

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko

Doktorska disertacija

**INTEGRIRAN MODEL OCENJEVANJA
USPEŠNOSTI PROGRAMSKIH OGRODIJ**

December 2008

Gregor Polančič



Fakulteta za elektrotehniko,
računalništvo in informatiko

Inštitut za informatiko

Smetanova ulica 17
2000 Maribor

Doktorska disertacija

Integriran model ocenjevanja uspešnosti programskih ogrodij

Avtor: Gregor Polančič

Mentorica: doc. dr. Romana Vajde Horvat

Somentor: red. prof. dr. Ivan Rozman

December 2008



UNIVERZA V MARIBORU



FAKULTETA ZA ELEKTROTEHNIKO,
RAČUNALNIŠTVO IN INFORMATIKO
2000 Maribor, Smetanova ulica 17

Številka: 1/64-podipl.-2006

Datum: 02. 11. 2006

TEMA ZA DOKTORSKO DISERTACIJO

Senat Fakultete za elektrotehniko, računalništvo in informatiko je na svoji 39. seji dne 26. oktobra 2006 v skladu s statutom Univerze v Mariboru

sprejel
temo za doktorsko disertacijo

z naslovom:

»INTEGRIRANI MODEL OCENJEVANJA USPEŠNOSTI PROGRAMSKIH OGRODIJ«,

ki jo je prijavil

Gregor POLANČIČ, univ. dipl. inž. rač. in inf.

Mentorica: doc. dr. Romana Vajde Horvat

Somentor: red. prof. dr. Ivan Rozman

Rok za oddajo disertacije v 8 izvodih je 4 leta (do 02. 11. 2010).

PREJMEJO:

- dr. Vajde Horvat
- dr. Rozman
- G. Polančič
- Univerza v Mariboru
- odloga



Ključne besede: programska ogrodja, ponovna uporaba, empirično programsko inženirstvo, TAM, učinkovitost IS, terenska raziskava, anketa, modeli strukturnih enačb

UDK: 004.43:659(043.3)

Povzetek:

Osrednji predmet disertacije so programska ogrodja, ki ob ustrezni uporabi spadajo med najbolj učinkovite tehnike ponovne uporabe programske opreme. Predhodne raziskave so ugotovile, da na uspešnost uporabe ogrodja bistveno vpliva obseg uporabe ogrodja oziroma število instanciranj ogrodja. Zato smo si za osnovni cilj disertacije zastavili preučitev dejavnikov, ki povečujejo obseg uporabe ogrodij in s tem njihovo uspešnost. Pri tem smo se osredotočili na uporabniški vidik dojemanj ogrodij, ki vpliva na sprejetje oziroma zavrnitev ogrodij. Raziskovalnega problema smo se lotili s terensko raziskavo, ki je identificirala faktorje, ki vplivajo na uporabnikove odločitve o izbiri ogrodja. Na osnovi rezultatov terenske raziskave smo definirali teoretične modele disertacije, ki smo jih zasnovali na najbolj splošnem in validiranem modelu sprejetosti informacijskih tehnologij - modelu TAM (*Technology Acceptance Model*). Veljavnost predpostavljenih teoretičnih modelov smo preizkusili z izvedbo ankete in z uporabo specialne statistične tehnike – modelov strukturnih enačb. Rezultate raziskav predstavljata izvirna modela sprejetosti in uspešnosti programskih ogrodij. Poglavitne ugotovitve, ki izhajajo iz analize teoretičnih modelov, so, da na sprejetost ogrodij vplivajo različne karakteristike ogrodij in posebnosti uporabnikov. V nasprotju z uporabniki končne programske opreme sta poglavitni determinanti sprejetosti ogrodij dojeta uporabnost ogrodja in zaupanje v ogrodje, kar sovпада z zahtevami po dolgoročni uporabi ogrodja. Pri interpretaciji rezultatov je vsekakor treba upoštevati omejitve metod raziskovanja, teoretičnih modelov, tehnik analize podatkov in vzorce, ki smo jih uporabljali v raziskavah disertacije.

Rezultati raziskav disertacije so lahko osnova za številne izvirne rešitve, ki lahko pozitivno vplivajo na razvoj in uporabo ogrodij. V okviru disertacije smo predstavili model za ocenjevanje sprejetosti ogrodij, ki omogoča napovedovanje sprejetosti ogrodij z uporabo linearne regresije. Druga aplikativna rešitev disertacije je metoda za izboljševanje sprejetosti ogrodij, ki je namenjena razvijalcem ogrodij in omogoča iterativno izboljševanje ogrodij. Izboljšave ogrodja temeljijo na postopnih modifikacijah abstraktnih sestavnih delov ogrodja. Tretja aplikativna rešitev je namenjena uporabnikom javno dostopnih ogrodij in omogoča sistematično vrednotenje in izbiro najprimernejšega ogrodja na osnovi ocenitve kakovosti, primernosti in finančnega modela alternativ.

Keywords: software frameworks, software reuse, empirical software engineering, TAM, IS success, field research, survey, structural equation models

Title: Integrated model for assessing the success of software frameworks

Abstract:

The focal research object of the thesis is software framework, which represents one of the most successful software reuse techniques, if used under proper conditions. Researchers already demonstrated that the success of using frameworks increases with the amount of framework use respectively number of framework instantiations. So, the main objective of the thesis was to investigate factors, which impact the amount of framework use and consequently the success of framework use. We oriented the thesis to users' perceptions which have an impact on the adoption or rejection of frameworks. The objective of the first part of the thesis was to identify major factors, which impact users' decisions to use frameworks. Using the results of previously performed field research, we defined theoretical models, which were based on the most popular and well proved acceptance model – TAM (*Technology Acceptance Model*). We used a survey method to test the validity of the hypothesized theoretical models, where the results were statistically analyzed using structural equation modelling techniques. The results support hypothesized theoretical models, which extended TAM to the context of frameworks. The main findings are following. We found that the acceptance of frameworks is influenced by two groups of factors: framework characteristics and individual differences. The direct determinants of framework acceptance are confidence and perceived usefulness. This represents a slight deviation to the end user acceptance models. Our explanation to this deviation is that framework users are aware of the required long-term framework use which is reflected in the confidence factor. Several research limits should be considered when interpreting the results of the thesis, including research methods limits, statistical techniques limits, models limits and limits of samples used in the thesis.

Based on the original scientific outcomes of the thesis, several practical applications for framework developers and users are presented and foreseen. First, we present a model, which is capable to predict the acceptance of frameworks based on acceptance determinants and linear regression. Second we present a method for framework acceptance improvement. The main idea behind the proposed method lies in an user-oriented, iterative and evolutionary process, which improves frameworks stepwise, on the level of their common constituent elements. The last applicative outcome of the thesis is a framework selection method. The method is designed for application developers which use public available frameworks. It evaluates framework alternatives from three viewpoints: framework quality, framework suitability and financial view.

Zahvale

*Številni dejavniki so pripomogli k temu, da sem napisal to disertacijo.
Izpostaviti želim ljudi, ki so bistveno pripomogli k temu.*

*V prvi vrsti se želim zahvaliti Sabini in Filipu.
Zahvala Sabini za veliko razumevanja in podpore.
Zahvala Filipu, mojemu največjemu sončku, viru energije.*

*Zahvala mojim staršem. Odkar pomnim, so me podpirali in usmerjali v mojih interesih -
področjih znanosti in tehnologije.*

Zahvala mentorici, dr. Romani Vajde Horvat, za podporo, vložen čas in energijo.

Zahvala somentorju, red. prof. dr. Ivan Rozmanu, za somentorstvo.

KAZALO

1	Uvod.....	1
1.1	Predstavitev problematike	3
1.2	Metodološki vidik disertacije.....	5
1.3	Pregled predlagane rešitve.....	7
1.3.1	Teza doktorske disertacije	9
1.3.2	Pričakovani izvirni in aplikativni prispevki disertacije	10
1.4	Struktura disertacije.....	11
2	Predstavitev raziskovalnega področja.....	12
2.1	Ponovna uporaba v programskem inženirstvu.....	13
2.2	Programska ogrodja	15
2.2.1	Zgodovina ogrodij	17
2.2.2	Prednosti in slabosti ogrodij	18
2.2.3	Ogrodja v procesu razvoja programske opreme.....	20
2.2.4	Vrste in klasifikacije ogrodij.....	22
2.2.5	Ogrodjem sorodni koncepti in tehnologije.....	24
2.3	Učinkovitost informacijskih tehnologij	29
2.3.1	D&M model uspešnosti IS.....	30
2.3.2	Seddonov model uspešnosti IS.....	32
2.4	Sprejetost informacijskih tehnologij.....	34
2.4.1	Model sprejetosti IT - TAM.....	35
2.4.2	Model ujemanja zahtev in lastnosti IT - TTF.....	39
2.5	Kakovost in ponovna uporabnost programske opreme.....	40
2.5.1	Standard ISO/IEC 9126	42
2.5.2	Model REBOOT.....	46
2.6	Povezava med karakteristikami, sprejetostjo in učinkovitostjo programske opreme... 48	
2.7	Povzetek predstavitve raziskovalnega področja	50
3	Raziskave	52

3.1	Zasnova raziskav.....	53
3.1.1	Analiza raziskovalnega problema	53
3.1.2	Izbira metod raziskovanja.....	54
3.1.3	Definiranje procesa izvedbe raziskav.....	60
3.2	Pregled literature: definiranje uspešnosti programskih ogrodij.....	61
3.3	Pregled literature: determinante uspešnosti ogrodij.....	63
3.4	Terenska raziskava: identifikacija determinant sprejetosti ogrodij.....	65
3.4.1	Načrtovanje fokusne skupine.....	65
3.4.2	Analiza rezultatov fokusne skupine	69
3.4.3	Interpretacija rezultatov fokusne skupine	83
3.4.4	Omejitve raziskave	84
3.5	Pregled literature: identifikacija determinant sprejetosti ogrodij.....	86
3.5.1	Proces in rezultati pregleda literature	86
3.5.2	Rezultati postopnega približevanja.....	88
3.5.3	Rezultati ponazorilne metode.....	90
3.5.4	Omejitve raziskave	91
3.6	Primerjava rezultatov pregleda literature in terenske raziskave	92
3.7	Anketa: empirična validacija determinant sprejetosti in uspešnosti ogrodij.....	96
3.7.1	Opredelevanje ankete	96
3.7.2	Faktorji in hipoteze RV_1	98
3.7.3	Faktorji in hipoteze RV_2	103
3.7.4	Faktorji in hipoteze RV_3	108
3.7.5	Teoretični modeli.....	109
3.7.6	Vprašalnik.....	113
3.7.7	Vzorčenje.....	117
3.7.8	Vzpostavitev in izvedba ankete.....	118
3.7.9	Metoda analize podatkov.....	119
3.7.10	Rezultati.....	125
3.7.11	Analiza veljavnosti rezultatov.....	153

3.7.12	Interpretacija rezultatov	154
3.8	Povzetek raziskav	160
3.8.1	Odgovori na raziskovalno vprašanje RV ₁	160
3.8.2	Odgovori na raziskovalno vprašanje RV ₂	160
3.8.3	Odgovori na raziskovalno vprašanje RV ₃	161
3.8.4	Odgovor na zastavljeno tezo disertacije.....	161
4	Aplikacije raziskav	162
4.1	Model za ocenjevanje sprejetosti programskih ogrodij.....	162
4.2	Osnove metode za ocenjevanje in izboljšanje sprejetosti programskih ogrodij	165
4.2.1	Identifikacija sestavnih delov ogrodja.....	165
4.2.2	Integriran konceptualni model ogrodij.....	170
4.2.3	Proces izboljševanja sprejetosti ogrodij.....	179
4.2.4	Povzetek in nadaljnji koraki razvoja metode	181
4.3	Metoda za izbiro ogrodja.....	182
4.3.1	Opredelevitev problema	182
4.3.2	Predlagana rešitev	182
4.3.3	Proces izbire ogrodja.....	184
4.3.4	Nadaljnji koraki razvoja metode.....	186
4.4	Povzetek aplikacij raziskav.....	187
5	Zaključki	189
6	Priloge	192
	Priloga A: Vprašalnika	193
	Vprašalnik fokusne skupine.....	193
	Vprašalnik ankete.....	194
	Priloga B: Podatkovna baza Sourceforge.....	200
	Priloga C: Predstavitev pomembnejših postopkov statistične analize podatkov.....	204
	Povprečje, varianca in standardni odklon.....	204
	Korelacija in kovarianca	205
	Regresijska analiza	207

Faktorska analiza.....	209
Modeli strukturnih enačb	210
Metrike prileganja.....	212
Priloga D: Rezultati statistične analize podatkov	216
Seznam vrednotenih ogrodij.....	216
Primerjava zgodnjih in poznih respondentov	223
Literatura.....	226
Stvarno kazalo	235
Življenjepis kandidata	237

Kazalo slik

Slika 1: Ogrodja in njihova povezava z drugimi tehnikami ponovne uporabe (Sangdon et al. 1999).....	2
Slika 2: Poglavitni faktorji, ki vplivajo na uspešnost ponovne uporabe (Morisio, Ezran, & Tully 2002).....	6
Slika 3: Osnutek kavzalnega modela sprejetosti ogrodiv.....	8
Slika 4: Osnutek kavzalnega modela uspešnosti uporabe ogrodiv.....	8
Slika 5: Taksonomija raziskovalnih področij v disciplinah računalništva (Glass, Ramesh, & Vessey 2004).....	12
Slika 6: Proces razvoja na osnovi ponovne uporabe (Sindre, Conradi, & Karlsson 1995).....	14
Slika 7: Umestitev ogrodiv v REBOOT klasifikacijo ponovne uporabe.....	15
Slika 8: Različni procesi razvoja ogrodja, prikazani v modelu BPMN.....	20
Slika 9: Evolucija ogrodiv, temelječa na vzorcih zasnove (Roberts & Johnson 1997).....	21
Slika 10: Vrste ogrodiv glede na razširljivost.....	23
Slika 11: Celovita klasifikacija ogrodiv (Krajnc & Heričko 2004).....	23
Slika 12: Povezava ogrodiv s sorodnimi koncepti in tehnologijami.....	24
Slika 13: Povezava med komponentami in ogrodivem.....	25
Slika 14: Uporaba knjižnice razredov.....	25
Slika 15: Razlika v uporabi ogrodja in knjižnic razredov.....	26
Slika 16: Osnovne aktivnosti produktivnih linij (Northrop 1999).....	27
Slika 17: D&M model uspešnosti IS (DeLone & Mclean 1992).....	30
Slika 18: Seddonov model uspešnosti IS (Seddon 1997).....	32
Slika 19: Osnovna filozofija modelov sprejetosti (Venkatesh, Morris, Davis, & Davis 2003).....	34
Slika 20: Konceptualni model Teorije utemeljene akcije (Fishbein & Ajzen 1975).....	35
Slika 21: Izviren model TAM (Davis 1989).....	35
Slika 22: Testiranja izvirnega modela TAM.....	36
Slika 23: Kategorije razširitev modela TAM (King & He 2006).....	38
Slika 24: Model TTF (Goodhue & Thompson 1995).....	39
Slika 25: Model FCM (ISO 9126 2001).....	41
Slika 26: Model notranje in zunanje kakovosti v ISO/IEC 9126-1.....	43
Slika 27: Model »kakovosti v uporabi« v ISO/IEC 9126-1.....	43
Slika 28: Povezava med različnimi vidiki kakovosti v ISO/IEC 9126-x.....	44
Slika 29: REBOOT model ponovne uporabnosti (Sindre, Conradi, & Karlsson 1995).....	46
Slika 30: REBOOT model kakovosti (Jung, Kim, & Chung 2004).....	47
Slika 31: Povezanost karakteristik, sprejetosti in učinkovitosti programske opreme.....	48
Slika 32: Vrste raziskovalnih vprašanj.....	54

Slika 33: Možnosti izvedbe fokusnih skupin.....	57
Slika 34: Proces izvedbe raziskav.....	60
Slika 35: Skupni stroški »običajnega razvoja« in razvoja na osnovi produktnih linij (Bockle, Clements, McGregor, Muthig, & Schmid 2004).....	64
Slika 36: Determinante uspešnosti ogrodij.....	64
Slika 37: Opisna statistika neobdelanih podatkov	69
Slika 38: Grafična ponazoritev postopnega približevanja (Neuman 2005).....	70
Slika 39: Grafična predstavitev ponazorilne metode (Meier 1998).....	71
Slika 40: Proces analize podatkov fokusne skupine.....	71
Slika 41: Zaslonska slika orodja Ethnograph s pojasnili.....	72
Slika 42: Postopek redukcije podatkov in odprtega kodiranja.....	72
Slika 43: Delež enakih kod v odvisnosti od njihove frekvence.....	76
Slika 44: Proces izvedbe postopnega približevanja.....	77
Slika 45: Taksonomija konceptov v drevesni obliki	79
Slika 46: Analiza klasifikacije kod glede na hierarhično strukturo konceptov	81
Slika 47: Analiza klasifikacije kod glede na apriorno in posteriorno definiranje konceptov.....	81
Slika 48: Taksonomija konceptov v drevesni obliki	89
Slika 49: Analiza klasifikacije kod glede na hierarhično strukturo konceptov	90
Slika 50: Deleži relevantnosti konceptov v določeni skupini	93
Slika 51: Povezava med karakteristikami ogrodij, kakovosti in ponovne uporabnosti	98
Slika 52: Grafična ponazoritev koncepta »ujemanje zahtev in ogrodja«	101
Slika 53: Grafična ponazoritev koncepta »tehnološki razkorak«.....	102
Slika 54: Teoretični model 1.....	109
Slika 55: Teoretični model 2A.....	110
Slika 56: Teoretični model 2B.....	110
Slika 57: Teoretični model 2C.....	111
Slika 58: Teoretični model 2D.....	111
Slika 59: Teoretični model 3.....	112
Slika 60: Grafična ponazoritev postopka vzorčenja	117
Slika 61: Odločitveno drevo za izbiro ustrezne statistične tehnike	121
Slika 62: Ilustracija delovanja SEM.....	122
Slika 63: Merljivi modeli in strukturni model v modelu SEM.....	123
Slika 64: Merljiv model, izpeljan iz teoretičnega modela 1	132
Slika 65: Rezultati začetnega strukturnega modela	134
Slika 66: Rezultati končnega strukturnega modela, izpeljanega iz teoretičnega modela 1	136
Slika 67: Rezultati analize teoretičnega modela 2A (vsi uporabniki)	137

Slika 68: Rezultati analize teoretičnega modela 2B (vsi uporabniki)	138
Slika 69: Rezultati analize teoretičnega modela 2C (vsi uporabniki).....	141
Slika 70: Rezultati analize teoretičnega modela 2D (vsi uporabniki)	142
Slika 71: Model sprejetosti ogrodij – rezultati strukturnega modela.....	146
Slika 72: Merljiv model teoretičnega modela 3	148
Slika 73: Rezultati analize teoretičnega modela 3 (vsi uporabniki)	151
Slika 74: Model sprejetosti ogrodij.....	158
Slika 75: Model uspešnosti ogrodij.....	159
Slika 76: Mreža determinant ogrodij	163
Slika 77: Povezava med sestavnimi deli ogrodja	165
Slika 78: Najpogostejši elementi na nivoju implementacije ogrodij.....	166
Slika 79: Implementacija vročih točk v ogrodju bele in črne škatle (Parsons et al. 2006).....	166
Slika 80: Koncept dinamičnega povezovanja, ki se uporablja v ogrodjih (Cunningham, Liu, & Zhang 2006)	167
Slika 81: Integriran konceptualni model ogrodij.....	170
Slika 82: Model FCM faktorja razumljivosti	172
Slika 83: Ponazoritev merjenja kompleksnosti razreda	173
Slika 84: Ponazoritev zunanje kompleksnosti razreda	174
Slika 85: Ponazoritev metrike zunanje kompleksnosti skupine razredov.....	175
Slika 86: Model FCM za faktor prenosljivosti.....	176
Slika 87: Model FCM faktorja učinkovitost.....	177
Slika 88: Model FCM faktorja zaupanje.....	177
Slika 89: Proces izboljševanja sprejetosti ogrodij.....	179
Slika 90: Podproces izboljšave determinant sprejetosti ogrodja.....	180
Slika 91: Ilustracija metode OTSO (Kontio et al. 1995).....	182
Slika 92: Proces izbire ogrodja	184
Slika 93: Podproces analize rezultatov vrednotenja alternativ	185
Slika 94: Predloga e-poštnega vabila za sodelovanje v anketi	194
Slika 95: Prošnja za dostop do baze Sourceforge 1/3.....	200
Slika 96: Prošnja za dostop do baze Sourceforge 2/3.....	201
Slika 97: Prošnja za dostop do baze Sourceforge 3/3.....	202
Slika 98: ER-model baze Sourceforge	203
Slika 99: Ilustracija koreacijske zveze med spremenljivkama X in Y	205
Slika 100: Ilustracija linearne regresije	207
Slika 101: Grafična ponazoritev multiple regresije	208
Slika 102: Grafična ponazoritev analize poti	208

Slika 103: Simboli, ki se uporabljajo v modelih strukturnih enačb.....	210
Slika 104: Prikaz sintakse strukturnega modela (Gefen, Straub, & Bordeau 2000)	211

Kazalo tabel

Tabela 1: Pregled metrik uspešnosti IS (DeLone & Mclean 1992)	30
Tabela 2: Izbira metod raziskovanja.....	55
Tabela 3: Prednosti in slabosti fokusnih skupin (Johnston & Shanks 2003;Neuman 2005).....	56
Tabela 4: Primerjava fizične in spletne izvedbe fokusne skupine (Chase & Alvarez 2000)	57
Tabela 5: Primerjava empiričnih metod raziskovanja (Freimut, Punter, Biffi, & Ciolkowski 2002)	58
Tabela 6: Kandidatne spletne skupnosti za izvedbo raziskave.....	66
Tabela 7: Seznam identificiranih kod in pripadajočih konceptov	73
Tabela 8: Seznam identificiranih konceptov, urejen po relevantnosti	78
Tabela 9: Opisna statistika relevantnosti konceptov	80
Tabela 10: Relevantnosti faktorjev po izvedbi združevanja konceptov.....	82
Tabela 11: Rezultati pregleda literature	86
Tabela 12: Seznam identificiranih kod, mnogoternosti in pripadajoči koncepti.....	88
Tabela 13: Opisna statistika relevantnosti konceptov	90
Tabela 14: Primerjava terenske raziskave in pregleda literature z uporabo metode GQM.....	92
Tabela 15: Številčna primerjava identificiranih konceptov in njihovih relevantnosti.....	93
Tabela 16: Rezultati Mann-Whitneyjevega U-testa	94
Tabela 17: Primerjava razvrstitve konceptov glede na relevantnost	94
Tabela 18: Predhodne raziskave, ki temeljijo na TAM	103
Tabela 19: Pregled raziskav, ki so uporabile model TAM v okolju izkušenih uporabnikov	107
Tabela 20: Likertova lestvica s sedmimi stopnjami.....	113
Tabela 21: Vprašanja operacionaliziranega teoretičnega modela.....	114
Tabela 22: Statusi prejetih vprašalnikov.....	119
Tabela 23: Odvisnost statistične tehnike od podatkov	119
Tabela 24: Opisna statistika respondenta - frekvence	125
Tabela 25: Opisna statistika - povprečja in odkloni.....	126
Tabela 26: Seznam desetih najpogosteje vrednotenih ogrodij.....	127
Tabela 27: Opisna statistika izkušenj respondenta z vrednotenim ogrodjem	128
Tabela 28: Metrike prileganja začetnega in končnega merljivega modela	129
Tabela 29: Opisna statistika indikatorjev končnega merljivega modela	130
Tabela 30: Zanesljivost konstruktov.....	131

Tabela 31: Diskriminacijska veljavnost.....	131
Tabela 32: Metrike prileganja začetnega in končnega strukturnega modela	133
Tabela 33: Vpliv tipa uporabe ogrodja na začetni strukturni model.....	134
Tabela 34 Vpliv tipa uporabe ogrodja na končni strukturni model.....	135
Tabela 35: Primerjava rezultatov modela 2A s sorodno raziskavo	138
Tabela 36: Vpliv tipa uporabe ogrodja na teoretična modela 2A in 2B.....	139
Tabela 37: Prileganje strukturnega modela 2A in 2B empiričnim podatkom	139
Tabela 38: Primerjava rezultatov modela 2A in modela 2C.....	143
Tabela 39: Primerjava metrik prileganja za model 2A in model 2C	143
Tabela 40: Primerjava rezultatov modela 2B in modela 2D	144
Tabela 41: Primerjava metrik prileganja za model 2B in model 2D.....	144
Tabela 42: Poglavitni rezultati modela sprejetosti ogrodiij	147
Tabela 43: Opisna statistika indikatorjev teoretičnega modela 3	149
Tabela 44: Zanesljivost konstruktov.....	149
Tabela 45: Diskriminacijska veljavnost teoretičnega modela 3	150
Tabela 46: Vpliv tipa uporabe ogrodja na teoretični model 3	152
Tabela 47: Rezultati hipotez.....	154
Tabela 48: Pregled analize hipotez po struturnih modelih	155
Tabela 49: Vprašalnik za ocenjevanje sprejetosti ogrodiij.....	162
Tabela 50: Vzorci načrtovanja, ki se pogosto uporabljajo v ogrodiij	168
Tabela 51: Smernice ogrodiij, ki so jih predlagali različni avtorji.....	168
Tabela 52: Matrika sosednosti A_1	171
Tabela 53: Pomen vrednosti CC	173
Tabela 54: Vpliv vzorcev načrtovanja na karakteristike programske opreme (Guéhéneuc, Guyomarc'h, Khosravi, & Sahraoui 2006).....	178
Tabela 55: Označevanje strukturnih modelov.....	211
Tabela 56: Seznam vrednotenih ogrodiij.....	216
Tabela 57: Rezultati T-testa, ki je bil izveden na neodvisnih vzorcih	223

1 UVOD

Programsko inženirstvo (*software engineering*) predstavlja sistematičen, discipliniran in merljiv pristop k razvoju, izvajanju in vzdrževanju programske opreme (IEEE 1990). Programsko inženirstvo velja za mlado¹ in spreminjajočo se vedo, ki je primorana

"Nečesa sem se naučil v dolgem življenju: v primerjavi s svetom je vsa naša znanost primitivna in otročja, a je vseeno najdragocenejše, kar imamo."

