loša organizacija posla i ljudi koji rade, Česte promene ideja, nepoznavanje domena problema, loša organizacija i podela zadataka, propust kod planiranja obima projekta, menadžment (ako se uzme u obzir da je kod dobro napisan, da je dizajn dobro urađen, da je prodaja uradila dobro svoj deo posla), Probijanje milestoneova, Losa procena zahteva I procena vremena, Loše rukovođenje, Lose planiranje, Waterfall pristup razvoju softvera, loše definisani taskovi, Menadžment koji prodaje projekat bez stručne konsultacije. Nejasan cilj, i prebrzo i nedovoljno planirano upuštanje u projekat. loše liderstvo, česte izmene specifikacije/funkcionalnosti, cesta izmena predefinisanog koncepta softvera, losa raspodela taskova po timu, nerealni ugovoreni constraints, gold plating, nedostatak kontrole na projektu (uključujući odgovarajući QA), Neorganizovanost, Nedovoljno precizni zahtevi korisnika, neuredne i nejasne specifikacije, neprecizna business analiza, Kratki rokovi ili nepridržavanje predviđene dinamike, Neuređenost, Prekratki rokovi za razvoj, gruba greška u proceni obima projekta, sporost u odlucivanju zbog nefleksibilne organizacije, nejasni ciljevi ili previse zeljenih feature-a, menjanje ciljeva u pokusaju da se svakom klijentu izadje u susret, Nerealni rokovi i ocekivanja klijenta, nedefinisani i nepotpuni zahtevi, ne sagledavanje kompletnog problema u ranim fazama razvoja, rad bez planiranja, Mijenjanje opsega projekta u toku projekta (mijenjanje scope-a), neefikacni menadžment i oversize tima, Nedovoljno pripremljena projektna dokumentacija, nedovoljno definisani zahtevi, drugi prioriterniji projekti, nedovolno planiranja i diskutovanja u slucaju timskog projekta , previse planiranja i diskutovanja, Promenljivi ciljevi projekta, neumeren rast apetita tj dodavanje novih zahteva, loš izbor redova</p>

	kompletiranja issue-a, Nerealan schedule (rokovi), prebukiranost SME (Subject Matter Expert)-jeva, fluktuacija seniora, amorfan projektni scope, Loš menadžment, odsustvo procenja/kontrole, odsustvo zacrtanih ciljeva (kada se zahtevi cesto menjaju), nerealni/preobimni zahtevi, neorganizovanost tima, nedostatak plana. Organizacija u toku projekta: ukoliko se projekat ne završi na vreme, naručilac šalje nove zahteve, isporučilac menja projekat, to uzrokuje novo kašnjenje, a zatim i novu promenu plana. I tako u krug, do potpune propasti projekta. Nepoznavanje trenutnih aktivnosti saradnika, lose planiranje, los timeline.
Активности у развоју софтвера	pogresna procena da li softver ispunjava ocekivanja, Lose testiranje, Neadekvatno testiranje softvera u neodgovarajućoj sredini, slaba kontrola kvaliteta, Preterana upotreba shotgun surgery refaktorisanja, preveliko cepkanje problema na delove, Ne korišćenje source control-a. Nepostojanje modela, izostanak testiranja softvera, korišćenje zastarelih tehnologija, nedostatak kontrole na projektu (uključujući odgovarajući QA), nedovoljna pokrivenost testovima, nedostatak manuelnog testiranja, losa izrada softvera, Loš izbor alata za rad, sporost realizovanja rešenja, slabo ili nikakvo testiranje, nedovoljno testiranje.
Карактеристике решења	slaba kontrola kvaliteta, pogresna procena da li softver ispunjava ocekivanja, Nedovoljno definisani HW I interfejs, Koriscenje novih tehnologija bez prave opravdane potrebe, dizajn ali u smislu kada se nadje ispred upotrebljivosti I funkcionalnosti, nefunkcionalnost, loš prototip, los kvalitet programskog koda, haotican kod, Korisnički interfejs, Kompleksnost projekta, losa arhitektura.
Документација	loša dokumentovanost projekta (napretka i problema koji se javljaju u realizaciji), Izostanak dokumentacije na projektima gde radi više od 10 ljudi, losa/nepotpuna dokumentacija.
Личне особине учесника пројекта	netimski rad, project manager, nezainteresovanost I slaba obucenost, prevelika ocekivanja. Nedovoljno ili neiskusni ljudski resursi, pogresna procena da li softver ispunjava ocekivanja, neiskustvo, Nekompetentan rukovodioc projekta, tim developera i test developera, iskustvo učesnika projekta, Tehnološka nekompetentnost i pritisak zbog budžeta i vremenskih rokova, prejaka disciplina, tj. Jasna granica izmedju sefa I ostatka tima u smislu "nize vrednosti", Nerazumevanje I nerad, nerealna ocekivanja "naručioca" projekta, nivo znanja (stručnosti) svih učesnika projekta, losi odnosi u timu, površnost, neorganizovanost,nesaradnja u timu, neiskustvo programera, stručnost projektnog tima, nekvalitetan PM, Dekoncentracija, aljkavost u radu, nekvalitetni saradnici, neupućenost ljudi (koji nisu programeri) u kompleksnost softverskog projekta, nedostatak motivacije, neiskustvo pisanja softvera, nejedinstvo u timu, lični stav prema projektu, međuljudski odnosi, Potreba za ličnim isticanjem. Ljudi koji ne znaju da programiraju na većim softverskim rešenjima.nestručnost tima ili pojedinaca, Menadzer projekta, neiskustvo, nedostatak saradnje između programera, nekvalitetni ucesnici (na prvom mestu project manager, na drugom developer), Sujeta, Sastav tima, Asocijalno delovanje članova tima, Nedovoljna sposobnost motivacije od strane tim lidera, Nesposobnost upravljanja projektom od strane vodje projekta, los tim, perfekcionizam (teznja za savrsenim softverom koja dovodi do kasnijeg izlaska na trziste), nemogucnost tima da se prilagodi klijentskim zahtevima,Los timski rad, los klijent, nespremnost na adaptiranje, Ljudi, nesaradnja u timu, neiskustvo projektnog tima, slab realizacioni tim (brojnost/iskustvo itd), Nefleksibilan stav prilikom izrade softvera, Nepoznavanje mogućnosti saradnika.
Ресурси	manjak resursa,bolje iskoristiti postojece resurse, Premali budzet, loša procena potrebnih resursa (vreme, novac, ljudski resursi), budžet projekta,nedostatak finansiranja etapa projekta, budžet.
Комуникација	nedovoljno informacija o nameni softvera od strane narucioca, losa komunikacija, losa komunikacija izmedju korisnika aplikacija I projektanata, losa komunikacija u timu, kontakt sa buducim korisnicima projekta, losa komunikacija sa klijentom, losa komunikacija, nerazumevanje korisnik - projektant, loša međusobna komunikacija, neadekvatna komunikacija među članovima tima. Ignorisanje perspektive korisnika jer korisnik je taj koji ce da koristi finalni softver a ne ljudi koji su ga razvili, Nedovoljno efikasna komunikacija, nejasna specifikacija i nerazumevanje domena problema koji se rešava, nedovoljna informisanost, loša komunikacija, Loša komunikacija u timu, Nerazumevanje i loša komunikacija članova tima unutar projekta, nejasna komunikacija, nedostatak kontakta sa klijentom koji je narucilac posla, loša

	komunikacija članova tima, loša komunikacija, neuspeh u razumevanju zahteva klijenta, loša komunikacija u timu, Nedovoljna komunikacija, Loša komunikacija, Nejasan cilj, Mogucnost lose komunikacije izmedju klijenta i PM-a, loša komunikacija među radnicima i predpostavljenim, Nejasnoca klijentskih zahteva, nedostatak komunikacije ili neefektivna komunikacija, nejasni ciljevi, loša komunikacija, pogotovo između business i development delova, Nejasna funkcionalna specifikacija, loša komunikacija, Neprecizne informacije, medjusobno nerazumevanje.
Алат за управљање пројектом	-

**Приказ појединачних одговора:**

Scope creep
Supervisor-oriented automation.
a) Over confidence b) Ignoring any suggestions from the lower / tail-end team mats. c) Insufficient knowledge about programming languages, tools etc.
* Wrong interpretation of user's requirements * Too long development period * Selection of wrong development tool
Lack of team work and collaboration
A gap in understanding requirements between a novice user and the software developer leads to failure. An analyst can add as a buffer between these two key players.
The lack of project full understanding and the lack of detailed elaboration of technical specifications
Incomplete requirements
Poorly implemented solutions or solutions that hit the market too late for their time
skills
Failure to link project objectives to organization strategic goals- role of OPM, lack of support and guidance from PMO, Lack of appropriate application/use of project management knowledge, skills, tools, techniques and appropriate resources.

**Питање 12 – Које су предности коришћења развијеног прототипа апликације [www.it-project.rs](http://www.it-project.rs)?**

ГРУПА КАРАКТЕРИСТИКА	ОПИС КАРАКТЕРИСТИКА
УТИЦАЈ НА УСПЕШНОСТ ПРОЦЕСА У УПРАВЉАЊУ ПРОЈЕКТИМА, ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ	
Контрола и организација	<p>pracenje razvoja od pocetka do kraja, pracenje razvoja softvera po fazama, pracenje realizacije projekta, kontrola nad softverskim projektom, pracenje razvoja, sta je sve uradjeno, pracenje svih faza razvoja softvera, Centralizovan pogled na tok realizacije faza projekta, Upravljanje softverskim projektom „sa jednog mesta“, sagledavanje celokupnog rada iz druge perspektive, Mogućnost kompletnog praćenja i upravljanja realizacijom projekta od njegovog početka do kraja. članovi tima moraju da budu odgovorniji jer svi mogu da vide ko je koliko uradio od posla koji im je dodeljen, lakše praćenje napredka u realizaciji projekta. Може да помогне у процени времена потребног за реализацију пројекта у целини, појединачних активности као и болјој организованости. Болја организација пројеката, омогућава да се прати пројекат од његовог почетка па све до његовог краја. pracenje rokova, pracenje gresaka, Mogućnost merenja razlicitih faktora, izvlacenje korisnih metrika za unapredjenje daljeg procesa. upravljanje projektom, upravljanje promenama, paralelna realizacija, bilo kakva organizacija je bolja od nikakve, a ako je jos u elektronskom obliku tim bolje, Pracenje i upravljanje procesom razvoja softvera, praćenje</p>

	<p>prograsa, Praćenje stepena razvoja pojedinačnih projekata. planiranje i praćenje realizacije projekta, Lakse je ispratiti napredak projekta, lakse se raspoređuju zadaci među članovima tima, Olakšava praćenje razvoja softvera, kontrola realizacije, informisanost o stepenu realizacije taskova članova tima radi preventivnog delovanja, fazni pregled svih nerealizovanih projekata radi adekvatne procene i planiranja sledećeg projekta, Prvenstveno to sto se u svakom momentu može pratiti napredak u razvoju određenog softvera, Mogućnosti upravljanja projektom i distribuiranim timovima. Svi članovi pojekata tačno imaju pregled dokle se stiglo sa projektom i koji su tekući problemi, Pracenje toka projekta. lakši i organizovaniji način upravljanja razvoja projekta. ovakva aplikacija doprinosi fluidnosti razvoja softvera, praćenje razvoja, merenje vremena izrade, prioritet taskova. dobra organizacija, sistematizacija, organizacija, Mogućnost praćenja razvoja softvera, Jasno praćenje projektnog ciklusa, Ubrzanje ispitivanja softvera, praćenje procesa razvoja, brža analiza rada i ispitivanja kvaliteta urađenog posla, Mogućnost pracenja zadataka, Uvid članova tima o trenutnom stanju projekta. mogućnost da se isprate koji su projekti pokrenuti, Razumljiva organizacija i navigacija, zna se ko radi na kom projektu, praćenje šta se uradilo u kojoj sekciji / iteraciji projekta, Upravljanje zadacima je mnogo lakse i menadzerima i radnicima, Organizacija ljudskih resursa i zadataka, Praćenje razvoja projekta u realnom vremenu, merenje uspešnosti razvoja i odgovaranja na zahteve klijenata. Podrška odlučivanju u vezi sa procesom razvoja softvera, Bolja organizacija i podela.</p>
Тимски рад	<p>podrska za timski rad, kolaboracija u radu na softverskim resenjima, razmena iskustava I pomoci, Saradnja sa članovima tima u realnom vremenu. saradnja među članovima tima bez obzira gde se nalaze. Omogućuje teorijski neograničenom broju ljudi, da saraduje i učestvuje u razvoju i dokumentovanju projekta, mogućnost kolaboracije, Članovi tima mogu da komuniciraju bez obzira gde se nalaze, lakša saradnja među članovima tima, Mogućnost odabira saradnika po više osnova, lako uključivanje novog člana tima, laka koordinacija, Kolaboracija i razmena iskustava učesnika različitih projekata. omogućen je kvalitetniji timski rad, pojednostavljeno uključivanje novih saradnika na projektu, olaksava timski rad, Timski rad. Lakša razmena informacija među učesnicima, mogućnost centralne komunikacije između klijenata, project manager-a i developer-a, kao vid kolaboracije. Uvid članova tima o trenutnom stanju projekta. Razumljiva organizacija i navigacija, zna se ko radi na kom projektu.</p>
Ресурси	<p>bolje upravljanje resursima, Brže upravljanje resursima, deljenje projektnih resursa, svi neophodni resursi su dostupni, pristupiti materijalu na jednom mestu, proveriti raspoloživost ljudskih resursa, kontrola resursa, Optimalnija upotreba resursa. raspored raspoloživih resursa, Organizacija ljudskih resursa.</p>
Комуникација, информисаност, прегледност, транспарентност	<p>spisak projekata, informacija o temama I napretku softverskih resenja, Transparentnost datoteka i dokumenata, Komunikacija, saradnja među članovima tima bez obzira gde se nalaze. brže generisanje izveštaja, ovakva aplikacija omogućuje objedinjavanje i praćenje velikog broja podataka bitnih za projekat, olakšava njihovo pretraživanje, praćenje i ažuriranje, mogućnosti detaljnijeg informisanja, Članovi tima mogu da komuniciraju bez obzira gde se nalaze, uvid o informacijama o određenom projektu, omogućavaju pojedincu ili čitavom projektnom timu da prikupi i obradi podatke, pretraži ih, nadgleda i ažurira, Pregled taskova i komunikacije u toku razvoja projekata, razmena informacija o statusu delova projekta, kolekcija dokumentacije i izvora podataka, kolekcija test podataka i očekivanih izlaza, Praćenje i dokumentacija resursa, zadataka, informacija, pohranjivanja projektne dokumentacije, Nudi sagledavanje svih projekata na jednom mestu, preglednost, Sveobuhvatni prikaz aktivnih i završenih projekata i njihovih nosilaca. formiranje baze znanja koja bi se kasnije koristila u praksi. records omogućuje da se vidi ko je na čemu radio, Svim učesnicima informacije su dostupne na jednom mestu, transparentnost, Konstantna i sveprisutna informisanost o tokovima razvoja softvera, jasan uvid u trenutno stanje projekta, Svi članovi pojekata tačno imaju pregled dokle se stiglo sa projektom i koji su tekući problemi, Dostupnost informacija svim članovima tima i zainteresovanim stranama u svakom trenutku. olaksava komunikaciju, Bolji pregled na projekat, Lakša razmena informacija među učesnicima, Dobar pregled raspoloživih projekata i njihovih delova,</p>



	ucesnika/resursa, uvid u stanje tekucih/završenih/stopiranih projekata, detalji o projektima (opis, realizatori, pridruzeni dokumenti, itd). mogućnost centralne komunikacije između klijenata, project manager-a i developer-a, kao vid kolaboracije.Uvid članova tima o trenutnom stanju projekta. Razumljiva organizacija i navigacija, zna se ko radi na kom projektu, pregled gde su problem.
Визуализација	Vizuelizacija faza razvoja.
Квалитет софтверског производа	kontrola softvera, U velikoj meri bi olakšala rad u smislu praćenja i kontrole kvaliteta izrade softvera.
<b>КАРАКТЕРИСТИКЕ ПРОТОТИПА WWW.IT-PROJEKT.RS, НАМЕНА И УТИЦАЈ НА УСПЕХ ПРОЈЕКТА</b>	
Једноставност коришћења прототипа	jednostavan za upotrebu, adekvatan nacin pristupa informacijama,Dobra preglednost zbog malog broj opcija je čine jednostavnom za upotrebu.Jednostavnost, jednostavnost koriscenja, Jednostavan ticketing sistem. Kroz nekoliko "klikova" može se doci do korisnih informacija o nekim temama,ucesnicima, relativno lak pregled statusa projekta, adresar učesnika, ... ukratko, važne informacije vezane za projekat/projekte na jednom mestu.
Брзина креирања прототипа	brza implementacija, Brzina realizacije.
Намена прототипа	za manje projekte u okviru naučnih organizacija, Organizacioni deo može imati koristi od ovakve aplikacije u slučaju da se radi o sistemu sa velikim brojem projekata. U većem timskom okruženju ovakva aplikacija doprinosi fluidnosti razvoja softvera, praćenje razvoja, merenje vremena izrade, prioritet taskova. Usko specijalizovana aplikacija, za upravljanje isključivo softverskim projektima. Većina sličnih aplikacija su pravljene za upravljanje bilo kojim projektima, pa je često potrebno, a nekada i nemoguće, prilagoditi ih softverskim projektima.
Коришћење прототипа убрзава рад	ubrzava rad, povećava produktivnost, brži način upravljanja razvoja projekta. korišćenje softvera ne treba da zahteva mnogo dodatnog vremena, usteda vremena, brže uočavanje problema na razvoju softvera, Ubrzanje ispitivanja softvera, praćenje procesa razvoja, brža analiza rada i ispitivanja kvaliteta urađenog posla.
Коришћење прототипа смањује грешке	Smanjuje mogućnost greske.
Доступност саме апликације прототипа и података	Mogućnost pristupa sa bilo kog mesta i bilo kog uređaja (pošto je web aplikacija u pitanju), Organizovan pristup pohranjivanju projektne dokumentacije koja je lako dostupna, Članovi tima mogu da komuniciraju bez obzira gde se nalaze, Dostupnost, Dostupnost na svakom mestu gde ima interneta, Svim učesnicima informacije su dostupne na jednom mestu, Lak pristup svemu potrebnom, Може свако да прати стање пројекта.Svi članovi pojekata tačno imaju pregled dokle se stiglo sa projektom i koji su текући проблеми, Dostupnost informacija svim članovima tima i zainteresovanim stranama u svakom trenutku. Могућност учествовања на пројекту независно од географске локације. Pristup aplikaciji bez instalacija, bez licenciranja softvera, bez muke, rađen u formi web aplikacije, ne trebaju dodatne instalacije za korišćenje, može se pristupiti sa bilo kog uređaja (računar, telefon...) koji je povezan na internet, Dostupnost,nezavisnost od platforme i sve ostale prednosti web aplikacija, ONLINE dostupnost, nema potrebe za instalaciju softvera na računar. Kroz nekoliko "klikova" može se doci do korisnih informacija o nekim temama,ucesnicima, Uvid članova tima o trenutnom stanju projekta. web-based aplikacija, Mobilnost, pristupačnost, relativno lak pregled statusa projekta, adresar učesnika, ... ukratko, važne informacije vezane za projekat/projekte na jednom mestu.
Свеобухватност функција, обједињеност података	sveobuhvatnost aplikacije,pracenje razvoja od pocetka do kraja, pracenje svih faza razvoja softvera, Objedinjenost podataka.
Прецизност	Preciznost.
Повећава успешност пројекта	povecanje uspesnosti projekta.

**Приказ појединачних одговора:**

<p>Program služi svrsi. U ovom vremenu kada je veoma aktuelno projektno planiranje jedan ovakav softver je velika pomoć jer sadrži potrebne podatke o učesnicima u projektu kao i o projektima koji su aktuelni. Program treba da ima dobar način pristupa informacijama, da je jednostavan za upotrebu i da se brzo implementira.</p>
<p>Potentially all members of the team could log in and check on project support tasks.</p>
<p>Get 24 hr /7days support and availability of different teams to address the issue tickets within a short notice.</p>
<p>Under globalization, all business are e-business, hence most of the vendors wants their business should be in global market. Therefore web projects are more in demand.</p>
<p>If the fish-bone workflow management is incorporated in this application, it could be the fine application which shall be used during distributed project management.</p>
<p>Project tracking aplikacije generalno imaju prednosti:                  * Sa individualne strane: uvek imas listu svojih taskova sa neophodnim informacijama. Takodje uvek znas fazu u kojoj se nalazi tvoj task                  * Sa organizacijske strane: uvek postoji informacija o tome koji ljudi su angazovani na kojim taskovima, da li je neko preopterecen ili obratno, prati se sama uspesnost realizacije celog tima i pojedinacno itd...</p>
<p>No advantages with respect to, e.g., Bugzilla and Redmine.</p>
<p>Provides a very simplistic view of some IT entity concepts (Participants, Topics -- not that clear in the IT world what these are; Projects/Tasks).</p>
<p>With the world becoming internet savvy, its an advantage to have web based appliations specially in business's spread across distances. The feasibility of usage increases along with updating correct data, using it at any time.</p>
<p>It may be useful for shared (in team) projects development, debugging, bug fixing.</p>
<p>Friendly interface.</p>

**Питање 13 – Који су недостаци коришћења развијеног прототипа апликације [www.it-project.rs](http://www.it-project.rs)?**

Табела 9.7.7. Аналитички приказ одговора о недостацима развијеног прототипа, класификовани према категоријама

ГРУПА КАРАКТЕРИСТИКА	ОПИС КАРАКТЕРИСТИКА
<p>POTREBNA INTEGRACIJA SA DRUGIM ALATIMA, OFF - LINE RAD SA MATERIJALOM I UPLOAD-DOWNLOAD</p>	<p>Download pa onda rad - pokusati da se napravi da moze online da se ispravi I dopuni materijal, svaka ispravka mora da se ponovo preuzima, bez mogucnosti online nadogradnje, Nepovezanost sa sistemom za upravljanje kodom (GIT, svn, bazaar,...), nepovezanost sa sistemom za pracenje bugova.Nema podrške za direktnu integraciju sa razvojnim okruženjima (svodi se na skini fajl, pročitaj, pa integriši) čini mi se da sistem za jednoznačno održavanje strukture rešenja (kada se baš razvija softver) ne postoji, kao što ima GitHub, SVN itd, ili je osnovna namena razvijene aplikacije da se koristi u kombinaciji sa prethodno spomenutim rešenjima (ako sam ja dobro razumeo koncept aplikacije)</p>

<p>VREME I NAPOR ZA UNOS PODATAKA (svrha i korisnost zavisi od velicine tima i projekta), POTREBA ZA ODRŽAVANJEM AŽURNOSTI PODATAKA</p>	<p>Mislim da je u slučaju nekog manjeg projekta vreme potrebno za ažuriranje ovakve aplikacije znatno veće od koristi koje on donosi. Svakodnevno ažuriranje tokom izrade projekta. Jedini nedostak bi bilo utrošene vreme za unos podataka u aplikaciju, ali s obzirom na potrebu pitanje je da li je to utrošeno vreme nedostatak...Potencijalno velik dodatan napor u održavanju stanja azurnim.Nedostatak bi mogao da bude nepotrebno komplikovanje procesa realizacije projekta sa dodatnim opterećenjem za inženjere. Jedino ako se radi veoma mali projekat od strane pojedinca u scenariju „one man show“ gde ne postoji interes ili potreba za informacijama ovog tipa, onda je evidentiranje podataka kroz ovakvu aplikaciju nepotrebno trošenje resursa, ali opet, i tada može da bude korisno, pogotovo gledano u širem vremenskom okviru. odbojnost malih timova koji to shvataju kao dodatno opterećenje i nepotrebnu papirologiju (mora biti krajnje jednostavna i prosta za upotrebu a ujedno kvalitetna i korisna što je svakako izazov implementatoru aplikacije), Na manjim projektima nije neophodna ovakva aplikacija, vreme potrebno za administraciju.Moze zahtevati dodatnu administraciju samog sistema, jer se mogu javiti problemi tipa neazurnosti I naravno moze zahtevati malo dodatnog vremena za sam unos progresa, veci utrosak vremena (Video sam da za manje projekte koriste I MS Excel), administracija (moze da optereti proces proizvodnje), Unosenje podataka moze da uspori projekat. Za prostije i kraće projekte može oduzeti previše vremena, a doprineti malo. Suvise dokumentacije moze da bude prepreka za brzu izradu softvera. Potreba za dodatnim resursima koji ce se baviti upravljanjem procesa i tumačenjem prikljupljenih podataka.</p>
<p>KOMUNIKACIJA ČLANOVA TIMA</p>	<p>komunikacija, nedostatak komunikacije.</p>
<p>BRZINA RADA APLIKACIJE</p>	<p>moze biti sporija od desktop aplikacije, obratiti pažnju na brzinu rada aplikacije.</p>
<p>BEZBEDNOST PODATAKA I TEHNIČKA POUZDANOST SISTEMA</p>	<p>zloupotreba podataka, pažnju treba obratiti na zloupotrebu softvera, u pitanju web aplikacija kojoj se moze pristupiti preko interneta moze da bude I problem ukoliko se ozbiljno ne poradi na bezbednosti iste, treba staviti pozornost na security i power strategiji jer bi cjelokupan projekat zavisio od servera koji je na internetu.Security (ima projekata koji barataju senzitivnim podacima koji moraju da se cuvaju na lokalnim serverima), Prekid interneta onemogućava pristup i rad sa podacima, Mogući sigurnosni propusti koji mogu prouzrokovati neautorizovan pristup projektnim dokumentima.</p>
<p>ANALIZA PODATAKA, PODRŠKA ODLUČIVANJU</p>	<p>Evidenciona komponenta je dominantna, za razliku od analitičke, komparativne i problemske.Nedovoljno razvijeni pretraživčki alati. omoguciti unos statusa projekata i drugih informacija koje su potrebne korisniku (u tom momentu konkretnom sistemu), kao i pretrage/filtere po tim podacima, Trebalo bi pojačati modul za praćenje napretka projekta, staticka analiza projekta, postaviti filteri za informacije, Da planiranje posebno bude odradjeno i posebno odvojeno od realizacije. Na osnovu uloge lakse bi pratili korisnici svoj deo. Postoje mnogobrojni načini tumačenja uspešnosti i napretka nekog softverskog projekta (.Ne postoji definicija za njih. Oni se mogu razlikovati od projekta do projekta. Obično ih definiše tim koji radi na tom projektu. Podložni su promeni tokom trajanja projekta.).</p>

<p>FLEKSIBILNOST U ODNOSU NA METODOLOGIJE I POSLOVNE PROCESSE</p>	<p>Ponekad nedovoljna fleksibilnost i prilagodljivost poslovnim procesima. Generalizovana metodologija upravljanja projektima, reklo bi se kao PMI (Project management institut) framework, bez mogućnosti upotrebe konkretne metodologije. Nedovoljna fleksibilnost, Sve aplikacije koje sam koristila su imale zajednočku manu, a to je fleksibilnost. Potrebno je osmisliti alat koji prati neku od poznatih i korišćenih metodologija za project management (npr PRINCE2, PMP, SCRUM etc). Postoje mnogobrojni načini tumačenja uspešnosti i napretka nekog softverskog projekta (.Ne postoji definicija za njih. Oni se mogu razlikovati od projekta do projekta. Obično ih definiše tim koji radi na tom projektu. Podložni su promeni tokom trajanja projekta.).</p>
<p>ZNANJE, SAVESNOST, MOTIVACIJA KORISNIKA ZA KORIŠĆENJE</p>	<p>Nezainteresovanost izvodjaca, Uglavnom se ne koriste ovakve aplikacije, Svi učesnici u projektu moraju da imaju visok nivo IT pismenosti da bi mogli da koriste ovakvu aplikaciju. Ukoliko se učesnici projekta motivišu da uniformno koriste samo ovu aplikaciju, onda njeno korišćenje be može da prouzrokuje nikakve nedostatke. ljudi koji rade na razvoju aplikacije su najveći nedostaci, jer ako se ne popunjava kvalitetnim ulaznim podacima o tome koliko da je dobar softver, davaće nepotpune rezultate. Učenje novog softvera.</p>
<p>DODATNE FUNKCIJE, MODULI</p>	<p>Ne deluje mi dovoljno detaljno, Aplikacija podržava samo najosnovnije funkcije vezane za upravljanje softverskim projektima. Mislim da softver trenutno nije dovoljno razvijen. Obzirom da je ovo prva verzija aplikacije, naredne verzije bi nudile dodatne funkcionalnosti izostavljene u prvoj verziji.uvesti ocenjivanje, odnosno pozitivne i negativne reference za učesnike koje bi mogle da daju kolege na istom projektu (poput onoga na sajtu www.kupujemprodajem.com), trebalo bi dodati još funkcionalnosti, potrebno: Praćenje aktivnosti u toku izrade task-a, kada je startovana izrada task-a, da li je task prebačen na nekog drugog, komentari.Dodeljivanje taskova odredjenoj osobi. Kalendar, Mogucnosti paralelizacije rada na taskovima/topicima. Pregled vise ljudi odjednom. postaviti filteri za informacije, omogućiti kod spiska projekta I prikaz detalja svakog projekta, razdvajanje pojednih delova softverskog projekta u posebne module. da se opiše detaljnije svaka uloga učesnika u projektu kao i širi opis stavki vezanih za projekat.</p>
<p>SLOŽENOST UNOSA PODATAKA U KASNIJEM UNAPREĐENJU</p>	<p>Moze da preraste u komplikovanu tj. Obimnu ukoliko se ne fokusira na nesto konkretno.</p>
<p>DOSTUPNOST ZBOG INFRASTRUKTURE</p>	<p>Jedino ako nema interneta ali to je sada retka pojava sobzirom na mobilne pakete koji omogućavaju neograničenu upotrebu interneta, možda jedino u inostranstvu zbog visokih cena rominga.</p>
<p>KORISNICKI INTERFEJS</p>	<p>poboljšati interfejs da bude više user friendly i vizuelno sofisticiraniji, los UI/UX, Korisnicki interfejs mora biti mnogo laksi za unos podataka i pregled, da bude optimizovan za mobilne uredjaje.Zbunjujuci GUI. UI, losa organizacija, graficki interfejs - vizualni deo, Estetika interfeca i intuitivnost pri koriscenju, nekonzistentnost interfejsa u razlicitim segmentima aplikacije.</p>

VIZUALIZACIJA I IZVEŠTAVANJE	Nedovoljna vizuelna preglednost, uglavnom lose opcije za stampanje odredjenih pregleda za potrebe team lidera (i dalje je papir sa skicama na stolu jednostavno nepremostiva potreba tim lidera), Nema nikakve vizuelizacije taskova, da li je task izvršen, u toku ili na čekanju, Da li je task kritičan ili je nice to have. Dodatna vizuelizacija informacija, veci broj podataka da bude vdljiv na prvi pogled, kada se korisnik uloguje ne vidi nikakve nove relevantne podatke. svi segmenti su u jednoj tabeli bez mogućnosti filtriranja.
PREGLEDNOST I INFORMATIVNOST ZA NACIN KORISCENJA (UPUTSTVA)	aplikacija je konfuzna, Ljudi teško prihvataju aplikacije za koji ne vide smisao ili ne znaju da ga koriste. Kome je namenjena ova app? ako komunikacija ide preko ovakve aplikacije, ona treba da bude što jednostavnija za upotrebu, pregledna, moguć nedostatak jasne navigacije kroz segmente aplikacije, neintuitivne skracenice (bez pop-up helpa), Nedostatak realnih primera za upoznavanje sa radom aplikacije, Nepreglednost.
NETAČNI ZAKLJUČCI	ne postojanje komunikacije (pisane i razmene dokumenata), ako komunikacija ide preko ovakve aplikacije, ona treba da bude što jednostavnija za upotrebu, pregledna.
GOTOVA REŠENJA	Pretpostavljam da postoji vec gotovih resenja na Internetu sto opet ne mora da bude lose jer moze da da inspiraciju za poboljsanja.

**Приказ појединачних одговора:**

Lack of control and Accountability.
Without information security, any web application will caused greater disadverages.
Bit langthy. No 'Reply to' option for records. All records are visible to all participants and accessible.
Sve aplikacije koje sam koristila su imale zajednocku manu, a to je fleksibilnost. Omogucavanje jedne aplikacije da bude fleksibilna znaci da ukoliko korisnik zeli sam da napravi plugin koji ce mu doprineti u procesu - kod mora biti razumljiv i prilagodljiv. Imala sam slucaj da smo krenuli Active Collab da prilagodjavamo svojim licnim potrebama a kod je trebalo mesec dana raspetljivati, kreirali su svoj framework bez ikakve dokumentacije i na kraju je ispao sizifov posao.
Today it does not supply any of above mentioned (in the survey) featues.
<ul style="list-style-type: none"> <li>• No graphical display of project progress, assignments</li> <li>• Poor quality graphics for a web application</li> <li>• Unclear way of use (no documentation), non-intuitive</li> <li>• Comparing to Microsoft Project or other Project Management applications, this application has a long way to go.</li> </ul>
<ol style="list-style-type: none"> <li>1. One of the major concern would be security and privacy of data</li> <li>2. Second would be develop the application in correct way which enables easy regular updates on the server.</li> </ol>
Uninformative interface, the order of actions intuitively understood. There are no any instruction/description of IT-project system description.
In my opinion, there is no task sequence progress, completed tasks, etc.

**Питање 14 – Које софтверске функције сматрате да треба да има оваква апликација, а нису подржане у постојећем решењу? Питање 15 – коментари и сугестије.**

**Сугестије у вези потребних софтверских функција:** Mogucnost testiranja gotovog soft, Online edit и update sadržaja, detaljni opisi svih projekata, ocene istih I rezultati svakog ponaosob, Planiranje, Problematika, Troškovi, Diskusija.aplikacija ovog tipa trebala bi da objedini sledeće funkcionalnosti: Document management za skladištenje projektne dokumentacije i modela (podržano), Task management, Bug tracking, Project management sa gantogramom za planiranje vremena. Grafički prikaz napretka softvera, jednostavniji pregled zadataka, mogućnost dodeljivanja prioriteta i osoba postojećim zadacima. Obaveštavanje o realizaciji putem e-maila, Administracija korisnika, Vizuelizacija podataka, Deo za budžet/troškove projekta, Opcije za dodavanje, editovanje i brisanje topic-a, mogućnost direct-link kontaktiranja učesnika.Funkcije koje omogućavaju vizuelnu komponentu praćenja razvoja projekta.Kako bi bila korisna i za manje sisteme potrebno je unaprediti preglednost i citljivost podataka o statusima projekata. Vizuelni prikaz u kojoj fazi je projekat. Nešto kao Scrum burn down chart, tj. koliko je procenjeno vreme trajanja projekta i u kojoj fazi se projekat trenutno nalazi (koliko je realizovano). Pretraga projekta po imenu, Deo za komunikaciju između samih članova projekta kao i kordinatora.neki grafik ili procenat obavljenog posla (urađenog projekta). uvid u to ko je kada izvršio promene nad dokumentom i opcija vraćanja na neko od prethodnih stanja, kao vid zaštite. vizualizaciju i praćenje postignutog progressa projekta, responsive web design, treba voditi računa da je aplikacija kompatibilna sa mobilnim uređajima.Podrska za task detalje koji se koriste za razne procene - nadam se da ce to imati. sistem za kolaboraciju učesnika pojedinačnih projekata (mogućnost poziva učesnika projekta B da bude učesnik i na projektu A), detaljni izveštaji o statusu projekta (datum počta, nosioci, dokument sa ciljem (project charter možda), burndown chart i slično...), metodologija koju timovi koriste za rad (agilna, tradicionalna), trebalo bi omogućiti praćenje svakog posla u toku realizacije projekta: evidencirati ko je šta radio, koliko je vremena utrošio, i klasifikovati poslove prema obimu i zahtevnosti. Takođe treba dodati podršku za evidentiranje problema u realizaciji projekata, pošto se problemi uvek javljaju, a mogu se iskoristiti da se uči iz grešaka i da se one kasnije izbegnu. Možda dodavanje nekog dashboard-a za reviziju koda i deo za klijente gde mogu da postavljaju zahteve i prijavljuju bug-ove ili traže izmene nečega. Neizostavno, a pogotovo za realizaciju distribuiranog razvoja, video konferencije sa podrškom za document management; ne samo hronološki, već i hijerarhijski pregled projekta razbijenog na taskove; možda neki sistem evaluacije gotovih projekata od strane krajnjih korisnika. Uvođenje datumski ograničenih i zavisno povezanih poslova u sistem praćenja IT projekta. nedostaje vizuelizacija preseka stanja projekata (u vidu dijagrama ili drugih vizuelnih objekata koji ne zahtevaju traženje podataka po tabelarnim prikazima). Vreme trajanja taskova I vreme predviđeno za realizaciju projekta, prikaz putem gantt chart dijagrama npr. Funkcionalnost flowchart-a praćenje koraka projekta, approval process, bolja podrška ITIL-a, check-in/out objekata/dokumenata. Orjentisanost ka Projektnoj metodologiji i njena podrška (da li podržava AGILE, WATERFALL, PRINCE) Svaka od tih metodologija ima svoje specifičnosti i funkcije koje softver podržava bi se u tom slučaju razlikovale. Za kompletniju listu moralo bi da se uradi detaljnija uporedna analiza. dodati neki slikoviti, grafički, procentualni, statistički prikaz realizacije softvera. Implementirati fleksibilan način definisanja i praćenja napretka projekta ugledajući se na metodologije kao što su Kanban i Scrum. Takođe omogućiti fleksibilan način definisanja kvaliteta softvera. Bilo bi dobro da postoji mogućnost da se gledaju zadaci iz svih projekata, kao i da se prikaz zadataka filtrira. dodati machineKey u web.config, redefinisati trajanje sesije i uraditi bolju obradu grešaka. Testirati stabilnost web aplikacije, Jasno definisanje grupe korisnika kojoj bi aplikacija bila namenjena i koristi koje mogu da ostvare njenom upotrebom je vrlo bitno. Mogućnost uploadovanja različitih fajl ekstenzija je izuzetno korisna pri razmeni i deljenju resursa koji mogu biti različite prirode. Sugerisala bih da liste projekata budu interaktivne (u obliku linkova koji bi vodili resursima vezanim za određeni projekat). Mogla bi se ubaciti podrška za: issue tracking system, time tracking, Gantt chart, web calendar, News, documents & files management, wiki. Povezati issue tracking sistem sa softverima za praćenje verzija koda SVN, Git itd... Eventualno statistika koliko se vremena troši na pojedine faze. detaljnije opcije o svakom prokojektu ( ne mora da bude mandatorno). Integracija sa MS Office aplikacijama u modulu Project records\Documenting. detaljnije opise taskova po projektima, vreme predviđeno za realizaciju projekta I taskove, kolaboracija, lista projekata (u kojima se ucestvuje, ne svi), lista taskova po projektima, upravljanje

resursima (ljudima, dokumentima). Integracija sa mailom, na primer ako sam pomenut u necijem komentaru ili je ostavljen komentar na deo za koji sam ja zaduzen dobro je da dobijem mail. Hierarhijska struktura i mogucnost pregleda, projekat, phase, segment. Ako kliknem na jedan projekat svi komentari su vezani za projekat, sta se desava ako je projekat veci i ima stotine komentara? Da li su komentari grupisani ili nasumicni ili su vezani za jedan zadatak koji je vezan za korisnika, koji je status zadatka? Koji je status celog projekta i kako ga pratiti? vise vizualizacije dobro би doslo SWOT analiza, deljenje projekta na manje taskove i odredjivanje prioriteta tih taskova. izvoza podataka za popularne formate za softvere za tabelarne proračune i eventualno u formate za uvoz u baze podataka. Ne postoji mogucnost da se odredjeni zadatak dodeli nekome, prioritizira ili odredi njegovo predvidjeno i stvarno trajanje Recimo detaljniji uvid u projekat, opisi, pratece dokumentacije, datoteke. Workflow sistem, izveštavanje i analiza, portfolio manager Resource tracking, PERT, CP, budget! Reporting za top level management. Mozda je potrebno da ima vise linkova, kada se stane na osobu (ime, prezime) da tu bude link ka njegovim licnim podacima. Nisam se bavio kreiranjem ovakvih aplikacija ali gledajuci neke slicne aplikacije vidim da bi ovde moglo biti toga malo vise. Mogucnost organizovanja taskova u sprint-ove, po Scrum metodologiji. Pracenje ucinka i kasnjenja na osnovu tezinskih procena zadataka od strane developer-a. Fleksibilan način definisanja napretka projekta i kvaliteta softvera. Integracija sa GitHub, SVN, Microsoft Team Foundation server-om, Kada se klikne na osobu u projektu da se prikaže profil učesnika, kao i lista projekata na kojima isti učestvuje. Možda da se doda još i opterećenje, recimo pera perić radi na projektu 1 2h dnevno, proekat 2 4h dnevno itd... Bolji UX/GUI, Merenje vremena, Mesto za kolaboraciju (taskovi, statusi, komentari) Nedostaje upravljanje rokovim (vremenskom ograničenju zadataka), podeljenost u timove, upravljanje kontrolom kvaliteta, detaljniji opisi zadataka i podrška za agilne pristupe razvoju softvera. Funkcije koje se tiču upravljanja projektima Grafički prikaz progressa. Funkcionalnost tipa brzog pregleda stanja i osnovnih informacija o projektu/projektima na jednoj stranici (nešto kao dashboard). Graficko predstavljanje statistike, kroz grafikone npr. Posto su tehnicka lica obavezna u ovakvim delovima projekta, integracija sa postojećim alatima bi bila jedna od olaksajucih stvari. Svakako detaljnije informacije. Recimo, za upravljanje projektima u IT ili bilo gde drugde bi bilo dobro znati kad je koji covek zauzet i kad je slobodan da bi PM mogao da organizuje njegovo vreme efikasnije, kao i spisak njegovih vestina i eventualno prethodnih projekata. Sto su informacije detaljnije (a opet zadržati i meru) to je mogucnost organizacije bolja. Bolja navigacija kroz aplikaciju, Detaljniji opis opcija (footer). Timeline Mogućnost grupisanja po segmentima, početni i žavršni datum zadataka, "milestones", Postaviti checkbox liste u segmentima da se lako može označiti kad je neki zadatak završen Ako se u opcije nekada budu dodali npr. napredak u razvoju i problemi oni bi se mogli prikazati vizuelno putem dijagrama. ubaci da se prati progres u % pojedinačno po taskovima, grafikon kompletiranosti projekta u odnosu na taskove - najbolje kombinacija bar graph + chart.

**Сугестије у вези дизајна корисничког интерфејса:** Dizajn prilagoditi današnjim standardima, Moderniji dizajn, vizuelizacije ticketa, Aplikaciju bi trebalo unaprediti sa UI strane kako bi bila "more user friendly". Pregled topica i projekata nije bas citljiv. Poraditi na grafickom dizajnu. Jako je vazno da ima dobar interfejs. vizuelna modernizacija web okruzenja za sta se moze koristiti I pomoc nekih postojećih frameworka. GUI bi trebao da bude intuitivniji. Sadržaj i korisnički interfejs bi trebali da budu optimizovani za upotrebu na mobilnim uređajima. Promeniti dizajn interfejsa razvijenog rešenja, ozbiljno poraditi na dizajnu I ne bi bilo lose da se aplikacija prilagodi I mobilnim uređajima (mobile responsive ili mozda bolje kao zasebna mobile aplikacija). ukoliko ovo ostane web aplikacija, ne bi bilo loše da se optimizuje za više platformi, odnosno rezolucija ekrana. S obzirom da su mobilni uređaji (telefon, tableti) sa raznim operativnim sistemima danas u velikoj upotrebi, trebalo bi uraditi neki „responsive“ prikaz, odnosno da aplikacija bude lepo vidljiva na svim tim uređajima.

#### **ПОЈЕДИНАЧНИ ОДГОВОРИ:**

\*Bilo bi veoma korisno da pri kreiranju projekta bira metodologija upravljanja projektima (Scrum, Kanban itd.) te da se i radno okruzenje, korisnicke role itd. prilagodjavaju izabranoj metodologiji (definisanje scrum sastanaka, napominjanje korisnika o sastancima, rokovima, taskovima, pregled bekloga, pregled radne ploce. za kanban pregled todo liste, pregled u toku i završenih taskova za odredjeni rollout itd.).

\* Pravljenje baseline projekta, te na osnovu njega pracenje napretka projekta (vecinom kasnjenja).

\* Upravljanje finansijama (procjenjena vrijednost projekta, stvarna vrijednost projekta, kovanje faza, taskova, resursa, obuka itd.).

\* Upravljanje rizicima (licno jos nisam naisao na aplikaciju za upravljanje projektima koja u sebi ima ugradjeno i upravljanje rizicima, te bi to mozda moglo biti inovativnost aplikacije).

Cisto da navedem sta ja zelim od jedne takve aplikacije: prvenstveno da se vidi sta, ko, kad I koliko dugo, agile metodologija scrum i/ili kanban board, time tracking I task size (fibonacci ili hours), github integracija, email integracija, task comments i attachments, mentions u komentarima kao github, mogucnost razbijanja epic taskova na subtaskove.

Iz mog iskustva kako sa klijentima tako i sa developerima i sa menadzmentom, mislim da ste se uhvatili sa ozbiljnim izazovom. Iako je ovo prototip vaše aplikacije morao bih da vidim malo više da bih mogao vam više reći. Prvenstveno mislim da bi vaša aplikacija imala primenu u industrijskim uslovima razvijanja softvera, morate da prikazete sta tačno će vaša aplikacija da ponudi drugacije od drugih softvera. Aplikacija koja treba da prikaze tačno šta ze developeri ili klijent, menadzment morate da vidite iz njihovih uglova na koji način je najbolje prikazati ono što ih zanima u vezi projekta. Kako se aplikacije razvijaju i kako podrzavaju sve više i više funkcionalnosti na jednom mestu, time se daje sve veća pažnja na intuitivnost aplikacije (lakše korišćenje za end-usera). Iz licnih iskustva, menadzment i klijenti, se dosta fokusiraju na vizualizaciju podataka jer im brzo i lako daju uvid u progres projekta. Da bi ste imali dovoljno podataka za prikaz takvih grafova i chartova, morate da ukljucite dosta faktora vezanih za projekat, pocev od vremenskog i finansijskog. Developeri takodje se fokusiraju na vizuelizaciju ali zbog prirode njihovog posla potreban im je i detaljan opis zadataka. Kategorizacija, podela posla u projektima je dobro dosla. Delegiranje, estimacija zadataka. Obratite paznju na izbor tehnologija koji mogu da vam olaksaju razvoj vizuelno intuitivnog front-enda sa dobrom podrskom u backendu.

1: Nisam sigurna sta se podrazumeva pod „informacioni sistem“ ovde. Osim predmeta „informacioni sistemi“ nikad se vise nisam susrela sa ovim izrazom.

2: www.it-project.rs pruza informacije gde se project records mogu kreirati, to meni izgleda kao „work items“ u TFS koji moze imati vrednosti: user story (requirement/feature), task, bug, support request, ili bilo koju custom vrednost koja ima smisla za proces koji se prati.

3: ovo resenje izgleda OK ako bi smo radili u waterfall software development lifecycle modelu gde se uglavnom radi na jednoj iteraciji istovremeno. Trend je da se ide na „agile“ gde su iteracije krace i manji delici posla se izvrsavaju, takodje vise iteracija mogu da se poklope i da se izvrsavaju istovremeno.

4: resenje ima fokus na projekat. Kao software developer, ja mogu da radim na vise projekata odjednom i bilo bi mi od koristi da vidim sta je trenutno moje da uradim: sta je blokirano (I cime), na cemu mogu odmah da pocnem da radim, sta sam odradila, itd. Za sve projekte gde imam nesto da uradim.

5: bilo bi dobro da za svaki od records postoji „due date“ ili kad treba da za zavrsi rad.

6: bilo bi dobro da za svaki od records postoji „priority“ i „stack rank“ koji ce mi pomoci da organizujem posao i da odredim sta treba prvo da se odradi.

7: filter/sorting – da mogu da sortiram records asc/desc; da mogu da filtriram i da vidim, na primer: samo records koje ja treba da odradim, ili samo records koji su u „user interface“ i „documenting“ phase.