Albert Einstein

slediti vse bolj kompleksnim zahtevam programske opreme. Zato se nenehno pojavljajo nove metode in tehnike, ki so namenjene učinkovitemu in kakovostnemu razvoju programske opreme. Z vidika produktivnosti uvršča Boehm (1999) omenjene metode in tehnike v tri osnovne strategije izboljšanja produktivnosti razvoja programske opreme:

- Strategija hitrejšega razvoja (*working faster*) se najpogosteje realizira z uporabo orodij, ki omogočajo avtomatizacijo delovno intenzivnih opravil.
- Strategija pametnejšega razvoja (*working smarter*) se najpogosteje realizira z izboljšanjem procesov razvoja, ki minimizirajo in optimizirajo število opravil.
- Strategija izogibanja razvoju (*work avoidance*) se najpogosteje realizira s ponovno uporabo že razvitih programskih izdelkov².

Empirične raziskave, ki jih je izvedel Boehm (1993), so pokazale, da ima izmed predstavljenih strategij največji potencial³ strategija izogibanja razvoju, ki se realizira s ponovno uporabo.

Ponovna uporaba v programskem inženirstvu je definirana kot primer dejanj, v katerih se enak programski izdelek uporabi v različnih kontekstih. Izmed številnih vrst ponovne uporabe (Leach 1996; Sindre, Conradi, & Karlsson 1995) spadajo produktne linije (*product line software*) med najučinkovitejše in po Morisiu et al. (2002) predstavljajo enega izmed kritičnih faktorjev uspeha ponovne uporabe. Filozofija produktnih linij temelji na razvoju množice izdelkov, ki so zgrajeni na enotni programski osnovi (*core asset base*). Pokazalo se je, da takšen pristop zagotavlja

¹ Termin programsko inženirstvo (*software engineering*) se je pojavil leta 1968 na konferenci »NATO Software engineering«.

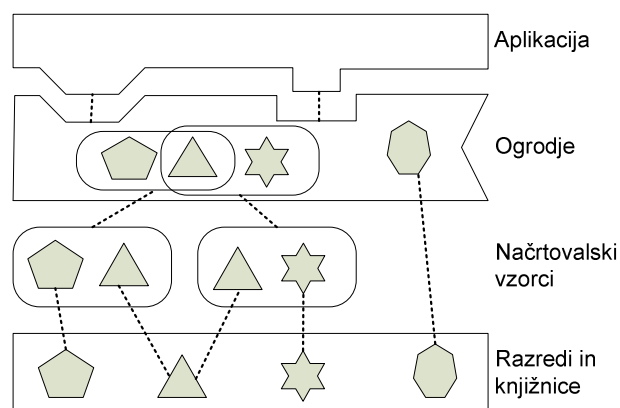
² Z besednima zvezama »programski izdelek« ali »programski artefakt« označujemo vsak izdelek, ki nastane v procesu razvoja programske opreme (Arevalo & Mens 2002; Fayad & Hamu 2000; Mathias, de Oliveira, & de Lucena 2004).

³ Prihranki pri razvoju so se v raziskavi, ki jo je izvedel Boehm, vrednotili relativno, glede na (v okviru raziskave) definiran »običajen« razvoj. Pri tem so bili empirični rezultati naslednji: (1) prihranki s strategijo pohitritve razvoja 8%; (2) prihranki s strategijo pametnejšega razvoja 17%; (3) prihranki s strategijo izogibanja razvoju 47%.

dolgoročne ekonomske prednosti glede na razvoj posameznih izdelkov, kar je razvidno iz ekonomskega modela produktnih linij (Bockle et al. 2004).

Produktne linije se najpogosteje implementirajo z razvojem na osnovi programskih ogrodij (Batory, Cardone, & Smaragdakis 2000; Cunningham, Liu, & Zhang 2006). Ta omogočijo produktnim linijam enotno osnovo s tem, da zagotavljajo generične rešitve za množico podobnih problemov v domeni produktne linije (Sheikh I. Ahmeh, Alex Pezewski, & Al Pezewski 2004).

Programska ogrodja (v nadaljevanju ogrodja) so nepopolni sistemi, ki vsebujejo gradnike, enotne vsem aplikacijam v produktni liniji, in gradnike, ki jih je možno prilagajati in predstavljajo edinstvene dele posameznih aplikacij v produktni liniji (Srinivasan 1999). Ogradja se razlikujejo od preostalih vrst ponovne uporabe v programskem inženirstvu (programske komponente, knjižnice, načrtovalski vzorci), saj težijo k ponovni uporabi večjih delov programskega koda in zasnove na višjem nivoju (Slika 1) (Morisio, Romano, & Stamelos 2002). V nasprotju z drugimi tehnikami ponovne uporabe programskega koda definirajo ogrodja tok izvajanja in zato delujejo kot osnova na njih temelječih aplikacij. Ogradja spadajo med učinkovitejše tehnike ponovne uporabe, saj poleg ponovne uporabe programskega koda enkapsulirajo še znanje načrtovanja (Oliveira et al. 2004).



SLIKA 1: OGRODJA IN NJIHOVA POVEZAVA Z DRUGIMI TEHNIKAMI PONOVNE UPORABE
(SANGDON ET AL. 1999)

Ogradja imajo osrednje mesto v programskem inženirstvu (Manolescu, Noble, & Voelter 2006) predvsem na področju razvoja produktnih linij in družin programske opreme (Batory, Cardone, & Smaragdakis 2000; Cunningham, Liu, & Zhang 2006). Ogradja združujejo znanje določene programske domene, zato predstavljajo specializacijo generičnih programskih jezikov za določeno področje. Razvijalcem programske opreme zagotavljajo koristi na številnih področjih razvoja programske opreme, in sicer: (1) na področju skupnih lastnosti v domeni njihovega delovanja (domenska ogradja), (2) na področju tehnik, ki jih vključujejo v razvoj programske opreme (orodna ogradja) in (3) na področju vrste aplikacij, ki jih izdelujejo (aplikacijska

ogrodja). Posledično je njihova uporaba močno zaželena (Sparks, Benner, & Faris 1996). Število ogrodij se je v zadnjih letih zelo povečalo, kar nakazuje tudi podatek, da je na spletnem repozitoriju odprtokodnih projektov »Sourceforge.net« več kot 4000 projektov umeščenih med ogrodja⁴. Povzeto po Fontouri (1999) »*obstajajo projekcije, da bodo ogrodja postala jedro tehnologij programskega inženirstva prihodnosti*« .

1.1 PREDSTAVITEV PROBLEMATIKE

Neodvisno od številnosti ogrodij, tehnološke zrelosti in zaželene vključenosti ogrodij v projekte razvoja programske opreme raziskovalci in razvijalci programske opreme že dalj časa ugotavljajo, da obstajajo številne težave pri razvoju, uporabi in vzdrževanju ogrodij (Bosch et al. 1997; Srinivasan 1999; van Gorp & Bosch 2001). Težave z ogrodji se v najširšem pogledu kažejo v neuspešnih ogrodjih, neuspešni uporabi ogrodij in izogibanju uporabi ogrodij.

- **Neuspešna ogrodja.** Med javno dostopnimi ogrodji je možno zaslediti veliko ogrodij, ki kljub drugačnim napovedim niso postala uspešna ali pa je razvoj obstal na mrtvi točki. Razlog je za neuspeh ogrodij pogosto v tem, da razvoj, uporaba in vzdrževanje uspešnega ogrodja zahtevajo veliko znanja in virov. Z vprašanjem, kako razviti uspešno ogrodje, se zato ukvarjajo številni raziskovalci in razvijalci (Todd 1997).
- **Neuspešna oziroma neučinkovita uporaba ogrodij.** Uporaba ogrodij velja za zapleten in drag proces z vidikov porabljenega časa, človeških in drugih virov (Oliveira, Alencar, Filho, de Lucena, & Cowan 2004). Uporabniki ogrodij pogosto pričakujejo takojšnje prednosti, ki izhajajo iz koncepta ponovne uporabe, med katerimi je najbolj pogosta zvišanje produktivnosti. Identifikacija faktorjev, ki imajo negativen ali pozitiven vpliv na uspešnost ponovne uporabe, je zato ključnega pomena (Lewis 1990). V domeni ogrodij je zato ključno vprašanje, kako zagotoviti uspešno uporabo ogrodja.
- **Izogibanje uporabi ogrodij.** Slabe izkušnje z uporabo ogrodij vplivajo na izogibanje nadaljnji uporabi ogrodij. Prav tako nekateri razvijalci programske opreme še vedno dajejo prednost »razvoju od začetka« (*development from scratch*), namesto da bi uporabili primerno ogrodje. Kako zagotoviti, da bodo uporabniki sprejeli ogrodje, je pomembno vprašanje za razvijalce ogrodij in za uporabnike ogrodij.

Zaradi navedenih težav so raziskovalci in razvijalci predlagali številne izboljšave ogrodij, ki so pretežno usmerjene na: (1) področje učenja ogrodij (Shull, Lanubile, & Basili 2000), (2) področje dokumentiranja ogrodij (Johnson 1992) in (3) področje tehničnih in organizacijskih smernic za uspešen razvoj in uporabo ogrodij (Froehlich et al. 1998; van Gorp & Bosch 2001).

⁴ Podatek je po Sourceforge.net povzet dne 13. 3. 2008.

Omenjene raziskave so vsekakor pripomogle k izboljšanju razvoja in uporabe ogrodij, vendar imajo tudi pomanjkljivosti, ki jih lahko strnemo v naslednje skupine:

- **Pomanjkanje empiričnih raziskav.** Osnovna pomanjkljivost raziskav na področju programskega inženirstva je, da ne temeljijo na verodostojnih empiričnih podatkih in teoretičnih osnovah (Shaw 1990). Takšno negativno lastnost imajo številne raziskave na področju ogrodij, ki predlagajo izboljšave brez analize empiričnih podatkov ali pa vplivov izboljšav ne ovrednotijo na osnovi empiričnih podatkov⁵. Pri takšnih raziskavah je zato neznano, kakšni so dejanski učinki predlaganih izboljšav.
- **Omejitve uporabljenih metod raziskav.** Večina raziskav na področju ogrodij so študije primerov (*case study*). Med študije primerov se uvrščajo vse tiste raziskave, ki temeljijo na izkustvih razvoja ali uporabe posameznega ali manjšega števila ogrodij⁶. Slabost takšnih raziskav je tvegano posploševanje rezultatov zunaj območja študije primera.
- **Neupoštevanje človeških faktorjev.** Smernice in izboljšave na področju ogrodij so predvsem tehnološko usmerjene, čeprav obstajajo empirične raziskave, ki so potrdile poglobljen vpliv človeških faktorjev na uspešnost ponovne uporabe (Morisio, Ezran, & Tully 2002). Zato bi bilo smiselno, da bi predlagane smernice in izboljšave ovrednotili neodvisni razvijalci oziroma uporabniki ogrodij.
- **Nepovezanost raziskav.** Kljub pomenu, ki ga imajo ogrodja v programskem inženirstvu, obstajajo le parcialne in nepovezane raziskave, ki dajejo odgovore na vprašanja, zastavljena v prejšnjem poglavju. Zato so vplivi, ki jih imajo predlagane tehnološke izboljšave na sprejetost ogrodij in na učinkovitost uporabe ogrodij, le domneve. Obstoječe raziskave temeljijo predvsem na pristopu »od spodaj navzgor« (*bottom-up*).
- **Pomanjkanje teoretičnih osnov.** Obstoječe raziskave pogosto ne temeljijo na modelih, metodah ali teorijah, ki so bili v preteklosti že preizkušeni. Pomanjkanje teoretičnih osnov vpliva na slabo povezanost raziskav področja ogrodij s preostalimi področji in na nekumulativen⁷ razvoj znanja na področju programskega inženirstva.

⁵ Primeri takšnih raziskav so: (Bosch, Molin, Mattsson, & Bengtsson 1997; Cunningham, Liu, & Zhang 2006; Deursen 1997; Fujiwara et al. 2003; Mamrak & Sinha 1999; Mattsson & Bosch 2006; Morisio, Romano, & Stamelos 2002).

⁶ Primeri študij primerov na področju ogrodij so: (Menzies & Di Stefano 2003).

⁷ Kumulativne vede se razvijajo tako, da nova spoznanja kot točnejša povsem zamenjajo tovrstna starejša, manj točna spoznanja in teorije.

1.2 METODOLOŠKI VIDIK DISERTACIJE

Osnovni cilj disertacije je, na »komplementaren način«, podati odgovore na temeljna vprašanja razvijalcev in uporabnikov ogrodi, ki so bila predstavljena v prejšnjem poglavju (glej poglavje 1.1). S terminom »komplementaren« označujemo naslednje:

- Pri raziskavah, ki bodo podale odgovore na zastavljena raziskovalna vprašanja, se bodo uporabili pristopi, ki minimizirajo ali odpravljajo splošne pomanjkljivosti obstoječih raziskav na področju ogrodi (glej poglavje 1.1).
- Raziskave bomo zasnovali tako, da se bo obstoječe znanje na področju ogrodi in programskega inženirstva kar najbolje vključevalo v naša spoznanja.

V skladu z omenjenim bomo pri disertaciji upoštevali naslednje: (1) empirični pristop, (2) možnost posploševanja rezultatov, (3) pristop »od zgoraj navzdol«, (4) človeške faktorje in (5) kumulativni raziskovalni pristop.

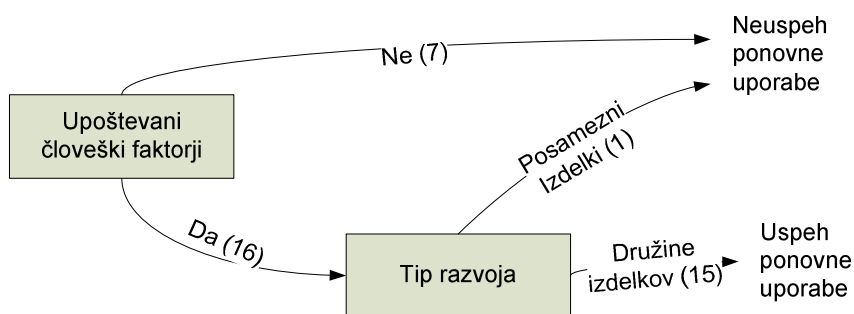
- **Empirični pristop.** Do novih spoznanj bomo poskušali priti z empiričnim raziskovalnim pristopom. Empirične študije v programskem inženirstvu ali eksperimentalnem programskem inženirstvu (*experimental software engineering*) so namenjene preučevanju prednosti in slabosti obstoječih metod, tehnik in orodij programskega inženirstva. Temeljijo na izkustvenih ali empiričnih podatkih⁸, katerih rezultat so empirično znanje ali preverjeni koncepti, ki se vključijo v obstoječe znanje programskega inženirstva.
- **Možnost posploševanja rezultatov.** Empirične študije se lahko izvedejo z različnimi metodami raziskovanja, med katerimi so najpogostejše: eksperiment (*experiment*), študija primera (*case study*) in anketa (*survey*). Dosedanje študije na področju ogrodi so v večini temeljile na študijah primera, katerih slabosti so: (1) slaba zmožnost repliciranja, (2) težavna generalizacija rezultatov in (3) neprimernost za uporabo statistične metode (Freimut et al. 2002). Slabosti študij primerov v veliki meri odpravljajo ankete (*survey*), katerih rezultate je, ob ustreznem raziskovalnem vzorcu, možno posplošiti.
- **Pristop »od zgoraj navzdol«.** Pristopa »od zgoraj navzdol« (*top-down*) in »od spodaj navzgor« (*bottom-up*) sta strategiji za obdelavo informacij in urejanje znanja v programski opremi in znanstvenih teorijah. Predstavljata vidik razmišljanja in učenja. Sinonima za pristop »od zgoraj navzdol« sta analiza in dekompozicija, za pristop »od spodaj navzgor« pa sinteza⁹. Naše raziskave bodo temeljile na pristopu »od zgoraj

⁸ Izkustven in empiričen sta sinonima.

⁹ Povzeto po http://en.wikipedia.org/wiki/Top-down_and_bottom-up_design.

navzdol«, ki predstavlja nasproten pristop od večine obstoječih raziskav na področju ogrodij. V pristopu »od zgoraj navzdol« se najprej pridobi oziroma določi pregled nad sistemom ali objektom raziskave. Ta se v nadaljevanju razgrajuje do vse večjih podrobnosti.

- **Upoštevanje človeških faktorjev.** Številne raziskave na področju ponovne uporabe so pokazale poglobljen vpliv dveh faktorjev na njihov uspeh: uporabe produktivnih linij in upoštevanja človeških faktorjev (Boehm 1999;Frakes & Kang 2005;Tomer et al. 2004).



SLIKA 2: POGLAVITNI FAKTORJI, KI VPLIVAJO NA USPEŠNOST PONOVNE UPORABE¹⁰ (MORISIO, EZRAN, & TULLY 2002)

Iz slike (Slika 2) je možno sklepati, da je v primeru uporabe ogrodij, ki realizirajo družine izdelkov, bistveno za zagotovitev uspeha prav upoštevanje človeških faktorjev.

- **Kumulativni raziskovalni pristop.** V disertacijo želimo vpeljati kumulativni raziskovalni pristop v smislu, da bodo nova, pridobljena znanja specializirala znanja na področju programskega inženirstva in sorodnih ved ter dopolnjevala obstoječa znanja na področju ogrodij. V tej točki je smiselno izpostaviti navezovanje na zagotavljanje uspešnosti ponovne uporabe, kot je definirano v metodologiji REBOOT (*REuse Based on Object-Oriented Techniques*) (Sindre, Conradi, & Karlsson 1995).

¹⁰ Na sliki je predstavljeno odločitveno drevo, ki je nastalo na osnovi CART- (*Classification And Regression Trees*) algoritma za podatkovno rudarjenje. Številke na puščicah označujejo število primerkov posamezne odločitve (Grimm & Yarnold 2004).

1.3 PREGLED PREDLAGANE REŠITVE

Na osnovi raziskovalnih vprašanj (glej poglavje 1.3.1), opredeljenega metodološkega pristopa (glej poglavje 1.1) in pregleda sorodnih raziskav (podan je v nadaljevanju) želimo podati odgovore z oblikovanjem povezanih kavzalnih modelov, ki pojasnjujejo sprejetost in uspešno uporabo ogrodi. Pri tem bodo kavzalni modeli temeljili na naslednjih postavkah¹¹ in predpostavkah¹²:

- **Postavka Po₁.** Obstajajo preverjeni modeli in teorije, ki pojasnjujejo sprejetost informacijskih tehnologij (IT) (Venkatesh et al. 2003).
- **Predpostavka Pr₁.** Splošne modele, ki pojasnjujejo sprejetost IT, je možno prilagoditi kontekstu ogrodi.
- **Postavka Po₂.** Sprejetost ogrodi še ni raziskana (Lederer et al. 2000). Zaradi posebnosti predmeta raziskave¹³ je natančnost obstoječih modelov sprejetosti v kontekstu ogrodi neznanka.
- **Predpostavka Pr₂.** Splošne modele, ki pojasnjujejo sprejetost IT, je možno nadgraditi v model sprejetosti ogrodi tako, da je ta natančnejši od splošnih modelov.
- **Postavka Po₃.** Obstajajo preverjeni modeli in teorije, ki definirajo uspešnost uporabe informacijskih sistemov (IS) (DeLone & Mclean 1992; Seddon 1997; Seddon et al. 1999).
- **Predpostavka Pr₃.** Splošne modele, ki definirajo uspešnost uporabe IS, je možno prilagoditi in uporabiti v kontekstu ogrodi.
- **Postavka Po₄.** Kavzalna povezava med uporabo IS in uspešnostjo uporabe IS ni možna brez postavitve povezave v kontekst. Več uporabe povzroči le več posledic, ki pa niso nujno pozitivne (Seddon 1997).
- **Predpostavka Pr₄.** V kontekstu ogrodi obstaja pozitiven vpliv med obsegom uporabe ogrodi in uspešnostjo njihove uporabe.
- **Postavka Po₅.** Sprejetost IT je dobra determinanta uporabe IT (Davis 1989).
- **Predpostavka Pr₅.** Obstaja povezava med sprejetostjo ogrodi in obsegom uporabe ogrodi.

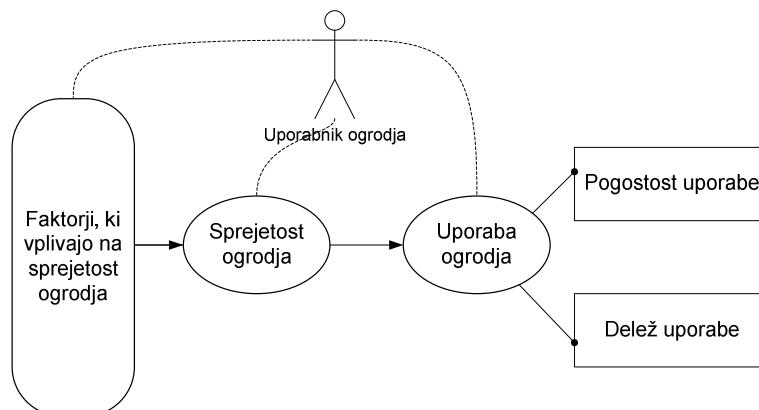
Na osnovi navedenih postavk in predpostavk je možno oblikovati osnutke kavzalnih modelov (Slika 3, Slika 4), ki z vidika »od zgoraj navzdol« definirajo: (1) osnovne koncepte, ki pojasnjujejo sprejetost ogrodi in uspešno uporabo ogrodi (ovalni), (2) predvidene kavzalne povezave med koncepti (puščice med ovalni) in (3) indikatorje, s katerimi je možno meriti koncepte

¹¹ Postavka - kar predstavlja osnovo, temelj čemu; izhodišče.

¹² Predpostavka - mnenje, trditev, ki se v danem primeru sprejme za izhodišče, ne glede na resničnost.

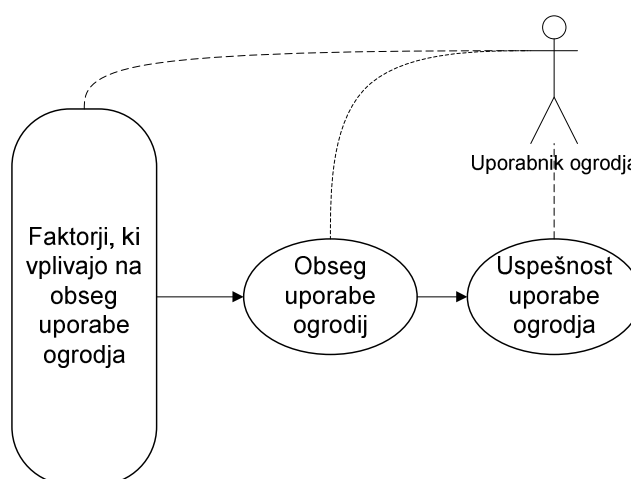
¹³ Posebnost predmeta raziskave se kaže v naslednjih vidikih: (1) uporabniki ogrodi niso končni uporabniki, temveč razvijalci programske opreme, (2) ogrodja niso končne rešitve, (3) ogrodja so kompleksnejša od preostalih tehnik ponovne uporabe.

(pravokotniki). Povezave med akterji in ovali določajo vidik, s katerega je smiselno vrednotiti posamezni kavzalni model.



SLIKA 3: OSNUTEK KAVZALNEGA MODELA SPREJETOSTI OGRODIJ

Na sliki (Slika 3) je prikazan osnutek modela sprejetosti ogrodij. Kot je razvidno iz modela, je sprejetost determinanta uporabe ogrodij, ki se najpogosteje meri s pogostostjo uporabe (na primer: število ur na teden) in deležem uporabe (na primer: delež razvoja na osnovi ogrodja glede na razvoj brez uporabe ogrodja). Naš cilj je, na osnovi obstoječih teorij in empiričnih podatkov identificirati poglobitve faktorje, ki vplivajo na sprejetost ogrodij. V empirične raziskave je smiselno vključiti uporabnike ogrodij. Pri obstoječih teorijah se bomo osredotočili na teorijo sprejetosti in na njenega najvidnejšega predstavnika - model TAM (*Technology Acceptance Model*) (Davis 1989).



SLIKA 4: OSNUTEK KAVZALNEGA MODELA USPEŠNOSTI UPORABE OGRODIJ

Na sliki (Slika 4) je prikazan osnutek modela uspešnosti uporabe ogrodij. Naš cilj je definirati uspešnost uporabe ogrodij in na osnovi obstoječih teorij in empiričnih podatkov potrditi povezavo med obsegom uporabe ogrodij in uspešnostjo uporabe ogrodij. Drugi cilj je

identificirati poglobitve faktorje, ki vplivajo na obseg uporabe ogrodij, in te povezati s konceptom sprejetosti. V empirične raziskave je smiselno vključiti uporabnike ogrodij. Pri obstoječih teorijah se bomo osredotočili na modele uspešnosti, ki izhajajo iz D&M modela uspešnosti IS (DeLone & Mclean 1992).

1.3.1 TEZA DOKTORSKE DISERTACIJE

Trditev, ki jo je v okviru doktorske disertacije treba utemeljiti in ki predstavlja vodilo raziskavi v okviru doktorske disertacije, je naslednja:

Z integracijo modelov, ki določajo sprejetost informacijskih tehnologij, uspešnost informacijskih sistemov in specifičnih faktorjev iz domene ogrodij, je možno definirati operacionaliziran model uspešnosti ogrodij, ki je natančnejši¹⁴ od obstoječih empiričnih modelov. Takšen model lahko predstavlja osnovo za model določanja sprejetosti ogrodij in uspešnosti ogrodij.

Teza doktorske disertacije je v nadaljevanju razčlenjena na deskriptivno¹⁵ (vprašanje RV₁) in pojasnjevalna¹⁶ (ostala vprašanja) raziskovalna vprašanja:

- **RV₁**: Kateri faktorji vplivajo na sprejetost ogrodij in kako jih lahko merimo?
- **RV₂**: Ali je predlagani model sprejetosti ogrodij natančnejši in primernejši od obstoječih modelov sprejetosti IT?
- **RV₃**: Ali obstaja povezava med sprejetostjo in uspešno uporabo ogrodja?

Pojasnjevalna raziskovalna vprašanja predstavljajo implicitno izražene hipoteze, ki bodo v okviru raziskave preizkušene s kvantitativnimi empiričnimi raziskavami (Čagran & Pšunder 2003).

¹⁴ Pri tem je natančnost definirana s stopnjo ujemanja (*goodness-of-fit*) strukturnega modela, stopnjo signifikantnosti kavzalnih povezav in stopnjo pojasnjene variance endogenih konstruktov modela (Hallsteinsen & Paci 1997).

¹⁵ Deskriptivna vprašanja so vprašanja opisa stanja pojava, osebe, situacije itd.

¹⁶ Eksplikativna vprašanja so vprašanja o odvisnih zvezah in razlikah.

1.3.2 PRIČAKOVANI IZVIRNI IN APLIKATIVNI PRISPEVKI DISERTACIJE

V okviru disertacije pričakujemo naslednje izvirne znanstvene prispevke (IZP):

- **IZP₁**: Operacionaliziran in empirično preizkušen model faktorjev, ki vplivajo na sprejetost ogrodij.
- **IZP₂**: Operacionaliziran in empirično preizkušen model faktorjev, ki vplivajo na uspešnost uporabe ogrodij.
- **IZP₃**: Integriran model sprejetosti in uspešnosti uporabe ogrodij, ki temelji na preizkušeni teoriji in empiričnih podatkih.

Na osnovi izvirnih znanstvenih prispevkov pričakujemo naslednje aplikativne rešitve:

- metodo za izbiro najustrežnejšega ogrodja,
- metodo za izboljšanje sprejetosti ogrodij in
- model za ocenjevanje sprejetosti ogrodij.

Poleg pričakovanih izvirnih znanstvenih prispevkov in aplikativnih prispevkov pričakujemo, da bodo rezultati disertacije vplivali na naslednja področja programskega inženirstva:

- **Projekti razvoja programskih ogrodij.** Predvidevamo, da bo »sprejetost ogrodij« kot determiniran in operacionaliziran konstrukt pozitivno vplivala na smernice razvoja ogrodij v smislu njihovega vrednotenja in povečanja njihove sprejetosti.
- **Projekti razvoja rešitev, temelječih na ogrođjih.** Predvidevamo, da bo **IZP₃** pozitivno vplival na odločitve o izbiri ogrodja in posledično pripomogel k višji stopnji uspešnosti takšne vrste projektov.
- **Metrike programskih ogrodij.** Predvidevamo, da bodo formalno verificirani faktorji in indikatorji kazalnih modelov prispevali znanje k naboru metrik uspešnosti ogrodij.
- **Definiranje predhodnikov eksogenih faktorjev modela.** Predvidevamo, da bodo operacionalizirani modeli disertacije predstavljali osnovo nadaljnjih raziskav v smislu identifikacije dejavnikov, ki vplivajo na eksogene konstrukte **IZP₁**.
- **Primerjava odprtokodnih in lastniških ogrodij.** **IZP₁** bo lahko osnova primerjalnim analizam uspešnosti odprtokodnih in lastniških ogrodij.
- **Faktorji uspešnosti ponovno uporabne programske opreme.** Izvirni znanstveni prispevki raziskave bodo lahko verodostojna osnova za bolj generične raziskave uspešnosti v kontekstu ponovno uporabne programske opreme in programskega inženirstva.

1.4 STRUKTURA DISERTACIJE

Vsebina disertacije je v nadaljevanju strukturirana, kot sledi. V **drugem poglavju** je zajeto raziskovalno področje disertacije, ki predstavlja in analizira obstoječe znanje, relevantno za disertacijo. Obstoječe znanje je strukturirano v dva osnovna sklopa: (1) predstavitev in vsebinska umestitev osrednjega predmeta disertacije – programskih ogrodij in (2) predstavitev in analiziranje teoretičnega osnov, na katerih temelji disertacija.

V **tretjem poglavju** so na osnovi obstoječega znanja s predstavljenega raziskovalnega področja izvedene parcialne in med sabo povezane raziskave, na osnovi katerih se nato izoblikujejo kavzalni modeli, ki podajajo odgovore na raziskovalna vprašanja disertacije.

V **četrtem poglavju** so predstavljene aplikacije raziskav in njihove implikacije za raziskovalce, razvijalce in uporabnike ogrodij.

V **petem, zadnjem poglavju** so podani zaključki disertacije, ki povzemajo ugotovitve prejšnjih poglavij. Podani so tveganja in omejitve disertacije in njihovi vplivi na ugotovitve disertacije. Na osnovi ugotovitev disertacije so podane smernice za bodoče temeljne in aplikativne raziskave na področju ogrodij in tudi širšem področju programskega inženirstva.

V **prilogah** disertacije so predstavljene podrobnosti raziskav, ki smo jih izvedli za potrebe disertacije in ki smo jih zaradi izboljšanja berljivosti izpustili iz vsebine disertacije. Priloge vključujejo: raziskovalne instrumente, pregled uporabljenih statističnih metod in rezultate statističnih metod.

2 PREDSTAVITEV RAZISKOVALNEGA PODROČJA

Raziskave na področju računalništva se v osnovi delijo na (Glass, Ramesh, & Vessey 2004):

- računalniško znanost (*computer science*)¹⁷,
- programsko inženirstvo (*software engineering*) in
- informacijske sisteme¹⁸ ali informatiko¹⁹ (*information systems*).

Na osnovi taksonomije raziskovalnih področij računalništva (Slika 5), ki so jo izvedli Glass et al. (2004), se naša disertacija nahaja v preseku programskega inženirstva in informatike.

"Znanost nas nenehno oddaljuje od opic. Seveda, če nekdo želi biti opica, se tovrstnemu odmikanju upira."

neznan avtor

Topic Categories				CS	SE	IS	Topic Categories				CS	SE	IS
1.0	Problem-Solving Concepts	14.7%	5.9%	5.9%	6.0	Systems/software management concepts				11.5%	6.8%		
1.1	Algorithms	5.8%	0.5%	0.2%	6.1	Project/product management (incl. risk management)	0.2%	3.3%	3.1%				
1.2	Mathematics/Computational Science	6.7%	-	-	6.2	Process management	-	2.2%	0.6%				
1.3	Methodologies (object, function/process, information/data, event, business rules, ...)	-	4.9%	0.8%	6.3	Measurement/metrics (development and use)	-	6.2%	0.8%				
1.4	Artificial Intelligence	2.4%	0.5%	4.9%	6.4	Personnel issues	-	0.3%	-				
					6.5	Acquisition of (Packaged/Custom) Software	0.2%	0.5%	2.3%				
2.0	Computer Concepts	28.7%	10.9%	0.0%	7.0	Organizational concepts	0.3%	1.9%	65.6%				
2.1	Computer/hardware principles/architecture	10.2%	-	-	7.1	Organizational Structure	-	0.5%	5.0%				
2.2	Intercomputer communication (networks, distributed systems)	17.7%	9.5%	-	7.2	Strategy	-	-	6.6%				
2.3	Operating systems (as an augmentation of hardware)	0.80%	1.4%	-	7.3	Alignment (incl. business process reengineering)	-	0.5%	6.9%				
2.4	Machine/assembler-level data/instructions	-	-	-	7.4	Organizational learning/knowledge management	-	-	1.4%				
					7.5	Technology transfer (incl. innovation, acceptance, adoption, diffusion)	0.1%	0.3%	19.4%				
3.0	Systems/software concepts	19.1%	54.8%	6.4%	7.6	Change management	-	-	1.6%				
3.1	System architecture/engineering	0.48%	1.9%	2.9%	7.7	Information technology implementation	-	-	1.6%				
3.2	Software life cycle/engineering (incl. requirements, design, coding, testing, maintenance)	-	8.7%	1.4%	7.8	Information technology usage/operation	-	-	24.4%				
3.3	Programming languages	3.8%	3.8%	1.4%	7.9	Management of "computing" function	0.2%	-	11.0%				
3.4	Methods/techniques (incl. reuse, patterns, parallel processing, process models, data models...)	3.8%	18.2%	0.2%	7.10	IT Impact	-	0.3%	15.3%				
3.5	Tools (incl. compilers, debuggers)	5.3%	12.2%	0.2%	7.11	Computing/information as a business	-	-	-				
3.6	Product quality (incl. performance, fault tolerance)	1.8%	8.4%	1.4%	7.12	Legal/ethical/cultural/political (organizational) implications	-	0.3%	3.4%				
3.7	Human-computer interaction	3.2%	1.1%	1.4%									
3.8	System security	0.80%	0.5%	0.2%									
4.0	Data/information concepts	15.4%	7.6%	3.0%	8.0	Societal concepts	-	0.3%	1.4%				
4.1	Data/file structures	1.9%	0.8%	-	8.1	Cultural implications	-	-	0.2%				
4.2	Data base/warehouse/mart organization	8.4%	4.6%	1.6%	8.2	Legal implications	-	-	0.2%				
4.3	Information retrieval	4.0%	1.4%	0.4%	8.3	Ethical implications	-	-	-				
4.4	Data analysis	0.64%	0.5%	0.6%	8.4	Political implications	-	0.3%	1.0%				
4.5	Data security	0.48%	0.3%	0.4%									
5.0	Problem domain-specific concepts	21.5%	2.7%	6.4%	9.0	Disciplinary issues	-	3.5%	4.3%				
5.1	Scientific/engineering (incl. bioinformatics)	0.48%	0.3%	-	9.1	"Computing" research	-	1.1%	3.3%				
5.2	Information systems (incl. decision support, group support systems, expert systems)	0.64%	1.6%	6.4%	9.2	"Computing" curriculum/teaching	-	2.4%	1.0%				
5.3	Systems programming	-	-	-									
5.4	Real-time (incl. robotics)	0.16%	0.5%	-									
5.5	Edutainment (incl. graphics)	20.2%	0.3%	-									

SLIKA 5: TAKSONOMIJA RAZISKOVALNIH PODROČIJ V DISCIPLINAH RAČUNALNIŠTVA
(GLASS, RAMESH, & VESSEY 2004)

¹⁷ Računalniška znanost je veja znanosti in tehnologije, ki obravnava obdelavo informacij s pomočjo računalniških sistemov (www.islovar.org).

¹⁸ Informacijski sistem (IS) je urejen in organiziran sistem, ki uporabnike oskrbuje z vsemi potrebnimi informacijami za odločanje. Osnovne aktivnosti informacijskega sistema so: zbiranje, shranjevanje, obdelava in posredovanje rezultatov končnim uporabnikom.

¹⁹ Informatika je znanost o sistematični obdelavi podatkov.

Natančneje gledano, predstavlja disertacija analizo organizacijskih konceptov (sprejetost, uporaba, uspešnost) na področju računalniških konceptov (ponovna uporaba, programska ogrodja). Iz slike (Slika 5) je možno razbrati, da spadajo omenjena področja disertacije med najbolj raziskovana področja (obkrožene vrednosti), kar je pozitivno za kumulativni raziskovalni pristop. Kljub temu smo identificirali zelo malo raziskav, ki so, podobno kot naša disertacija, naslovile presek omenjenih raziskovalnih področij. S tega vidika je naša disertacija izvirna.

2.1 PONOVNA UPORABA V PROGRAMSKEM INŽENIRSTVU

Projekti razvoja programske opreme se ubadajo z vedno večjimi konkurenčnimi pritiski, ki vladajo na trgu programske opreme. Ustrezen odziv na vse večjo in kakovostno ponudbo zahteva hiter razvoj novih programskih proizvodov oziroma nadgradenj obstoječih proizvodov, širitev ponudbe, zagotavljanje skladnosti s standardi in visoko stopnjo povezljivosti z drugimi proizvodi. Ponovna uporaba je učinkovito sredstvo za doseganje omenjenih ciljev.

Poglaviten namen ponovne uporabe programske opreme je izboljšanje aktivnosti programskega inženirstva s povečanjem produktivnosti, kakovosti in hitrosti izdelave izdelkov (Mili 1995). Leach (1996) tako definira ponovno uporabo v programskem inženirstvu:

Ponovna uporaba je primer stanj, v katerih se je določen programski izdelek uporabil v več kontekstih. Pri tem se pod terminom programski izdelek lahko obravnava vsak izdelek, ki nastane v procesu razvoja programske opreme.

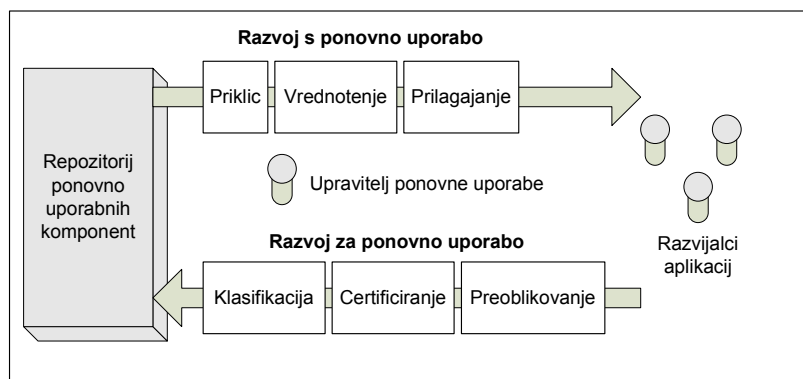
Ponovna uporaba je proces razvoja programske opreme na osnovi obstoječih programskih proizvodov. Nasprotje je razvoj programske opreme z začetne točke (*building from scratch*). Termin, ki se pogosto navezuje na ponovno uporabo, je »stopnja ponovne uporabe« (*reusability*), ki je definirana kot (IEEE 1990):

Stopnja ponovne uporabe je zmožnost uporabe določenega programskega modula ali drugega razvojnega dokumenta v več kot enem programskem sistemu.

S ponovno uporabo se ukvarja inženirstvo ponovne uporabe. Cilj inženirstva ponovne uporabe je ponovna uporaba vseh informacij, ki so bile izdelane v procesu razvoja programske opreme.

Koncept delovanja ponovne uporabe je preprost. Na eni strani vlagamo dodaten napor (človeške, finančne in ostale vire), da zagotovimo ustrezno okolje, namenjeno razvoju ponovno uporabnih izdelkov (*reusable artefacts*), medtem ko na drugi strani pričakujemo dolgoročne

koristi, ki se izražajo v manjšem naporu, potrebnem za razvoj novih proizvodov, ki temeljijo na ponovno uporabnih izdelkih. Ponovno uporabni izdelek v programskem inženirstvu ni le programski kod, ampak se mednje uvrščajo vsi izdelki, ki nastajajo v procesu razvoja programske opreme (na primer: arhitektura, modeli načrtovanja, algoritmi, testni primeri in skripte za prevajanje). Celovit pogled na proces razvoja programske opreme na osnovi ponovne uporabe prikazuje proces razvoja programske opreme REBOOT (*REuse Based on Object-Oriented Techniques*) (Slika 6).



SLIKA 6: PROCES RAZVOJA NA OSNOVI PONOVNE UPORABE (SINDRE, CONRADI, & KARLSSON 1995)

Ponovna uporaba zagotavlja pogoje za številne prednosti pri razvoju programske opreme, med katere spadajo (Sindre, Conradi, & Karlsson 1995): višja kakovost izdelkov, nižji stroški razvoja, večja produktivnost, krajši čas razvoja, večja predvidljivost razvoja in enostavnejše vzdrževanje.

Kljub prednostim ponovna uporaba ni izboljšala programskega inženirstva v tolikšni meri, kot je bilo pričakovano (Lewis 1990). Najpogosteje omenjeni vzroki za njen neuspeh so (Leach 1996; Sindre, Conradi, & Karlsson 1995): visoki začetni stroški, težavno obvladovanje ponovno uporabnih komponent, nezaupanje v tuje komponente²⁰, problem odgovornosti v primeru uporabe tujih komponent in pomanjkanje podpore vodstva pri vpeljavi sistematične ponovne uporabe. Iz navedenega je možno predpostavljati, da obstajajo težave na tehnološkem in netehnološkem področju (organizacija, človeški dejavniki, pravni vidik).

Prav zaradi slabosti in neuspešnih projektov ponovne uporabe ostaja poglobljena naloga področja identificirati poglobljene faktorje, ki povzročajo neuspeh ponovne uporabe, kakor tudi faktorje, ki vplivajo na uspešnost ponovne uporabe (Lewis 1990).

Ponovne uporabe v programskem inženirstvu je več vrst. Prav tako obstaja več klasifikacij ponovne uporabe, in sicer klasifikacija glede na (Sindre, Conradi, & Karlsson 1995): (1) vrsto

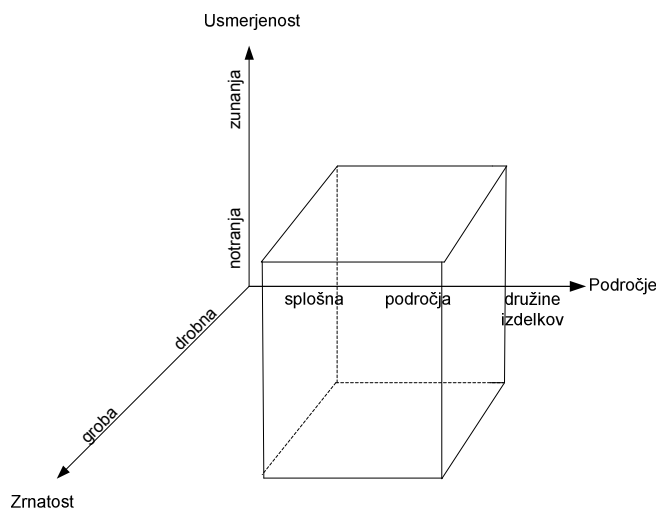
²⁰ NIH (*Not-Invented-Here*) syndrome.

adaptacij ponovno uporabne komponente, (2) obseg ponovne uporabe, (3) ciljne uporabnike in (4) zrnatost.

2.2 PROGRAMSKA OGRODJA

Med tehnike ponovne uporabe, ki imajo največji potencial, spadajo programska ogrodja (Krajnc & Heričko 2004;Lajoie & Keller 1994;van Gorp & Bosch 2001). Programska ogrodja (v nadaljevanju ogrodja) so vnaprej izdelani sistemi, ki vsebujejo znanje o reševanju določene problemske domene, na podlagi katerega se razvijajo končne programske rešitve oziroma primerki ogrodij (*framework instances*).

Ogrodja predstavljajo tehniko ponovne uporabe, zato je njihov osnovni cilj dvig produktivnosti v programskem inženirstvu (Mattsson 1996). Z vidika obsega ponovne uporabe se programska ogrodja uvrščajo med tehnike področne ponovne uporabe (*domain reuse*) oziroma ponovne uporabe, ki je namenjena družinam programskih izdelkov (*product families*). Z vidika zrnatosti se programska ogrodja nahajajo na visokem (grobem) nivoju zrnatosti (glej območje kvadra na Slika 7).



SLIKA 7: UMESTITEV OGRODIJ V REBOOT KLASIFIKACIJO PONOVNE UPORABE

Najpogosteje uporabljena definicija ogrodja je naslednja (Johnson & Foote 1988):

Ogrodje je množica razredov, ki vključujejo abstrakten načrt rešitve za družino povezanih problemov²¹.

²¹ A framework is a set of classes that embodies an abstract design for solutions to a family of related problems.

Večina sodobnih ogrodij je objektno orientirana (*Object-Oriented Framework - OOF*) (Krajnc 2006). Objektno orientirana definicija ogrodja je povzeta po Gamma et al. (Gamma et al. 1995) in se glasi:

Ogrodje je množica sodelujočih razredov, ki sestavljajo ponovno uporaben načrt za specifično vrsto programske opreme. Ogrodje določa arhitekturne smernice z razdelitvijo načrta v abstraktne razrede in z definiranjem njihovih odgovornosti in sodelovanj. Razvijalec prilagaja ogrodje za posamezno aplikacijo s povezovanjem primerkov razredov ogrodja.

Obstaja še več definicij ogrodij, ki jih je analiziral Mattsson (1996) in na njihovi osnovi oblikoval lastno, generično definicijo ogrodja:

Ogrodje predstavlja generično arhitekturo, ki je zasnovana z namenom zviševanja ponovne uporabnosti. Ogrodja vključujejo množico sodelujočih abstraktnih in konkretnih razredov, ki enkapsulirajo obnašanje podedovanih specializacij²².

Ogrodja so torej vzorci, ki vsebujejo zamisel načrta rešitve določene problemske domene in množico gradnikov, ki vsak zase izpolnjujejo posamezno vlogo v ogrodju. Poenostavljeno povedano, predstavljajo ogrodja programske opreme, ki jo lahko programer uporabi, prilagodi in razširi z namenom ustrezati zahtevam končne programske rešitve. Ogrodja (objektno orientirana) temeljijo na uveljavljenih vzorcih in izkoriščajo prednosti treh konceptov objektno paradigme: abstraktnih podatkov (razredov), polimorfizma in dedovanja. Takšnim ogrodjem so skupne naslednje karakteristike (Johnson & Foote 1988):

- **Razredi odjemalci** (*client classes*) – končne programske rešitve se ogrodju prilegajo na t. i. razširitvenih točkah.
- **Sodelovanje objektov** (*collaboration of objects*) – (abstraktni) razredi ogrodja definirajo model obnašanja (*model of interaction*). Končne programske rešitve se zato obnašajo po definiranem modelu.
- **Zamenjava nadzora** (*inversion of control*) – Model obnašanja ogrodja določa način vključevanja razredov odjemalcev, kar pomeni, da igra ogrodje vlogo glavnega programa (v nasprotju z vključevanjem programskih knjižnic). Koncept je poznan kot »Hollywoodsko načelo« (*Hollywood principle*).

²² A (*generative*) architecture designed for maximum reuse, represented as a collective set of abstract and concrete classes; encapsulated potential behavior for sub-classed specializations.