Ovaj softver, po meni, je specifično namenjen za programere a ne za managere. Smatram da u tom polju postoje aplikacije koje su mnogo bolje iz managerske perspektive, koje mnogo bolje podrzavaju aspekte i faze projektnog menadzmenta. Na žalost u mojoj karijeri, dok sam radio na razvoju softvera nisam koristio ovakve aplikacije jer nisam imao potrebe. Sada kada radim kao projektni manager mnogo više vidim smisao ovakvih aplikacija. Na žalost posao projektnog menadzera se više vrti oko resursa, budžeta i gantograma, nego oko specifičnih faza razvoja softvera. Mislim da programeri nemaju potrebe za ovakvim softverom, i da njihov rad i dokumentovanje razvoja i programerskog doprinosa projektu treba automatizovati. Takve funkcije treba da su podržane od strane alata za razvoj softvera, koji bi trebalo na neki način preslikati u softver za praćenje projektnih aktivnosti. Projektni menadžeri po mom iskustvu imaju potrebe za praćenje drugih parametara za vreme projekta nego programeri. U našoj firmi se za razvoj softvera koristi HOUSTON metodologija koja u ciklusu sadrži faze i pre i nakon implementacije



softvera. Postoje razni softverski paketi koji pokrivaju sve aspekte ove metodologije.

- Nije mi jasna svrha ove webapp, jer nije napravljena da bude cloud app, i ako ima samostalan domen (veorvatno za potrebe testiranja), vidim je kao neku intranet app, tj stand alone app.
- Dizajn je los, ne treba izmisljati nista novo, postoji recimo <http://getbootstrap.com/> (css framework, ima ih jos) koji ce da ti pomogne oko dizajna.
- UI/UX je takodje los, forma sa 4 dugmeta je uvek lose resenje.
- Najmanje bitne stvari za jedan projekat su home sa opisom projekta i profile, home mora da bude dashboard na kome ce da se nalaze sveze informacije o stanju projekta (promene, komentari, poruke). Ovo kazem jer su prve dve stavke home i profile. Ostatak navigacije je takodje konfuzan. Stavke nisu jasne. Po licnom misljenju treba da da postoji navigacija za Home, Projects (lista aktivnih projekata), Tasks (lista otvorenih taskova), Messages, Profile.
- Kontrola vidljivosti podataka, u malim timovima je ok da svi vide sve, u velikim nije, dovoljno su imena, nekom adminu mozda je ok da vidi ali obicnom korisniku ne.
- Ne postoji deo aplikacije za kolaboraciju izmedju clanova projekta (razemna poruka, fajlova)
- moderan proj. man. soft. mora da ima kolaboraciju (komunikacija i deljenje fajlova na raznim nivoima, licni, projekat, task), upravljanje projektima/taskovima, vodjenje dokumentacije, regulaciju nivoa pristupa, upravljanje ljudskim resursima.

Da bi aplikacija ponudila nesto novo ili kvalitetnije morala bi se pre svega prilagoditi platformi za koju je namenjena. Aplikacije koje su za sve platforme su obicno light verzije koje bi mogle da posluze malim timovima ili jednom coveku ili vec postoje kao sto je Mantis i koje su uz to besplatne.

Sto se tice tacki koje su navedene

a) Merenje kvaliteta softvera

Prvo treba odrediti sta se meri, funkcionalnost softwera, kvalitet pisanja koda ili performance ili sve ukupno. Postoje alati koji to vec rade tako da bi aplikacija trebala vise biti bazirana na integraciju sa postojećim alatima.

b) Merenje uspešnosti procesa razvoja

Recenica je malo diskutabilna. Na koji nacin uspesnot razvoja moze da se meri, opet mora da postoji neki kriteriju. A kriterijum moze opet biti vezan za metodologiju. Da li se koristi iterativna metodologija kao Scrum ili metodlogija uskog grla kao Kanban ili nijedna ili kombinacija.

c) Vizualizacija podataka o progresu

Ovo je opet vezano za metodologiju I postoje alati kao Jira koji to vec dobro prikazuju.

d) Adaptivnost na promene

Adaptivnost bi znacila da pomocu pluginova ili CMS moze da se postojeca aplikacija prilagodi novoj metodologiji razvoja software ili da se napravi nova sa svojim pravilima

e) Podrška distribuiranom razvoju

Postojanje aplikacije za razvoj software znaci da je moze koristi bilo koji korsinik bez obzira ne njegovu lokaciju jedino u slucaju da je napravljena samo da radi u lokalnoj mrezi. Tako da samo postojanje aplikacije koju moze koristiti korisnik van lokalne mreze znaci njen rad u distribuiranom sistemu. Ako se tu takodje misli na razlicitu vrstu posla u okviru projekta to opet ne menja stvar. Samo postojanje aplikacije doprinosi distributivnosti projekta.

f) Primena poslovnih pravila

Poslovna pravila su opet neka vrsta custom-izacije aplikacije, tako da u sutini sama aplikacija mora da ima taj deo gde moze da se prosiri ili podese odredjeni rulovi- vezani za metodologiju ili pravila. Sigurno bitna stavka i alati kao sto su Jira i mantis imaju neki skup pravila ali mislim da su oni vise vezani za metodologiju nego za poslovna pravila.

Aplikacija koja bi mogla da bude dovoljno fleksibilna u kreiranju i editovanju poslovnih pravila i metodologije rada, i time omogucila pravljenje na neki nacin sopstvene metodlogije bi imala trenutno najvise smisla, morala bi da ima podrsku za integracija sa postojećim alatima za merenje kvaliteta koda i funkcionalnosti koda ili da se bar omoguci unos njihovih rezultata. Takodje pracenje rezultata na DTAP serverima mora biti omoguceno.

Za pravni poslovnu aplikaciju ovo je vise nego krupan zalogaj, neka light verzija sta bi moglo da se uradi za razvoj softwara vise kao ideja nego kao full realizacija moze da se odradi ali tu treba da se napravi fokus sta bi u stvari developeri zeleti od takvog alata ili ako se pravi za Project managere sta oni ocekuju od takvog alata. Naci sredinu je jako tesko i

developeri generalno ne vole alate sa kojima moraju da se puno maltretiraju ako ne rade nesto automatski jer smatraju da im to oduzima vreme.

Za sada vidim da je aplikacija neka vrsta baze podataka projektnih artefakata (modela podataka, requiremenata, dokumentacije etc), ali nemam više informacija šta je zamišljeno da aplikacija radi. Za sada mi se čini kao neka vrsta repozitorijuma, ali ne vidim kako se pomoću ove aplikacije može upravljati npr kvalitetom softvera, procesom razvoja.

Iskustveno, najveći izazov za softwer ovog tipa je da prikaže realnu sliku projekta. Razlog ovome je što su u relanom okruženju retke čiste situacije gde se svaki korak razvoja može jasno definisati i predvideti, već u toku razvoja dolazi do promena delova projekta u skladu sa trenutnom situacijom, resursima... Ovde mogu pomoći trenutno aktuelne agilne metode razvoja...

stara tehnologija: WebForms

- loš dizajn
- nepreglednost podataka i nejasnoća istih
- sam prikaz podataka i navigacija istih bi trebala da se reši asinhronim prikazivanjem ili prikazivanjem u tkz. „dialog box,, što bi doprinelo manjem protoku, većem korisničkom doživljaju i lakšoj navigaciji
- vizuelno aplikacija nije privlačna, deluje previše amaterski.

Kao neko ko se bavi izuzetno ozbiljnim razvojem informacionih sistema mogu da vam kazem par stvari:

1. Svaki projekat ima svoje faze/etape, i to je ključna podela od koje se kreće. Ovaj deo ne vidim kod vas. Po mom mišljenju nakon davanja imena projektu to je sledeći korak.
2. Svaka faza sadrži veliki broj celina, pod celina, delova, ...
3. Svaki deo iz stavke 2 treba da ima mogućnost dodavanja zahteva, data modela, use case dijagrama, ... (Mogu da postoje i globalno).
4. Procena vremena potrebnog za razvoj, pracenje utrosenog vremena.
5. Dodeljivanje zadataka.
6. Analiza uspesnosti procena\stvarno utroseno vreme.
7. Vizuelni prikaz mora biti mnogo bolji.

Kada sam se zaposlio u firmi, imali smo software koji je bio katastrofa, osim liste zadataka, kome su dodeljeni i pracenje vremena nije još bog zna šta radio. Ubrzo smo poceli da trazimo nesto bolje i nasli smo FogBugz. Ako imate vremena pogledajte, mislim da ce vam dosta pomoci. Znam da nama jeste. Ceo projekat se rasclanjuje na niz manjih celina, ali je vizuelno sve odlicno uradjeno (sto je izuzetno bitno). Jako dobra stvar je sto svaki CASE ima statuse, pa u jednom trenutku je zahtev, u drugom je zahtev za testiranje, u trecem je bug u cetvrtom ...

Evo kako bi ovo trebalo da izgleda, vasi zuti dugmici po mom:

Phase 1 – Base development

1. Web site design
  - a. Logo Design
  - b. Header design
  - c. Footer Design
  - d. Button Design
  - e. ...
2. Security
  - a. Forms authentication
    - i. Log in
    - ii. Log out
    - iii. Forgot my password
    - iv. ...
3. Deployment
4. Documentation

Unutar svakog dela, 1. 2. a. b. i. ii. ... se stavljaju Requirements, data model, use case, UI Tokom razvoja se upisuje vreme potrebno za programing, testing...

Kao sto sam i rekao globalni dokumenti i data base dijagrami se mogu drzati na globalnom

<p>nivou, da se u svakom trenutku moze pogledati kompletna slika, mada lep vizuelni prikaz grupa i podgrupa sa dobrim nazivima pruza upravo to.</p>
<p>Veliki broj IT firmi i jaka konkurencija na trzistu razvoja softvera otvara jedan novi trend u razvoju softvera a to je sve veća paralelizacija projekata i paralelizacija rada projektnog tima na više projekata istovremeno. Bez obzira na iskustvo vodje projekta (najčešće projekata) i na njegovu veliku sposobnost u paralelizaciji poslova, apsolutno neophodno je postojanje dobrog softvera za praćenje projekata. Nedostatak vremena zbog preopterećenosti velikim brojem paralelnih projekata zahteva da upotreba aplikacije za vođenje projekata mora biti krajnje jednostavna ali sa dovoljno pruženih (bitnih) informacija za preventivno i korektivno delovanje vodje projekta. Apsolutna potreba vizuelizacije grafickim elementima (jedna slika vredi kao hiljadu reči i tabela) kako bi u par efikasnih rečenica vodja projekta mogao da koordinira tim lidere. Analiza problema ne sme da traje vecnost, a uočavanje potencijalnog problema u realizaciji projekta mora biti u prvih 15% realizacije projekta (za ovo je vrlo bitna vremenska dimenzija koja mora biti pridružena projektima). Ovome mora doprineti ovakav softver.</p>
<p>Da bi sistem zaživeo u praksi, korišćenje treba da je lako i brzo, sa optimalnim brojem relevantnih informacija.</p>
<p>Bilo bi dobro da aplikacija ima izbor jezika (srpski i engleski). Učesnici ili deo tima nekog softverskog projekta mogu vrlo često biti i ljudi koji nisu iz IT struke i koji možda ne znaju dovoljno dobro strani jezik ili ako ga i znaju ne poznaju toliko ovaj „stručni“ deo. Menadžeri takođe više vole da je to „lokalizovano“. A i postaje neki opšti vid „kulture“ da sve bude i na jeziku regionalne sredine (sam Microsoft to već godinama radi).</p> <p>Kao što sam napisao kod pitanja broj 14, možda ne bi bilo loše da postoji administracija korisnika. Da se svakom korisniku dodele određena prava, šta može samo da gleda, šta može da menja, ko može da dodaje a ko ne i tome slično...</p> <p>Ne znam da li će postojati deo o budžetu, troškovima projekta i ta, da tako kažem, ekonomska strana evidencije nekog projekta, ali možda treba i o toj mogućnosti razmisliti. U realnom „svetu“ i ova softverska funkcija ovakve aplikacije može biti veoma bitna.</p> <p>Ljudima iz struke je bitna suština ovakve aplikacije, ali onima koji nisu iz struke, vrlo je bitan i sam izgled aplikacije. Dakle, koliko god aplikacija bila dobra i imala „suštinu“ vrlo je bitan njen dizajn tj. izgled. Posveti dovoljno pažnje i tome (naučeno iz iskustva rada sa ljudima koji nisu iz IT sveta i koje često više može da zadivi dobar izgled nečega nego to šta suštinski radi „to nešto“).</p>
<p>Umesto id-a da ima neku generisanu šifru, tipa prva tri slova da budu od naziva projekta, a ostalo da budu brojevi, smatram da korisnik ne treba da vidi id ni u kakvom obliku.</p> <p>Želim da vidim samo moje taskove u nekom pogledu. Možda sa nekim filterom rešiti.</p> <p>Dodati unos komentara na taskovima, komentare mogu dodavati i ostali učesnici na isti task.</p> <p>Prenos taska na nekog drugog, ovo bi trebalo biti ispraćeno logovanjem i možda nekim opcionim razlogom.</p> <p>GUI vizuelizacija na taskovima, zelen kvadratić da je task završen, žut da je na čekanju, crven da je u izradi možda.</p> <p>Pratiti kada je krenuto sa izradom task-a, kada je završen.</p> <p>Po neki izveštaj, dijagram ili tako nešto, praćenje po mesecima, danima, godišnji.</p> <p>Za neki kasniji deo da se ubaci praćenje izmena na tasku, recimo ako neko promeni opis taska da se to zabeleži kao promena pa da se može to videti.</p>
<p>Trebala bi da ima sopstveni repozitorijum za version controll koda (nesto poput Git-a). Bilo bi odlicno da postoji version controll i za sve ostalo sto čini informacioni sistem. Dakle projektni dijagram, korisničku dokumentaciju sa spiskom mogućnosti IS-a, test alate i dokumentaciju za testiranje, skripte za baze, konfiguracione skripte za servere (OS, WEB I DB) i mrežu. Trebalo bi jasnije razgraničiti mjesto na kome customer podnosi requirement-e od onoga na kome bi to recimo project manager pretvarao u „adekvatne tehničke task-ove“. Naknadno dodavanje kolone u tabeli npr. afektuje gotovo sve djelove IS-a, a customer ili odjeljenje koje prima zahtjeve od customer-a obično nema pojma o tome.</p>
<p>www.it-project.rs po meni je prvi korak ka konacnoj aplikaciji. Koliko sam videla skoro sve funkcije PM tool-a su izostavljene ali napomenuli ste da se jos razvija. Pre svega treba dobro razmisliti o fleksibilnosti, a takodje i o tome sta drugacije ponuditi sto ce diferencirati ovaj proizvod od najvise koriscenih PM toolova.</p>

ukoliko je zamisljena kao klasican project management tool, nalik TFS-u, morala bi se „vrteti“ oko dve ideje, Kanban i SCRUM. Koliko shvatam, postoji ticket sistem, gde se svaki ticket moze dodeliti osobi, medjutim, ne vidim mogucnost prebacivanja stanja tipa „not started“ → „in progress“ → „in testing“ → „complete“. Sa tim bi funkcionisao kao Kanban, ali je u tom slucaju vizuelizacija ticketa obavezna. SCRUM, koliko vidim nije implementiran.

Svaki project records moze da zavisi od nekog drugog (dependent), takodje neki su podskup od nekog drugog, ili mozda na osnovu jednog se stvori vise, primer: od jednog dokumenta moze vise tasks da se napravi.

#### **ПОЈЕДИНАЧНИ ОДГОВОРИ на енглеском језику:**

overall, i found something missing or in-complete in the usage of the project. It needs more information on how to use the dissoft , target users and the projects launched using which base requirements.

Add an tips for fields and buttont, add an instruction. May be add AJAX technologies to exclude web pages reload.

Might be worth to include the finance part of the project.

User's simple trouble shooting/diagnostic functionality

Systems security and Information security.

Task assignment to participants function should be added.

Now it does really only participation user lists, their roles, and some of project activities. There are no implemented features regarding most of the questions 9 and 10.

Trello

- Graphical representation of progress
- One view of a project (graphical). Tables are hard to read and will not help in a professional environment
- Documentation

Would be usefula adding a HELP section on how to use the application.

This is wide field and no single solution exist for distributed software management. General concepts

may be highlighted and customizing functionality be available to enable organizations to customize the product to meet their organizational strategic goals for their portfolios.

Web Application based project are more in demand in the current globla market. It has a better scope in years to come.

Look at Redmine, Bugzilla, trac (<http://trac.edgewall.org/>), Github (Assembla, Bitbucket) and Docker.com for new ideas.

- While the application represents a good first form of what could become a project management tool, the development team would need to include a User Experience person and (perhaps the same person) a UI designer for creating powerful user interfaces.
- I'm wary that there are lots of such applications out there, so this one would have to really bring something new to the mix to even be considered for professional use.

The project is an actuall.

#### **ПОЈЕДИНАЧНИ КОМЕНТАРИ:**

Dopada mi se ideja, i nadam se da će uskoro ova probna verzija prerasti u verziju koja će se moći koristiti i u realnom radu. Na žalost, proveo sam relativno malo vremena upoznavajući se sa Vašim proizvodom ali smatram da može imati upotrebnu vrednost. Takodje, po mom skromnom mišljenju, treba povesti računa da rad sa ovom aplikacijom ne bude dodatni teret učesnicima u smislu obaveza i vremena koje treba da posvete radu sa aplikacijom. U svakom slučaju, smatram da ste na dobrom putu, i da ideja zasluzuje pažnju.

prvi deo ankete sa brojem softvera je nekako nerealan – izjednačava velike i male projekte... mislim da bi ovo trebalo nekako drugačije da se formuliše. da li duži rad na razvoju jednog velikog sajta može da se poredi sa serijskom izradom malih sajtova? Kako pobrojati sve pomoćne aplikacije koje su nastele usput? Ovi brojevi mogu jako da se razlikuju u zavisnosti od delatnosti firme i verujem da je tačne cifre teško pogoditi nekome ko se više godina aktivno time bavi.

Smatram da nije lako upustiti se u izradu softvera za praćenje razvoja projekta, ali da je

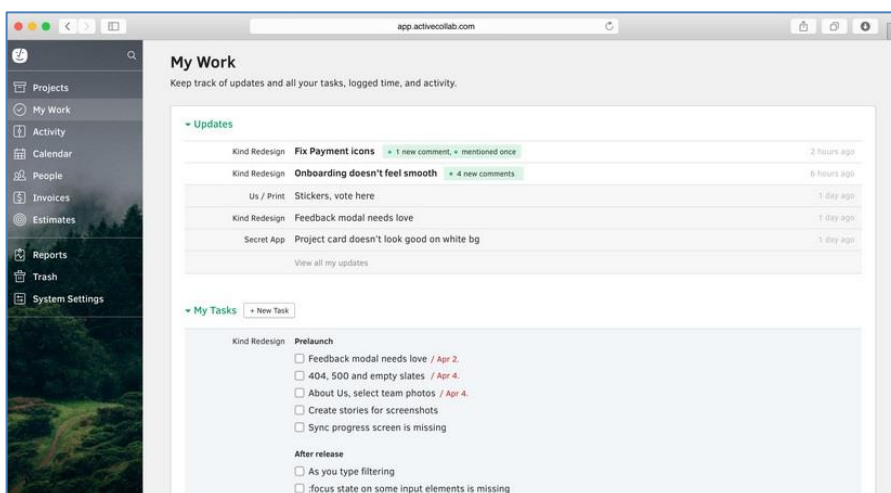
takav softver vrlo koristan i da će se koristiti svakodnevno.
Na osnovu ove pilot ankete bilo bi dobro da se sastavi jedna detaljnija anketa sa konkretnim funkcionalnostima i to distribuira na većem uzorku npr. po firmama i obrazovnim institucijama. Mada je to poprilično ozbiljan i obiman posao.
mного функција недостаје, тек је започет сајт.
Kao i sve u zivotu sve se pokusava generalizovati radi smanjenja troskova proizvodnje i drugih faktora , a na ustrb funkcionalnosti proizvoda. Veoma mi se svidja prica sa aplikacijom za upravljanje IT projekata (jer ma ko sta pricao, razlika je izmedju gradnje zgrade i gradnje softvera).
Код рангирања значаја карактеристика софтверске подршке управљању софтверским пројектима, сматрам да IS за праћење успешности софтверског пројекта треба да садржи све набројане особине, и све су подједнако важне, али у одређеним фазамa животног циклуса пројекта. Npr. приликом пројектовања софтвера треба унапред укalkulisати d); код same израде софтверског пројекта треба водити рачуна о f), e), b); за менаџере пројекта свакако су битни c), b); у realној примени важно је а) као повратна информација за даља побољшања itd.
Jako mi se sviđa ideja o jedinstvenom sistemu za praćenje i podršku softverskim projektima. Vođenje softverskog projekta je veoma složen zadatak i zahteva iskusnog i predanog project managera. S obzirom da dve godine učestvujem u timu u kojem radimo isključivo u agilnom okruženju, mislim da je to jedini način za uspešno vođenje softverskih projekata, jer su oni drugačiji od drugih projekata (na primer u građevinarstvu).
Sviđa mi se što su obuhvaćene sve faze razvoja softvera od zahteva, modela podataka, ... testiranja... do finalizacije.
Zanimalo bi me da vidim aplikaciju kad bude 50-60% gotova da vidim njene mogućnosti ako je moguće pošto ne mogu mnogo da zaključim u trenutnom stanju.

## 9.8. Подаци о алатима из бенчмаркинг анализе

Опис софтверских функција и карактеристика алата који су према анкети, најчешће коришћени у управљању софтверским пројектима у дистрибуираном окружењу дат је у наставку.

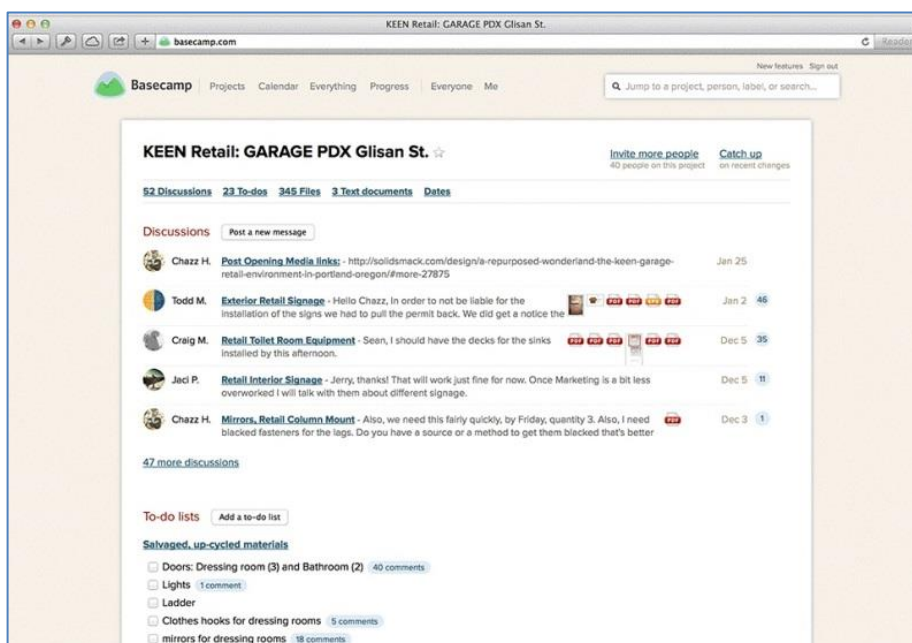
### 9.8.1. Општи алати за управљање пројектима

Алат „Active Collab“ [activecollab] - могућност инсталирања (Desktop апликација), чување података у cloud-у, могућност вођење евиденције за више пројеката у исто време, могућност проширења могућности (updates); функционалне могућности: филтрирање података, календар, дискусије са више учесника, колаборативно писање, израчунавање времена проведеног у раду на пројекту, наплате, процена трошкова.