2.2.1 ZGODOVINA OGRODIJ

Pojem »programska ogrodja« ni nov, saj se je koncept ogrodij pojavil že v osemdesetih letih prejšnjega stoletja, in sicer v okoljih Smalltalk (Adele 1984) in Apple Inc. (Kurt 1986). Prvo široko uporabljeno ogrodje je bil uporabniški vmesnik Smalltalk-80, znan pod imenom model-pogled-nadzornik (*Model-View-Controller*) ali krajše MVC.

V devetdesetih letih prejšnjega stoletja so se ogrodja iz domene uporabniških vmesnikov razširila na preostale programske domene. Med pomembnejša ogrodja, razvita v devetdesetih, spadajo CommonPoint²³, HotDraw²⁴, ACE²⁵, JAWS²⁶, CORBA (*Common Object Request Broker Architecture*) in MFC (*Microsoft Foundation Classes*). Ogradje MVC je bilo ob ogrodju OWL (*Object Windows Library*) kar nekaj časa dejansko industrijski standard za razvoj grafičnih aplikacij za osebne računalnike.

K razširitvi in uveljavitvi ogrodij v devetdesetih je pripomogel tudi programski jezik Java. Večina ogrodij za Java nastaja znotraj delovnih skupin v procesu JCP (*Java Community Process*), ki ga upravlja podjetje Sun. Med ogrodja, ki so nastala v okviru JCP, spadajo: EJB (*Enterprise JavaBeans*), RMI (*Remote Method Invocation*) AWT (*Abstract Window Toolkit*), Swing, JFC (*Java Foundation Classes*), JSP (*JavaServer Pages*), JSF (*JavaServer Faces*), Collection Framework, JMF (*Java Media Framework*) in JAF (*JavaBeans Activation Framework*). Veliko ogrodij na osnovi Jave nastaja tudi v odprtokodnih projektih. Primeri takšnih ogrodij so: Struts, Spring, Hibernate, JUnit, Avalon in JCorporate Espresso (Krajnc & Heričko 2004).

Med najpomembnejša in najpogosteje uporabljana ogrodja zadnje generacije spadajo: Microsoft .NET, Spring, Jakarta Struts, Django, Hibernate, Ruby on Rails in Eclipse framework²⁷.

Poleg omenjenih ogrodij, ki so javno dostopna, številna podjetja razvijajo še lastna ogrodja (*in-house framework*). Takšna ogrodja se uporabljajo zgolj interno in niso namenjena za prodajo ali javno uporabo. V zadnjem času je mogoče zaslediti tudi uporabo ogrodij za razvoj programskih orodij (Krajnc et al. 2005). Primer takšnega orodja je Eclipse IDE.

²³ Množica ogrodij za hitrejši razvoj aplikacij.

²⁴ Ogradje za izgradnjo grafičnih urejevalnikov, napisano v jeziku Smalltalk.

²⁵ *ADAPTIVE Communication Environment* - objektno orientirano ogrodje, namenjeno za komunikacijsko programsko opremo.

²⁶ *Adaptive Web Server* - spletni strežnik in ogrodje za izgradnjo drugih vrst strežnikov.

²⁷ Povzeto po opisni statistiki ankete, ki smo jo izvedli v okviru disertacije.

2.2.2 PREDNOSTI IN SLABOSTI OGRODIJ

Programerji in vodje projektov se za razvoj na osnovi ogrodij odločajo predvsem zaradi: (1) minimiziranja obsega implementacije, ki je potrebna za razvoj aplikacij, in (2) lažjega obvladovanja znanja domene, v kateri organizacija razvija aplikacije (Mattsson 1996). Prednosti ogrodij so torej naslednje:

- **Hitrejši in učinkovitejši razvoj.** Z uporabo ogrodja aplikacije nikoli ne gradimo od začetka, ampak programski kod oziroma storitve, ki jih zagotavlja ogrodje, ponovno uporabimo. Ker aplikacije, ki temeljijo na enakem ogrodju, gradimo po ustaljenem vzorcu, se učinkovitost razvoja še dodatno poveča.
- **Boljša kakovost programske opreme.** Izvorni kod ogrodij običajno temelji na preizkušenih programskih vzorcih in je zaradi večkratne uporabe izpostavljen obsežnim testiranjem.
- **Ogrodja omogočajo ponovno uporabo izvornega koda in načrtovanja.**
- **Omogočajo preusmeritev fokusa s področja sistemskih problemov na področje domene.** Sistemski problemi so rešeni v ogrodjih, zato se razvijalcem aplikacij z njimi ni treba ubadati.
- **Zagotavljajo visoko stopnjo medizvedljivosti (*interoperability*).** Aplikacije, ki temeljijo na enakem ogrodju, so si glede arhitekture sorodne. Skupaj s podporo uveljavljenim standardom imajo na ogrodju temelječe aplikacije zagotovljeno visoko stopnjo medsebojne izvedljivosti.

Avtorja (Fayad & Schmidt 1997) navajata, da izhajajo prednosti uporabe ogrodij iz naslednjih lastnosti ogrodij:

- **Ponovna uporabnost (*reusability*).** Stopnja ponovne uporabe ogrodja je običajno višja kot pri preostalih tehnikah ponovne uporabe²⁸. Ogrodja omogočajo ponovno uporabo na nivoju programskega koda, vzorcev in opisa konceptov, potrebnih za reševanje določenega problema. Z opisovanjem konceptov definirajo ogrodja slovar za problemsko področje. Razvijalec, ki uporablja ogrodje, vidi problemsko področje prav skozi slovar ogrodja. S tem zagotavljajo ogrodja še ponovno uporabo konceptov analize (Roberts & Johnson 1997).
- **Modularnost (*modularity*).** Ogrodja zvišujejo modularnost programske opreme z ločevanjem vmesnikov od implementacije. Zaradi večje modularnosti je identifikacija napak in sprememb v takšni programski opremi lažja. S tem se: (1) zmanjša napor za

²⁸ Ogrodja predstavljajo tudi do 80% kode končnega izdelka (Seddon, Staples, Patnayakuni, & Bowtell 1999).

razumevanje in vzdrževanje programske opreme in (2) zvišuje kakovost programske opreme.

- **Razširljivost (*extensibility*).** Ogrodja povečujejo razširljivost programske opreme z zagotavljanjem standardnih razširitvenih točk (*hook methods*). Razširitvene točke zagotavljajo stabilnost vmesnikov ogrodij z njihovimi najpogostejšimi implementacijami. Zaradi razširitvenih točk so ogrodja lahko hkrati stabilna in razširljiva.

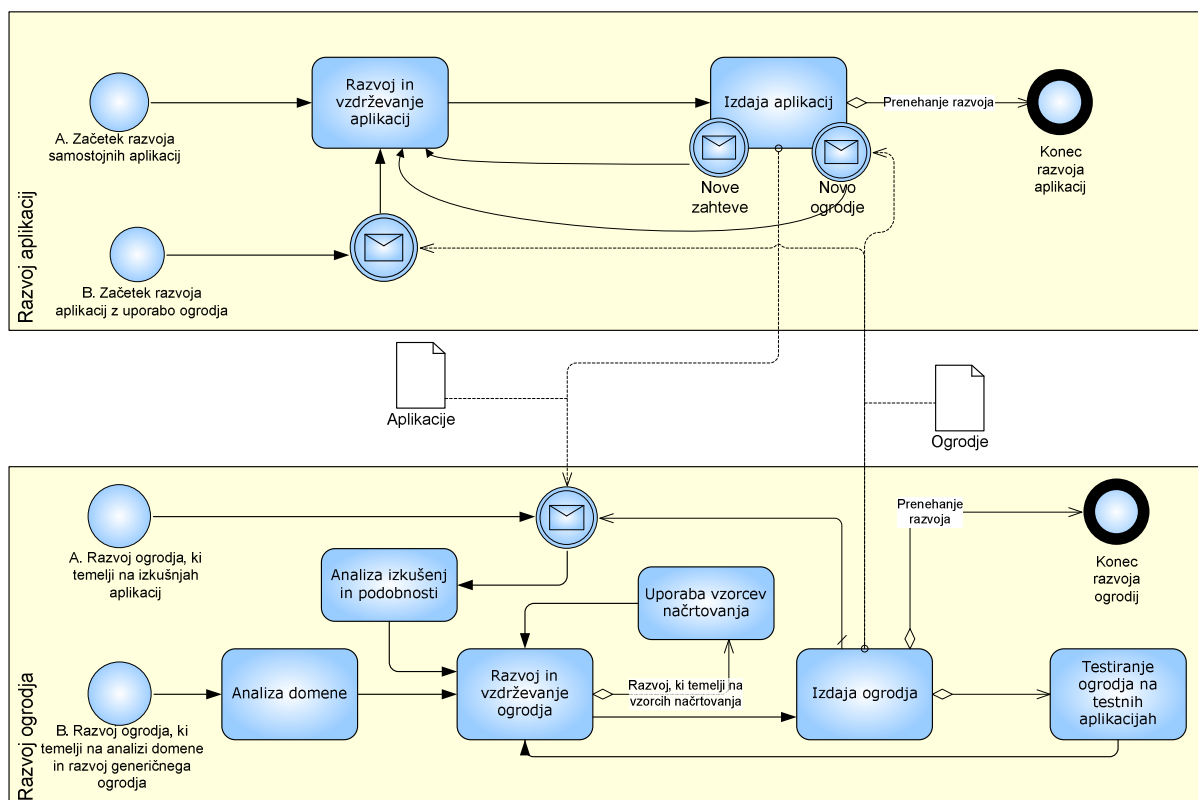
Potencialne slabosti uporabe ogrodij so naslednje (Mattsson 1996; Roberts & Johnson 1997):

- **Zapleten razvoj ogrodij.** Razvoj kakovostnega ogrodja je težaven in običajno zahteva bogate izkušnje v arhitekturni zasnovi aplikacij in problemski domeni.
- **Težavno dokumentiranje ogrodij.** Zaradi kompleksnosti je ogrodja težko dokumentirati. Če ogrodja niso ustrezno dokumentirana, jih razvijalci aplikacij ne uporabljajo.
- **Težavno zagotavljanje povezljivosti.** Ogrodja se nenehno razvijajo in spreminjajo, zato je težavno zagotavljati kompatibilnost s predhodniki in njihovimi primerki.
- **Zmanjšana učinkovitost aplikacij.** Splošnost in prožnost ogrodij lahko predstavljata omejitve za razvijalce aplikacij in učinkovitost razvitih aplikacij.
- **Težavno razhroščevanje.** Razhroščevanje ogrodij in primerkov ogrodij je težavno, saj pogosto ni mogoče lokalizirati napak («Ali se napaka pojavlja v ogrodju ali v aplikaciji?»). Prav tako so lahko napake, ki se pojavijo v ogrodju, neodpravljljive za uporabnike ogrodij²⁹.
- **Pomanjkanje standardov.** Na področju ogrodij standardi ne obstajajo ali so šele v povojih, kar zmanjšuje njihovo zamenljivost (*replaceability*). V zadnjem času se podjetja lotevajo omenjene težave z delovnimi skupinami, v katere so vključena različna podjetja in odprtokodne skupnosti.
- **Odvisnost od programskega jezika.** Ker so ogrodja napisana v določenem programskem jeziku, so vezana nanj. Namestitev ogrodja v okolje, ki temelji na drugačnem programskem jeziku, zato ni možno.

²⁹ V primeru lastniških, zaprtokodnih ogrodij.

2.2.3 OGRODJA V PROCESU RAZVOJA PROGRAMSKE OPREME

Proces razvoja programske opreme na osnovi ogrodij predstavlja poseben primer procesa razvoja na osnovi ponovno uporabne programske opreme, kot je na primer proces REBOOT (glej Slika 6). Procesov oziroma modelov razvoja programske opreme na osnovi ogrodij je več (Mattsson 1996). Na spodnji sliki (Slika 8) so v obliki modela BPMN³⁰ predstavljeni štirje procesi razvoja ogrodij in na ogrodju temelječih aplikacij:



SLIKA 8: RAZLIČNI PROCESI RAZVOJA OGRODJA, PRIKAZANI V MODELU BPMN

- **Proces razvoja ogrodja, ki temelji na izkušnjah razvoja aplikacij.** Takšen proces se prične z razvojem (družin) aplikacij (glej začetna dogodka A). Na osnovi podobnosti med aplikacijami in izkušenj razvijalcev se lahko razvijalci odločijo skupne funkcionalnosti prenesti v ogrodje. Po izdaji ogrodja se nato vse samostojne aplikacije preoblikujejo v aplikacije, ki temeljijo na ogrodju. Izkušnje pri razvoju takšnih aplikacij se nato ponovno prenesejo v razvoj ogrodja in proces se ponovi.
- **Proces razvoja ogrodja, ki temelji na analizi domene.** Začetek takšnega procesa predstavlja analiziranje abstrakcij v domeni, ki se nato vključijo v razvoj ogrodja (glej začetna dogodka B). Na osnovi ogrodja se nato začnejo razvijati končne aplikacije.

³⁰ BPMN je akronim za *Business Process Modelling Notation*.

Izkušnje z razvojem končnih aplikacij in odzivi uporabnikov se nato upoštevajo pri vzdrževanju ogrodja.

- **Proces razvoja ogrodja, ki temelji na uporabi vzorcev načrtovanja.** Proces se prične z razvojem samostojne aplikacije (glej začetna dogodka A). Na osnovi analiziranja aplikacij se nato z upoštevanjem vzorcev načrtovanja prične razvoj ogrodja. V nadaljevanju se ogrodje posodablja na osnovi izkušenj, ki se pridobijo pri razvoju aplikacij in njihovih uporabnikov.
- **Proces razvoja generičnega ogrodja.** Razvoj generičnega ogrodja se prične z analiziranjem domene (glej začetna dogodka B). Na osnovi analize domene se prične razvoj generičnega ogrodja. Testiranje ogrodja se izvede s testnimi aplikacijami, na osnovi katerih se nato izboljšuje ogrodje. Takšen proces je lahko popolnoma neodvisen od razvoja aplikacij.

Vzporedno s procesom razvoja in vzdrževanja so ogrodja izpostavljena tudi evoluciji oziroma zorenju ogrodja. Roberts in Johnson (1997) sta na osnovi vzorcev zasnove ogrodij opredelila naslednje stopnje zrelosti ogrodij:



SLIKA 9: EVOLUCIJA OGRODIJ, TEMELJEČA NA VZORCIH ZASNOVE (ROBERTS & JOHNSON 1997)³¹

- **Ogrodja bele škatle** (*white-box framework*). Instanciranje takšnega ogrodja temelji na modifikacijah izvornega koda ogrodja in na dedovanju razredov ogrodja.
- **Knjižnice komponent** (*component library*). Razredi, ki so skupni aplikacijam v domeni ogrodja, so v ogrodje vključeni v obliki knjižnic.

³¹ Prvo fazo v vzorcih evolucije ogrodij predstavljajo trije primerki razvoja aplikacij v domeni, v kateri naj bi se razvilo ogrodje. Takšen pristop razvoja ogrodij je skladen s procesom razvoja ogrodja na izkušnjah razvoja aplikacij (Slika 8).

- **Vročne točke** (*hot spots*). V takšnih orodjih je koda, ki se pogosto spreminja, ločena od kode, ki se ne spreminja (*frozen spots*). Zaradi lažjega obvladovanja je koda, ki se spreminja, združena v razredih, ki se najpogosteje razširjajo s kompozicijo.
- **Vključitveni objekti** (*pluggable objects*). Namesto trivialnih podrazredov vsebuje takšno ogrodje podrazrede, ki jih je možno parameterizirati.
- **Fino zrnate komponente** (*fine-grained components*). S ciljem povečanja stopnje ponovne uporabnosti so razredi in knjižnice v takšnih ogrodjih fino zrnati.
- **Ogrodja črne škatle** (*black-box frameworks*). V takšnih ogrodjih so knjižnice strukturirane na osnovi dedovanja, medtem ko se za njihovo povezovanje uporablja kompozicija.
- **Grafična razvojna okolja** (*visual building tools*). Grafična razvojna okolja so v pomoč razvijalcem aplikacij pri specifikaciji in povezovanju objektov v primerkih ogrodja.
- **Programskim jezikom sorodna orodja** (*language tools*). Takšnim ogrodjem so dodana orodja za njihov nadzor izvajanja in pomoč pri razhroščevanju.

2.2.4 VRSTE IN KLASIFIKACIJE OGRODIJ

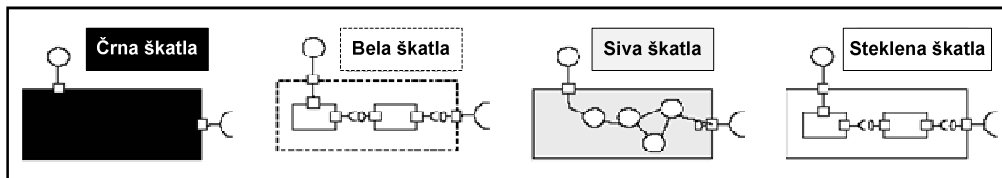
Ogrodja se po svoji zasnovi, obsežnosti in namenu zelo razlikujejo. Čeprav obstajajo najrazličnejše klasifikacije ogrodij, se ogrodja najpogosteje delijo na (Johnson & Foote 1988):

- **domenska ogrodja** (*domain framework*) – naslavljajo določene problemske domene (na primer: zavarovalništvo, računovodstvo in upravljanje človeških virov),
- **orodna ogrodja** (*utility framework*) – naslavljajo določene programske domene (na primer: trajnost podatkov, uporabniški vmesnik in testiranje kode),
- **aplikacijska ogrodja** (*application framework*) – obsežna ogrodja, ki so uporabna za različne problemske domene in naslavljajo številne programske domene.

V zadnjem času je vse več poskusov uveljavljanja t. i. organizacijskih ogrodij (*enterprise frameworks*), ki zaokrožujejo posamezno problemsko domeno poslovanja. Če jih primerjamo z ostalimi ogrodji, so organizacijska ogrodja po obsegu večja in bolj kompleksna; v njih so lahko vsebovane najrazličnejše komponente in druga ogrodja. Organizacijska ogrodja vključujejo infrastrukturni, domenski in arhitekturni vidik (Fayad & Hamu 2000).

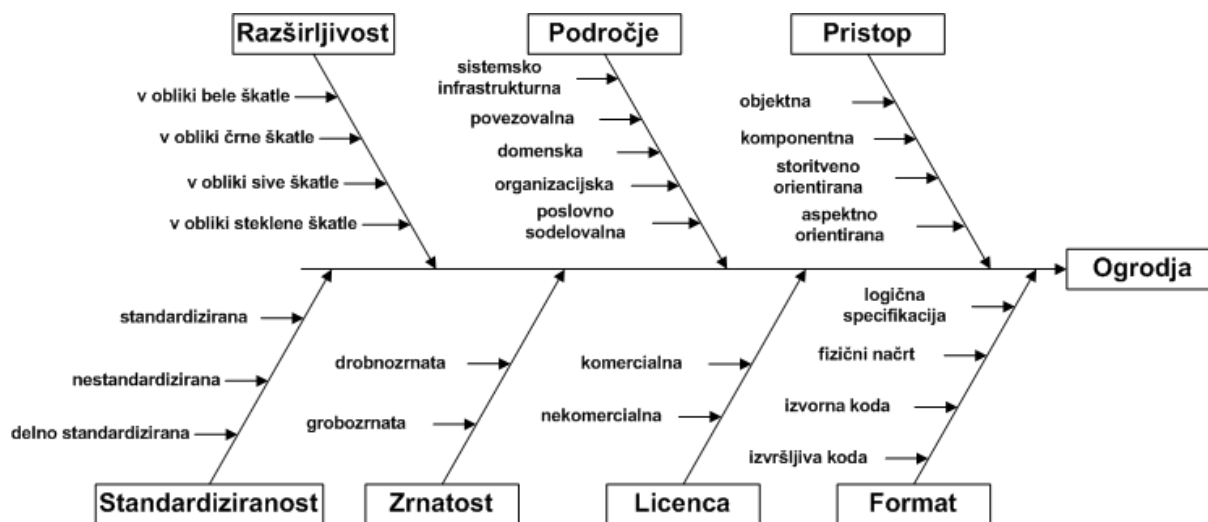
Poleg predstavljene delitve se ogrodja pogosto delijo glede na tip razvoja, in sicer na: (1) odprtokodna (*open source frameworks*), (2) lastniška (*proprietary frameworks*) in (3) ogrodja, ki so razvita za lastne potrebe (*in-house frameworks*). Glede na tehniko uporabe lahko ogrodja razvrstimo na ogrodja bele škatle in ogrodja črne škatle. Za ogrodja bele škatle je značilno, da se za doseganje razširljivosti močno opirajo na lastnosti objektivno orientiranih jezikov, kot sta

dedovanje ali povezovanje v času izvajanja (*dynamic binding*). Ogradja črne škatle podpirajo razširljivost skozi definicijo vmesnikov za komponente, ki jih lahko nato vključimo v ogradje z uporabo kompozicije objektov. Obstajajo še ogradja sive škatle in steklene škatle, ki predstavljajo vmesne rešitve (Slika 10).



SLIKA 10: VRSTE OGRODIJ GLEDE NA RAZŠIRLJIVOST

Na spodnji sliki (Slika 11) je prikazan model celovite klasifikacije ogradij, ki temelji na deskriptivni študiji predhodno izvedenih klasifikacij ogradij (Krajnc & Heričko 2003).

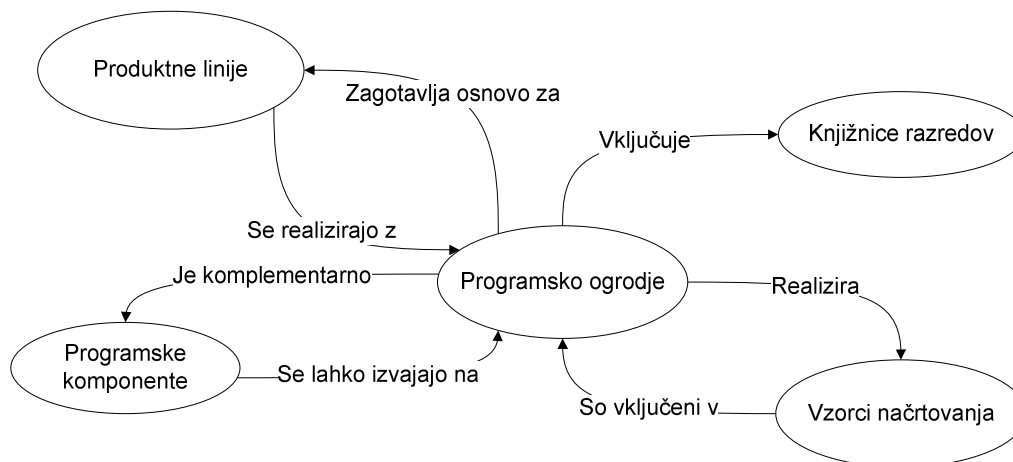


SLIKA 11: CELOVITA KLASIFIKACIJA OGRODIJ (KRAJNC & HERIČKO 2004)

Zgornji model klasifikacije ogradij deli ogradja glede na: način instanciranja, področje uporabe, pristop razvoja ogradij, standardiziranost ogradij, znatost ogradij, licenčni vidik ogradij in format zapisa ogradij. Značilnost modela celovite klasifikacije ogradij je, da so posamezne kategorije med sabo odvisne.

2.2.5 OGRODJEM SORODNI KONCEPTI IN TEHNOLOGIJE

Ker so se ogrodja razvila iz drugih, objektno orientiranih konceptov in tehnik ponovne uporabe, so ostala tesno povezana z njimi (Slika 12).



SLIKA 12: POVEZAVA OGRODIJ S SORODNIMI KONCEPTI IN TEHNOLOGIJAMI

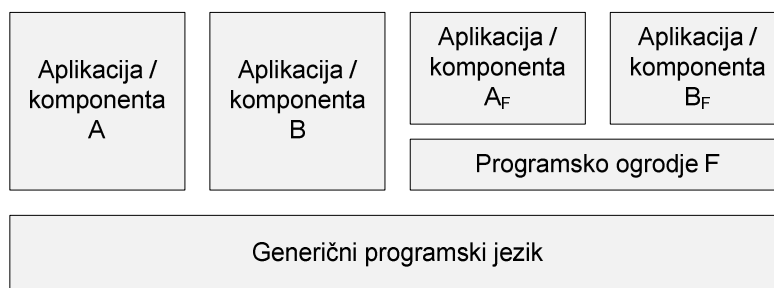
V nadaljevanju so izpostavljene podobnosti in razlike med ogrodji in sorodnimi tehnologijami oziroma koncepti.

2.2.5.1 PROGRAMSKE KOMPONENTE

Programska komponenta (v nadaljevanju komponenta) je del sistema, ki zagotavlja določeno storitev ali dogodek in je sposobna komunicirati z drugimi komponentami. Komponenta mora zadostiti naslednjim kriterijem (Messerschmitt & Szyperski 2003):

- večkratna uporaba,
- neodvisnost od konteksta,
- zmožnost sodelovanja z drugimi komponentami,
- skrite podrobnosti in dostopanje preko vmesnikov,
- neodvisnost od namestitve in verzioniranja.

Komponente zagotavljajo ponovno uporabo na nivoju implementacije (*code reuse*), medtem ko zagotavljajo ogrodja še ponovno uporabo načrtovanja (*design reuse*) in konceptov analize.

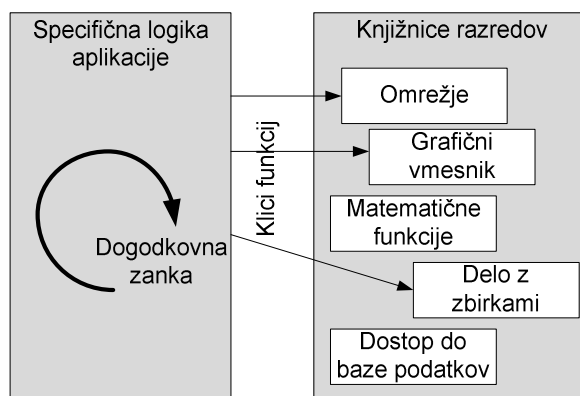


SLIKA 13: POVEZAVA MED KOMPONENTAMI IN OGRODJEM

Ogrodja in komponente sta v več pogledih komplementarni tehnologiji. Ogrodja pogosto zagotavljajo ponovno uporaben kontekst za komponente z zagotavljanjem storitev, kot so: izmenjava podatkov, obvladovanje izjem, beleženje dnevnikov in podobno. Druga možnost sodelovanja je, da lahko ogrodja zagotovijo osnovo za izdelovanje družine komponent (Slika 13) (Roberts & Johnson 1997).

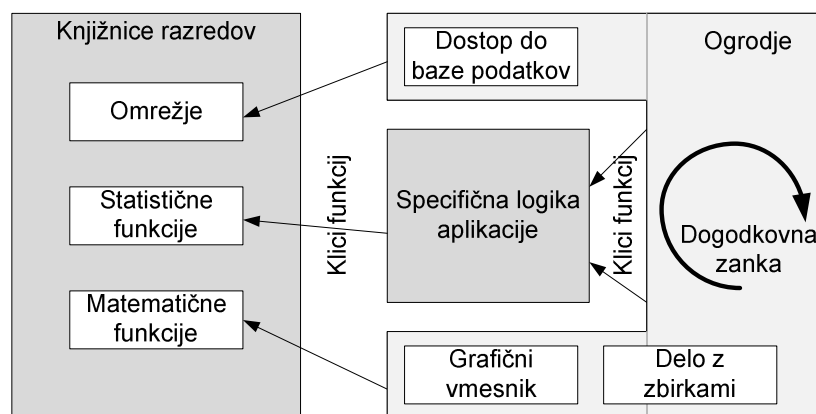
2.2.5.2 KNJIŽNICE RAZREDOV

Z vidika ogrodij ponujajo knjižnice razredov majhno zrnatost (*granularity*) in nižjo abstrakcijo ponovne uporabe. Kot prikazuje Slika 14, so knjižnice razredov nizkonivojske, relativno neodvisne in splošne komponente kot na primer: pripomočki za matematično in statistično obdelavo, delo z zbirkami, razredi za delo z omrežji, dostop do podatkovne baze. Podobno kot komponente zagotavljajo knjižnice razredov le ponovno uporabo na nivoju implementacije.



SLIKA 14: UPORABA KNJIŽNICE RAZREDOV

V nasprotju s knjižnicami razredov ogrodja definirajo delne rešitve, ki vključujejo domensko specifične objektne strukture in funkcionalnosti. Obseg ponovne uporabe ogrodij je zato večji kot pri uporabi knjižnic razredov (Morisio, Romano, & Stamelos 2002; Moser & Nierstrasz 1996).



SLIKA 15: RAZLIKA V UPORABI OGRODJA IN KNJIŽNIC RAZREDOV

Razlika med ogrodji in knjižnicami razredov je tudi v kontroli izvajanja aplikacij. Knjižnice razredov so "pasivne", kar pomeni, da nimajo glavne kontrole nad tokovi izvajanja aplikacije, temveč izvajanje programa kontrolira koda aplikacije (Slika 14). Ogradja so "aktivna", kar pomeni, da se pri izvajanju aplikacije kontrola izvajanja prenese na ogrodje, ki potem po potrebi kliče programsko kodo aplikacij. Takšna arhitektura temelji na povratnih klicih (Slika 15) in se imenuje obrat kontrole (*inversion of control*) ali "Hollywoodsko načelo"³².

Ogradja in knjižnice razredov sta lahko komplementarni tehnologiji. Ogradja pri izvajanju programske kode uporabljajo knjižnice razredov. Uporaba je lahko interna (kot del ogrodja) ali eksterna (preko povratnih, aplikacijsko specifičnih klicev) (Krajnc 2006).

2.2.5.3 VZORCI NAČRTOVANJA

V programskem inženirstvu so vzorci načrtovanja definirani kot splošne ponovno uporabne rešitve za pogoste probleme, ki se pojavljajo v fazi načrtovanja programske opreme³³. Objektno orientirani vzorci načrtovanja najpogosteje prikazujejo povezave in sodelovanje med splošnimi razredi in objekti. Vzorci načrtovanja se od algoritmov razlikujejo po tem, da rešujejo načrtovalske probleme in ne računskih problemov.

Ogradja se od vzorcev načrtovanja razlikujejo predvsem po tem, da zagotavljajo še ponovno uporabo na nivoju implementacije (*code reuse*). Vzorcev načrtovanja ne moremo izraziti kot razrede v objektnih programskih jezikih in jih ponovno uporabiti z uporabo dedovanja oziroma kompozicije. Vzorci so torej bolj abstraktni kot ogrodja (Krajnc 2006).

Ogradja in vzorci sta komplementarni tehniki. V ogrodju je pogosto uporabljenih več vzorcev (glej tudi Slika 1), na primer, v ogrodju MVC zasledimo uporabo treh načrtovalskih vzorcev: vzorec Opazovalec (*Observer*), vzorec Kompozicija (*Composite*) in vzorec Strategija (*Strategy*). Na

³² »Ne kličite nas, mi bomo poklicali vas.« (*Hollywood principle*).

³³ Povzeto po: [http://en.wikipedia.org/wiki/Design_pattern_\(computer_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science)).

vzorci načrtovanja lahko gledamo kot na mikroarhitekturne elemente ogrodij, saj predstavljajo rešitev, ogrodja pa konkretno implementacijo (Johnson 1997).

2.2.5.4 PRODUKTNE LINIJE

Produktne linije so definirane naslednje (Bosch 2000):

»Produktna linija programske opreme je množica sistemov programske opreme, ki si delijo skupno upravljano množico funkcionalnosti, ki zadovoljujejo specifične potrebe določenega tržnega segmenta ali poslanstva in so razvite iz skupne množice osnovnih pridobitev na predpisan način.«

in kot (Northrop 1999):

»Produktna linija programske opreme je množica produktov, ki si delijo skupno arhitekturo programske opreme in množico ponovno uporabnih komponent.«

Razvoj na osnovi produktnih linij vključuje voden in nadzorovan razvoj množice ponovno uporabnih izdelkov (*core asset development*) in razvoj množice končnih izdelkov (*product development*) (Slika 16). Pri tem vrstni red razvoja izdelkov ni določen³⁴. Produktne linije naj bi zagotavljale dolgoročne ekonomske prednosti glede na razvoj posameznih izdelkov oziroma glede na nesistematično ponovno uporabo (Amar & Coffey 2005; Bockle, Clements, McGregor, Muthig, & Schmid 2004).



SLIKA 16: OSNOVNE AKTIVNOSTI PRODUKTNIH LINIJ (NORTHROP 1999)

³⁴ Ponovno uporabni izdelki lahko nastajajo na osnovi končnih izdelkov in obratno.

Produktne linije in ogrodja so med sabo povezani, saj se skupna osnova produktnih linij (ponovno uporabni izdelki) najpogosteje realizira z uporabo ogrodja (Batory, Cardone, & Smaragdakis 2000; Morisio, Romano, & Stamelos 2002; Philippow & Riebisch 2001). Pojem produktnih linij je torej širši od pojma ogrodij.

2.2.5.5 DRUGI SORODNI KONCEPTI IN TEHNOLOGIJE

Ogrodja se pogosto primerjajo še z domensko specifičnimi jeziki in generatorji aplikacij. Domensko specifični jeziki (*domain-specific language*) ali krajše DSL so programski jeziki, ki so namenjeni za uporabo v določeni domeni. DSL, kot je, na primer, skriptni jezik lupine Unix, predstavljajo nasprotje generičnim programskim jezikom, kot sta, na primer, C++ in Java. Ogrodja in DSL imajo podoben cilj - dosežati boljše rezultate pri razvoju programske opreme v določeni domeni. Kljub temu se ogrodja in DSL z vidika zasnove in tudi njunih lastnosti močno razlikujejo (Deursen 1997).

Ogrodjem so sorodni tudi generatorji aplikacij (*application generator*). Generatorji aplikacij so programska oprema, ki generira aplikacije (delno ali v celoti) na osnovi opisa problema. Generatorji aplikacij običajno temeljijo na visokonivojskem domensko specifičnem jeziku (Roberts & Johnson 1997) medtem ko temeljijo ogrodja na generičnih programskih jezikih.

2.3 UČINKOVITOST INFORMACIJSKIH TEHNOLOGIJ

Osnovni cilj uporabe informacijskih tehnologij (IT) v organizacijah je povečati učinkovitost informacijskih sistemov³⁵ (IS). Ker povečanja učinkovitosti oziroma uspešnosti³⁶ IS ni mogoče določiti brez merjenja, je to osrednje področje v raziskavah in praksi na področju IS (Seddon, Staples, Patnayakuni, & Bowtell 1999). Pri tem je uspešnost IS definirana naslednje³⁷:

Uspešnost IS je zunanja značilnost IS, ki jo merimo po rezultatih uporabe in po posledicah, ki jih IS povzroči v okolju.

Že od začetkov je merjenje učinkovitosti IS povezano s številnimi omejitvami. V literaturi je možno zaslediti veliko raziskav, ki so vrednotile učinkovitost IS in analizirale faktorje, ki vplivajo na učinkovitost IS. Prva težava nastopi pri primerjavi omenjenih raziskav, saj so različni raziskovalci različno definirali učinkovitost IS. Druga težava je povezana z načinom merjenja učinkovitosti IS. Raziskovalci so poskušali določati učinkovitost IS izključno z objektivnimi in kvantitativnimi metrikami, kot sta, na primer: obseg zmanjšanih stroškov ali nivo tehnološke odličnosti. Pokazalo se je, da imajo takšne metrike pomanjkljivosti, kot sta: (1) težavna povezava metrik z dejanskim objektom merjenja (vpeljavo novega IS) ali (2) praktično neizvedljivo kvantitativno vrednotenje. Omenjene metrike so v večini primerov vrednotili eksperti (*expert judgement*), zato dobljeni rezultati niso bolj objektivni od kvalitativnih metrik. Iz navedenih pomanjkljivosti objektivnih metrik so se merjenju učinkovitosti IS sčasoma dodale še subjektivne metrike (Blake, Margrethe, & Jack 1983).

Več urejenosti na področje merjenja učinkovitosti IS je prinesla študija DeLona in McLeana (DeLone & Mclean 1992). Avtorja sta na osnovi analize množice znanstvenih objav³⁸ oblikovala taksonomijo metrik učinkovitosti IS. Pri tem sta jedro taksonomije vzpostavila na teoriji informacij, ki sta jo leta 1949 objavila Shannon in Weaver (Shannon & Weaver 1949). Po omenjeni teoriji se lahko informacija, ki predstavlja izhod IS, meri na tehničnem nivoju³⁹, semantičnem nivoju⁴⁰ in nivoju učinkovitosti ali vpliva⁴¹. Nivo učinkovitosti je še dodatno razčlenil Mason (1978), in sicer na: prejemnika, vpliv na prejemnika in vpliv na sistem.

³⁵ V islovar.org je informacijski sistem definiran kot »sistem, ki omogoča zbiranje, obdelavo, shranjevanje, distribucijo ter uporabo podatkov in informacij«.

³⁶ Uspešnost in učinkovitost sta v disertaciji obravnavani kot sinonima, kar je enakovredno Seddonovi raziskavi (Seddon, Staples, Patnayakuni, & Bowtell 1999).

³⁷ Prevod povzet po www.islovar.org.

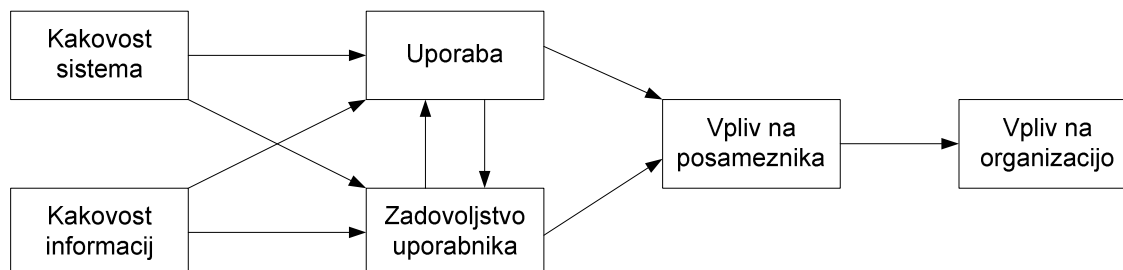
³⁸ Avtorja sta analizirala 180 objav v znanstvenih publikacijah in na znanstvenih konferencah, ki so bile objavljene med letoma 1981 in 1987. V raziskavo sta vključila empirične raziskave, katerih cilj je bil merjene določenega vidika učinkovitosti IS.

³⁹ Tehnični nivo sta avtorja definirala kot »natančnost in učinkovitost sistema, ki proizvaja informacije«.

⁴⁰ Semantični nivo sta avtorja definirala kot »uspešnost informacije, da prenese namenjen pomen«.

 2.3.1 D&M MODEL USPEŠNOSTI IS

Pomembnejši od DeLonove in McLeanove taksonomije metrik učinkovitosti IS (Tabela 1) je konceptualni model uspešnosti IS (t. i. D&M model uspešnosti IS ali krajše D&M) (Slika 17).



SLIKA 17: D&M MODEL USPEŠNOSTI IS (DELONE & MCLEAN 1992)

Model D&M temelji na Masonovi ugotovitvi, da uspešnosti IS ni mogoče meriti z eno dimenzijo, saj informacija v svoji življenjski dobi prepotuje naslednje faze: produkcijo, uporabo in vpliv informacije na posameznike in organizacije. Na tej ugotovitvi temeljijo dimenzije D&M modela uspešnosti IS in DeLonova in McLeanova ugotovitev, da so prav različni možni vidiki merjenja uspešnosti IS razlog za številne metrike, ki jih je možno zaslediti v literaturi.

Avtorja sta v model vključila še subjektivno metriko »zadovoljstvo uporabnika«, ki je definirana kot (Blake, Margrethe, & Jack 1983):

Zadovoljstvo uporabnika predstavlja obseg uporabnikovih spoznanj, da informacijski sistem zadovoljuje njegove informacijske zahteve.

V nadaljevanju (Tabela 1) je podana (delna) klasifikacija metrik uspešnosti, ki sta jih avtorja identificirala v obstoječih virih in jih nato klasificirala v dimenzije uspešnosti.

TABELA 1: PREGLED METRIK USPEŠNOSTI IS (DELONE & MCLEAN 1992)

Kakovost sistema	Kakovost informacij	Uporaba (informacij)	Zadovoljstvo uporabnika	Vpliv na posameznika	Vpliv na organizacijo
Natančnost podatkov	Pomembnost	Obseg uporabe	Zadovoljstvo s specifikami	Razumevanje informacij	Portfelj aplikacij
Enostavnost uporabe	Relevantnost	Število uporabljenih funkcij	Splošno zadovoljstvo	Učenje	Doseg in obseg aplikacij
Enostavnost	Uporabnost	Število	Enotočkovno	Natančnost	Število

⁴¹ Nivo učinkovitosti sta avtorja definirala kot »vpliv informacije na sprejemnika«.

Kakovost sistema	Kakovost informacij	Uporaba (informacij)	Zadovoljstvo uporabnika	Vpliv na posameznika	Vpliv na organizacijo
učenja	informacij	dostopnih zapisov	merilo	interpretacije	kritičnih aplikacij
Človeški faktorji	Razumljivost	Pogostost dostopanja	Večtočkovno merilo	Informacijska zavest	Zmanjšanje stroškov izvajanja
Realizacija uporabniških zahtev	Berljivost	Pogostost dostopanja do poročil	Zadovoljstvo z informacijami	Ponovni priklic informacij	Zmanjšanje števila zaposlenih
Uporabnost funkcij sistema	Jasnost	Število generiranih poročil	Razlika med zahtevanimi in prejetimi informacijami	Identifikacija problema	Celotno izboljšanje produktivnosti
Natančnost sistema	Natančnost vsebine	Število povpraševanj	Zadovoljstvo	Učinkovitost odločitev	Povečanje dohodkov
Fleksibilnost sistema	Konsistentnost	Trajanje povezanega časa	Zadovoljstvo s programsko opremo	Kakovost odločitev	Povečanje tržnega deleža
Zanesljivost sistema	Zanesljivost	Uporaba (binarno)	Zadovoljstvo s sprejetimi odločitvami	Izboljšanje analize odločitev	Povečanje dobičkov
Zmožnost integracije sistema	Unikatnost	Dejanska/poročana uporaba		Pravilnost odločitev	Povračilo investicije
Koriščenje virov	Skladnost	Primerna uporaba		Čas do izvedbe odločitve	Razmerje med stroški in prihodki
Odzivni čas	Kvantitativnost	Tip uporabljenih informacij		Zaupanje v odločitev	Kakovost izdelkov
Obračalni čas	Zgoščenost	Sprejetost poročil		Izboljšanje posameznikove učinkovitosti	Učinkovitost storitev

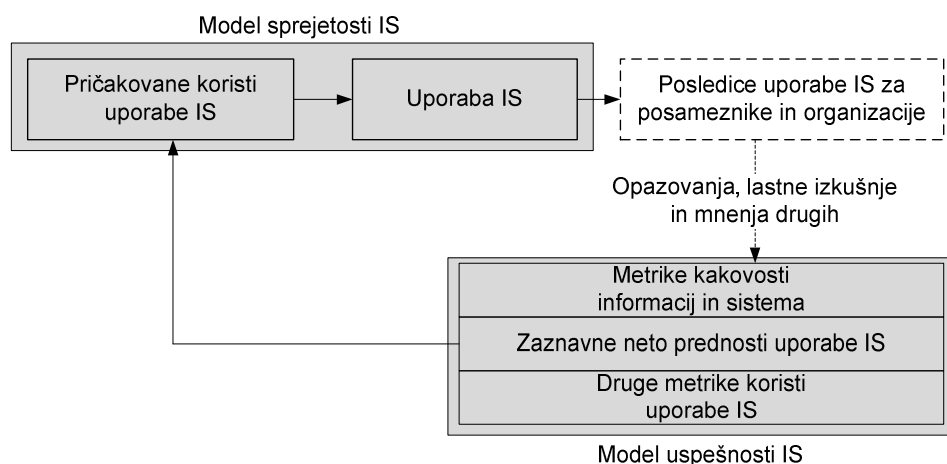
D&M model uspešnosti IS je postal široko uveljavljen, saj ga je od objave v letu 1992 do leta 2002 citiralo preko 150 znanstvenih objav (DeLone & Mclean 2002). Kljub temu so avtorja modela D&M in drugi raziskovalci identificirali pomanjkljivosti modela, ki so predstavljene v Seddonovem modelu uspešnosti IS.

2.3.2 SEDDONOV MODEL USPEŠNOSTI IS

Seddon je analiziral model D&M in pri tem identificiral naslednje pomanjkljivosti modela (Seddon 1997):

- **Različne interpretacije povezav.** Ugotovil je, da si raziskovalci interpretirajo povezave v modelu D&M na dva načina: (1) kot povezave v procesnem modelu⁴² in (2) kot povezave v kavzalnem modelu⁴³. Ker imajo povezave v procesnem in kavzalnem modelu različne pomene, jih ni mogoče poenotiti oziroma združiti v enem modelu.
- **Različne interpretacije koncepta »uporaba«.** Ugotovil je, da je koncept »uporaba« v modelu D&M nejasno definiran, zato so si ga raziskovalci interpretirali na tri načine: (1) uporaba kot nadomestni faktor za prednosti uporabe, (2) uporaba kot odvisna spremenljivka v kavzalnem modelu D&M in (3) uporaba kot dogodek v procesnem vidiku modela D&M.

Seddon je pomanjkljivosti modela D&M odpravil s preoblikovanjem modela in definiranjem konceptov modela, kot je prikazano na spodnji sliki (Slika 18).



SLIKA 18: SEDDONOV MODEL USPEŠNOSTI IS (SEDDON 1997)

Seddonov model uspešnosti uporabe IS (v nadaljevanju Seddonov model) pojasnjuje dinamiko uspešnosti uporabe programske opreme z dvema podmodeloma: parcialnim modelom sprejetosti IS in modelom uspešnosti IS. Iz kavzalne interpretacije Seddonovega modela (neprekinjene puščice) je možno razbrati, da je uspešnost IS možno zagotoviti le, če so uporabniki zadovoljni s posledicami njegove uporabe. Pri tem se je pokazalo, da je povezava med uporabo IS in uspešnostjo IS posredna (bistvena razlika glede na model D&M), saj ni mogoče posplošiti sklepanja, da več uporabe pomeni več prednosti uporabe; pomeni le več

⁴² V procesnem modelu pomenijo povezave zaporedje izvajanja aktivnosti.

⁴³ V kavzalnem modelu pomenijo povezave vpliv neodvisne spremenljivke na odvisno spremenljivko.

posledic uporabe (prekinjena puščica). Seddon et al. (Seddon, Staples, Patnayakuni, & Bowtell 1999) trdijo, da je omenjena povezava (prekinjena puščica) odvisna predvsem od tipa programske opreme in tipa uporabnika⁴⁴. Iz enakih razlogov Seddonov model uspešnosti ne predpisuje konkretnih metrik uspešnosti, temveč le njene poglavitne dimenzije, ki so naslednje: (1) metrike kakovosti informacij in sistema, (2) zaznavne neto prednosti uporabe IS (dojeta uporabnost, zadovoljstvo uporabnikov) in (3) druge metrike koristi uporabe IS. Seddon podaja še natančne definicije konceptov njegovega modela. Osrednje mesto med definicijami ima IS, ki je definiran naslednje:

Informacijski sistem predstavlja aspekt aplikacije informacijske tehnologije (IT), individualno aplikacijo, skupino aplikacij ali aplikacijo določene vrste IT.

Na osnovi navedene definicije lahko sklepamo, da Seddonov model ni namenjen le vrednotenju informacijskih sistemov⁴⁵, temveč tudi vrednotenju posameznih aplikacij⁴⁶ v fazi njihove uporabe. Uspešnost IS je definirana naslednje:

Uspešnost IS je mera, s katero oseba, ki vrednoti IS, ugotavlja stopnjo prednosti, ki izhajajo iz uporabe IS, za osebo, za katero se izvaja vrednotenje.

Če bi se neto prednosti lahko vrednotile z visoko natančnostjo, bi bile neto prednosti enake uspešnosti IS. Neto prednosti so definirane naslednje:

Neto prednosti predstavljajo idealno metriko, sestavljeno iz vsote vseh preteklih in prihodnjih prednosti, ki izhajajo iz uporabe IS, zmanjšanih za vsoto vseh preteklih in prihodnjih stroškov, ki izhajajo iz uporabe IS. Pri tem predstavlja stroške vsaka uporaba virov, povezanih z razvojem, učenjem in uporabo sistema.

⁴⁴ Na primer: večanje uporabe elektronske pošte zmanjšuje stroške komunikacije, medtem ko avtomatizacija določenega poslovnega procesa (zmanjšanje uporabe z vidika uporabnika) poveča učinkovitost poslovanja.

⁴⁵ Informacijski sistem je urejen in organiziran sistem, ki uporabnike oskrbuje z vsemi potrebnimi informacijami za odločanje. Osnovne aktivnosti informacijskega sistema so: zbiranje, shranjevanje, obdelava in posredovanje rezultatov končnim uporabnikom.

⁴⁶ Aplikacija je računalniški program, namenjen uporabniku za opravljanje določenih nalog (www.islovar.org).

Iz Seddonovega modela (Slika 18) je še razvidno, da se o izbiri in posledično uporabi IS odločamo predvsem na osnovi pričakovanih koristi, ki izhajajo iz uporabe določenega IS, kar se sklada z miselnostjo modelov sprejetosti (*user acceptance*), predstavljenih v nadaljevanju.