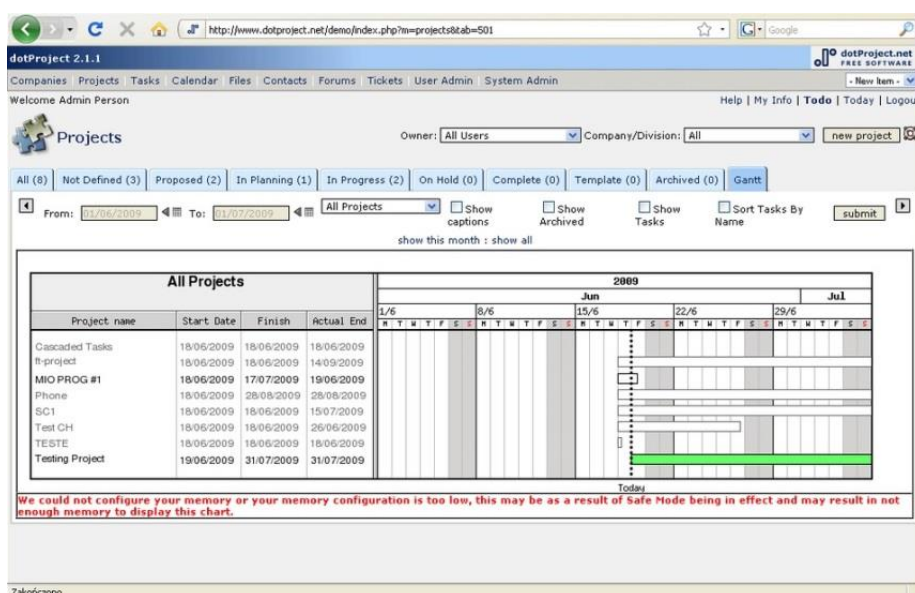
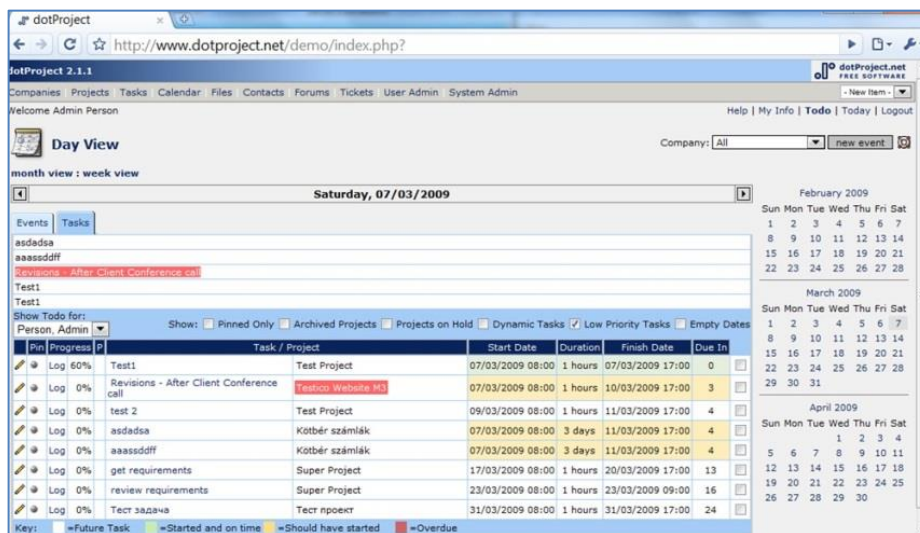
Слика 9.8.1.1. Радно окружење алата „Active Collab“

Алат „Basecamp“ [basecamp]- web апликација која снима податке у cloud-у, подршка за мобилне уређаје, функционалне могућности: е-маил, комуникација учесника пројекта, опис пројекта, додавање фајлова, дискусије, календар, колаборативно писање, чување и приказ мултимедијалних фајлова



Слика 9.8.1.2. Радно окружење алата „Basecamp“

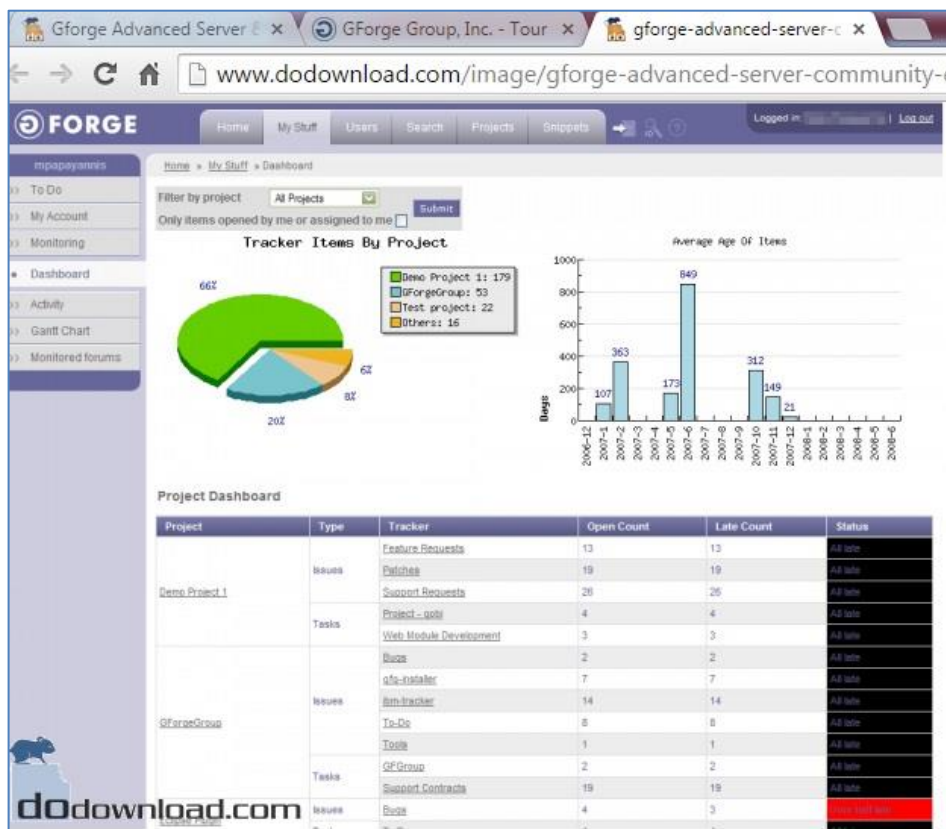
Алат „dotProject (или dP)” [dotproject]– web апликација уз могућност инсталирања целе web апликације на серверу за хостинг, имплементиран користећи PHP/MYSQL, проширив (могућност додавања модула за језике и теме - графички стилови, измена икона, додатни модули за форум, гантограме), функције: аутоматско слање е-маила, управљање корисницима, управљање тикетима, управљање клијентима, листинг таскова, архива фајлова, листа контаката, календар, дискусионни форуми



Слика 9.8.1.3. Радно окружење алата „dotProject”

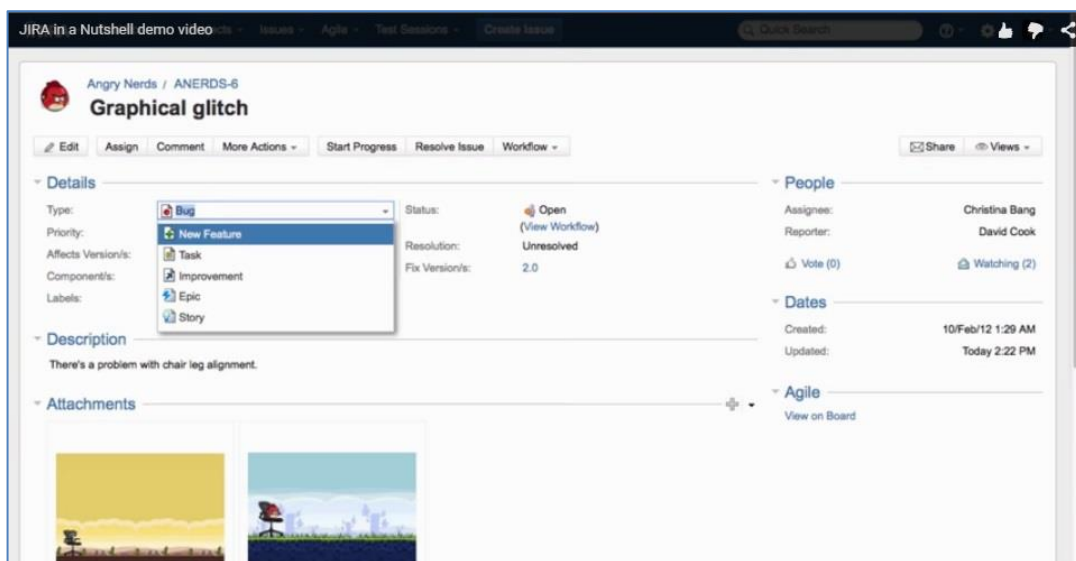
## 9.8.2. Алати за управљање софтверским пројектима

Алат „GForge Advanced Server” [gforge] [gforge inria] [gforge group]– инсталира се као VMWare виртуална машина или као посебна инсталација, потребан је Linux оперативни систем, а може се и користити хостинг. Проширив ради омогућавања интероперабилности, тј. 1) комуницирање система са развојним окружењима: Visual Studio, Eclipse, Jenkins и Microsoft Project.; 2) експорт података у MS Excel и XML формат. Најновија верзија подржава агилну методологију, односно подрсте као што је Kanban. Функционалне могућности: праћење задатака, грешака, пројектни менаџмент, управљање документима, дискусионни форум, wiki, контролу изворног кода, анализу података и графички приказ, мониторинг процеса, управљање фајловима, вестима, задацима, форум, извештаји, chat, размену фајлова, праћење промена (tracker) и филтрирање података. Добијање обавештења и технике „спомињања” („mentions”).



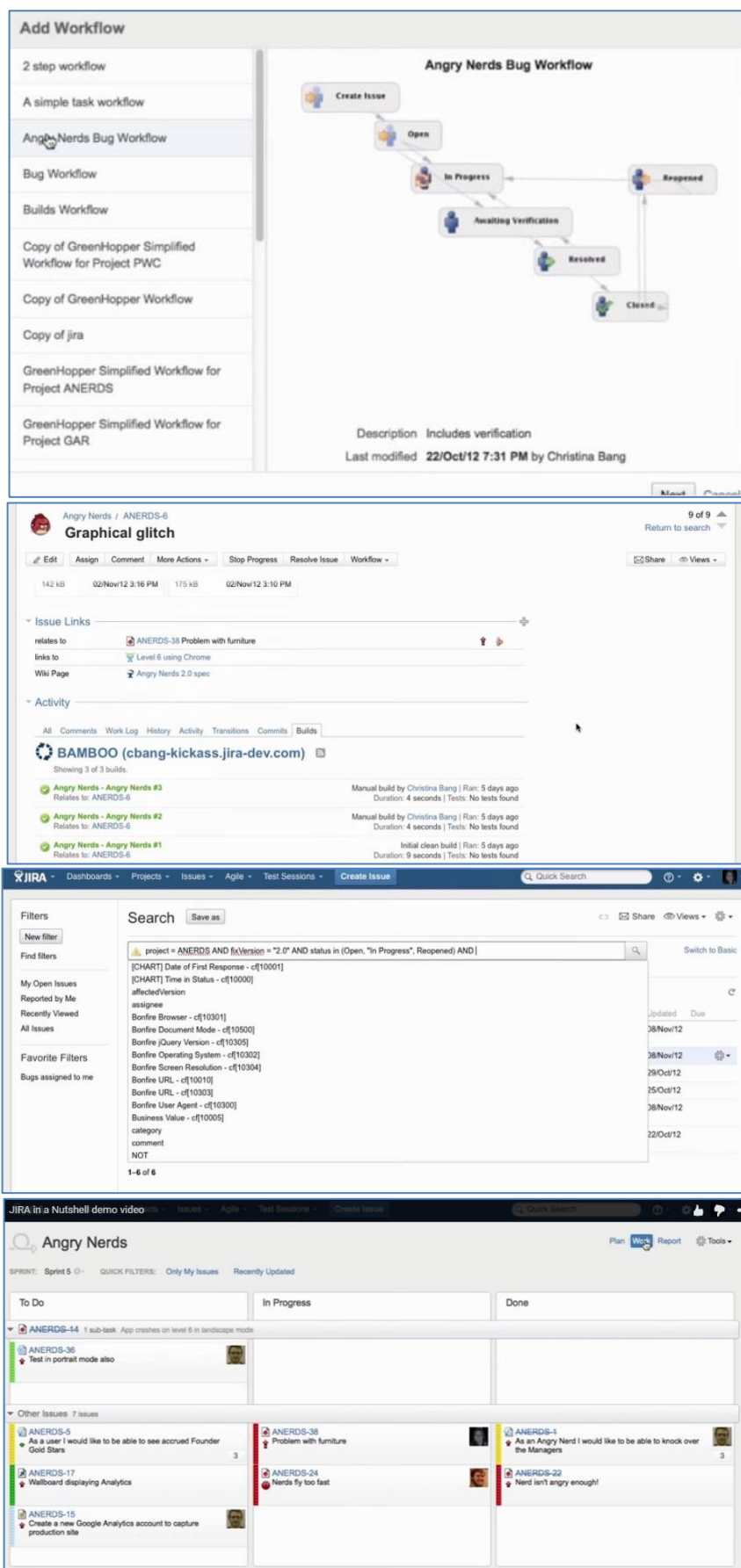
Слика 9.8.2.1. Радно окружење алата „GForge Advanced Server“

Алат „JIRA“ [jira] – функционалне могућности: анализа података и приказ за одлучивање (dashboards), евиденција пројеката, евиденција проблема и грешака, подршка агилном приступу (SCRUM, Kanban), повезивање са тестирањем, графички приказ радног тока (workflow) са могућношћу дефинисања сопствених радних токова и активности, претрага података, интероперабилност, тј. повезаност са развојним окружењима и алатима (BitBucket, Bamboo), претрага и филтрирање података (JIRA query language JQL).



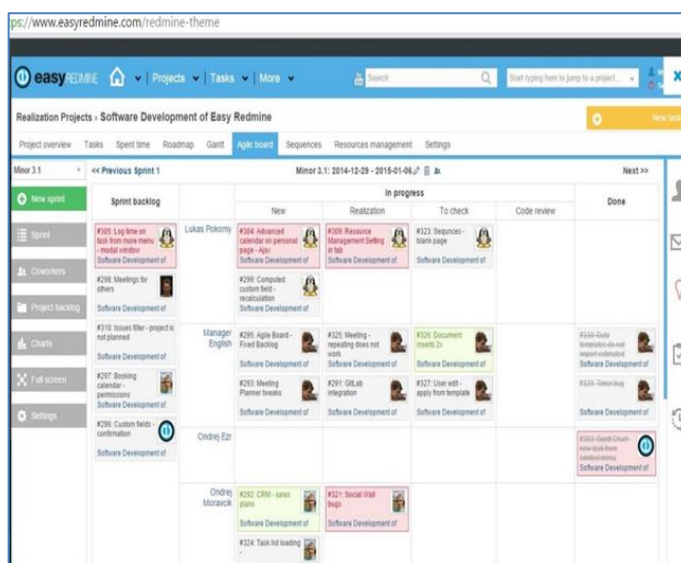
Слика 9.8.2.2. Радно окружење алата „JIRA“ – додавање активности и материјала



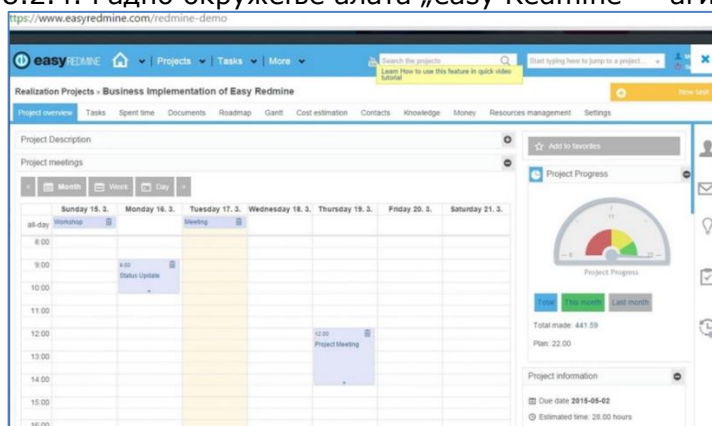


Слика 9.8.2.3. Радно окружење алата „JIRA“ – дефинисање радног тока, примена JQuery језика и подршка за Kanban

Алат „RedMine“ и „easyRedMine“ [redmine] [easyredmine]– креиран у технологији Ruby on Rails Framework, подржава инсталацију или хостинг, омогућава додатне модуле (plugins), измена графичког дизајна избором теме (боје и организација опција екрана). Функционалне могућности: рад са више пројеката истовремено, флексибилна контрола приступа у зависности од радних улога, садржи систем праћења проблема (issue tracking system), интегрише управљање вестима, документима и фајловима, омогућава нотификације о променама путем е-маила и web feed, за сваки појединачни пројекат омогућује wiki и форуме, праћење времена, прилагодљив дизајн и дефинисање потребних података за евиденцију проблема, временских података, пројеката и корисника, омогућава интеграцију са другим системима за развој, праћење промена софтвера и репозиторијуме кода (SVN, CVS, Git, Mercurial, Bazaar and Darcs), омогућава аутентификацију и саморегистрацију корисника, омогућава подршку за 34 говорна језика, даје могућност коришћења различитих база података. Додатни модули: – Управљање ресурсима, Управљање буџетом и финансијско извештавање, Наплата, Help desk, CRM, агилни прегледи. Управљање ресурсима – задавање радних задатака, планирање рада, евидентирање присуства на послу. Агилни приступи – Kanban, Scrum. Имплементиран је концепт базе знања (прикупљање знања из разних елемената управљања пројектима, сортирање, приказ). Реализована је подршка за гантограме, календар, временске листе (time sheets - дневне евиденције рада на пројектима). Реализован је систем за рано упозоравање о потенцијалном ризику неуспеха пројекта. Реализован је модул за подршку друштвеној мрежи (слично као Facebook). Омогућава дизајн радних токова дефинисањем низа активности. Омогућава креирање формула за израчунавање вредности на основу других вредности (слично као MS Excel). Омогућава интероперабилност, тј. повезивање са другим апликацијама и сервисима користећи REST API.

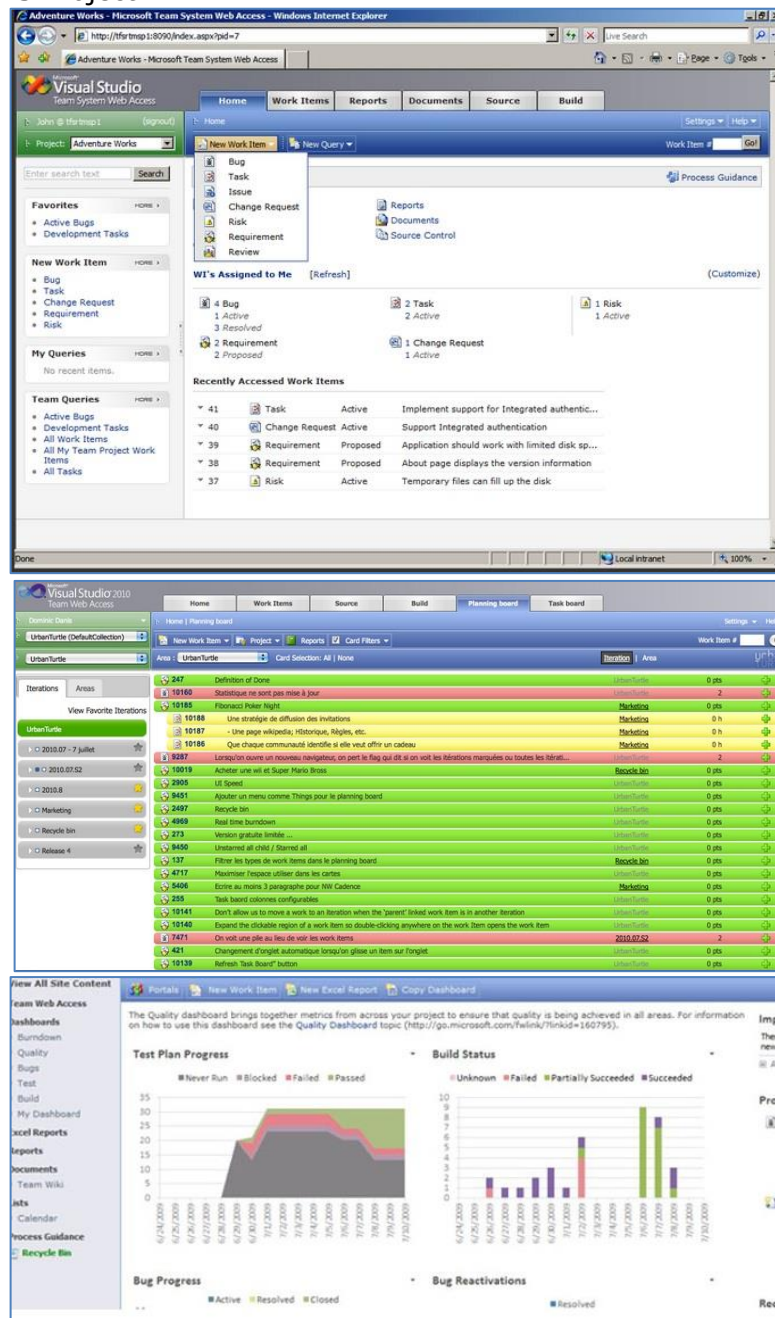


Слика 9.8.2.4. Радно окружење алата „easy Redmine“ – агилна табла



Слика 9.8.2.5. Радно окружење алата „easy Redmine“ –временски план и граф прогреса

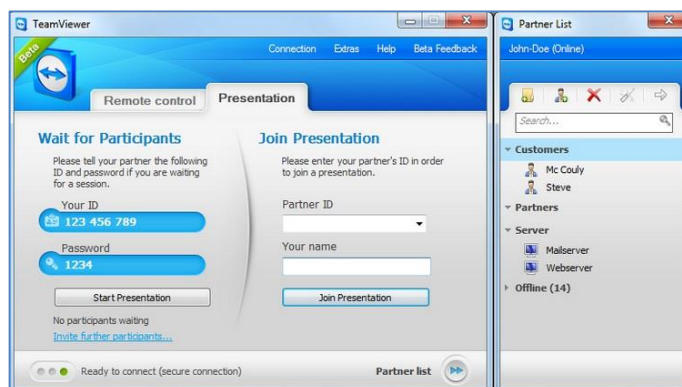
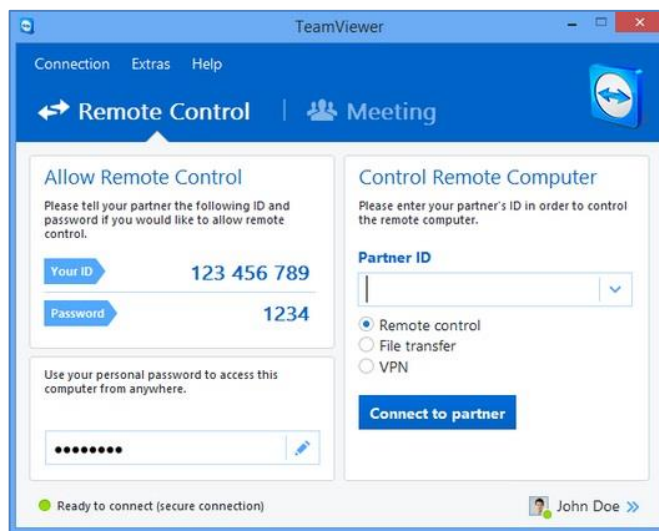
Алат „Microsoft Team Foundation Server (или алтернативно Visual Studio Online)“ –[tfs vstudio] [fts overview] проширење уз MS Visual Studio, намењен за подршку комплетном животном циклусу развоја софтвера у оквиру MS Visual Studio. Функционалне могућности: контрола верзија изворног кода и извршних верзија централизовано (Team Foundation Version Control) или дистрибуирано (Git), подршка за агилне методологије и радне токове, омогућава преузимање готових темплејта који се односе на радне токове. Омогућава проналажење грешака раније у току развоја и интеграцију. Подржава веб базирано тестирање. Омогућава креирање извештаја и графикана о напретку рада. Интегрисан је са другим алатима за развој – Eclipse, GIT. Повезивање са другим апликацијама и сервисима користећи REST API. Омогућава евидентирање и обавештавање о догађајима, евидентирање података о раду, креирање извештаја и анализу података, управљање захтевима. Омогућава телеметрију, тј. мерење перформанси, коришћења и доступности имплементираних апликација. Модул Монако омогућава директно програмирање апликација користећи Web browser или мобилни уређај. Омогућава експорт података и интеграцију са MS Office фајловима - MS Excel, MS PowerPoint, MS Project.



Слика 9.8.2.6. Радно окружење „Microsoft Team Foundation Server/VS Online“

### 9.8.3. Алат за комуникацију и подршку тимском раду

Алат „TeamViewer“ [teamviewer] – омогућава евиденцију корисника и група, управљање конекцијама и сесијама приступа и приступ удаљеном рачунару, размену података и састанке у дистрибуираном окружењу. Омогућава интеграцију са другим алатима помоћу REST API.



Слика 9.8.3.1. Радно окружење алата „Team Viewer“

### 9.8.4. Подаци из бенчмаркинг анализе

У наставку је дат приказ основних питања за анализу и алата који су вредновани на основу датих питања. Посебно су обележена питања где одговарајући алат има имплементирану подршку.

Табела 9.8.4.1. Анализа карактеристика издвојених алата путем анкете, применом бенчмаркинг модела

ОБЛ АСТ	КАРАКТЕРИСТИКА	Active Collab	Basecamp	dotProject	GForge	Jira	RedMine	Microsoft Team Foundation Server	TeamViewer
PMBOK	Интероперабилност са другим алатима, посебно развојним окружењима и		Да		Да	Да	Да	Да	Да

Љубица Кази - Докторска дисертација

	различитим форматима фајлова?								
	експорт података?				Да		Да	Да	Да
PMBO K	Приказ дефинисаног обухвата потребних софтверских функција? Измене обухвата потребних софтверских функција? Приказ реализованог обухвата потребних софтверских функција?					Да Да Да	Да Да Да	Да Да Да	
PMBO K	Процену и планирање времена реализације? Приказ временских карактеристика тока реализације?	Да Да	Да Да	Да Да	Да Да	Да Да	Да Да	Да Да	
PMBO K	Мерење квалитета људи – учесника пројекта? Мерење квалитета тима у учешћу пројекта? Мерење квалитета резултата? Мерење квалитета тока резултата процеса?					- Да Да Да	- Да Да Да		
PMBO K	Евиденцију учесника пројекта? Евиденцију реализованог рада? Персонализацију функција за различите профиле корисника у односу на различите улоге у реализацији пројекта?	Да Да Да	Да Да Да	Да Да Да	Да Да Да	Да Да Да	Да Да Да	Да Да Да	
PMBO K	Размену фајлова између учесника? Комуницирање – размену порука, идеја, питања?	Да Да	Да Да	Да Да	Да Да	Да Да	Да Да	Да Да	Да Да
PMBO K	Евидентирање ризика?						Да Да		

Љубица Кази - Докторска дисертација

	Мерење ризика? Приказ ризика?						Да		
PMBO K	Унапређење постојећег решења додатним модулима?			Да	Да	Да	Да	Да	
PMBO K	Комуникацију са заинтересованим странама?	Да	Да	Да	Да	Да	Да	Да	Да
<b>PMBOK: Max=21</b>		8	9	9	11	13	20	17	5
SWEB OK	Евиденцију захтева? Евидентирање промене захтева? Праћење и приказ промена захтева? Евидентирање степена усклађености решења са захтевима?				Да Да Да	Да Да Да	Да Да Да	Да Да Да	
SWEB OK	Интеграцију са алатима за дизајн система?				Да	Да		Да	
SWEB OK	Интеграцију са алатима за развој?				Да	Да	Да	Да	
SWEB OK	Интеграцију са алатима за тестирање?				Да	Да		Да	
SWEB OK	Приказ тока процеса реализације? Мерење успеха фазе или тока реализације?	Да	Да	Да	Да	Да	Да Да	Да Да	
SWEB OK	Примену различитих приступа и методологија управљања пројектима? Примену различитих приступа и методологија развоја софтвера?				Да	Да Да	Да Да	Да Да	
SWEB OK	Мерење квалитета резултата? Мерење квалитета тока резултата процеса?						Да Да	Да Да	
<b>SWEBOK: Max=13</b>		1	1	1	8	9	10	12	0
BSC	Финансијски показатељи успеха фазе или процеса? Финансијски	Да					Да Да		

Љубица Кази - Докторска дисертација

	показатељи успеха производа?								
BSC	Комуникацију са корисницима? Креирање извештаја? Мерење задовољства корисника? Евидентирање потреба и захтева корисника?	Да Да	Да	Да	Да Да	Да	Да Да	Да Да Да	Да
BSC	Приказ тока процеса? Планирање тока процеса? Мерење успеха тока процеса?		Да	Да	Да	Да Да Да	Да Да Да	Да Да Да	
BSC	Евиденцију и приказ искустава? Евидентирање и приказ проблема и одговора?				- Да	- Да	Да Да		
<b>BSC:</b> <b>max=10</b>		<b>3</b>	<b>2</b>	<b>2</b>	<b>4</b>	<b>5</b>	<b>9</b>	<b>7</b>	<b>1</b>
	<b>УКУПНО:</b> <b>max=44</b>	<b>12</b>	<b>12</b>	<b>12</b>	<b>23</b>	<b>27</b>	<b>39</b>	<b>36</b>	<b>6</b>

## 9.9. Подаци из анализе ефикасности коришћења прототиша анализом „истовремених вишеструких активности“

Табела 9.9.1. Приказ укупног броја радних ставки за реализоване пројекте у шк. 2013/14

Ознака пројекта	Пројекат	Бр. Рад. ставки
2014_1	ASPX апликација за подршку евиденције Alumni (дипломираних студената)	23
2014_2	ASPX апликација за подршку евиденцији опреме факултета	14
2014_3	ASPX апликација за подршку организацији међународне конференције	17
2014_4	PHP апликација за групно слање е-mail обавештења	6
2014_5	ASPX апликација за праћење евиденције наставног и научног рада на факултету	32
2014_6	ASPX апликација за праћење евиденције конкурса и запослених на факултету	
2014_7	ASPX апликација за подршку организацији одбрана завршних радова	21
2014_8	PHP апликација за Web форум	12
2014_9	ASPX апликација за евиденцију правних докумената факултета	15
2014_10	PHP апликација за евиденцију рада библиотеке факултета	12
2014_11	ASPX апликација за подршку раду скриптарнице факултета	14
2014_12	ASPX апликација за on-line CIP евиденцију	11
2014_13	Windows C# апликација за израду профактура	12
2014_14	ASPX апликација за евиденцију предиспитних обавеза	20
2014_15	ASPX апликација за евиденцију научног финансијског фонда факултета	11
2014_16	ASPX апликација за евиденцију записника са седница	13
2014_17	ASPX апликација за евиденцију оцењивања наставника	14
2014_18	ASPX апликација за подршку организацији рада на факултету	25
2014_19	ASPX апликација за евиденцију података о понашању студената	11

Табела 9.9.2. Приказ броја записа са једном радном активности у оквиру пројеката реализованих у шк. 2013/14

Ознака пројекта	Укупно записа са једном радном активности
2014_1	11
2014_2	8
2014_3	7
2014_4	5
2014_5	10
2014_6	
2014_7	5
2014_8	6
2014_9	2
2014_10	6
2014_11	10
2014_12	2
2014_13	6
2014_14	11
2014_15	8



2014_16	7
2014_17	3
2014_18	6
2014_19	7

Табела 9.9.3. Приказ броја радних ставки у односу на вишеструки унос – правилан начин

Ознака пројекта	Број уноса у једном дану					Укупно записа	Укупно рад. акт.
	2	3	4	5	6		
2014_1	3	0	0	0	0	6	9
2014_2	2	0	1	0	0	8	11
2014_3	3	1	0	0	0	9	16
2014_4	0	1	0	0	0	3	4
2014_5	3	2	1	0	0	13	23
2014_6							
2014_7	1	1	0	0	0	5	7
2014_8	2	1	0	0	0	7	11
2014_9	1	0	0	1	0	7	14
2014_10	2	0	0	0	0	4	6
2014_11	2	1	0	0	0	6	7
2014_12	1	0	1	0	0	6	10
2014_13	1	1	0	0	0	5	8
2014_14	4	1	0	0	0	11	17
2014_15	2	0	0	0	0	4	4
2014_16	0	1	1	0	0	7	10
2014_17	2	1	0	0	0	7	13
2014_18	0	2	2	0	0	14	24
2014_19	2	0	0	0	0	0	0

Табела 9.9.4. Приказ броја радних ставки у односу на композитан унос – неправилан начин

Ознака пројекта	Број уноса у једном дану			Укупно записа	Укупно радних активности
	2	3	4		
2014_1	11	1	0	12	25
2014_2	4	0	0	4	8
2014_3	8	2	0	10	22
2014_4	1	0	0	1	2
2014_5	15	6	1	22	52
2014_6					
2014_7	16	0	0	16	32
2014_8	6	0	0	6	12
2014_9	13	0	0	13	26
2014_10	6	0	0	6	12
2014_11	4	0	0	4	8
2014_12	8	1	0	9	19
2014_13	5	1	0	6	13
2014_14	9	0	0	9	18
2014_15	3	0	0	3	6
2014_16	6	0	0	6	12
2014_17	11	0	0	11	22
2014_18	19	0	0	19	38
2014_19	4	0	0	4	8

## 9.10. Категоризација фактора успеха, неуспеха и ризика у оквиру истраживања софтверских, дистрибуираних и агилних софтверских пројеката

У овом одељку приказано је распоређивање (у одговарајуће категорије у оквиру предлога метричког модела) фактора успеха, неуспеха и ризика у реализацији софтверских пројеката општег типа, као и дистрибуираних софтверских пројеката и агилних софтверских пројекта, који су наведени у претходним истраживањима која су наведена у табели 3.4.1.1.

Табела 9.10.1. Категоризација фактора успеха, неуспеха и ризика у оквиру истраживања софтверских пројеката, дистрибуираних и агилних софтверских пројеката према предлогу метричког модела процеса

Озн. Кар.	Карактеристика
S1	Корисност производа за организацију (додаје вредност), задовољавање корисникових потреба, Решавање проблема корисника, Корисник користи производ, Задовољство корисника, Комерцијални успех, Креирање удела тржишта, пораст профита, уговарање
S2	унапређена ефикасност рада, Ресурси – минимизација губитака времена, Ефикасност процеса развоја – у складу са временским планом и буџетом, дефинисана методологија развоја, контрола промена <ul style="list-style-type: none"> <li>тип организације: Програмирање у пару, виртуелни тимови</li> <li>карактеристике организације: Подела посла на локације, додељивање задатака, организациони облици(Програмирање у пару, виртуелни тимови – кооперативни, делегацијски, консултативни), менторство, координација, синхронизација, Величина организације, агилна, адаптивни менаџмент, самоорганизујући тимови</li> <li>Организациона култура: не традиционална, не политички орјентисана</li> <li>планирање: Минимизација обухвата посла, распоред рада, процена трајања активности, јасна дефиниција пројекта, пројектни план, визија и обухват пројекта, оптимизација, Обухват, захтеви, планирање пројекта, Јасност задатка, реални план – коректно процењено трајање активности, буџет, развијање функција и интерфејса који је одговарајући, измене распореда рада, претпоставке о расположивости ресурса, адекватна додела активности, довољно јасно дефинисани елементи испоруке, одговорност на пројекту, не превише функција обећано</li> <li>контрола: Краћи интервали контролних тачака, оптимизациони оквир, довољно ревизије, контроле и одобравања, на време корективне акције да буду урађене, праћење прогреса, довољно заступљена контрола, Аутоматизована контрола изворног кода</li> <li>организациона култура: Кооперативна организациона култура уместо хијерархијске</li> <li>менаџмент: Подршка вишег менаџмента, посвећеност и подршка спонзора, подршка прековременом раду, подршка оперативног менаџмента, власништво над пројектом, извршно спонзорство, посвећеност менаџмента, укључен менаџмент, подршка менаџмента,</li> <li>процес развоја: редовна испорука, испорука прво најважнијих особина, тестирање, методологија развоја, покривеност фаза развоја, Ефикасност, ефективност</li> </ul>
S3	Трошкови
S4	организационо учење, Утицај на животну средину, лични развој, Размена знања,
T1	Интеграција резултата,
T2	<ul style="list-style-type: none"> <li>Прецизност спецификације, промена захтева, јасни пословни циљеви, јасни почетни захтеви</li> <li>Карактеристике пројекта: Елементи дефиниције пројекта, Величина, флексибилност,</li> </ul>
T3	Рок израде, реалистичан временски план
T4	Адекватан буџет

T5	Минимизација неисправног кода
T6	<ul style="list-style-type: none"> <li>Учесник: Мотивација, Фокус, Подноси притисак, способност, високе перформансе, Компетентан, Емоционална зрелост, тимски рад, нема отпор, мотивација кроз награђивање, компетенција, експертиза, Конвергенција, културне разлике, идеологија, однос према хијерархији, познавање, поверење, нема страха, нема отпора променама, способности, искуства, знање (функционално, технолошко, управљачко), Број људи, измене/стабилност кључних људи – пројектни спонзор, пројектни менаџер..., особине одговорности, знања и способности, технолошка компетентност</li> <li>Тимске особине учесника: Групна свесност, друштвене везе, фамилијарност, ментални модели, когнитивна дивергенција, поверење, колаборација, лидерство, индивидуална различитост</li> <li>Девелопери: Укљученост (активност), задовољство, Предзнање и искуство учесника из технологије, Технички изазови, технички тренинг</li> <li>Пројектни менаџери: Лидерство, способност фокусирања, мотивисања, ауторитет, комуникационе способности, разумевање проблема, компетентан, емоционална зрелост, експертиза, Одговорност, знања и способности, адекватна компетенција, искуство, ауторитет</li> </ul>
T7	Расположивост средстава комуникације, повратна информација од корисника и пројектног менаџера, комуникација са корисником, комуникација лицем у лице усмено, редовно извештавање, метрике комуникације, конфликти, Управљање непрецизношћу, Спецификација захтева, јасно изражена добит-корист од система, Везе, удаљеност, брзина, размена, језик, координација, преговарање, временске зоне, стил комуникације (прецизна, формална, флексибилна), управљање информисањем (документовање, обавештавање, истицање рокова, проблема, потребног знања), недостатак неформалне комуникације, језик комуникације Документација: Расположена, одржава се, правни аспекти
T8	Ризици се јављају у свим другим сегментима
T9	ресурси: Распоположивост, Екстерна зависност - Екстерне компоненте, довољан буџет, довољни ресурси Технологије: За комуникацију, споре и непоуздане рачунарске мреже, усклађеност верзија алата, компатибилност формата података, Погодност технолошке основе, Неодговарајућа технологија и алати, Алати у развоју и управљању, инфраструктура, примењена технологија, Непознате, прецењене могућности нових технологија
T10	ефикасно управљање заинтересованим странама
AV1	комуникација са корисником
AV2	Квалитет софтвера
AV3	повратна информација од корисника, дефинисана улога корисника, присутност корисника, јасно изражена добит-корист од система
AV4	Спецификација захтева, Континуални ток промена, измене обухвата технологије, функционалности или пословног проблема
AP1	Спецификација захтева
AP2	Спецификација захтева
AP3	Интерни процес
AP4	Комуникације
AP5	Људски ресурси
AP6	Комуникације
AP7	Квалитет софтвера
AP8	Заштита животне средине
AP9	Комуникације
AP10	Квалитет софтвера, адекватан дизајн: Погодност за проширење, за измене, за интеграцију са постојећим решењима, Рефакторисање кода, једноставан дизајн
AP11	Интерни процес планирања
AP12	Тимски дух, Управљање знањем: Дељење, координација, ток размене, оперативно учење
AP13	Учење и развој

## 9.11. Подаци из примене метричких модела у евалуацији артефакта развоја софтвера у области развоја информационах система

### 9.11.1. Евалуација модела пословних процеса

Табела 9.11.1.1. Оцене радова студената у области моделовања пословних процеса применом метричког модела – шк. 2013/14

Школска 2013/14				
РБ РАДА	НАЗИВ ЕЛЕМЕНТА %	ОРГАНИЗАЦИЈА ЕЛЕМЕНТА %	РЕЧНИК ПОДАТАКА %	ПРОСЕК %
1	100	78	100	93
2	0	0	0	0
3	67	44	60	57
4	83	0	20	34
5	100	67	100	89
6	50	0	60	37
7	0	0	0	0
8	83	0	0	28
9	50	22	60	44
10	50	22	60	44
11	100	100	100	100
12	17	0	80	32
13	67	22	80	56
14	0	0	0	0
15	83	56	80	73
16	17	0	40	19
17	50	0	100	50
18	33	56	80	56
19	83	56	80	73
20	83	67	80	77
21	0	0	0	0
22	67	100	80	82
23	100	67	20	62
24	17	0	20	12
25	0	0	0	0
26	17	0	0	6
27	67	56	100	74
28	0	0	0	0
29	83	44	40	56
30	0	0	0	0
31	0	0	0	0
32	100	0	80	60
33	67	0	60	42
34	100	100	100	100
35	0	0	0	0
36	100	100	80	93
37	67	33	100	67
38	83	89	80	84
39	83	67	80	77
40	0	0	0	0
41	67	0	60	42
42	17	0	40	19
43	17	0	80	32
44	33	0	60	31
45	33	0	80	38
46	83	0	80	54
47	50	33	100	61

48	50	22	80	51
49	0	0	0	0
50	67	67	100	78
51	50	0	60	37
52	83	89	80	84
53	33	0	80	38
54	67	0	60	42
55	100	100	80	93
56	100	11	80	64
57	33	0	20	18
58	50	0	40	30
59	0	0	0	0
60	17	0	40	19
61	100	100	80	93
62	67	33	100	67
63	50	0	20	23
64	83	22	100	68
65	33	0	80	38
66	83	89	40	71
<b>ПРОСЕК %</b>	<b>52</b>	<b>27</b>	<b>55</b>	<b>45</b>

Табела 9.11.1.2. Оцене радова студената у области моделовања пословних процеса применом метричког модела – шк. 2014/15

<b>Школска 2014/15</b>				
<b>РБ РАДА</b>	<b>НАЗИВ ЕЛЕМЕНТА %</b>	<b>ОРГАНИЗАЦИЈА ЕЛЕМЕНТА %</b>	<b>РЕЧНИК ПОДАТАКА %</b>	<b>ПРОСЕК %</b>
1	100	94	100	98
2	80	94	100	91
3	60	94	93	82
4	40	0	93	44
5	90	94	100	95
6	80	0	71	50
7	80	94	100	91
8	100	94	86	93
9	80	94	100	91
10	100	100	100	100
11	50	0	93	48
12	70	0	100	57
13	70	100	100	90
14	90	100	100	97
15	80	100	86	89
16	100	100	100	100
17	90	94	86	90
18	90	50	71	70
19	70	0	86	52
20	70	50	93	71
21	50	0	0	17
22	80	100	50	77
23	100	100	93	98
24	70	0	0	23
25	70	94	86	83
26	50	0	71	40
27	90	100	93	94
28	100	100	93	98
<b>ПРОСЕК %</b>	<b>79</b>	<b>66</b>	<b>84</b>	<b>76</b>

**9.11.2. Евалуација модела дизајна софтверских функција**

Табела 9.11.2.1. Оцене радова студената у области модела дизајна софтверских функција применом метричког модела – шк. 2013/14

<b>Школска 2013/14</b>				
<b>РБ РАДА</b>	<b>ТАБЕЛА ПРЕСЛИКАВАЊА</b>	<b>ДИЈАГРАМ СЛУЧАЈЕВА КОРИШЋЕЊА</b>	<b>СПЕЦИФИКАЦИЈА СЛУЧАЈА КОРИШЋЕЊА</b>	<b>ПРОСЕК %</b>
1	80	100	100	93
2	100	100	100	100
3	100	75	25	67
4	70	113	92	92
5	100	88	8	65
6	80	63	100	81
7	0	0	0	0
8	80	63	25	56
9	0	0	0	0
10	100	113	100	104
11	100	100	100	100
12	100	100	100	100
13	0	0	0	0
14	80	88	100	89
15	0	0	0	0
16	100	113	92	102
17	80	75	92	82
18	90	88	83	87
19	60	75	92	76
20	0	0	0	0
21	100	75	100	92
22	90	88	50	76
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	90	100	100	97
27	0	0	0	0
28	100	63	67	77
29	0	0	0	0
30	0	0	0	0
31	80	75	50	68
32	0	0	0	0
33	0	0	0	0
34	70	100	58	76
35	100	113	100	104
36	0	0	0	0
37	70	88	67	75
38	60	75	83	73
39	90	100	92	94

40	100	113	100	104
41	0	0	0	0
42	90	100	100	97
43	0	0	0	0
44	15	6	0	7
45	100	88	0	63
46	0	0	0	0
47	80	113	100	98
48	100	113	100	104
49	80	113	58	84
50	0	0	0	0
51	100	113	92	102
52	60	100	0	53
53	90	100	100	97
54	90	113	92	98
55	95	100	0	65
56	100	113	100	104
57	100	113	92	102
58	100	100	100	100
59	100	88	50	79
60	0	0	0	0
61	0	0	0	0
62	100	100	100	100
63	90	113	83	95
64	80	88	0	56
65	90	113	42	82
66	70	100	0	57
67	100	100	83	94
<b>ПРОСЕК %</b>	<b>61</b>	<b>66</b>	<b>50</b>	<b>59</b>

Табела 9.11.2.2. Оцене радова студената у области модела дизајна софтверских функција применом метричког модела – шк. 2014/15

<b>Школска 2014/15</b>				
<b>РБ РАДА</b>	<b>ТАБЕЛА ПРЕСЛИКАВАЊА</b>	<b>ДИЈАГРАМ СЛУЧАЈЕВА КОРИШЋЕЊА</b>	<b>СПЕЦИФИКАЦИЈА СЛУЧАЈА КОРИШЋЕЊА</b>	<b>ПРОСЕК %</b>
1	95	88	100	94
2	100	88	100	96
3	90	100	100	97
4	80	88	75	81
5	50	50	0	33
6	100	100	83	94
7	100	75	100	92
8	100	0	0	33
9	95	0	0	32
10	100	88	50	79

<b>11</b>	95	100	100	98
<b>12</b>	95	100	100	98
<b>13</b>	100	88	100	96
<b>14</b>	100	100	100	100
<b>15</b>	90	100	0	63
<b>16</b>	70	88	67	75
<b>17</b>	100	100	100	100
<b>18</b>	95	88	100	94
<b>19</b>	95	88	100	94
<b>20</b>	0	0	0	0
<b>21</b>	100	100	100	100
<b>22</b>	85	50	100	78
<b>23</b>	100	75	92	89
<b>24</b>	80	75	17	57
<b>25</b>	80	100	100	93
<b>26</b>	95	100	100	98
<b>27</b>	95	100	50	82
<b>28</b>	85	88	0	58
<b>29</b>	65	50	0	38
<b>30</b>	90	88	100	93
<b>ПРОСЕК %</b>	<b>88</b>	<b>79</b>	<b>68</b>	<b>78</b>

### 9.11.3. Евалуација концептуалног модела података

Табела 9.11.3.1. Оцене радова студената у области концептуалног модела података применом метричког модела – шк. 2013/14

<b>Школска 2013/14</b>					
<b>РБ РАДА</b>	<b>ЦЕЛИНА</b>	<b>ЕНТИТЕТ</b>	<b>АТРИБУТ</b>	<b>ПОВЕЗНИК</b>	<b>ПРОСЕК %</b>
<b>1</b>	100	96	89	100	96
<b>2</b>	78	67	67	67	70
<b>3</b>	89	100	100	94	96
<b>4</b>	78	58	67	89	73
<b>5</b>	100	83	100	100	96
<b>6</b>	89	75	100	100	91
<b>7</b>	0	0	0	0	0
<b>8</b>	89	92	89	94	91
<b>9</b>	0	0	0	0	0
<b>10</b>	100	100	100	100	100
<b>11</b>	100	96	100	94	98
<b>12</b>	89	92	78	94	88
<b>13</b>	0	0	0	0	0
<b>14</b>	100	92	56	94	86
<b>15</b>	0	0	0	0	0
<b>16</b>	78	79	67	94	80
<b>17</b>	100	100	89	94	96



<b>18</b>	100	96	89	94	95
<b>19</b>	100	75	89	94	90
<b>20</b>	0	0	0	0	0
<b>21</b>	100	96	89	89	94
<b>22</b>	78	83	78	89	82
<b>23</b>	0	0	0	0	0
<b>24</b>	0	0	0	0	0
<b>25</b>	0	0	0	0	0
<b>26</b>	100	100	100	100	100
<b>27</b>	0	0	0	0	0
<b>28</b>	100	92	89	83	91
<b>29</b>	0	0	0	0	0
<b>30</b>	0	0	0	0	0
<b>31</b>	89	67	44	89	72
<b>32</b>	89	83	100	94	92
<b>33</b>	100	96	67	94	89
<b>34</b>	0	0	0	0	0
<b>35</b>	78	75	67	94	79
<b>36</b>	100	88	89	89	92
<b>37</b>	89	92	89	78	87
<b>38</b>	100	88	100	94	96
<b>39</b>	0	0	0	0	0
<b>40</b>	67	67	67	83	71
<b>41</b>	0	0	0	0	0
<b>42</b>	78	67	78	94	79
<b>43</b>	0	0	0	0	0
<b>44</b>	89	96	89	100	94
<b>45</b>	0	0	0	0	0
<b>46</b>	100	100	89	94	96
<b>47</b>	0	0	0	0	0
<b>48</b>	100	100	100	94	99
<b>49</b>	0	0	0	0	0
<b>50</b>	100	96	89	72	89
<b>51</b>	100	88	78	94	90
<b>52</b>	89	88	100	100	94
<b>53</b>	100	92	100	89	95
<b>54</b>	89	96	89	89	91
<b>55</b>	100	100	89	94	96
<b>56</b>	89	75	56	89	77
<b>57</b>	0	0	0	0	0
<b>58</b>	0	0	0	0	0
<b>59</b>	0	0	0	0	0
<b>60</b>	78	83	100	94	89
<b>61</b>	89	92	78	94	88
<b>62</b>	0	0	0	0	0
<b>63</b>	89	100	89	100	95

<b>64</b>	100	92	89	100	95
<b>65</b>	89	100	89	89	92
<b>ПРОСЕК %</b>	<b>61</b>	<b>58</b>	<b>56</b>	<b>61</b>	<b>59</b>

Табела 9.11.3.2. Оцене радова студената у области концептуалног модела података применом метричког модела – шк. 2014/15

<b>Школска 2014/15</b>					
<b>РБ РАДА</b>	<b>ЦЕЛИНА</b>	<b>ЕНТИТЕТ</b>	<b>АТРИБУТ</b>	<b>ПОВЕЗНИК</b>	<b>ПРОСЕК %</b>
1	94	75	83	92	86
2	100	100	100	100	100
3	100	94	83	100	94
4	100	88	100	100	97
5	89	75	92	96	88
6	94	94	100	92	95
7	100	100	92	100	98
8	100	94	92	100	97
9	94	94	83	100	93
10	94	100	100	100	99
11	100	88	50	85	81
12	100	94	50	92	84
13	94	100	100	100	99
14	94	69	58	85	77
15	83	63	67	77	73
16	89	94	67	96	87
17	94	75	83	100	88
18	94	75	83	100	88
19	94	75	83	100	88
20	100	100	100	100	100
<b>ПРОСЕК %</b>	95	87	83	96	90

#### 9.11.4. Евалуација софтверске апликације

Пример примене модела евалуације софтверске апликације илустроваће могућност примене овог модела на конкретном примеру софтвера, односно софтверског пројекта 2014-1 „Alumni“.