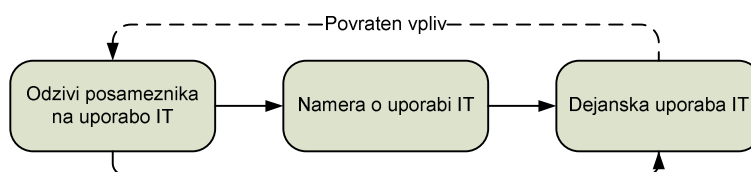
2.4 SPREJETOST INFORMACIJSKIH TEHNOLOGIJ

Uspešnost IS je tesno povezana s sprejetostjo (*user acceptance*) in uporabo IT, kar so potrdile številne raziskave (DeLone & Mclean 1992; Garrity et al. 2005; Venkatesh, Morris, Davis, & Davis 2003). Dillon in Morris definirata pojem uporabniške sprejetosti naslednje (Dillon & Morris 1996):

Uporabniška sprejetost je nazorna pripravljenost (skupine uporabnikov) uporabiti informacijsko tehnologijo za izvedbo opravila, za katero je ta namenjena.

Sprejetost in uporaba IT sta ključnega pomena za organizacije, saj sta prvi pogoj za izboljšanje produktivnosti v organizacijah (Venkatesh, Morris, Davis, & Davis 2003). Prav zaradi navedenega spadajo raziskave na področju sprejetosti med najbolj zrela področja IS. Raziskave na področju sprejetosti temeljijo na teorijah in modelih s področja informacijskih sistemov, psihologije in sociologije. Rezultati raziskav so kavzalni modeli, ki konceptualizirajo in operacionalizirajo uporabo IT. Modeli so namenjeni predvsem razumevanju in napovedovanju uporabe IT. Posledično je uporaba takšnih modelov ključnega pomena za ocenitev verjetnosti uporabe, kakor tudi za proaktivno delovanje v smislu izboljšanja sprejetosti (*acceptance*), uporabe in uspešnosti IT (Venkatesh, Morris, Davis, & Davis 2003).

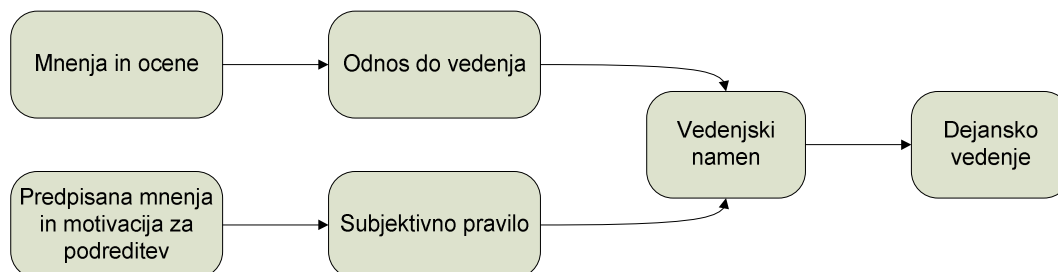
Obstaja več smeri raziskav na področju sprejetosti IT (Venkatesh, Morris, Davis, & Davis 2003): (1) posameznikova sprejetost IT, kjer predstavljata namera o uporabi ali dejanska uporaba odvisno spremenljivko, (2) implementacija uspešnosti na nivoju organizacije in (3) skladanje zahtev in tehnologije (*task-technology-fit*). Večina raziskav na področju sprejetosti se osredotoča na posameznikovo sprejetost in uporabo IT. V takšnih modelih sta sprejetost ali uporaba predstavljeni kot odvisni spremenljivki (Slika 19).



SLIKA 19: OSNOVNA FILOZOFIJA MODELOV SPREJETOSTI (VENKATESH, MORRIS, DAVIS, & DAVIS 2003)

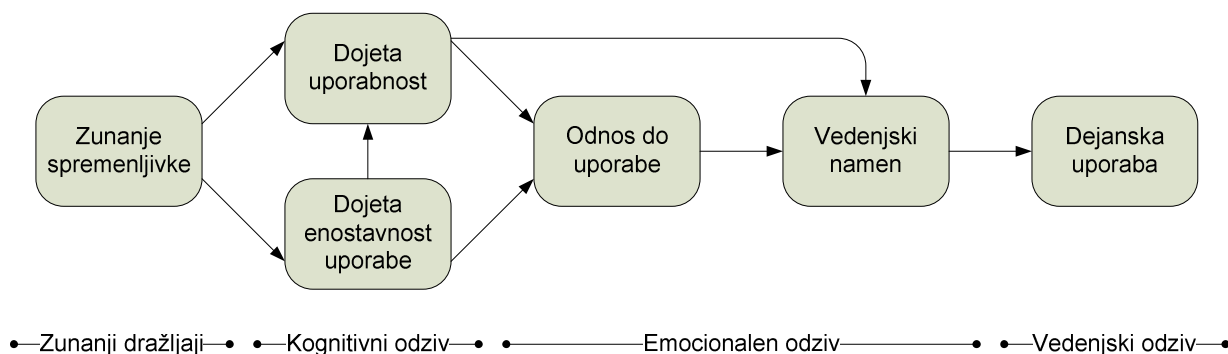
2.4.1 MODEL SPREJETOSTI IT - TAM

Model TAM (*Technology Acceptance Model*) je največkrat citiran model sprejetosti IT (Sharp 2006). TAM temelji na Ajzenovi in Fishbeinovi teoriji utemeljene akcije – TRA (*theory of reasoned action*) (Fishbein & Ajzen 1975) in jo prilagaja za kontekst IT.



SLIKA 20: KONCEPTUALNI MODEL TEORIJE UTEMELJENE AKCIJE (FISHBEIN & AJZEN 1975)

TAM je informacijskosistemska teorija, ki razlaga, kako uporabniki sprejemajo in uporabljajo IT. Teorija pravi, da na uporabnike, ki spoznajo novo IT, vplivata dva poglobitna dejavnika: dojeta enostavnost uporabe – PEOU (*perceived ease of use*) in dojeta uporabnost - PU (*perceived usefulness*) IT. Vse ostale spremenljivke (na primer: karakteristike sistema in karakteristike posameznika) so obravnavane kot zunanje spremenljivke z neposrednim vplivom na zgoraj predstavljena kognitivna konstrukta modela TAM (Slika 21).



SLIKA 21: IZVIREN MODEL TAM (DAVIS 1989)

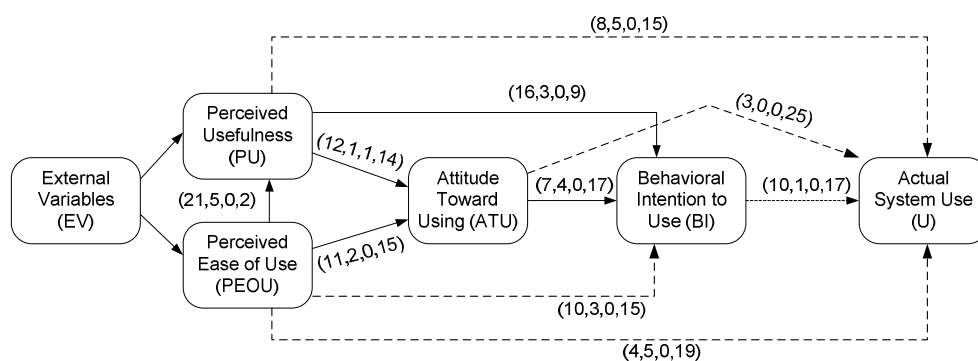
Dojeta uporabnost (*Perceived Usefulness - PU*) je definirana naslednje (Davis 1989):

Dojeta uporabnost je stopnja, do katere posameznik verjame, da bo uporaba določenega sistema izboljšala njegovo delovno storilnost.

Dojeta enostavnost uporabe (*Perceived Ease Of Use - PEOU*) je definirana naslednje (Davis 1989):

Dojeta enostavnost uporabe je stopnja, do katere posameznik verjame, da bo uporaba določenega sistema enostavna oziroma njegova uporaba ne bo zahtevala dodatnih naporov.

Na spodnji sliki (Slika 22) je predstavljen model TAM s konstrukti v izvirnem, angleškem jeziku. Neprekinjene puščice ponazarjajo kavzalne povezave izvirnega modela TAM, prekinjene puščice pa kavzalne povezave, ki so bile testirane v nadaljnjih študijah.



SLIKA 22: TESTIRANJA IZVIRNEGA MODELA TAM

Številčne vrednosti na povezavah modela predstavljajo rezultate metaanalize modela TAM, ki so jo izvedli (Legris, Ingham, & Colletette 2003). Številčne vrednosti (\vec{x}) imajo naslednji pomen: (x_1) število odkritih signifikantnih pozitivnih relacij, (x_2) število odkritih nesignifikantnih pozitivnih relacij, (x_3) število odkritih signifikantnih negativnih relacij in (x_4) število relacij, ki v študijah niso bile obravnavane.

Empirična testiranja izvirnega modela TAM so pokazala signifikanten vpliv PEOU in PU na »vedenjski namen«, pri čemer se je pokazalo, da ima PU večji vpliv kakor PEOU. Kot drugo so študije ugotovile, da »odnos do uporabe« le delno posreduje vplive med uporabnikovimi dojemami in namero o uporabi. Na osnovi teh ugotovitev sta se razvili dve izpeljanki modela TAM: poenostavljeni TAM in TAM2.

- Poenostavljeni TAM (*parsimonious TAM*). Ta izpeljanka modela TAM ne vključuje konstrukta »odnos do uporabe«, predvsem iz razloga nekonsistentnih rezultatov predhodnih raziskav, kjer je bil konstrukt uporabljen.
- TAM2 (Venkatesh & Davis 2000). Ta izpeljanka modela TAM temelji na konsistentnih rezultatih predhodnih študij, ki so potrdili, da je PU poglavitna determinanta »vedenjskega namena«. Na osnovi teh ugotovitev je bil osnovni cilj oblikovanja TAM2 poiskati konstrukte, ki vplivajo na PU.

Empirične raziskave so potrdile veljavnost modela TAM na različnih vrstah IT (Lederer, Maupin, Sena, & Zhuang 2000). Model TAM se je izkazal za uspešnega predvsem z vidika določanja namere o uporabi, kjer se lahko izkaže s povprečno 45% pojasnjene variance (R^2).

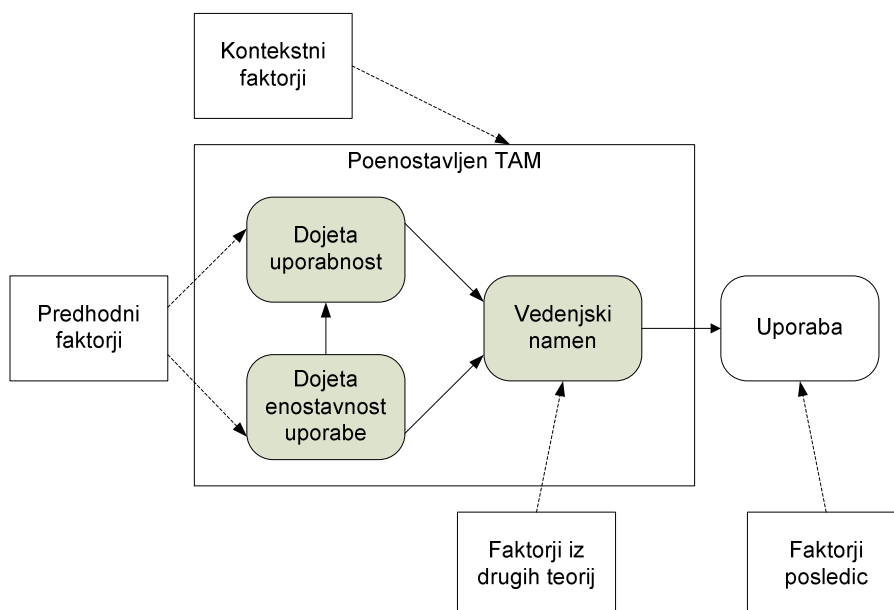
Zaradi obsežnosti uporabe modela TAM v znanstvenih raziskavah⁴⁷ so se identificirale številne pomanjkljivosti modela TAM. Avtorji Speier et al. (2005) ugotavljajo, da predstavlja osnovno pomanjkljivost modela TAM njegova široka uporabnost. Po njihovem prepričanju je pojem IT preširok za TAM, zato so tudi rezultati modela nekonsistentni. Speier et al. (2005) zato predlagajo, da se za potrebe modelov sprejetosti tehnologija kategorizira glede na: (1) individualno ali organizacijsko uporabo, (2) obvezujočo ali prostovoljno uporabo, in (3) radikalno ali postopno vpeljavo nove tehnologije.

Ostale pogosto izpostavljene pomanjkljivosti modela TAM so naslednje (Legris, Ingham, & Colletette 2003):

- **Vzorci iz študentov.** Večina raziskav modela TAM temelji na anketiranih študentov, kar zmanjšuje zunanjo veljavnost rezultatov.
- **Vrste aplikacij.** Obstoječe raziskave TAM v večji meri preučujejo preprostejše IT rešitve, kot so pisarniške ali preproste spletne aplikacije (Hong et al. 2001).
- **Samoocenitev uporabe.** Večina obstoječih študij meri uporabo s samoocenitvijo, kar lahko predstavlja kvečjemu relativni indikator uporabe.

Zaradi pomanjkljivosti modela TAM, so se pojavile številne raziskave, ki so model TAM prilagodile ali razširile za potrebe specifičnega predmeta raziskave (King & He 2006).

⁴⁷ Povpraševanje po bazi virov Science Direct (<http://www.sciencedirect.com/>) je za področje »Computer Science« dne 16. 4. 2008 vrnilo 144 objav, ki vključujejo (v naslovu, ključnih besedah ali povzetku) besedno zvezo »Technology Acceptance Model«, in 79 objav, ki vključujejo akronim TAM.



SLIKA 23: KATEGORIJE RAZŠIRITEV MODELA TAM (KING & HE 2006)

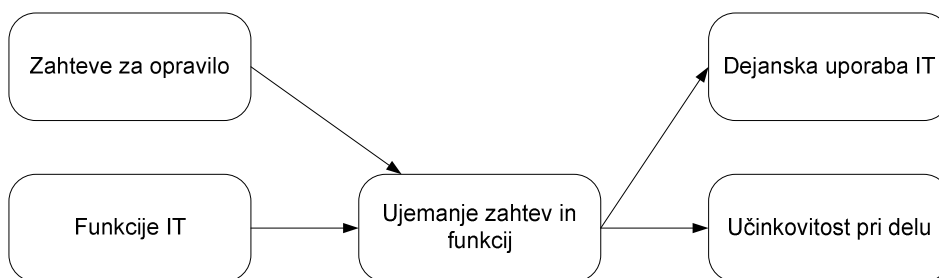
Na zgornji sliki (Slika 23) so ponazorjene štiri pogoste razširitve modela TAM (King & He 2006):

- Vključevanje predhodnikov modela TAM, kot so: karakteristike sistema (*system characteristics*) in posebnosti posameznika (*individual differences*) (Hong, Thong, Wong, & Tam 2001).
- Vključevanje faktorjev, ki so povzeti iz drugih teorij, s katerimi se poskuša povečati moč napovedovanja (*predictive power*) modela TAM. Primeri te vrste faktorjev so: subjektivne norme (*subjective norm*), skladanje zahtev s funkcijami (*task-technology fit*), tveganje (*risk*) in zaupanje (*trust*).
- Vključevanje kontekstnih faktorjev, za katere se predpostavlja, da imajo posreden vpliv (*moderator effect*) na kavzalne povezave modela. Primeri te vrste faktorjev so: spol, kultura in starost.
- Vključevanje posledičnih faktorjev, kot sta dojeta uporaba ali dejanska uporaba.

Osnovni cilj raziskav, ki razširjajo model TAM, je pokazati, da je razširjeni model v določenem kontekstu natančnejši in ustrežnejši od modela TAM.

2.4.2 MODEL UJEMANJA ZAHTEV IN LASTNOSTI IT - TTF

Druga pomembna smer raziskav na področju določanja sprejetosti oziroma uporabe IT se osredotoča na odločitev o uporabi določene programske opreme, ki je posledica ujemanja uporabnikovih zahtev in funkcij programske opreme. Ta smer raziskav je konceptualizirana v modelu TTF (*task-technology fit model*) (Slika 24) (Goodhue & Thompson 1995). Model TTF pojasnjuje uporabo IT z ujemanjem funkcionalnosti IT in zahtev za izvedbo zahtevanih opravil. TTF temelji na predpostavki, da bodo uporabniki izbrali IT, ki jim bo pri izvajanju opravil zagotovila največji obseg prednosti (*net benefits*). Goodhue in Thompson (1995) sta poleg tega ugotovila signifikanten vpliv koncepta TTF na učinkovitost pri delu, ki uporablja vrednoten sistem.



SLIKA 24: MODEL TTF (GOODHUE & THOMPSON 1995)

Metriko ujemanje zahtev in funkcij (TTF) sta Goodhue in Thompson (1995) konceptualizirala z osmimi faktorji: kakovost (*quality*), zmožnost lokalizacije (*localability*), avtorizacija (*authorization*), skladnost (*compatibility*), enostavnost uporabe (*ease of use*), produkcijski čas (*production timeline*), zanesljivost sistema (*system reliability*) in povezanost z uporabniki (*relationship with users*). Vsak faktor se vrednoti na Likertovi lestvici (*Likert scale*) z dvema do osmimi likertovimi elementi (vprašanji).

Model TTF je primarno namenjen individualni uporabi, vendar sta ga Zigurs in Buckland (1998) preoblikovala v model, ki je uporaben tudi na organizacijskem nivoju.

V primerjavi z modelom TAM je model TTF manj zrel, kar se kaže predvsem v nestabilni operacionalizaciji modela. Prav tako se je pokazalo, da so konstrukti modelov TAM in TTF v veliki meri komplementarni (Dishaw & Strong 1999; Klopping & McKinney 2004).

2.5 KAKOVOST IN PONOVA UPORABNOST PROGRAMSKE OPREME

Modeli sprejetosti, kot je na primer TAM, ne določajo lastnosti IT, ki vplivajo na njihovo sprejetost, saj so te odvisne od številnih dejavnikov, med katerimi se najpogosteje izpostavljata vrsta IT in tip uporabnika.

V domeni programskega inženirstva so lastnosti, ki naj bi jih imel programski izdelek, opredeljene s kakovostjo programske opreme, ki je definirana naslednje (Kitchenham & Pfleeger 1996):

Kakovost so vse karakteristike entitete, ki uporabniku zagotavljajo zadostitev eksplicitno navedenih in implicitnih zahtev.⁴⁸

Kakovost programske opreme torej predstavlja kompozit vseh lastnosti programske opreme, ki ovrednotijo njeno kakovost in vplivajo nanjo. Zato je kakovost kompleksen in večplasten koncept, ki ga lahko opazujemo z različnih vidikov (McCall, Richards, & Walters 1977):

- **Transcendentni vidik** (*transcendental view*). Transcendentni vidik kakovosti je nerazgradljiv pogled na celoto, ki temelji na izkušnjah in zaznavanju. Kakovost v takšnem vidiku se lahko zazna, vendar je ni možno definirati ali izmeriti.
- **Uporabniški vidik** (*user view*). Predstavlja ustrežanje programskega izdelka uporabniku. Vidik je usmerjen na uporabnikova opravila in ga je možno delno izmeriti z uporabo proizvoda. V standardu ISO/IEC 9126 je takšen vidik kakovosti opredeljen kot »kakovost v uporabi« (*quality in use*).
- **Proizvodni vidik** (*manufacturing view*). Predstavlja stopnjo, do katere programski izdelek ustreza formalnim zahtevam oziroma specifikacijam. Kakovost se definira, na primer, s štetjem napak in evalvacijo stroškov odprave napak. V standardu ISO/IEC 9126 je takšen vidik kakovosti opredeljen kot »zunanja kakovost« (*external quality*). Proizvodni vidik kakovosti je odvisen od zrelosti delovnih procesov organizacije, ki razvija programsko opremo. Zrelost delovnih procesov podrobneje opredeljujejo standardi, kot je na primer zrelostno-zmožnostni model ali krajše ZZM (SEI 2002). ZZM predstavlja pot izboljšav in priporočil za organizacije, ki razvijajo programsko opremo in želijo povečati zmožnost svojih procesov. Modelu ZZM je podoben ITIL (*Information Technology Infrastructure Library*)⁴⁹, ki predstavlja množico konceptov in tehnik za obvladovanje IT infrastrukture, razvoja in storitev. Procesi organizacije se lahko na

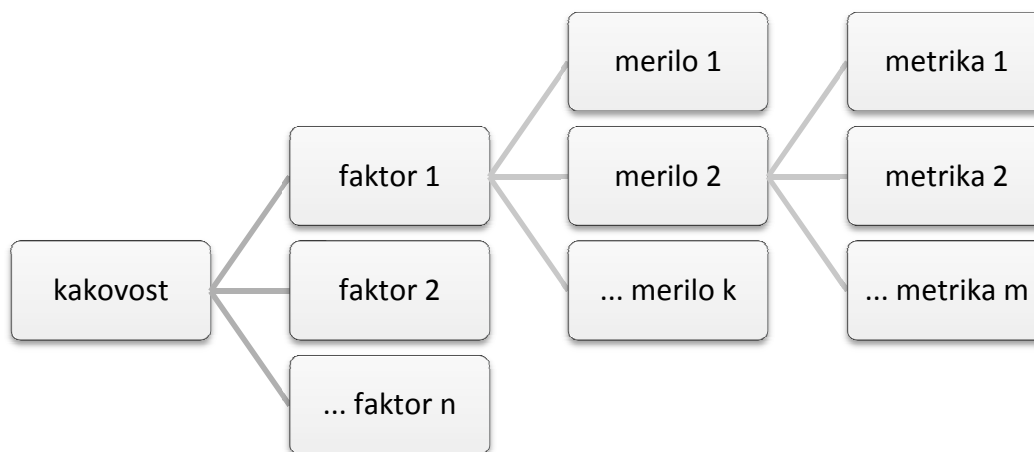
⁴⁸ www.islovar.org definira kakovost kot »stopnjo, s katero sistem, komponenta ali proces zadovoljuje zahteve«.

⁴⁹ Uradna stran ITIL: <http://www.itil-officialsite.com>.

standardiziran način vrednotijo s standardoma ISO/IEC 15504 oziroma SPICE (*Software Process Improvement and Capability dEtermination*)⁵⁰.

- **Vidik programskega izdelka** (*product view*). V tem vidiku izhaja kakovost iz neposrednih lastnosti programskega izdelka, kot so: zanesljivost, prenosljivost, prožnost, zmožnost ponovne uporabe, zmožnost vzdrževanja, testiranja in druge. Kakovost se meri posredno z različnimi metrikami, kot sta kompleksnost in sklopljenost, za katere se privzame, da merijo vrednosti pripadajočih atributov programskega izdelka. V standardu ISO/IEC 9126 je takšen vidik kakovosti opredeljen kot »notranja kakovost« (*internal quality*).
- **Vidik, temelječ na vrednosti** (*value-based view*). Vidik, temelječ na vrednosti, zazna poglede uporabnika, proizvodnje in proizvoda. Kakovost se meri kot razmerje med prednostmi in stroški programskega izdelka.

Zaradi kompleksnosti je kakovost najpogosteje opredeljena v modelih kakovosti, ki določajo lastnosti kakovosti in njihove relacije. Prva modela kakovosti programske opreme, ki sta nastala v sedemdesetih letih prejšnjega stoletja, sta McCallov model kakovosti (McCall, Richards, & Walters 1977) in Boehmov model kakovosti (Boehm et al. 1978). Oba modela temeljita na uporabniško usmerjenem identificiranju ključnih lastnosti programskih izdelkov, ki jih avtorji imenujejo »faktorji kakovosti« (*quality factors*). Avtorji obeh modelov so prav tako ugotovili, da so faktorji kakovosti preveč abstraktni za neposredno vrednotenje, zato so razgrajeni na t. i. merila kakovosti (*quality criteria*). Nadaljnji nivo razgradnje poveže merila kakovosti z dejanskimi metrikami kakovosti (*quality metrics*). Navedena modela razgradnje kakovosti sta poznana pod akronimom FCM (*Factor-Criteria-Metrics*)(Slika 25).



SLIKA 25: MODEL FCM (ISO 9126 2001)

⁵⁰ <http://www-sqi.cit.gu.edu.au/spice/>.

2.5.1 STANDARD ISO/IEC 9126

Med modeli kakovosti programske opreme sta v zadnjem času najodmevnejša standard ISO/IEC 9126 (ISO 9126 2001) in njegova novejša revizija, ki deli standard na štiri podstandarde:

- **standard ISO/IEC 9126-1**, ki definira tri modele kakovosti programskih izdelkov: (1) model notranje kakovosti, (2) model zunanje kakovosti in (3) model kakovosti v uporabi.
- **standard ISO/IEC 9126-2**, ki definira množico zunanjih metrik,
- **standard ISO/IEC 9126-3**, ki definira množico notranjih metrik, in
- **standard ISO/IEC 9126-4**, ki definira množico metrik v uporabi izdelka.