Табела 9.11.4.1. Примена метричког модела за оцену софтверске апликације у области развоја информационог система на примеру „Alumni“

<b>ГРУПА ЗАХТЕВА / ЗАХТЕВ КВАЛИТЕТА</b>	<b>ИМПЛЕМЕНТАЦИОНИ ДЕТАЉИ РЕАЛИЗАЦИЈЕ ЗАХТЕВАНЕ КАРАКТЕРИСТИКЕ</b>	<b>Max poena</b>	<b>ПРОЈЕКАТ: 2014_1</b>
ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ			
Функционалност софтвера	Софтвер функционише	1	1
Функционална усклађеност са потребама	Усклађеност решења са захтевима (случајеви коришћења USE CASE или „user stories“), потребама и очекивањима корисника	1	1
Подршка основним функцијама			

Љубица Кази - Докторска дисертација

<input type="checkbox"/>	Унос и аквизиција података	Унос података подршком за различите уређаје - tastatura, mis, 2) drugo	2	1
<input type="checkbox"/>	Ажурирање података	брисање, измена	2	0.5
<input type="checkbox"/>	Верификација података	Контрола исправности података приликом уноса/измене	1	0.5
<input type="checkbox"/>	Чување података	База података, различити формати (XML), дистрибуиране базе података, бекап уређаји	4	1
<input type="checkbox"/>	Размена података	Интероперабилност – експорт података, учитавање података других формата	2	0
<input type="checkbox"/>	Обрада података	Сумарна и статистичка обрада података	2	0
<input type="checkbox"/>	Презентовање података	Графичко представљање, Визуализација података,	1	0
<input type="checkbox"/>		Графикони, Табеларни прикази	2	1
<input type="checkbox"/>	Претрага	Филтрирање података, сортирање, навигација	3	1
<b>КВАЛИТЕТ ПОДАТАКА</b>				
<input type="checkbox"/>	Комплетност	Усклађеност базе података, екранских форми и извештаја са базом података, захтевима корисника	3	3
<input type="checkbox"/>	Прецизност	Одређени типови података, ограничења	2	2
<input type="checkbox"/>	Конзистентност	Усклађеност података из различитих табела базе података (транзакције)	1	0
<input type="checkbox"/>	Доступност	Поузданост апликације у LAN, развој web и мобилних апликација повећава доступност	3	2
<input type="checkbox"/>	Поузданост - кредибилитет	Извор података – аквизиција, директни унос	2	1
<input type="checkbox"/>		Профили корисника, Овлашћени приступ,	2	2
<b>НЕФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ</b>				
<input type="checkbox"/>	Погодност за одржавање	Издавање шифарника, довољан ниво апстракције дизајна базе података, модуларизација, вишеслојно програмирање	4	0
<input type="checkbox"/>	Модуларност	Издавање модула – Dynamic Link Libraries, Web Services	1	0
<input type="checkbox"/>	Конзистентност	Усклађеност делова апликације	1	1
<input type="checkbox"/>	Поновна искористивост	Модули са општим решењима, семантички независним, Генератори класа, Генератори корисничког интерфејса	3	0
<input type="checkbox"/>	Оптимално коришћење ресурса	Брзина рада кода, минимизација процесорског времена и меморијских ресурса	3	1
<input type="checkbox"/>	Поузданост	Верификација података, Коректност формула код израчунавања	2	1
<input type="checkbox"/>	Валидност	Минимизација грешака функционисања	1	1
<input type="checkbox"/>	Стабилност	Верификација података, Бекап података	2	1
<input type="checkbox"/>	Безбедност података	Профили корисника, Овлашћени приступ, криптографија	3	2
<input type="checkbox"/>	Прецизност	Тачност приликом обраде података (израчунавања, интеграција итд.)	2	1
<input type="checkbox"/>	Лакоћа коришћења	Интуитивни кориснички интерфејс, Аутоматизми, Изглед корисничког интерфејса у складу са навикама (менталним моделима) корисника	3	3

<input type="checkbox"/> Лакоћа учења коришћења апликације	Минимизација потребних корака у коришћењу апликације за решавање неког пословног задатка	1	1
<input type="checkbox"/> Прилагодљивост различитим оперативним окружењима	Технолошко решење које може бити коришћено на различитим оперативним системима рачунара и мобилних уређаја, у оквиру различитих web browser-a	2	2
<input type="checkbox"/> Интероперабилност са другим апликацијама	Могућност да коегзистира са другим апликацијама	1	1
	Могућност да читава податке из других app	1	0
	Могућност да доставља податке другим apps	1	0
	Могућност да аутоматски интегрише свој рад са радом у оквиру других апликација разменом података	1	0
<input type="checkbox"/> Могућност прилагођавања различитим захтевима – персонализација	Измене структуре података, изгледа извештаја, izgleda korisni;kog interfejsa	3	0
<input type="checkbox"/> Проширивост	Могућност додавања нових функционалности уз интеграцију са постојећим	1	1
<input type="checkbox"/> Ергономска погодност	Усклађеност боја са потребама корисника, величина фонта, распоред контрола, подршка за различите уређаје за улаз (тастатура, миш, читачи картица) и излаз података (екран, штампач...)	5	5
<input type="checkbox"/> Подршка за продуктивност	Аутоматизми, Минимизација потребних корака у коришћењу апликације за решавање неког пословног задатка	2	2
<input type="checkbox"/> Управљивост	Могућност да се врше измене у структури и начину функционисања апликације ради прилагођавања потребама пословања – кастомизација	2	0
<input type="checkbox"/> Усклађеност са стандардима	ИТ, семантички стандарди примене	2	0
<input type="checkbox"/> Квалитет програмског кода	Читљивост, рефакторисање, модуларност, коментари	4	2
<input type="checkbox"/> Документовано решење	Корисничка и техничка документација	2	2
БРОЈ ПОЕНА:		87	44
ПРОЦЕНАТ:			51%

Сумарни приказ резултата примене метричког модела за оцену квалитета софтвера, односно оцена квалитативних карактеристика реализованих студентских пројеката у области унапређења информационог система Техничког факултета „Михајло Пупин“ Зрењанин (детаљно описани у прилогу 9.5.), према групама карактеристика представљен је у наставку.

Табела 9.11.4.2. Сумарни приказ резултата примене метричког модела за оцену студентских софтверских апликација реализованих у шк. 2013/14 и 2014/15

ПРОСЕК:	9.28	10.33	31.50	51.12	58.75
<b>ОЗНАКА ПРОЈЕКТА</b>	<b>ФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ – број поена</b>	<b>КВАЛИТЕТ ПОДАТАКА – број поена</b>	<b>НЕФУНКЦИОНАЛНЕ КАРАКТЕРИСТИКЕ – број поена</b>	<b>УКУПНО</b>	<b>ПРОЦЕНАТ %</b>
Мах поена:	21.00	13.00	53.00	87	100

Љубица Кази - Докторска дисертација

2014_1	7.00	10.00	27.00	44	51
2014_2	11.00	11.00	38.00	60	69
2014_3	9.00	11.00	33.00	53	61
2014_4	8.00	11.00	30.00	49	56
2014_5	11.50	11.00	38.00	60.5	70
2014_6	12.50	11.00	40.00	63.5	73
2014_7	12.00	11.00	38.00	61	70
2014_8	5.00	9.50	25.50	40	46
2014_9	13.00	11.00	41.00	65	75
2014_10	12.00	11.00	40.00	63	72
2014_11	9.50	10.00	33.00	52.5	60
2014_12	8.00	9.00	23.00	40	46
2014_13	8.00	9.00	27.50	44.5	51
2014_14	8.50	10.50	31.00	50	57
2014_15	6.00	9.50	25.50	41	47
2014_16	9.50	11.00	32.00	52.5	60
2014_17	4.00	7.50	21.00	32.5	37
2014_18	7.50	9.50	29.00	46	53
2014_19	6.50	10.50	31.00	48	55
2015_1	12.00	11.00	32.00	55	63
2015_2	12.00	11.00	34.00	57	66
2015_3	13.00	11.00	36.50	60.5	70
2015_4	17.00	11.00	37.50	65.5	75
2015_5	12.00	11.00	36.50	59.5	68
2015_6	10.00	11.00	31.00	52	60
2015_7	6.00	8.00	23.00	37	43
2015_8	8.50	10.50	26.50	45.5	52
2015_9	9.50	11.00	32.00	52.5	60
2015_10	4.50	10.00	26.00	40.5	47
2015_11	5.50	10.50	26.50	42.5	49

Табела 9.11.4.3. Сумарни приказ резултата примене метричког модела за оцену студентских софтверских апликација реализованих у шк. 2013/14 и 2014/15 - након примене тежинских фактора

ПРОСЕК:	44.21	79.49	59.43	58.89
<b>ОЗНАКА ПРОЈЕКТА</b>	<b>ПРОЦЕНАТ ФУНКЦИОНАЛНЕ карактеристике</b>	<b>ПРОЦЕНАТ КВАЛИТЕТ ПОДАТАКА</b>	<b>ПРОЦЕНАТ НЕФУНКЦИОНАЛНЕ карактеристике</b>	<b>Укупно проценат применом ТЕЖИНСКОГ ФАКТОРА</b>
<b>Тежински фактор:</b>	<b>1.00</b>	<b>0.70</b>	<b>0.50</b>	<b>2.20</b>
2014_1	33.33	76.92	50.94	51.21
2014_2	52.38	84.62	71.70	67.03
2014_3	42.86	84.62	62.26	60.55
2014_4	38.10	84.62	56.60	57.10
2014_5	54.76	84.62	71.70	68.11
2014_6	59.52	84.62	75.47	71.13

Љубица Кази - Докторска дисертација

2014_7	57.14	84.62	71.70	69.19
2014_8	23.81	73.08	48.11	45.01
2014_9	61.90	84.62	77.36	72.64
2014_10	57.14	84.62	75.47	70.05
2014_11	45.24	76.92	62.26	59.19
2014_12	38.10	69.23	43.40	49.21
2014_13	38.10	69.23	51.89	51.14
2014_14	40.48	80.77	58.49	57.39
2014_15	28.57	73.08	48.11	47.17
2014_16	45.24	84.62	60.38	61.21
2014_17	19.05	57.69	39.62	36.02
2014_18	35.71	73.08	54.72	51.92
2014_19	30.95	80.77	58.49	53.06
2015_1	57.14	84.62	60.38	66.62
2015_2	57.14	84.62	64.15	67.48
2015_3	61.90	84.62	68.87	70.71
2015_4	80.95	84.62	70.75	79.80
2015_5	57.14	84.62	68.87	68.55
2015_6	47.62	84.62	58.49	61.86
2015_7	28.57	61.54	43.40	42.43
2015_8	40.48	80.77	50.00	55.46
2015_9	45.24	84.62	60.38	61.21
2015_10	21.43	76.92	49.06	45.37
2015_11	26.19	80.77	50.00	48.97

## 9.12. Подаци из примене метричких модела за мониторинг успеха и процену трајања софтверског пројекта

### 9.12.1. Примена модела за мониторинг успеха процеса развоја софтвера, односно успеха софтверског пројекта у области развоја информационих система

У наставку ће бити приказана илустрација могућности коришћења метричког модела за оперативни ниво планирања и праћења успеха реализације софтверског пројекта у области развоја информационог система на примеру реализације софтвера „Алумни“. Матрични приказ кључних елемената за праћење успеха реализације софтверског пројекта у области информационих система који представља основ метричког модела на оперативном нивоу приказан је у одељку 3.4.2., а подаци за примену наведеног модела су усклађени са догађајима и записима у оквиру пројекта „Алумни“ који су описани у табели 9.3.2.1.1. у одељку са прилозима.

Први облик представљања резултата софтверског пројекта у области развоја информационих система представљен је у кумулативном начину уноса.

Табела 9.12.1. Приказ примене модела за оперативни ниво мониторинга успеха софтверског пројекта у области развоја информационог система заснованог на агилном приступу – кумулативан приказ

<b>ДАТУМ:</b> <b>13.3.2014.</b>	Дато: спецификација захтева Задато: Израда бизнис процес модела, дијаграма случаја коришћења и концептуалног модела података				
<b>ФАЗЕ</b>	<b>КОНЦЕПТ РЕШЕЊА</b>			<b>ЗАДАЦИ</b>	
ЕЛЕМЕНТ	примитивних процеса <b>задато (13.3.)</b>	Екранских форми	табела у бази података <b>задато (13.3.)</b>	„user stories“	Задатака <b>Задато (13.3.) =5</b>
	софтверских функција <b>задато (13.3.)</b>	Извештаја	Други формати фајлова	спецификација захтева <b>дато (13.3.): 1</b>	
	случајева коришћења <b>задато (13.3.)</b>	Процедура обраде			
	Профила корисника <b>задато (13.3.)</b>				
	ОСТАЛО				
	Тестирање	Документовање	Инсталација	Обука	Верификација од стране корисника /менаџмента

<b>ДАТУМ:</b> <b>25.3.2015.</b>	Верификовани: BPM дијаграм, USE CASE дијаграми, CDM дијаграми Задато: Дизајн корисничког интерфејса				
<b>ФАЗЕ</b>	<b>КОНЦЕПТ РЕШЕЊА</b>			<b>ЗАДАЦИ</b>	
ЕЛЕМЕНТ	примитивних процеса <b>задато (13.3.) верификовано</b>	Екранских форми <b>Задато (25.3.)</b>	табела у бази података <b>задато(13.3) верификовано</b>	„user stories“	Задатака <b>Задато (25.3.) =1</b>

	<b>= 3 (25.3.)</b>		<b>=1 (25.3.)</b>		
	софтверских функција – <b>задато (13.3.)</b> <b>верификовано =7 (25.3.)</b>	Извештаја	Други формати фајлова	спецификација захтева <b>дато: 1 (13.3.)</b>	
	случајева коришћења <b>задато (13.3.)</b> <b>верификовано =7 (25.3.)</b>	Процедура обраде			
	Профила корисника <b>задато (13.3.)</b> <b>верификовано = 3 (25.3.)</b>				
	ОСТАЛО				
	Тестирање	Документовање	Инсталација	Обука	Верификација од стране корисника /менаџмента

<b>ДАТУМ:</b> <b>1.4.2015.</b>	Верификовани: Дизајн корисничког интерфејса Задато: Унапређење дизајна корисничког интерфејса				
<b>ФАЗЕ</b>	<b>КОНЦЕПТ РЕШЕЊА</b>			<b>ЗАДАЦИ</b>	
ЕЛЕМЕНТ	примитивних процеса <b>задато (13.3.)</b> <b>верификовано = 3 (25.3.)</b>	Екранских форми <b>Задато (25.3.)</b> <b>Верификовано (1.4.)</b> <b>Задато (1.4.)</b>	табела у бази података <b>задато(13.3)</b> <b>верификовано =1 (25.3.)</b>	„user stories“	Задатака <b>Задато (1.4.) = 1</b>
	софтверских функција – <b>задато (13.3.)</b> <b>верификовано =7 (25.3.)</b>	Извештаја	Други формати фајлова	спецификација захтева <b>дато: 1 (13.3.)</b>	
	случајева коришћења <b>задато (13.3.)</b> <b>верификовано =7 (25.3.)</b>	Процедура обраде			
	Профила корисника <b>задато (13.3.)</b> <b>верификовано = 3 (25.3.)</b>				
	ОСТАЛО				
	Тестирање	Документовање	Инсталација	Обука	Верификација од стране корисника /менаџмента

<b>ДАТУМ:</b> <b>2.4.2015.</b>	Верификовани: Дизајн корисничког интерфејса Задато: Измене дизајна корисничког интерфејса				
<b>ФАЗЕ</b>	<b>КОНЦЕПТ РЕШЕЊА</b>			<b>ЗАДАЦИ</b>	
ЕЛЕМЕНТ	примитивних процеса <b>задато (13.3.)</b> <b>верификовано (25.3.)</b> <b>= 3</b>	Екранских форми <b>Задато (25.3.)</b> <b>Верификовано (1.4.)</b> <b>Задато (1.4.)</b> <b>Верификовано (2.4.)</b>	табела у бази података <b>задато(13.3)</b> <b>верификовано (25.3.)</b> <b>=1</b>	„user stories“	Задатака <b>Задато (2.4.) = 1</b>



		<b>Задато (2.4.)</b>			
	софтверских функција – <b>задато (13.3.) верификовано (25.3.) =7</b>	Извештаја	Други формати фајлова	спецификација захтева <b>дато: 1 (13.3.)</b>	
	случајева коришћења <b>задато (13.3.) верификовано (25.3.) =7</b>	Процедура обраде			
	Профила корисника <b>задато (13.3.) верификовано (25.3.) = 3</b>				
	ОСТАЛО				
	Тестирање	Документовање	Инсталација	Обука	Верификација од стране корисника /менаџмента <b>Верификовано (2.4.)</b>

Претходне 4 табеле представљају почетне резултате примене предложеног метричког приступа у мониторингу успеха реализације софтверског пројекта „Alumni Web“. Због просторних ограничења у овом документу, нису наведене све табеле у мониторингу, већ само почетне 4 табеле, како би илустровао начин коришћења предложеног приступа.

У наставку представљен је ток целокупног процеса развоја софтвера „Алумни“ у другој форми матрице. Свака матрица представља појединачни приказ активности које су реализоване само тог дана.

Табела 9.12.2. Приказ примене модела за оперативни ниво мониторинга успеха софтверског пројекта у области развоја информационог система заснованог на агилном приступу – појединачни приказ

Датум:		13.3.2014.									
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 4	1	1	1	1					
1		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум:		25.3.2015.									
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато					1				
		Урађено 4	1	1	1	1					
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 1.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 1					1				
		Урађено 1					1				
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 2.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 1					1				
		Урађено 1					1				
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1					1				

Датум: 3.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 1					1				
		Урађено 1					1				
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 9.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1			1						

Датум: 16.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 1					1				
		Урађено 1					1				
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 18.4.2014.											
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено 1					1				
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1			1						

Датум: 22.4.2014.											
-------------------	--	--	--	--	--	--	--	--	--	--	--

Љубица Кази - Докторска дисертација

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 2		1					1		
		Урађено 2					1			1	
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 6.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 4		2					2		
		Урађено 2		1					1		
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 8.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено 4		2					2		
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено									

Датум: 13.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато 2		1					1		
		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1	1								

Датум: 24.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1		1							

Датум: 27.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено 2		1					1		
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 1		1							

Датум: 28.5.2014.

Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено							1		

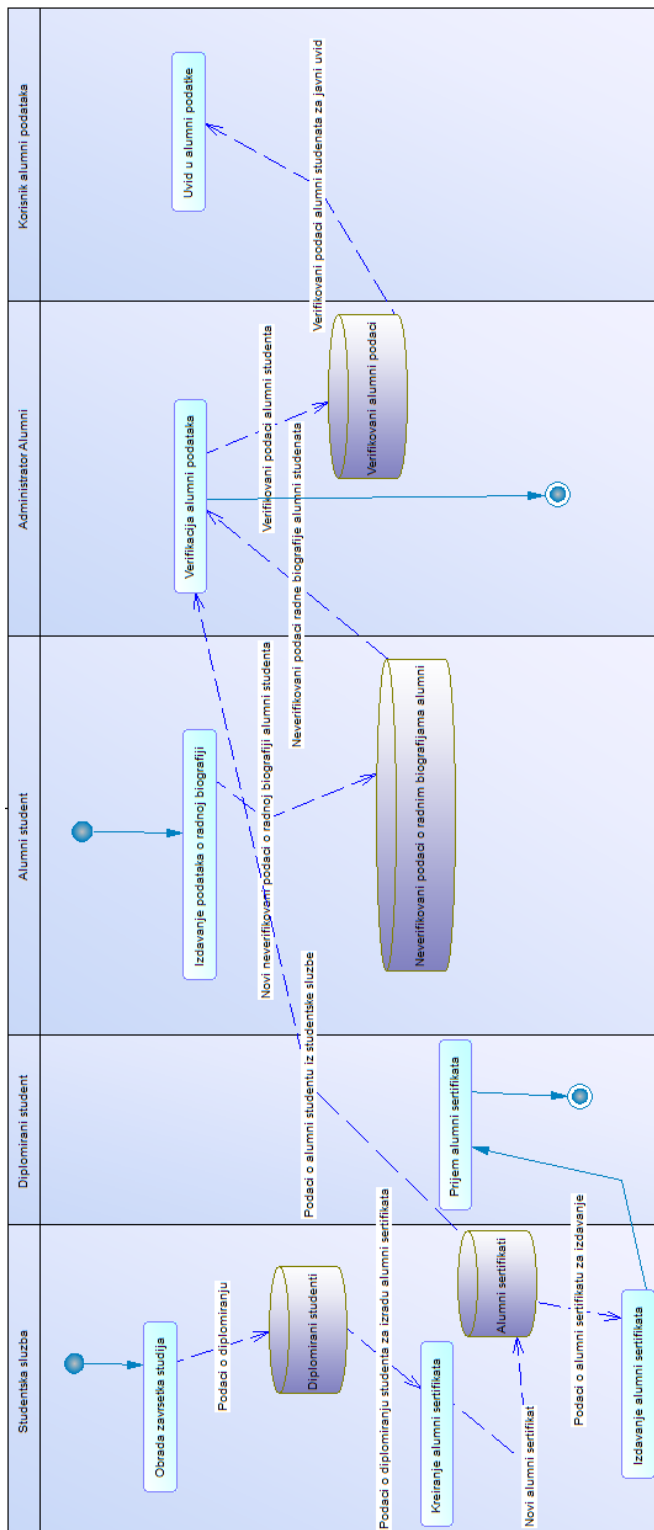
Љубица Кази - Докторска дисертација

		1									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено	1								

Датум:	4.7.2014.										
Спец. Зах.	User stories	Задатака	Прим. Проц	Софт. фја	Случ. Кор.	Проф. кор.	Екр. форма	Изв.	Проц. Обр.	Табела бп	Др. ф.п.
		Задато									
		Урађено									
			Тест	Док.	Инстал.	Обука	Вериф.				
		Задато									
		Урађено 2	1		1						

### 9.12.6. Примена модела за процену трајања софтверског пројекта на основу модела пословних процеса

У наставку ће бити дата илустрација начина примене метричког модела за процену трајања софтверског пројекта на основу једноставног модела пословних процеса на примеру „Алумни“.



Слика 9.12.6.1. Пример бизнис процес модела на примеру „Алумни“

У оквиру наведеног примера, можемо узети као пример пројекта где је дефинисано да су изван граница обухвата пројекта организационе јединице („вертикале дијаграма“)

модела које се називају „студентска служба“ и „дипломирани студент“. У оквиру граница обухвата пројекта су процеси који се односе на алумни студента и администратора. Наравно, у неком другом пројекту могао би се обухватити целокупан модел пословних процеса.

Табела 9.12.6.1. Приказ примене метричког модела за процену трајања софтверског пројекта на основу модела пословних процеса на примеру „Алумни“

<b>Бизнис процес модел</b>			
<b>ПРОЦЕС</b>	<b>ТОК ПОДАТАКА</b>	<b>СКЛАДИШТЕ ПОДАТАКА</b>	<b>ОРГАНИЗАЦИОНА ЈЕДИНИЦА</b>
<p><b>У оквиру граница обухвата:</b></p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> <li>- Верификација алумни података</li> <li>- Увид у алумни податке</li> </ul> <p>-----</p> <p><b>Може да се аутоматизује:</b></p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> <li>- Верификација алумни података</li> <li>- Увид у алумни податке</li> </ul> <p>-----</p> <p><b>УЛАЗНИ</b> – коеф. Тежине: 1</p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> </ul> <p><b>ИЗЛАЗНИ</b> - коеф. Тежине: 2</p> <ul style="list-style-type: none"> <li>- Увид у алумни податке</li> </ul> <p><b>ОБРАДА</b> – коеф. Тежине: 3</p> <ul style="list-style-type: none"> <li>- Верификација алумни података</li> </ul> <p>-----</p> <p><b>Број токова података са којима је процес повезан</b></p> <p><b>1 – коефицијент 1</b></p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> <li>- Увид у алумни податке</li> </ul> <p><b>“&gt;1” - коефицијент 2</b></p> <ul style="list-style-type: none"> <li>- Верификација алумни података</li> </ul> <p>-----</p> <p><b>Број складишта података са којима је процес повезан</b></p> <p><b>1 – коефицијент 1</b></p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> <li>- Увид у алумни податке</li> </ul> <p><b>“&gt;1” - коефицијент 2</b></p> <ul style="list-style-type: none"> <li>- Верификација алумни података</li> </ul>	<p><b>Број елементарних података,</b></p> <p><b>“do 10” – коефицијент 1</b></p> <ul style="list-style-type: none"> <li>- Подаци о радној биографији алумни студента</li> <li>- Неверификовани подаци радне биографије алумни студената</li> <li>- Верификовани подаци алумни студента</li> <li>- Верификовани подаци алумни студената за јавни увид</li> </ul> <p><b>“&gt;10” – коефицијент 2</b></p> <p>-----</p> <p>повезаност елементарних података – синтаксна анализа <b>jednostavan – 1</b></p> <ul style="list-style-type: none"> <li>- Подаци о радној биографији алумни студента</li> <li>- Неверификовани подаци радне биографије алумни студената</li> <li>- Верификовани подаци алумни студента</li> <li>- Верификовани подаци алумни студената за јавни увид</li> </ul> <p>-----</p> <p>podstruktura 1:M -2</p>	<p><b>број елементарних података,</b></p> <p><b>“do 10” – коефицијент 1</b></p> <ul style="list-style-type: none"> <li>- Подаци о радним биографијама алумни</li> <li>- Верификовани алумни подаци</li> </ul> <p><b>“&gt;10” – коефицијент 2</b></p> <p>-----</p> <p>повезаност елементарних података – синтаксна анализа <b>jednostavan – 1</b></p> <ul style="list-style-type: none"> <li>- Подаци о радним биографијама алумни</li> <li>- Верификовани алумни подаци</li> </ul> <p>podstruktura 1:M -2</p>	<p><b>Алумни студент:</b></p> <p>Број процеса:</p> <ul style="list-style-type: none"> <li>- Издавање података о радној биографији</li> </ul> <p>Број складишта:</p> <ul style="list-style-type: none"> <li>- Неверификовани алумни подаци</li> </ul> <p>Број токова података:</p> <ul style="list-style-type: none"> <li>- Нови неверификовани подаци о радној биографији алумни студента</li> </ul> <p><b>Администратор алумни:</b></p> <p>Број процеса:</p> <ul style="list-style-type: none"> <li>- Верификација алумни података</li> </ul> <p>Број складишта:</p> <ul style="list-style-type: none"> <li>- Верификовани алумни подаци</li> </ul> <p>Број токова података:</p> <ul style="list-style-type: none"> <li>- Неверификовани подаци радне биографије алумни студената</li> <li>- Верификовани подаци алумни студента</li> </ul> <p><b>Корисник алумни података:</b></p> <p>Број процеса:</p> <ul style="list-style-type: none"> <li>- Увид у алумни податке</li> </ul> <p>Број складишта: коефицијент 3</p> <p>Број токова података:</p> <ul style="list-style-type: none"> <li>- Верификовани подаци алумни студената за јавни увид</li> </ul> <p>-----</p> <p><b>Тip:</b></p> <p><b>Интерна</b> (унутар организације)</p> <ul style="list-style-type: none"> <li>- Администратор алумни</li> </ul> <p><b>Екстерна</b> (ван организације)</p> <ul style="list-style-type: none"> <li>- Алумни студент</li> <li>- Корисник алумни података</li> </ul>

## 9.13. ЛИСТИНГ КОДА

### 9.13.1. Листинг кључних делова кода почетног прототипа

#### 9.13.1.1. Табеларни приказ пројеката пријављеног корисника

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
// Dodato
using System.Data;
using System.Configuration;
using System.Data.SqlClient;

public partial class ProjectListUser : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack)
        {
            // ponovno pokretanje
        }
        else
        {
            // prvo pokretanje
            string ImePrezime = Session["glKorisnik"].ToString();
            string Ime = clsOpste.IzdvojIme(ImePrezime);
            string Prezime = clsOpste.IzdvojPrezime(ImePrezime);

            //
            SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
            Veza.Open();
            String strSQL;
            strSQL = "Select distinct tbProject.ID, tbProject.ProjTitle as [Project title],
CONVERT(VARCHAR(20), DATENAME(MM, tbProject.StartDate)) + ' ' + CAST(Day(tbProject.StartDate) as
nvarchar(2)) + ', ' + CAST(Year(tbProject.StartDate) as nvarchar(4)) as [Proj. start date],
tbProject.DurationMonths as [Duration (months)], tbPartRole.RoleName as [MY ASSIGNMENT] from tbProject,
tbUcesnik, tbPartProjAssign, tbPartRole where tbPartProjAssign.IDproj=tbProject.ID and
tbPartProjAssign.Idpart= tbUcesnik.IDpart and tbPartProjAssign.IDRole = tbPartRole.ID and
tbUcesnik.Surname='" + Prezime + "' and tbUcesnik.Name='" + Ime + "' and tbProject.EndDate IS NULL order
by tbProject.ProjTitle";
            SqlCommand Komanda = new SqlCommand(strSQL, Veza);
            SqlDataReader dr = Komanda.ExecuteReader();
            gvMojiProjekti.DataSource = dr;
            gvMojiProjekti.DataBind();
            dr.Close();
            Veza.Close();
            Veza.Dispose();
        }
    }

    protected void gvMojiProjekti_SelectedIndexChanged(object sender, EventArgs e)
    {
        GridViewRowCollection gvRowColl = gvMojiProjekti.Rows;

        if (gvRowColl[gvMojiProjekti.SelectedIndex].Cells[0].HasControls())
        {
            string celija = gvRowColl[gvMojiProjekti.SelectedIndex].Cells[1].Text ;

            Session.Add("IDSelectedProject", celija);
            Response.Redirect("ProjectRecords.aspx");
        }
        else
        {
        }
    }
}

```

### 9.13.1.2. Снимање записа у оквиру изабраног пројекта

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
// Dodato
using System.Data;
using System.Configuration;
using System.Data.SqlClient;

public partial class ProjectRecordAdd : System.Web.UI.Page
{
    private void NapuniComboSegment()
    {
        SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
        Veza.Open();
        String strSQL;
        strSQL = "Select ProjSegName from tbProjSegment order by OrderNo";
        SqlCommand Komanda = new SqlCommand(strSQL, Veza);
        SqlDataReader dr = Komanda.ExecuteReader();

        ddlSegment.Items.Clear();
        ddlSegment.Items.Add("choose...");
        while (dr.Read())
        {
            ddlSegment.Items.Add(dr["ProjSegName"].ToString());
        }
        dr.Close();
        Veza.Close();
        Veza.Dispose();
    }

    private void NapuniComboArtifakt()
    {
        SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
        Veza.Open();
        String strSQL;
        strSQL = "Select SoftArtName from tbSoftArtifact order by OrderNo";
        SqlCommand Komanda = new SqlCommand(strSQL, Veza);
        SqlDataReader dr = Komanda.ExecuteReader();

        ddlArtifakt.Items.Clear();
        ddlArtifakt.Items.Add("choose...");
        while (dr.Read())
        {
            ddlArtifakt.Items.Add(dr["SoftArtName"].ToString());
        }
        dr.Close();
        Veza.Close();
        Veza.Dispose();
    }

    private void NapuniComboFaza()
    {
        SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
        Veza.Open();
        String strSQL;
        strSQL = "Select SofDevPhName from tbSoftDevPhase order by OrderNo";
        SqlCommand Komanda = new SqlCommand(strSQL, Veza);
        SqlDataReader dr = Komanda.ExecuteReader();

        ddlFaza.Items.Clear();
        ddlFaza.Items.Add("choose...");
        while (dr.Read())
        {
            ddlFaza.Items.Add(dr["SofDevPhName"].ToString());
        }
    }
}

```



```

dr.Close();
Veza.Close();
Veza.Dispose();

}
private void PrikaziNazivProjekta()
{
    // PREUZIMANJE ID PROJEKTA
    string IDprojekta = Session["IDSelectedProject"].ToString();

    // PREUZIMANJE NAZIVA PROJEKTA
    SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
    Veza.Open();
    String strSQL;
    strSQL = "Select tbProject.ProjTitle from tbProject where tbProject.ID=" + IDprojekta;
    SqlCommand Komanda = new SqlCommand(strSQL, Veza);
    SqlDataReader dr = Komanda.ExecuteReader();

    if (dr.Read())
    {
        if (dr["ProjTitle"] != null)
        {
            // PRIKAZ NAZIVA PROJEKTA
            lblProjectName.Text = dr["ProjTitle"].ToString();
        }
    }

    dr.Close();
    Veza.Close();
    Veza.Dispose();
}

private void PrikaziUcesnikeProjekta()
{
    // PREUZIMANJE ID PROJEKTA
    string IDprojekta = Session["IDSelectedProject"].ToString();

    // PREUZIMANJE NAZIVA PROJEKTA
    SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
    Veza.Open();
    String strSQL;
    strSQL = "Select tbUcesnik.Surname + ' ' + tbUcesnik.Name as Ucesnik from
tbUcesnik,tbPartProjAssign, tbProject where tbUcesnik.IDpart=tbPartProjAssign.Idpart and
tbPartProjAssign.IDproj=tbProject.ID and tbProject.ID=" + IDprojekta + " and tbUcesnik.IDpart <>1";
    SqlCommand Komanda = new SqlCommand(strSQL, Veza);
    SqlDataReader dr = Komanda.ExecuteReader();

    string ucesnici = "";
    while (dr.Read())
    {
        if (dr["Ucesnik"] != null)
        {
            // PRIKAZ NAZIVA PROJEKTA
            if (ucesnici.Equals(""))
            {
                ucesnici = dr["Ucesnik"].ToString();
            }
            else
            {
                ucesnici = ucesnici + "; " + dr["Ucesnik"].ToString();
            }
        }
    }
    lblParticipants.Text = ucesnici;
    dr.Close();
    Veza.Close();
    Veza.Dispose();
}

private void PrikaziDanasnjiDan()
{
    txtDan.Text = DateTime.Now.Day.ToString();
    txtGodina.Text = DateTime.Now.Year.ToString();
    ddlMesec.Text = clsOpste.DajNazivMeseca(DateTime.Now.Month);
}

```

```

}
protected void Page_Load(object sender, EventArgs e)
{
    if (IsPostBack)
    {

    }
    else
    {
        PrikaziDanasnjiDan();
//
        PrikaziNazivProjekta();
        PrikaziUcesnikeProjekta();
        NapuniComboSegment();
        NapuniComboArtifakt();
        NapuniComboFaza();
    }
}
protected void btnSubmit_Click(object sender, EventArgs e)
{

// -----AUTOMATSKO PREUZIMANJE -----
string IDKorisnika = Session["IDKorisnika"].ToString();
//
string IDprojekta = Session["IDSelectedProject"].ToString();
int IDpartproj = clsAzuriraj.DajIdPartProjektaZaID(IDprojekta, IDKorisnika);
string IDpartprojStr = IDpartproj.ToString();
string DatumSnimanja = "" + clsOpste.DajBrojMeseca (ddlMesec.Text) + "/" + txtDan.Text + "/" +
txtGodina.Text + """;
//
// -----SA KORISNICKOG INTERFEJSA-----
string naslov = txtTitle.Text;
string opis = txtDescription.Text;
string segment = ddlSegment.Text;
string artifakt = ddlArtifakt.Text;
string faza = ddlFaza.Text;
// *****
// -----DODAVANJE -----
// *****
int brojslogova = 0;

if (naslov.Length > 0)
{
    brojslogova = clsAzuriraj.DodajProjectRecordZapisCombo(IDprojekta, IDpartprojStr,
DatumSnimanja,naslov, opis, segment, artifakt, faza);
}

// *****
// ----- snimanje u attachment -----
// *****

if (!txtFileDesc.Text.Equals(""))
{

    int brojRecord = clsAzuriraj.DajBrojRecord();
    string strbrojRecord = brojRecord.ToString();
    string opisfajla = txtFileDesc.Text;
    string kracaputanja = "~/fajlovi/" + FileUpload1.FileName;
    //

    int brojslogova1 = clsAzuriraj.DodajAttachment(strbrojRecord, opisfajla, kracaputanja);

    try
    {
        FileUpload1.PostedFile.SaveAs(Server.MapPath("~/") + "fajlovi/" + FileUpload1.FileName);
    }
    catch
    {

    }
}

lblUspeh.Visible = true;

if (brojslogova > 0)

```

```

    {
        lblUspeh.Text = "Successful add!";

        // Msg box
        string myStringVariable = "Successful add!";
        ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myStringVariable +
        "');", true);
    }
    else
    {
        lblUspeh.Text = "Error in adding!";
        // Msg box
        string myStringVariable = "Error in adding!";
        ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myStringVariable +
        "');", true);
    }
}

```

### 9.13.1.3. Табеларни приказ записа (радних ставки) изабраног пројекта са подршком визуализацији статуса реализације фазе пројекта

```

using System;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
// Dodato
using System.Data;
using System.Configuration;
using System.Data.SqlClient;

public partial class ProjectRecords : System.Web.UI.Page
{
    private void ObojiTaster(string filter)
    {
        string IDprojekta = Session["IDSelectedProject"].ToString();
        SqlConnection Veza1 = new
        SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
        Veza1.Open();
        String strSQL1;

        strSQL1 = "select tbSoftDevPhase.SofDevPhName as [Phase] from ((((((tbUcesnik inner join
        tbPartProjAssign on tbUcesnik.IDpart=tbPartProjAssign.Idpart) inner join tbProject on
        tbPartProjAssign.IDproj=tbProject.ID) inner join tbProjectRecord on tbProjectRecord.IDproj=tbProject.ID
        and tbProjectRecord.IdProjPartic=tbPartProjAssign.ID) inner join tbProjSegment on tbProjSegment.ID=
        tbProjectRecord.IDProjSegm) inner join tbSoftArtifact on tbProjectRecord.IDSoftArtif=tbSoftArtifact.ID)
        inner join tbSoftDevPhase on tbProjectRecord.IdSoftPhase=tbSoftDevPhase.ID) left join tbAttachments on
        tbAttachments.IDprojRecord=tbProjectRecord.ID where tbProject.ID=" + IDprojekta + " and
        tbSoftDevPhase.SofDevPhName='" + filter + "'";
        SqlCommand Komanda1 = new SqlCommand(strSQL1, Veza1);
        SqlDataReader dr1 = Komanda1.ExecuteReader();

        if ((dr1.HasRows))
        {
            switch (filter)
            {
                case "requirements":
                    btnRequirements.BackColor = System.Drawing.Color.Yellow;
                    break;
                case "data model":
                    btnDataModel.BackColor = System.Drawing.Color.Yellow;
                    break;
                case "use case":
                    btnUseCaseModel.BackColor = System.Drawing.Color.Yellow;
                    break;
                case "database":
                    btnDatabase.BackColor = System.Drawing.Color.Yellow;
                    break;
                case "user interface":
                    btnUserInterface.BackColor = System.Drawing.Color.Yellow;
                    break;
                case "programming":
                    btnProgramming.BackColor = System.Drawing.Color.Yellow;
                    break;
            }
        }
    }
}

```

```

        case "testing":
            btnTesting.BackColor = System.Drawing.Color.Yellow;
            break;
        case "documenting":
            btnDocumenting.BackColor = System.Drawing.Color.Yellow;
            break;
        case "deployment":
            btnDeployment.BackColor = System.Drawing.Color.Yellow;
            break;
        case "finalization":
            btnFinalization.BackColor = System.Drawing.Color.Yellow;
            break;
    }
}
dr1.Close();
Veza1.Close();
Veza1.Dispose();
}

private void ObojiTastere()
{
    btnRequirements.BackColor = System.Drawing.Color.LightGray;
    btnDataModel.BackColor = System.Drawing.Color.LightGray;
    btnUseCaseModel.BackColor = System.Drawing.Color.LightGray;
    btnUserInterface.BackColor = System.Drawing.Color.LightGray;
    btnDatabase.BackColor = System.Drawing.Color.LightGray;
    //
    btnProgramming.BackColor = System.Drawing.Color.LightGray;
    btnTesting.BackColor = System.Drawing.Color.LightGray;
    btnDocumenting.BackColor = System.Drawing.Color.LightGray;
    btnDeployment.BackColor = System.Drawing.Color.LightGray;
    btnFinalization.BackColor = System.Drawing.Color.LightGray;
    ObojiTaster("requirements");
    ObojiTaster("data model");
    ObojiTaster("use case");
    ObojiTaster("database");
    ObojiTaster("user interface");
    //
    ObojiTaster("programming");
    ObojiTaster("testing");
    ObojiTaster("documenting");
    ObojiTaster("deployment");
    ObojiTaster("finalization");
}

private void PrikaziNazivProjekta()
{
    // PREUZIMANJE ID PROJEKTA
    string IDprojekta = Session["IDSelectedProject"].ToString();

    // PREUZIMANJE NAZIVA PROJEKTA
    SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
    Veza.Open();
    String strSQL;
    strSQL = "Select tbProject.ProjTitle from tbProject where tbProject.ID=" + IDprojekta;
    SqlCommand Komanda = new SqlCommand(strSQL, Veza);
    SqlDataReader dr = Komanda.ExecuteReader();

    if (dr.Read())
    {
        if (dr["ProjTitle"] != null)
        {
            // PRIKAZ NAZIVA PROJEKTA
            lblProjectName.Text = dr["ProjTitle"].ToString();
        }
    }
    dr.Close();
    Veza.Close();
    Veza.Dispose();
}

private void PrikaziUcesnikeProjekta()
{
    // PREUZIMANJE ID PROJEKTA

```

```

string IDprojekta = Session["IDSelectedProject"].ToString();

// PREUZIMANJE NAZIVA PROJEKTA
SqlConnection Veza = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
Veza.Open();
String strSQL;
strSQL = "Select tbUcesnik.Surname + ' ' + tbUcesnik.Name as Ucesnik, tbUcesnik.Email from
tbUcesnik,tbPartProjAssign, tbProject where tbUcesnik.IDpart=tbPartProjAssign.Idpart and
tbPartProjAssign.IDproj=tbProject.ID and tbProject.ID=" + IDprojekta + " and tbUcesnik.IDpart <>1 and
tbPartProjAssign.DateFinalized IS NULL";
SqlCommand Komanda = new SqlCommand(strSQL, Veza);
SqlDataReader dr = Komanda.ExecuteReader();

string ucesnici = "";
string emailovi = "";
while (dr.Read())
{
    if (dr["Ucesnik"] != null)
    {
        // PRIKAZ NAZIVA PROJEKTA
        if (ucesnici.Equals(""))
        {
            ucesnici= dr["Ucesnik"].ToString();
            emailovi = dr["Email"].ToString();
        }
        else
        {
            ucesnici = ucesnici + "; " + dr["Ucesnik"].ToString();
            emailovi = emailovi + "; " + dr["Email"].ToString();
        }
    }
}
lblParticipants.Text = ucesnici;
lblEmails.Text = emailovi;
dr.Close();
Veza.Close();
Veza.Dispose();
}

private void PrikaziRecordsProjekta()
{
    string IDprojekta = Session["IDSelectedProject"].ToString();
    SqlConnection Veza1 = new
SqlConnection(ConfigurationManager.ConnectionStrings["WebKonekcija"].ConnectionString);
Veza1.Open();
String strSQL1;

    strSQL1 = "select tbProjectRecord.ID, CONVERT(VARCHAR(20), DATENAME(MM,
tbProjectRecord.DateSubmitted)) + ' ' + CAST(Day(tbProjectRecord.DateSubmitted) as nvarchar(2)) + ', ' +
CAST(Year(tbProjectRecord.DateSubmitted) as nvarchar(4)) as [Date], tbUcesnik.Surname + ' ' +
tbUcesnik.Name as [FROM], tbProjectRecord.Title as [Title], tbProjSegment.ProjSegName as [Segment],
tbSoftArtifact.SoftArtName as [Artifact], tbSoftDevPhase.SofDevPhName as [Phase], count(tbAttachments.ID)
as [Attachments] from (((((tbUcesnik inner join tbPartProjAssign on
tbUcesnik.IDpart=tbPartProjAssign.Idpart) inner join tbProject on tbPartProjAssign.IDproj=tbProject.ID)
inner join tbProjectRecord on tbProjectRecord.IDproj=tbProject.ID and
tbProjectRecord.IdProjPartic=tbPartProjAssign.ID) inner join tbProjSegment on tbProjSegment.ID=
tbProjectRecord.IDProjSegm) inner join tbSoftArtifact on tbProjectRecord.IDSoftArtif=tbSoftArtifact.ID)
inner join tbSoftDevPhase on tbProjectRecord.IdSoftPhase=tbSoftDevPhase.ID) left join tbAttachments on
tbAttachments.IDprojRecord=tbProjectRecord.ID where tbProject.ID=" + IDprojekta + " group by
tbProjectRecord.ID, tbProjectRecord.DateSubmitted, tbUcesnik.Surname + ' ' + tbUcesnik.Name,
tbProjectRecord.Title, tbProjSegment.ProjSegName, tbSoftArtifact.SoftArtName, tbSoftDevPhase.SofDevPhName
order by tbProjectRecord.DateSubmitted desc, tbProjectRecord.ID desc";
    SqlCommand Komanda1 = new SqlCommand(strSQL1, Veza1);
    SqlDataReader dr1 = Komanda1.ExecuteReader();

    if (dr1.HasRows)
    {
        gvProjectRecord.DataSource = dr1;
        gvProjectRecord.DataBind();
    }

    dr1.Close();
    Veza1.Close();
    Veza1.Dispose();
}