V standardu ISO/IEC 9126-1 sta notranja kakovost in zunanja kakovost definirani naslednje (Cote, Suryan, & Georgiadou 2006):

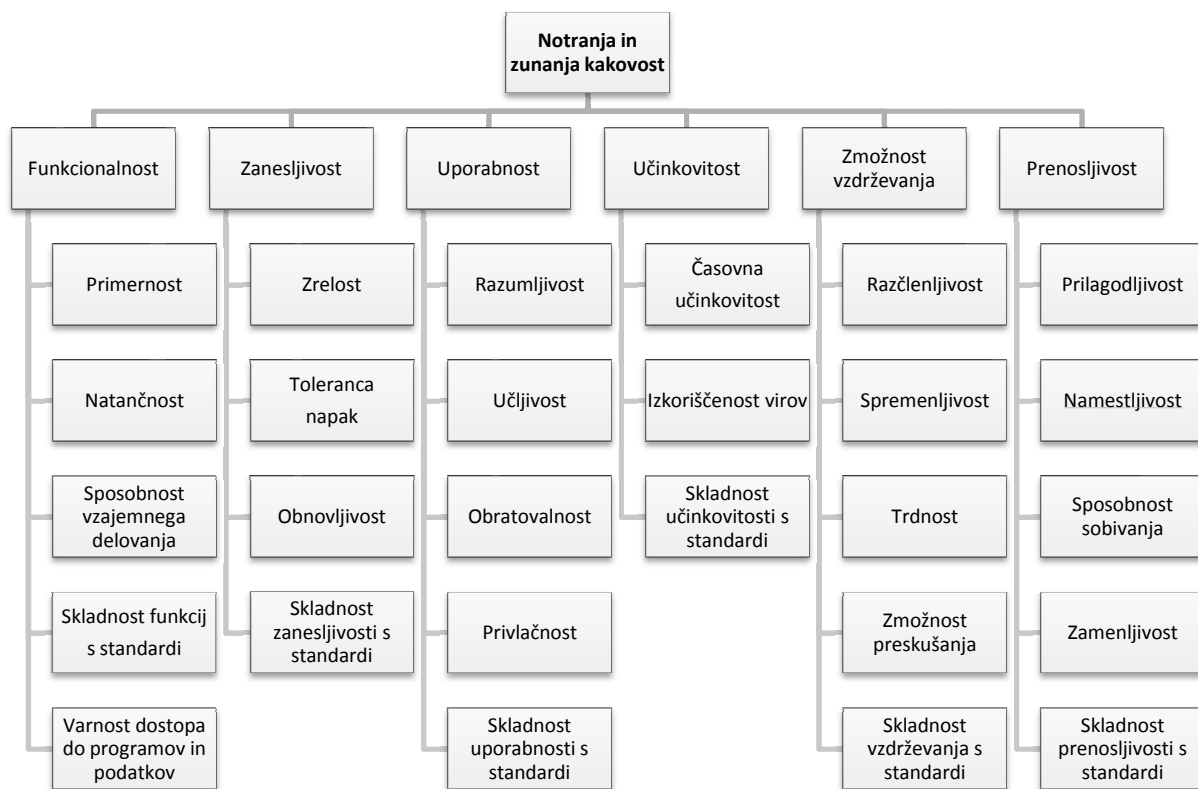
- definicija notranje kakovosti:

.....
Notranja kakovost je skupek karakteristik programskega izdelka, ki določajo njegovo zmožnost zadostiti eksplicitno navedenim in implicitnim zahtevam pod določenimi pogoji uporabe.
.....

- definicija zunanje kakovosti:

.....
Zunanja kakovost predstavlja obseg, v katerem programski izdelek pod določenimi pogoji uporabe zadovoljuje eksplicitno navedene in implicitne zahteve.
.....

Modela notranje in zunanje kakovosti standarda ISO/IEC 9126-1 temeljita na McCallovem in Boehmovem modelu kakovosti. Modela sestavlja šest karakteristik in 27 podkarakteristik kakovosti programskih izdelkov (Slika 26).



SLIKA 26: MODEL NOTRANJE IN ZUNANJE KAKOVOSTI V ISO/IEC 9126-1

»Kakovost v uporabi« je definirana naslednje:

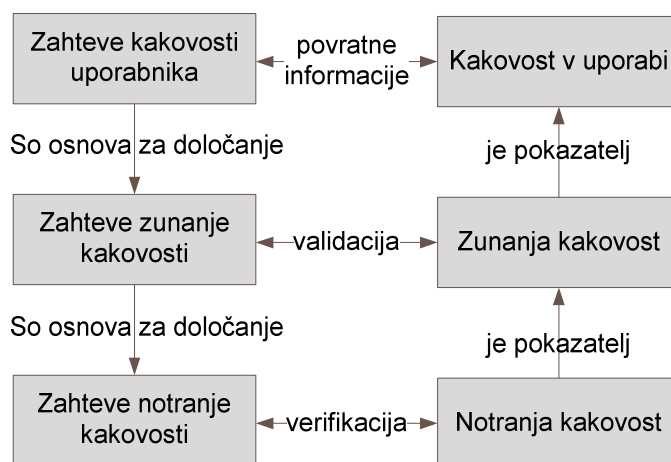
Kakovost v uporabi predstavlja obseg, v katerem programski izdelek zadostuje potrebam določenega uporabnika, ki uporablja izdelek z namenom doseganja določenega cilja.

Model »kakovosti v uporabi« je razdeljen le na nivo karakteristik, ki so: učinkovitost, produktivnost, varnost in zadovoljstvo (Slika 27).



SLIKA 27: MODEL »KAKOVOSTI V UPORABI« V ISO/IEC 9126-1

Posamezni vidiki oziroma modeli kakovosti v standardu ISO/IEC 9126-X (notranja kakovost, zunanja kakovost in kakovost v uporabi) so med sabo povezani, kot prikazuje naslednja slika (Slika 28).



SLIKA 28: POVEZAVA MED RAZLIČNIMI VIDIKI KAKOVOSTI V ISO/IEC 9126-X

Preostali podstandardi (ISO/IEC 9126-2, ISO/IEC 9126-3, in ISO/IEC 9126-4) določajo metrike za posamezne podkarakteristike. Združitevna metoda, ki bi omogočila vrednotenje podkarakteristik na osnovi metrik in karakteristik na osnovi podkarakteristik, v standardu ni definirana.

Model ISO/IEC 9126-1 je splošen model in predstavlja najbolj celovit model kakovosti programskih izdelkov, saj pokriva vse vidike kakovosti (Kitchenham & Pfleeger 1996). Kot pomanjkljivosti modela ISO/IEC 9126-1 (in sorodnih modelov) so se pokazale naslednje:

- **Neutemeljenost modela.** Model ne pojasnjuje, zakaj so njegove karakteristike sestavni del kakovosti. Prav tako ni pojasnjeno, zakaj so podkarakteristike sestavni del določene karakteristike. S tega vidika je struktura modela neutemeljena, prav tako ni mogoče dokazati konsistentnosti in celovitosti modela (Jung, Kim, & Chung 2004). V kavzalni oziroma statistični interpretaciji modela predstavlja vsaka karakteristika in podkarakteristika modela kakovosti statistični faktor ali dimenzijo. Rezultati empirične raziskave modela ISO/IEC 9126-1 so pokazali, da se struktura dejanskega modela kakovosti razlikuje od strukture modela ISO/IEC 9126-1 (Kitchenham & Pfleeger 1996).
- **Nepovezanost elementov modela.** V modelu ni opredeljeno, kako so posamezne metrike povezane s podkarakteristikami in karakteristikami modela. Metrik modela zato ni mogoče preveriti. Prav tako je nepreverljiv celoten model kakovosti (Kitchenham & Pfleeger 1996).

Raziskovalci, ki preučujejo kakovost programske opreme, ugotavljajo, da ni mogoče zagotoviti enotnega in merljivega modela kakovosti za vse vrste programske opreme in za vse tipe uporabnikov. Čeprav je merjene kakovosti pomembno, so razvijalci programske opreme primorani oblikovati svoje modele kakovosti za lastne, specifične potrebe (Fenton & Pfleeger 1998). Modeli kakovosti, kot je ISO/IEC 9126-1, so lahko namenjeni le podajanju splošnih smernic kakovosti pri razvoju programske opreme.

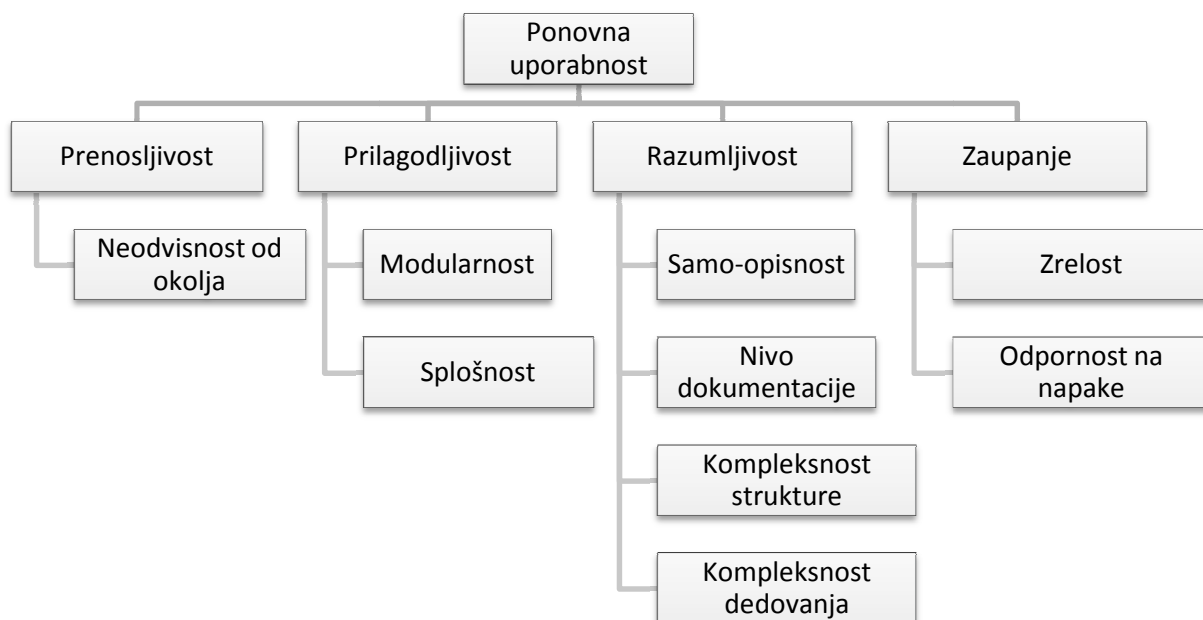
Iz navedenega je razvidno, da se lahko kakovost programske opreme vrednoti na dva načina: (1) z uporabo obstoječega modela kakovosti (*fixed quality model*) ali (2) z definiranjem svojega modela kakovosti (*defined-by-your-own quality model*), v katerem sami opredelimo, kateri atributi kakovosti so nam pomembni (Poulin 1994).

2.5.2 MODEL REBOOT

Obstaja več modelov in taksonomij merjenja ponovne uporabnosti programske opreme, izmed katerih je najbolj dodelan model, ki je nastal v okviru projekta REBOOT (*REuse Based on Object-Oriented Techniques*) (Sindre, Conradi, & Karlsson 1995).

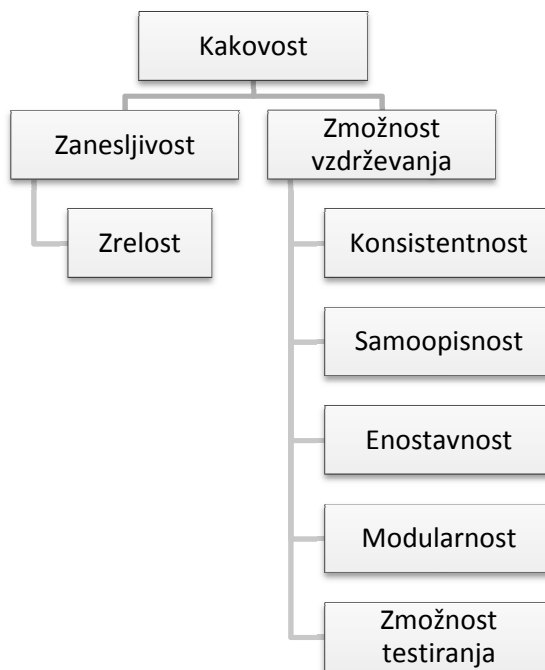
V okviru projekta REBOOT sta med drugim nastala komplementarna modela ponovne uporabnosti in kakovosti, namenjena ponovno uporabni programski opremi. Modela sta nastala na osnovi empiričnih podatkov in metode za konstrukcijo modela FCM, ki se je uporabila tudi pri razvoju modela kakovosti ISO/IEC 9126-1 (ISO 9126 2001).

Model ponovne uporabnosti se je izoblikoval na osnovi vprašanja, kateri faktorji so pomembni za ponovno uporabnost programskih izdelkov, ki so namenjeni ponovni uporabi. Model, ki je pri tem nastal, sestavljajo štiri faktorji in devet meril ponovne uporabnosti (Slika 29).



SLIKA 29: REBOOT MODEL PONOVNE UPORABNOSTI (SINDRE, CONRADI, & KARLSSON 1995)

Model kakovosti, namenjen ponovno uporabni programski opremi, je nastal na osnovi vprašanja, kateri faktorji so pomembni za kakovost programskih izdelkov, ki so namenjeni ponovni uporabi. Model, ki je pri tem nastal, je komplementaren modelu ponovne uporabnosti. Sestavljajo ga dva faktorja in šest meril kakovosti (Slika 30).



SLIKA 30: REBOOT MODEL KAKOVOSTI (JUNG, KIM, & CHUNG 2004)

Podobno kot model ISO/IEC 9126-1 vključujeta tudi modela REBOOT metrike, ki se lahko uporabijo za merjenje posameznih meril (podkarakteristik) modela kakovosti in ponovne uporabnosti. Poglavitna razlika med modeli REBOOT in modeli ISO/IEC 9126 je, da modela REBOOT opredeljujeta povezavo metrik ponovno uporabne programske opreme z merili in faktorji kakovosti in ponovne uporabnosti. REBOOT prav tako predpisuje:

- način vrednotenja faktorjev in meril modela kakovosti in ponovne uporabnosti,
- način predstavitve rezultatov vrednotenja in
- proces izboljšave ponovno uporabne programske opreme na osnovi modelov FCM.

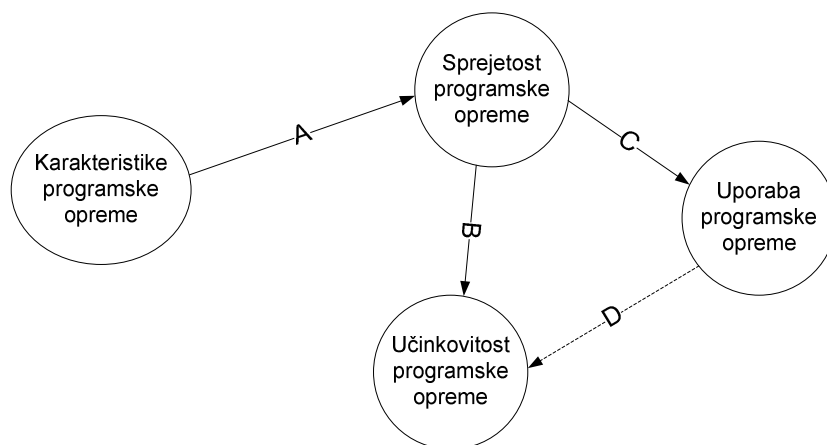
Modela REBOOT sta namenjena predhodnemu (*a priori*) in kasnejšemu (*a posteriori*) vrednotenju. Predhodno vrednotenje modelov se izvaja v fazi razvoja ponovno uporabne programske opreme. Izvede se z ustreznimi metriki, ki se nato, skupaj z utežmi, združujejo v merila, faktorje in ponovno uporabnost oziroma kakovost. Predhodno vrednotenje je namenjeno oceni kakovosti oziroma ponovne uporabnosti in opredelitvi smernic za izboljšave na področju ponovne uporabe oziroma kakovosti. Kasnejše vrednotenje modelov se izvaja v fazi uporabe ponovno uporabne programske opreme na osnovi podatkov, zbranih od uporabnika komponente.

REBOOT navaja tudi smernice za vrednotenje različnih tipov ponovne uporabe, pri čemer definira specifične metrike in podmodele za različne faze razvoja in vrste ponovno uporabne programske opreme.

2.6 POVEZAVA MED KARAKTERISTIKAMI, SPREJETOSTJO IN UČINKOVITOSTJO PROGRAMSKE OPREME

Koncepti kakovosti, ponovne uporabnosti, sprejetosti in uspešnosti uporabe so v kontekstu programske opreme najpogosteje obravnavani ločeno. Kljub temu je možno najti povezave med njimi.

Osnovo za določitev povezav med koncepti kakovosti, ponovne uporabnosti, sprejetosti in učinkovitosti (uspešnosti uporabe) programske opreme lahko predstavljajo kavzalni modeli (*causal model*)⁵¹ sprejetosti in uspešnosti ter kavzalna interpretacija modela kakovosti (Gallego, Luna, & Bueno 2008) in ponovne uporabnosti. Na osnovi kavzalnih interpretacij modelov so povezave med njimi naslednje (Slika 31):



SLIKA 31: POVEZANOST KARAKTERISTIK, SPREJETOSTI IN UČINKOVITOSTI PROGRAMSKE OPREME

- Povezava med karakteristikami programske opreme (kakovost, ponovna uporabnost) in sprejetostjo programske opreme je bila analizirana v številnih raziskavah sprejetosti programske opreme, kjer je bila demonstrirana signifikantna povezava med karakteristikami kakovosti in sprejetostjo programske opreme. Gallego et al. (2008) so na osnovi modela TAM ugotovili signifikanten vpliv med kakovostjo odprtega programja in dojeto uporabnostjo odprtega programja. Prav tako je bil ugotovljen signifikanten vpliv med kakovostjo in dejansko uporabo programske opreme (McFarland & Hamilton 2006). Drugi dve raziskavi (Al-Gahtani 1998; Hong, Thong, Wong, & Tam 2001) sta potrdili signifikanten vpliv med karakteristikami sistema in sprejetostjo programske opreme.

⁵¹ Kavzalni modeli so abstraktni modeli, ki ponazarjajo obnašanje sistemov z uporabo logike vzrokov in posledic.

- Viri, ki trdijo, da je sprejetost kritičnega pomena za učinkovitost implementacij programske opreme, so: (Davis 1989;Pikkarainen et al. 2004;Succi, Succi, & Walter 1999;Venkatesh & Davis 1996).
- Povezava med sprejetostjo in uporabo programske opreme je opredeljena v modelu TAM (Davis 1989). Številne potrditvene študije modela TAM so potrdile, da je sprejetost najboljši napovednik uporabe programske opreme (Lederer, Maupin, Sena, & Zhuang 2000).
- Povezava med sprejetostjo in učinkovitostjo programske opreme je pojasnjena v Seddonovem modelu uspešnosti (Seddon 1997). Seddon trdi, da obstaja kavzalna povezava med uporabo in uspešnostjo uporabe v primeru programske opreme, pri kateri več uporabe ali večje število uporab predstavlja več neto prednosti za uporabnika. Avtorji Behrens et al. (2005) so prišli do podobne ugotovitve kot Seddon. Ugotovili so, da je za napovedovanje uspešnosti uporabe možno uporabiti TAM, in sicer v primeru, ko je možno uspešnost uporabe programske opreme meriti z njeno uporabo. Skladno s to ugotovitvijo so Garrity et al. (2005) vrednotili uspešnost spletnih informacijskih sistemov kar z modelom TAM.

2.7 POVZETEK PREDSTAVITVE RAZISKOVALNEGA PODROČJA

V predstavitvi raziskovalnega področja smo predstavili in analizirali obstoječe znanje, ki je relevantno za področje disertacije.

Predstavili smo osrednji predmet raziskave, programska ogrodja, in ga umestili v področje programskega inženirstva, ponovne uporabe in procesa razvoja programske opreme. Predstavili smo osnovne značilnosti ogrodij, vrste ogrodij in zgodovinski razvoj ogrodij. Izpostavili smo podobnosti in razlike ogrodij, glede na druge tehnike ponovne uporabe, kot so: vzorci, komponente in produktne linije. Predstavili smo prednosti ogrodij, pogloblitve slabosti ogrodij in aktualne raziskave na področju ogrodij. Skupaj z uvodnim poglavjem disertacije je bilo ugotovljeno, (1) da so ogrodja zelo razširjena in vključena v velik del obstoječih projektov razvoja programske opreme; (2) da imajo ogrodja pomanjkljivosti, ki jih razvijalci in raziskovalci niso odpravili vse od njihovega pojava; in (3) da so aktualne raziskave na področju ogrodij metodološko pomanjkljive in nepovezane.

Učinkovitost oziroma uspešnost informacijskih sistemov (IS) je bila predstavljena kot poglobliten vidik oziroma odvisna spremenljivka uporabe in primerjave informacijskih sistemov. Predstavljeni so bili osnovni modeli določanja učinkovitosti IS: Shannonov in Weaverjev model, Masonov model in Delonov in McLeanov model (D&M). Podrobneje smo predstavili Seddonov model, ki predstavlja nadgradnjo modela D&M v smislu odprave njegovih pogloblitvenih pomanjkljivosti, kot so: mešanje kavzalnih in procesnih povezav in različne interpretacije koncepta »uporaba«. Iz Seddonovega modela je prav tako razvidna povezava med sprejetostjo, uporabo in uspešnostjo IS.

Sprejetost informacijskih tehnologij (IT) smo predstavili kot prvi pogoj za uspešnost IS in kot ključni faktor, ki vpliva na izboljšanje produktivnosti v organizacijah. Predstavili smo osnovne koncepte in povezave modelov sprejetosti. Podrobneje smo se osredotočili na model TAM, ki je najbolj priljubljen model sprejetosti na področju IT. Predstavljeni so bili koncepti in kavzalne povezave modela TAM in njihove dosedanje validacije. Izpostavili smo prednosti, pomanjkljivosti in variacije modela TAM. Na osnovi dosedanjih raziskav smo predstavili najpogostejše skupine razširitev modela TAM. V manjšem obsegu smo predstavili model ujemanja zahtev in lastnosti IT – TTF, ki ponazarja drugo smer raziskav na področju sprejetosti IT. Pri modelu TTF smo izpostavili možnosti integracije modela TTF z modelom TAM.

V podpoglavju Kakovost in ponovna uporabnost programske opreme smo predstavili koncepta kakovosti in ponovne uporabnosti programske opreme. Oba koncepta smo na osnovi modela FCM (*factor-criteria-metrics*) predstavili do nivoja meril, in sicer na osnovi standarda ISO/IEC

9126 in modela ponovne uporabnosti REBOOT. Predstavili smo pogloblitve pomanjkljivosti modelov, ki se kažejo predvsem v splošnosti in nedoločenosti povezav med faktorji, merili in metrikami modelov. Iz teh razlogov je pogosto potrebna specializacija modelov za določeno področje uporabe.

3 RAZISKAVE

V tem poglavju so predstavljeni raziskave, s katerimi smo dobili odgovore na tezo doktorske disertacije, in iz nje izhajajoča raziskovalna vprašanja. V raziskavah smo uporabili različne znanstvene metode^{52,53} in pri njihovi izvedbi upoštevali osnovne značilnosti znanstvene metodologije⁵⁴ (Ivanko 2007):

- jasno in precizno ter jezikovno, stilistično in terminološko pravilno formuliranje trditev,
- obrazložitev in povezovanje novih znanstvenih spoznanj s spoznanji s sorodnih področij in
- testiranje izidov znanstvenega raziskovanja v praksi.

»Vsaka raziskava je povezana s tveganji, drugače ne gre za raziskavo.«

»Če smo spoznali nekaj novega, še ne pomeni, da smo raziskovali.

Temu pravimo učenje.«

»Samo zato ker smo zaposleni, še ne pomeni da raziskujemo.