```

```

protected void Page_Load(object sender, EventArgs e)
{
    PrikaziNazivProjekta();
    PrikaziUcesnikeProjekta();
    PrikaziRecordsProjekta();
    ObojiTastere();
}

protected void gvProjectRecord_SelectedIndexChanged(object sender, EventArgs e)
{
    GridViewRowCollection gvRowColl = gvProjectRecord.Rows;

    if (gvRowColl[gvProjectRecord.SelectedIndex].Cells[0].HasControls())
    {
        //System.Web.UI.WebControls.DataControlFieldCell celija = new
        System.Web.UI.WebControls.DataControlFieldCell();
        string celija = gvRowColl[gvProjectRecord.SelectedIndex].Cells[1].Text;

        Session.Add("SelectedIDprojectrecord", celija);
        Response.Redirect("ProjectRecordView.aspx");
    }
    else
    {
    }
}
}

```

## 9.13.2. Листинг кода унапређења прототипа

### 9.13.2.1. ASPX Web сервис који користи XML и XQuery

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
//
using System.Data;
using Microsoft.Xml.XQuery;
using System.IO;

namespace WSADIS
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
    line.
    [System.Web.Script.Services.ScriptService]
    public class clsWSFaze : System.Web.Services.WebService
    {

        [WebMethod]
        public int DajBrojFaza(string pomNazivMetode)
        // izdvajanje naziva faza na osnovu naziva metoda
        // spajanjem sadrzaja 3 XML fajla, odnosno pokretanjem XQuery
        {
            XQueryNavigatorCollection col = new XQueryNavigatorCollection();
            col.AddNavigator(Server.MapPath("~/") + "XML/Faze.XML", "faze");
            col.AddNavigator(Server.MapPath("~/") + "XML/Metode.XML", "metode");
            col.AddNavigator(Server.MapPath("~/") + "XML/MetodeFaze.XML", "metfaz");
            string query = "<kolekcija> { for $f in document(\"faze\")//SpisakFaza/Faza, $mf in
            document(\"metfaz\")//SpisakMetodaFaza/MetodaFaza [RBFaze=$f/RedniBroj], $m in
            document(\"metode\")//SpisakMetoda/Metoda [RedniBroj = $mf/RBMetode] + \" where $m/Naziv='\" +
            pomNazivMetode + \"' return \" + <izlaz> + \"{\" + \"$f/Naziv\" + \"}\" + </izlaz> } </kolekcija>";

            //string query = "";

```

```

XQueryExpression xexpr = new XQueryExpression(query);
string result = xexpr.Execute(col).ToXml();
MemoryStream mstream = new MemoryStream(System.Text.ASCIIEncoding.ASCII.GetBytes(result));
//
DataSet dsFaze = new DataSet();
// RADI NA STARI NACIN
//dsFaze.ReadXml(Server.MapPath("~/") + "XML/Faze.XML");

// NOVO
dsFaze.ReadXml(mstream);
return dsFaze.Tables[0].Rows.Count;
}

[WebMethod]
public string DajPrvuFazu(string pomNazivMetode)
// izdvajanje naziva faza na osnovu naziva metoda
// spajanjem sadrzaja 3 XML fajla, odnosno pokretanjem XQuery
{
    XQueryNavigatorCollection col = new XQueryNavigatorCollection();
    col.AddNavigator(Server.MapPath("~/") + "XML/Faze.XML", "faze");
    col.AddNavigator(Server.MapPath("~/") + "XML/Metode.XML", "metode");
    col.AddNavigator(Server.MapPath("~/") + "XML/MetodeFaze.XML", "metfaz");
    string query = "<kolekcija> { for $f in document(\"faze\")//SpisakFaza/Faza, $mf in
document(\"metfaz\")//SpisakMetodaFaza/MetodaFaza [RBFaze=$f/RedniBroj], $m in
document(\"metode\")//SpisakMetoda/Metoda [RedniBroj = $mf/RBMetode] + \" where $m/Naziv='\" +
pomNazivMetode + \"' return \" + <izlaz>\" + \"{\" + \"$f/Naziv\" + \"}\" + </izlaz> } </kolekcija>";

    //string query = "";
    XQueryExpression xexpr = new XQueryExpression(query);
    string result = xexpr.Execute(col).ToXml();
    MemoryStream mstream = new MemoryStream(System.Text.ASCIIEncoding.ASCII.GetBytes(result));
    //
    DataSet dsFaze = new DataSet();
    // RADI NA STARI NACIN
    //dsFaze.ReadXml(Server.MapPath("~/") + "XML/Faze.XML");

    // NOVO
    dsFaze.ReadXml(mstream);
    return dsFaze.Tables[0].Rows[0].ItemArray[0].ToString ();
}

[WebMethod]
public string DajNtuFazu(string pomNazivMetode, int rbFaze)
// - krece od 1, pa zato ide -1
// izdvajanje naziva faza na osnovu naziva metoda
// spajanjem sadrzaja 3 XML fajla, odnosno pokretanjem XQuery
{
    XQueryNavigatorCollection col = new XQueryNavigatorCollection();
    col.AddNavigator(Server.MapPath("~/") + "XML/Faze.XML", "faze");
    col.AddNavigator(Server.MapPath("~/") + "XML/Metode.XML", "metode");
    col.AddNavigator(Server.MapPath("~/") + "XML/MetodeFaze.XML", "metfaz");
    string query = "<kolekcija> { for $f in document(\"faze\")//SpisakFaza/Faza, $mf in
document(\"metfaz\")//SpisakMetodaFaza/MetodaFaza [RBFaze=$f/RedniBroj], $m in
document(\"metode\")//SpisakMetoda/Metoda [RedniBroj = $mf/RBMetode] + \" where $m/Naziv='\" +
pomNazivMetode + \"' return \" + <izlaz>\" + \"{\" + \"$f/Naziv\" + \"}\" + </izlaz> } </kolekcija>";

    //string query = "";
    XQueryExpression xexpr = new XQueryExpression(query);
    string result = xexpr.Execute(col).ToXml();
    MemoryStream mstream = new MemoryStream(System.Text.ASCIIEncoding.ASCII.GetBytes(result));
    //
    DataSet dsFaze = new DataSet();
    //
    dsFaze.ReadXml(mstream);
    return dsFaze.Tables[0].Rows[rbFaze-1].ItemArray[0].ToString();
}

[WebMethod]
public DataSet DajFaze(string pomNazivMetode)
// izdvajanje naziva faza na osnovu naziva metoda
// spajanjem sadrzaja 3 XML fajla, odnosno pokretanjem XQuery
{
    XQueryNavigatorCollection col = new XQueryNavigatorCollection();
    col.AddNavigator(Server.MapPath("~/") + "XML/Faze.XML", "faze");
    col.AddNavigator(Server.MapPath("~/") + "XML/Metode.XML", "metode");

```

```

col.AddNavigator(Server.MapPath("~/") + "XML/MetodeFaze.XML", "metfaz");
string query = "<kolekcija> { for $f in document(\"faze\")//SpisakFaza/Faza, $mf in
document(\"metfaz\")//SpisakMetodaFaza/MetodaFaza [RBFaze=$f/RedniBroj], $m in
document(\"metode\")//SpisakMetoda/Metoda [RedniBroj = $mf/RBMetode] + \" where $m/Naziv='\" +
pomNazivMetode + \"' return \" + \"<izlaz>\" + \"{\" + \"$f/Naziv\" + \"}\" + \"</izlaz> } </kolekcija>";

//string query = "";
XQueryExpression xepr = new XQueryExpression(query);
string result = xepr.Execute(col).ToXml();
MemoryStream mstream = new MemoryStream(System.Text.ASCIIEncoding.ASCII.GetBytes(result));
//
DataSet dsFaze = new DataSet();
//
dsFaze.ReadXml(mstream);
return dsFaze;
}
}
}

```

### 9.13.2.2. XML који се користи у оквиру web сервиса

#### Faze.xml

```

<?xml version="1.0" standalone="yes"?>
<SpisakFaza>
  <Faza>
    <RedniBroj>1</RedniBroj>
    <Naziv>Requirements</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>2</RedniBroj>
    <Naziv>Architecture</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>3</RedniBroj>
    <Naziv>Modeling</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>4</RedniBroj>
    <Naziv>Implementation</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>5</RedniBroj>
    <Naziv>Testing</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>6</RedniBroj>
    <Naziv>Documenting</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>7</RedniBroj>
    <Naziv>Deployment</Naziv>
  </Faza>
  <Faza>
    <RedniBroj>8</RedniBroj>
    <Naziv>Training</Naziv>
  </Faza>
</SpisakFaza>

```

#### Metode.XML

```

<?xml version="1.0" standalone="yes"?>
<SpisakMetoda>
  <Metoda>
    <RedniBroj>1</RedniBroj>
    <Naziv>Waterfall</Naziv>
  </Metoda>
  <Metoda>
    <RedniBroj>2</RedniBroj>
    <Naziv>Agile</Naziv>
  </Metoda>
</SpisakMetoda>

```

#### MetodeFaze.XML

```

<?xml version="1.0" standalone="yes"?>
<SpisakMetodaFaza>

```



```

<MetodaFaza>
  <RBMetode>1</RBMetode>
  <RBFaze>1</RBFaze>
</MetodaFaza>
<MetodaFaza>
  <RBMetode>1</RBMetode>
  <RBFaze>2</RBFaze>
</MetodaFaza>
<MetodaFaza>
  <RBMetode>1</RBMetode>
  <RBFaze>3</RBFaze>
</MetodaFaza>
<MetodaFaza>
  <RBMetode>2</RBMetode>
  <RBFaze>1</RBFaze>
</MetodaFaza>
</SpisakMetodaFaza>

```

### 9.13.2.3. ASPX Web апликација која користи ASPX Web сервис

```

using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
//
using System.Data;

namespace ADIS
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnUcitajFaze_Click(object sender, EventArgs e)
        {
            DataSet dsFaze = new DataSet ();
            WSfaze.clsWSFaze objFaze = new WSfaze.clsWSFaze();
            dsFaze = objFaze.DajFaze(txtMetoda.Text);
            for (int i = 0; i <= dsFaze.Tables[0].Rows.Count - 1; i++)
            {
                lsbSpisakFaza.Items.Add (dsFaze.Tables[0].Rows[i].ItemArray[0].ToString());
            }
        }
    }
}

```

### 9.13.2.4. PHP апликација која користи ASPX Web сервис, уз динамичку промену структуре корисничког интерфејса у односу на вредности из ASPX Web сервиса

#### **Index.php**

```

<?php
session_start();

// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="sr-RS" xml:lang="sr-RS">
<meta charset="UTF-8">
<head>
<title>IT project</title>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="css/style.css" media="screen">
</head>

```

```

<body>

<table class="no-spacing" style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0"
border="0" style="border-spacing: 0;">

<!-- ZAGLAVLJE ----->
<?php include 'delovi/zaglavljeindex.php';?>

<!-- DONJI DEO ----->
<tr style="padding:0px;">
<td style="width:10%;">
</td>

<!-- SREDINA DONJEG DELA SA SADRZAJEM pocinje ovde ----->
<td align="center" valign="middle">
<table style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0" border="0"
bgcolor="#003366">
<tr>
<td style="width:1%;">
</td>

<td style="width:20%;padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">

<?php include 'delovi/menilevo.php';?>

</td>

<td style="width:2%;">
</td>

<td style="padding:0" cellspacing="0" cellpadding="0" border="0" valign="top">

<!-- GLAVNI SADRZAJ desno ----->
<?php include 'delovi/desnopocetna.php';?>
</td>

<td style="width:2%;">
</td>

</tr>
</table>

</td>
<!-- SADRZAJ zavrsava ovde ----->

<td style="width:10%;">
</td>
</tr>
<!-- DONJI DEO zavrsava ovde ----->

<tr style="padding:0px;">
<td style="width:10%;"></td>
<td align="center" valign="middle"></td>
<td style="width:10%;"></td>
</tr>
<!-- DONJI DEO sa donjom ivicom zavrsava ovde ----->
<!-- footer panel starts here -->
<?php include 'delovi/footer.php';?>

</table>

</body>
</html>

```

### **Desnopocetna.php**

```

<meta charset="UTF-8">
<!-- SADRZAJ STRANICE DESNO pocinje ovde ----->
->


```

```

<table style="width:100%;style="width:100%; padding:0" align="center" cellspacing="0" cellpadding="0"
border="0" bgcolor="#D8E7F4">
<tr>
<td style="width:5%;">
</td>

<td align="left">
<br/>
<font face="Trebuchet MS" color="darkblue" size="4px">
<b>VESTI</br>
<?php

//POVEZIVANJE PHP KODA SA ASP.NET WEB SERVISOM
require_once('lib/nusoap.php');
$wsdl='http://localhost:1433/WSFaze.asmx?wsdl';
$client = new nusoap_client($wsdl, 'WSDL');
$error = $client->getError();
if ($error) {
    die("client construction error: {$error}\n");
}
// OVO RADI ZA PRVU FAZU
//$param = array('pomNazivMetode' => 'Waterfall');
//$answer = $client->call('DajPrvuFazu', array('parameters' => $param), '', '', false, true);

// ----- UKUPAN BROJ FAZA
$param = array('pomNazivMetode' => 'Waterfall');
$answer =array();
$answer = $client->call('DajBrojFaza', array('parameters' => $param), '', '', false, true);
$brojfaza =$answer['DajBrojFazaResult'];
print_r('broj faza:'.$brojfaza);

// ISPIS VISESTRUKIH VREDNOSTI
echo "<ul>";
for ($row = 0; $row < $brojfaza; $row++) {
    echo "<p><b>Row number $row</b></p>";

    // ----- NTA FAZA
    $param = array('pomNazivMetode' => 'Waterfall', 'rbFaze' => $row+1);
    $answer =array();
    $answer = $client->call('DajNtuFazu', array('parameters' => $param), '', '', false, true);
    $nazivfaze =$answer['DajNtuFazuResult'];
    echo "<li>". 'naziv faze:'. $nazivfaze. "</li>";
    echo "<form ACTION=\"unosfazeform.php\" METHOD=\"POST\">";
    echo "<input type=\"hidden\" name=\"nazivfaze\" value=\"".$nazivfaze.\">";
    echo "<b><font face=\"Trebuchet MS\" color:#3F4534 size=\"2px\"><input TYPE=\"submit\" name=\"unosfaze\"
value=\"".$nazivfaze.\"></font></b>";
    echo "</form>";
}
    echo "</ul>";

$error = $client->getError();
if ($error) {
    print_r($client->response);
    print_r($client->getDebug());
    die();
}

?>
</td>
<td style="width:5%;">
</td>
</tr>
</table>

```

### 9.13.2.5. Адаптибилност слоја података – креирање структуре XML као носиоца података

У наставку је дат пример програмског кода којим се може креирати XML датотека, где би се као параметри могли задати: назив датотеке и називи елемената структуре.

```

using System;
using System.Data;

```

```
using System.Xml;

...

XmlTextWriter writer = new XmlTextWriter("Podaci.xml", System.Text.Encoding.UTF8);
writer.WriteStartDocument(true);
writer.Formatting = Formatting.Indented;
writer.Indentation = 2;
writer.WriteStartElement("Table");
KreirajElement("1", "Naziv1", writer);
writer.WriteEndElement();
writer.WriteEndDocument();
writer.Close();
}
private void KreirajElement(string pID, string pNaziv, XmlTextWriter writer)
{
writer.WriteStartElement("Element");
writer.WriteStartElement("Element_id");
writer.WriteString(pID);
writer.WriteEndElement();
writer.WriteStartElement("Element_naziv");
writer.WriteString(pNaziv);
writer.WriteEndElement();
writer.WriteEndElement();
}
}
}
```

### 9.13.2.6. ASPX web сервис за анализу концептуалnog модела података

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
//
using System.Text;
using System.IO;
using System.Collections;
using System.Xml;

namespace WSCDMmetrics
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
    line.
    // [System.Web.Script.Services.ScriptService]
    public class AnalizaCDMEntiteta: System.Web.Services.WebService
    {

        private System.IO.StreamReader fileModel;
        private String vCeoModel;
        private System.Collections.ArrayList listaEntiteta;
        String vDeoModelaEntiteta;
        int UkupnoEntiteta;

        private string IzdvojDeoModelaEntiteta()
        {
            int pozicijaPocetkaDelaEntiteta;
            int pozicijaKrajaDelaEntiteta;
            //
            string vModel = vCeoModel;
            pozicijaPocetkaDelaEntiteta = vModel.IndexOf("<c:Entities>"); // duzina =12
            pozicijaKrajaDelaEntiteta = vModel.IndexOf("</c:Entities>"); // duzina =13
            // izdvajamo untrasnji deo gde su entiteta, ali bez granicnih tagova entities
            vDeoModelaEntiteta = vModel.Substring(pozicijaPocetkaDelaEntiteta + 12,
            pozicijaKrajaDelaEntiteta - pozicijaPocetkaDelaEntiteta - 13);
            return vDeoModelaEntiteta;
        }
    }
}
```

```

private void IzdvojEntitete()
{
    String vModel;

    String vEntitet;
    String vEntityIDAttributCode;
    String vEntityIDAttributName;

    int pozicijaPocetkaEntiteta;
    int pozicijaKrajaEntiteta;

    int pozicijaPocetkaIDEntiteta;
    int pozicijaKrajaIDEntiteta;

    int duzinaModela;

    listaEntiteta = new ArrayList();

    // -----
    vModel = vCeoModel; // zapravo je samo deo modela sa entitetima
    // -----

    // pocetno citanje kako bi se omogucio while
    pozicijaPocetkaEntiteta = vModel.IndexOf("<o:Entity Id");
    pozicijaKrajaEntiteta = vModel.IndexOf("</o:Entity");
    //
    UkupnoEntiteta = 0;
    while ((pozicijaPocetkaEntiteta > 0) & (pozicijaKrajaEntiteta > 0))
    {
        // IZDVAJANJE JEDNOG ENTITETA
        pozicijaPocetkaEntiteta = vModel.IndexOf("<o:Entity Id");
        pozicijaKrajaEntiteta = vModel.IndexOf("</o:Entity");
        vEntitet = vModel.Substring(pozicijaPocetkaEntiteta, pozicijaKrajaEntiteta -
pozicijaPocetkaEntiteta + 11);
        UkupnoEntiteta++;

        // IZDVAJANJE NAZIVA ENTITETA
        pozicijaPocetkaIDEntiteta = vEntitet.IndexOf("<a:Code");
        pozicijaKrajaIDEntiteta = vEntitet.IndexOf("</a:Code");
        vEntityIDAttributCode = vEntitet.Substring(pozicijaPocetkaIDEntiteta,
pozicijaKrajaIDEntiteta - pozicijaPocetkaIDEntiteta + 9);
        pozicijaPocetkaIDEntiteta = vEntityIDAttributCode.IndexOf(">");
        pozicijaKrajaIDEntiteta = vEntityIDAttributCode.IndexOf("</");
        vEntityIDAttributName = vEntityIDAttributCode.Substring(pozicijaPocetkaIDEntiteta + 1,
pozicijaKrajaIDEntiteta - pozicijaPocetkaIDEntiteta - 1);

        // DODAVANJE NAZIVA U LISTU
        listaEntiteta.Add(vEntityIDAttributName);

        // ----- PRIPREMA ZA SLEDECU ITERACIJU
        duzinaModela = vModel.Length;
        vModel = vModel.Substring(pozicijaKrajaEntiteta + 11, duzinaModela - pozicijaKrajaEntiteta
- 12);

        // - citanje za while uslov - da li imamo i sledeci entitet
        pozicijaPocetkaEntiteta = vModel.IndexOf("<o:Entity Id");
        pozicijaKrajaEntiteta = vModel.IndexOf("</o:Entity");
    }
}

[WebMethod]
public int BrojEntitetaCDM(String CDMmodel)
{
    vCeoModel = CDMmodel;
    vCeoModel = IzdvojDeoModelaEntiteta();
    IzdvojEntitete(); // pripremljeni entiteti u globalnoj ArrayListi listaEntiteta
    return UkupnoEntiteta;
}
}
}

```

### 9.13.2.7. ASPX web сервис који позива и користи други ASPX web сервис

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Data;
using Toolkit.Net; // dodato

namespace WSRules
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
    line.
    // [System.Web.Script.Services.ScriptService]
    public class clsAdaptibilnost : System.Web.Services.WebService
    {
        private string DinamickoPozivanjeWebServisa(string pom)
        {
            List<WebServiceClient.Parameter> lstParameters = new List<WebServiceClient.Parameter>();
            lstParameters.Add(new WebServiceClient.Parameter { Name = "pomNazivMetode", Value = pom});
            WebServiceClient client = new WebServiceClient
            {
                WebMethod = "DajPrvuFazu",
                Url = "http://localhost:1433/WSFaze.asmx",
                WSServiceType = WebServiceClient.ServiceType.Traditional,
                WCFContractName = "",
                Parameters = lstParameters
            };
            string returnFromService = client.InvokeService();
            return returnFromService;
        }

        [WebMethod]
        public string PrimenaAkcije(string pom)
        {
            string rezultat = DinamickoPozivanjeWebServisa(pom);
            return rezultat;
        }
    }
}

```

НАПОМЕНА: У овом програмском коду на URL <http://localhost:1433/WSFaze.asmx> треба да се налази активан web сервис, чија је једна од метода "DajPrvuFazu", а параметар позива те методе је "pomNazivMetode", при чему се вредност „pom“ добија динамички, приликом позива web сервиса WSRules који позива наведени web сервис. У овом програмском коду користе се класе и методе из библиотеке класа Toolkit.Net: WebServiceClient и WebServiceClient.Parameter.

## 9.14. БИОГРАФИЈА КАНДИДАТА

Мр Љубица Кази



Рођена је 10.12.1974. године у Зрењанину. Основне студије на Техничком факултету «Михајло Пупин» у Зрењанину, смер: «Професор информатике», завршила је 1998. године, са просечном оценом 9.60. Магистарске студије на Техничком факултету «Михајло Пупин» у Зрењанину, научна област: «Информатика», завршила је 2005. године са просечном оценом 10.00. Мастер студије на смеру «Информатичко инжењерство» завршила је 2009. године са просечном оценом 10.00. Активан је учесник програма и семинара стручног усавршавања у области информатике. Члан је ACM (Association for Computing Machinery) од 2011. године.

Добитник је више награда на основу постигнутих резултата у току основних студија, међу којима су најзначајније: награда најбољем студенту генерације 1995/96 Техничког факултета «Михајло Пупин» у Зрењанину и награда „Стјепан Хан“ за најбољи дипломски рад из информатичких дисциплина у школској 1997/98 години. Била је стипендиста више фондова за талентоване студенте.

Радни однос на Техничком факултету «Михајло Пупин» у Зрењанину засновала је 1998. године. Радила је на радном месту асистента у звању асистента приправника од 1998. године, а од 2005. године у звању асистента у области Информационих технологија. У школској 2002. години добитник је награде Министарства просвете Републике Србије као један од два најбоља асистента запослених на Техничком факултету «Михајло Пупин» у Зрењанину, према успеху са основних студија. Према мишљењу студената и резултатима студентске годишње анкете, у школској 2013/14. и 2014/15. години најбољи је асистент Катедре за Информационе технологије, а други (у 2013/14. години), односно седми асистент (у 2014/15. години), од укупно 40 асистената на Техничком факултету «Михајло Пупин» у Зрењанину.

Учествовала је у бројним активностима у раду факултета. Била је активан учесник 12 научно-стручних пројеката националног нивоа. Коаутор је 4 рада објављених у часописима са SCI (*Thomson Reuters*) листе (M23), 18 радова у међународним часописима (M53), 58 радова објављених на међународним конференцијама (M33), 25 радова објављених на националним конференцијама (M63). Коаутор је 9 уџбеника, 3 софтвера (M85), као и 19 софтвера реализованих у научно-стручном раду.

Почев од 2014. године, истовремено са наставним и научним радним ангажовањем на пословима асистента, ангажована је и на пословима администратора апликативног софтвера информационог система Техничког факултета «Михајло Пупин» у Зрењанину, односно пословима координатора интеграције информационог система Факултета са информационом системом Универзитета у Новом Саду. У школској 2014/15. години руководилац је пројекта „Професионално усавршавање студената у оквиру развоја информационог система Факултета“ који омогућава унапређење стручног знања студената, као и унапређење информационог система Техничког факултета «Михајло Пупин» у Зрењанину.