Temu pravimo delo.«

(Johnston & Shanks 2003)

V doktorski disertaciji smo izvedli več raziskav in uporabili različne metode raziskovanja iz naslednjih razlogov:

- Vsaka metoda raziskovanja ima svoje prednosti in pomanjkljivosti. Z uporabo več metod raziskovanja⁵⁵ na enakem predmetu raziskave in z navzkrižno primerjavo dobljenih rezultatov dosežemo večjo zanesljivost rezultatov (Neuman 2005).
- Obsežne raziskovalne probleme je smiselno razgraditi na manjše, ki so lažje obvladljivi. Rezultati parcialnih raziskav se lahko nato združijo in tako podajo odgovor na kompleksnejše raziskovalno vprašanje.
- Narava raziskovalnega problema je zahtevala longitudinalni raziskovalni pristop, kar pomeni, da je za podajanje odgovora na zastavljena raziskovalna vprašanja bilo najprej treba odgovoriti na predhodna oziroma vmesna vprašanja.

⁵² Metoda je način raziskovanja in prikazovanja spoznanj o raziskovalnem predmetu (Ivanko 2007).

⁵³ Znanstvena metoda je celota postopkov, ki jih znanost uporablja v znanstvenoraziskovalnem delu, da bi prikazala in obrazložila izide znanstvenega raziskovanja določenega znanstvenega področja ali discipline (Ivanko 2007).

⁵⁴ Znanstvena metodologija je veda o metodah znanstvenega raziskovanja (Ivanko 2007).

⁵⁵ Poznano pod besedno zvezo »triangulacija metod« (*triangulation of methods*).

3.1 ZASNOVA RAZISKAV

Pri zasnovi raziskav smo upoštevali (1) predstavitev problematike, (2) metodološki vidik raziskave in (3) predlagano rešitev. Vsa tri področja so bila predstavljena v uvodnem poglavju.

3.1.1 ANALIZA RAZISKOVALNEGA PROBLEMA

Cilj raziskav v naši disertaciji je potrditi tezo doktorske disertacije in podati odgovore na raziskovalna vprašanja, ki izhajajo iz teze (glej poglavje 1.3.1). Zato temelji analiza raziskovalnega problema na tezi doktorske disertacije in podanih raziskovalnih vprašanjih:

Teza doktorske disertacije: »Z integracijo modelov, ki določajo sprejetost IT, uspešnost IS in specifičnih faktorjev iz domene ogrodij, je možno definirati operacionaliziran model uspešnosti ogrodij, ki je natančnejši od obstoječih formalnih modelov. Takšen model lahko predstavlja osnovo za model določanja sprejetosti ogrodij in uspešnosti ogrodij.«

Raziskovalno vprašanje RV₁: »Kateri faktorji vplivajo na sprejetost ogrodij in kako jih lahko merimo?«

Za pridobitev odgovora na RV₁ bomo izvedli dve raziskavi. Prva raziskava (Ra₁) bo identificirala potencialne faktorje, ki vplivajo na sprejetost ogrodij. Druga raziskava (Ra₂) bo temeljila na rezultatih prve raziskave in bo potrdila oziroma zavrgla vpliv posameznih faktorjev na sprejetost ogrodij.

Raziskovalno vprašanje RV₂: »Ali je predlagani model sprejetosti ogrodij natančnejši od obstoječih modelov sprejetosti IT?«

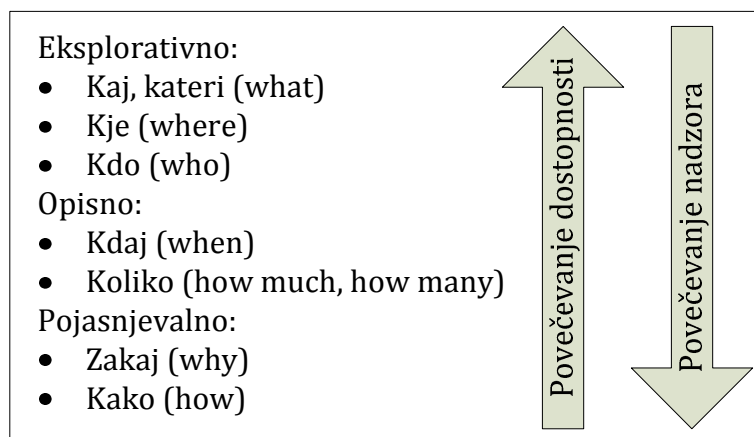
Odgovor na RV₂ bomo dobili tako, da bomo primerjali rezultate natančnosti modela sprejetosti ogrodij in splošnega modela sprejetosti IT (Ra₃).

Raziskovalno vprašanje RV₃: »Ali obstaja povezava med sprejetostjo in uspešno uporabo ogrodja?«

Za podajanje odgovora na RV₃ bomo izvedli raziskavo (Ra₄), v kateri bomo analizirali vplive med koncepti sprejetosti ogrodij in koncepti uspešnosti ogrodij.

3.1.2 IZBIRA METOD RAZISKOVANJA

Metodo raziskovanja za posamezne raziskave smo izbrali glede na (Johnston & Shanks 2003): (1) pripadajoče raziskovalno vprašanje (Slika 32), (2) namen raziskave in (3) faze v procesu raziskovanja (Tabela 2)^{56,57}.



SLIKA 32: VRSTE RAZISKOVALNIH VPRAŠANJ

Obstajajo trije osnovni nameni raziskav (Neuman 2005): eksplorativen, opisen in pojasnjevalen. Osnovni namen eksplorativnega raziskovanja (*exploratory research*) je preučiti slabo raziskan pojav ali predmet in nato na njegovi osnovi oblikovati natančnejšo raziskovalno vprašanje. Eksplorativno raziskovanje običajno temelji na kvalitativnih tehnikah raziskovanja.

Namen opisnega raziskovanja (*descriptive research*) je podati sliko določenega vidika pojava ali predmeta, ki ga nameravamo raziskati. Opisno raziskovanje zahteva fokusirano raziskovalno vprašanje. Pogosto predstavlja osnovo za pojasnjevalno raziskovanje.

Namen pojasnjevalnega raziskovanja (*explanatory research*) je razložiti pojave ali predmete raziskav. Pojasnjevalno raziskovanje služi za oblikovanje, ovrednotenje, razširitev in testiranje obstoječih teorij, metod in tehnik.

⁵⁶ Nivo dostopnosti (*level of access*) – večja ko je dostopnost, večja je verjetnost, da dobivamo pravilne rezultate, in obratno (Johnston & Shanks 2003).

⁵⁷ Nivo nadzora (*level of control*) – večji ko je nadzor, večja je verjetnost, da so naši sklepi veljavni, in obratno (Freimut, Punter, Biffel, & Ciolkowski 2002).

TABELA 2: IZBIRA METOD RAZISKOVANJA

Raziskava	Ra1	Ra2	Ra3	Ra4
Raziskovalno vprašanje		RV1	RV2	RV3
Namen raziskave	Eksplorativen (exploratory)	Pojasnjevalen (explanatory)	Pojasnjevalen (explanatory)	Pojasnjevalen (explanatory)
Faza v procesu raziskovanja	Začetna	Vmesna (Predpogoj Ra1)	Vmesna (Predpogoj Ra2)	Končna
Metoda raziskovanja	Študija literature in fokusna skupina	Anketa	Anketa	Anketa

V nadaljevanju so predstavljene in analizirane izbrane metode raziskovanja.

3.1.2.1 ŠTUDIJA LITERATURE

Študija literature predstavlja bistven korak v vsakem procesu raziskovanja. Namen študije literature je identificirati informacije, ki so povezane z našo raziskavo. Cilji študije literature so naslednji (Neuman 2005):

- nazorna demonstracija seznanjenosti z obstoječim znanjem in vzpostavitev kredibilnosti raziskave,
- ponazoritev smeri predhodnih raziskav in povezave predhodnih raziskav z našo raziskavo,
- integracija znanja na določenem področju in
- učenje na izkušnjah drugih in oblikovanje novih raziskovalnih vprašanj.

Študij literature je več vrst (Neuman 2005), pri čemer potrebam naše raziskave (Ra₁) ustreza metaanaliza.

Metaanaliza (*meta analysis*) je posebna vrsta študije literature, ki prestavlja dopolnilni pregled ali metodološki pregled, v okviru katerega raziskovalec objavi podatke, ki jih je zbral in analiziral iz večjega števila sorodnih raziskav. Metaanalize lahko temeljijo na kvalitativnih ali kvantitativnih podatkih.

3.1.2.2 FOKUSNA SKUPINA

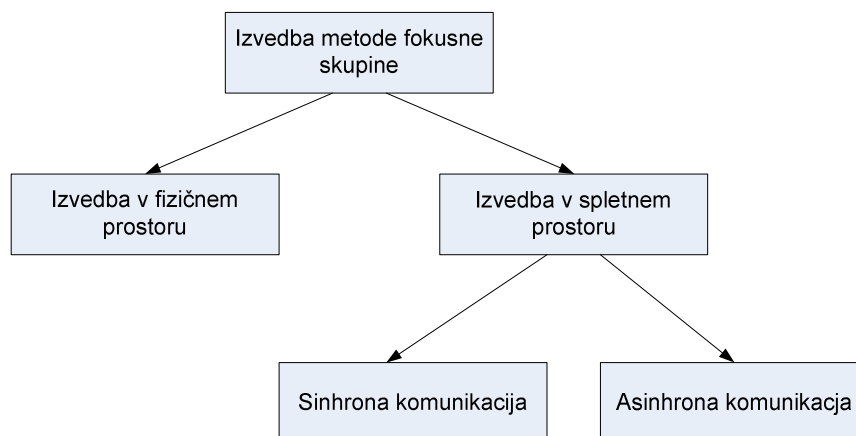
Fokusna skupina (*focus group*), ki je doživela nagel vzpon v zadnjih dvajsetih letih, predstavlja ključno tehniko za zbiranje kvalitativnih podatkov (Johnston & Shanks 2003). Fokusna skupina je specialna, empirična, kvalitativna in terenska raziskovalna tehnika, v kateri je skupina ljudi neformalno intervjuvana v okolju diskusijske skupine (Neuman 2005). Skupino običajno sestavlja okoli 10 ljudi, ki imajo interes in znanje diskutirati o podani tematiki. Idealna skupina je sestavljena iz homogenih članov, ki se med seboj ne poznajo (minimiziranje medsebojnih vplivov). Fokusno skupino vodi moderator, čigar cilj je ostati nevtralen. Za doseganje boljših rezultatov se pogosto zaporedoma izvede več fokusnih skupin. Prvi pogoj za učinkovitost fokusne skupine je njeno skrbno načrtovanje.

Fokusne skupine je v okviru celotne raziskave smiselno dopolnjevati z drugimi metodami raziskovanja. S tem se minimizirajo slabosti omenjene raziskovalne metode (Tabela 3).

TABELA 3: PREDNOSTI IN SLABOSTI FOKUSNIH SKUPIN
(JOHNSTON & SHANKS 2003; NEUMAN 2005)

Prednosti	Slabosti
<ul style="list-style-type: none"> - Naravno okolje, v katerem se nahajajo udeleženci, omogoča svobodno in neobremenjujoče izražanje zamisli. - Udeleženci se počutijo sposobne in posledično motivirane. - Udeleženci se lahko medsebojno sprašujejo in medsebojno pojasnjujejo odgovore. - Odprti odgovori zagotavljajo velike količine podatkov. - Možnost podajanja vprašanj na že podane odgovore. 	<ul style="list-style-type: none"> - Pojavi se lahko učinek »polarizacije«. - V okviru fokusne skupine se lahko naenkrat diskutira o zelo majhnem številu vprašanj. - Moderator lahko nehote omeji svobodno diskusijo. - Udeleženci v fokusni skupini običajno proizvedejo manj zamisli kot intervjuvani posamezniki. - Postopek izvedbe raziskave fokusne skupine je redko pojasnjen. - Težko je pojasniti razlike, ki lahko nastanejo med intervjuvanimi posamezniki in fokusno skupino. - Prevlada posameznih udeležencev. - Omejitve v primeru generalizacije pridobljenih rezultatov na širšo populacijo. - Odprte odgovore je težko organizirati in interpretirati.

Načinov izvedbe fokusnih skupin je več. V zadnjem času se vse bolj pogosto uporabljajo spletne fokusne skupine (*online focus group*). Pri tem se lahko spletna fokusna skupina izvede z uporabo sinhrono ali asinhrono komunikacije (Slika 33).



SLIKA 33: MOŽNOSTI IZVEDBE FOKUSNIH SKUPIN

Fizična izvedba metode fokusne skupine je primerna pri udeležencih z dobrimi retoričnimi sposobnostmi, medtem ko je spletna izvedba metode primerna za udeležence z dobro računalniško pismenostjo in udeležence, ki so sposobni svoje ideje izražati z besedilom (Chase & Alvarez 2000). V spodnji tabeli (Tabela 4) je podana primerjava med fizično in spletno fokusno skupino.

TABELA 4: PRIMERJAVA FIZIČNE IN SPLETNE IZVEDBE FOKUSNE SKUPINE (CHASE & ALVAREZ 2000)

	Fizična fokusna skupina	Spletna fokusna skupina
Splošni vidiki	Primerna za ljudi z dobrimi retoričnimi sposobnostmi. Na nivo udobja vpliva okolje.	Primerna za računalniško pismene ljudi. Nivo udobja je visok, ker se udeleženci nahajajo v lastnem okolju.
Značilnosti diskusije	Moderator lahko z dodatnimi vprašanji odkrije in odpravi dvoumnosti. Pridobijo se lahko poglobljene informacije.	Moderator težje odkriva dvoumnosti. Poglobljene informacije se težje dosežejo.
Komunikacijski tok	Na odgovore udeležencev lahko vplivajo drugi udeleženci. Enostavno vzdrževanje fokusa. Lahko je potreben poseg moderatorja, da se diskusija premakne z mrtve točke.	Anonimnost in fizična dislociranost povečujeta iskrenost in spontanost. Izguba fokusa je možna tudi v primeru intervencije moderatorja. Značilen je hiter prehod med predmeti diskusije.
Neverbalna komunikacija	Zaznavna je govorica telesa.	Zaznavajo se odzivni čas, pogostost podajanja odgovorov, črkovanje in stil

	Fizična fokusna skupina	Spletna fokusna skupina
Verbalna komunikacija	Iz tona in govora so razvidna čustva.	pisanja. Izkazovanje čustev je omejeno s simboli komunikacijske aplikacije.

3.1.2.3 ANKETA

Metoda anketiranja ali krajše anketa je postopek, s katerim na podlagi vprašalnika, intervjujev ali literature zbiramo podatke, informacije ali mnenja o raziskovalnem problemu. Anketna vprašanja se postavijo skupini ljudi, ki pripada določeni populaciji. Cilj ankete je identifikacija povezav med koncepti raziskovalnega področja. Ankete se, glede na njihov namen, delijo na opisne, eksplorativne in pojasnjevalne.

Poleg eksperimenta in študije primera spada anketa med najpogostejše metode raziskovanja v empiričnem programskem inženirstvu (Freimut, Punter, Biffel, & Ciolkowski 2002). V spodnji tabeli (Tabela 5) so podane značilnosti ankete in razlike med anketo in drugimi empiričnimi metodami raziskovanja.

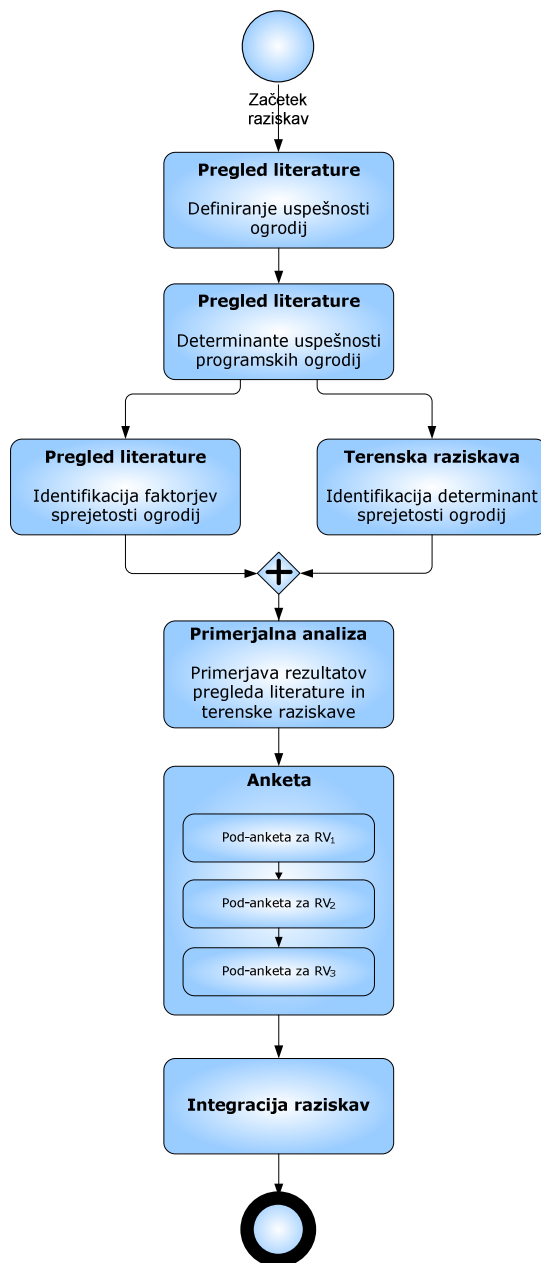
TABELA 5: PRIMERJAVA EMPIRIČNIH METOD RAZISKOVANJA
(Freimut, Punter, Biffel, & Ciolkowski 2002)

	Anketa	Študija primera	Eksperiment
Namen raziskave	Preučevanje informacij, zbranih od skupine ljudi, projektov, organizacij ali literature.	Preučevanja primera v realnem in reprezentativnem okolju.	Vzpostavitev kavzalnih povezav, potrditev teorij.
Zahtevan nadzor	Nizek	Srednji	Visok
Ustreznost metode	Analiza sprememb v velikem obsegu.	Vrednotenje spremembe v dejanskem okolju.	Določanje prednosti in slabosti uvedbe sprememb.
Okolje	Realno	Realno	Umetno (laboratorij)
Tip sklepanja	Statistični	Analitični	Statistični
Prednosti	Metoda je primerna za dejanske projekte, omogočena je generalizacija rezultatov, omogoča uporabo statističnih metod.	Metoda se vključuje v običajna delovna opravila, enostavna jo je načrtovati, omogoča odgovore na vprašanja »Zakaj?« in »Kako?«.	Metoda omogoča potrditev teorij in kavzalnih povezav.
Slabosti	Nimamo nadzora nad spremenljivkami,	Težavna ponovitev raziskave, težavni	Težko apliciranje metode v dejanske

	Anketa	Študija primera	Eksperiment
	oblikovanje vprašalnika je težavno, izvedba intervjujev je časovno potratna, slaba odzivnost korespondentov.	generalizacija in interpretacija, pomanjkanje standardnih postopkov izvedbe metode.	projekte.
Tip pridobljenih podatkov	Predvsem kvantitativni podatki, ki se pridobijo z uporabo vprašalnikov, intervjujev, metrik projektov ali literature.	Kvantitativni in kvalitativni podatki, ki se pridobijo z uporabo vprašalnikov, intervjujev ali metrik procesov in izdelkov.	Kvantitativni podatki, ki se pridobijo z uporabo vprašalnikov ali z merjenjem procesov in izdelkov.

3.1.3 DEFINIRANJE PROCESA IZVEDBE RAZISKAV

Na spodnji sliki (Slika 34) je prikazan proces izvedbe raziskav, ki smo jih izvedli v okviru disertacije.



SLIKA 34: PROCES IZVEDBE RAZISKAV

Iz slike je razvidno, da smo pred izvedbo zastavljenih raziskav $Ra_1 - Ra_4$ izvedli še dva pregleda literature, s katerima smo definirali uspešnost ogrođij in determinante uspešnosti ogrođij. Dodatno smo izvedli še primerjalno analizo, s katero smo primerjali rezultate terenske raziskave in pregleda literature. Ankete, ki smo jih izvedli v okviru disertacije, smo zaradi medsebojne podobnosti in odvisnosti izvedli in analizirali enotno.

3.2 PREGLED LITERATURE: DEFINIRANJE USPEŠNOSTI PROGRAMSKIH OGRODIJ

Koncept »uspešnost ogrodij« ni nov pojem, saj so ga uporabili že številni raziskovalci in strokovnjaki na področju ogrodij, kot so: Todd (1997), Meier (1998) in Shai (1997). Avtorji teh prispevkov so analizirali in določali različne konceptualne, tehnološke in organizacijske dejavnike, za katere so predpostavljali ali trdili, da vplivajo na uspešnost ogrodij. Rezultati njihovih raziskav so najpogosteje predstavljeni v obliki smernic za razvoj uspešnih ogrodij.

Tudi najbolj elegantno ogrodje se ne bo uporabljalo, če bodo programerjevi dojeti stroški spoznavanja in uporabe ogrodja višji kot programerjevi dojeti stroški razvoja iz začetne točke.

(Langelett 2003)

Gledano z vidika »uspešnosti ogrodij«, imajo našete raziskave naslednje pomanjkljivosti:

- Koncept uspešnosti ogrodij v navedenih prispevkih ni bil definiran, zaradi česar so (lahko) različni avtorji dojemali uspešnost ogrodij na različne načine.
- Avtorji v omenjenih prispevkih niso ločili uspešnosti z vidika razvijalcev in uporabnikov ogrodij. Zato so nekatere smernice namenjene razvijalcem ogrodij, medtem ko so druge namenjene uporabnikom ogrodij.

V naši disertaciji smo odpravili opisani omejitvi z definiranjem koncepta uspešnosti ogrodij:

Uspešnost ogrodja je mera, ki določa objektivne in dojete prednosti, izhajajoče iz uporabe ogrodja.

Zapisana definicija uspešnosti ogrodja vključuje tudi subjektivni vidik uspešnosti in se tako ujema s Seddonovo definicijo uspešnosti IS (Seddon 1997). Argument za uporabo Seddonove definicije uspešnosti IS izhaja iz dejstva, da so ogrodja ponovno uporabni programski izdelki, ki jih razvijalci končnih rešitev uporabljajo v procesu razvoja programske opreme. Zato jih lahko obravnavamo kot izdelke, namenjene razvijalcem končnih rešitev (Morisio, Romano, & Stamelos 2002). Razvijalci končnih rešitev so v tem primeru uporabniki ogrodja.

Definicija uspešnosti ogrodja implicitno vključuje objektivne (neto prednosti) in subjektivne faktorje. Najpogosteje uporabljeni objektivni faktorji uspešnosti ogrodij so (Morisio, Romano, & Stamelos 2002): kakovost končnih izdelkov, produktivnost razvoja izdelkov in čas izdelave izdelkov. Pri tem so empirične raziskave, ki so preučevale prednosti uporabe ogrodij, najpogosteje vrednotile produktivnost in kakovost. Ugotovitve teh raziskav so bile naslednje:

- V kombinaciji študije primera in nadzorovanega preizkusa sta Moser in Nierstrasz (1996) ugotovila, da imajo ogrodja pozitiven vpliv na produktivnost razvoja in kakovost končnih izdelkov.
- Terenska raziskava, ki je preučevala izboljšanje produktivnosti razvoja na osnovi ogrodiv (Mamrak & Sinha 1999), je potrdila pozitiven vpliv ogrodiv na produktivnost. Pri tem se je produktivnost vrednotila na osnovi metod funkcijskih točk in metode systemskega metra.
- V študiji primera, ki so jo izvedli Blake et al. (1983), je bilo ugotovljeno, da uporaba ogrodiv povečuje kakovost izdelkov in produktivnost razvoja.

Poleg predstavljenih objektivnih faktorjev uspešnosti je za določanje uspešnosti IS pomembna tudi subjektivna dimenzija, ki se najpogosteje določa z zadovoljstvom uporabnikov (Seddon 1997). Z vključitvijo subjektivnega faktorja je možno uspešnost ogrodiv definirati z naslednjimi faktorji:

- produktivnost razvoja končnih rešitev,
- kakovost končnih rešitev,
- čas razvoja končnih rešitev in
- zadovoljstvo uporabnikov ogrodiv.

Zadovoljstvo uporabnikov smo po Seddonu (1997) definirali:

.....
»Zadovoljstvo uporabnikov ogrodiv predstavlja subjektivno oceno različnih posledic uporabe ogrodiv, ki se vrednotijo na semantičnem diferencialu.«
.....

3.3 PREGLED LITERATURE: DETERMINANTE USPEŠNOSTI OGRODIJ

Predhodne študije so ugotovile, da je uspešnost uporabe IS odvisna od sprejetosti in pogojno od uporabe IS (glej poglavje 2.6), za katero Seddon trdi, da povečuje le posledice uporabe, ki niso nujno pozitivne. Zato je možno definirati povezavo med uporabo in uspešnostjo le, če povezavi določimo kontekst oziroma tip programske opreme (Michael 1999; Morisio, Romano, & Stamelos 2002; Moser & Nierstrasz 1996).

Obstoječe empirične študije, ki so preučevale razvoj na osnovi ogrodi, so ugotovile povezavo med obsegom uporabe ogrodi in uspešnostjo njihove uporabe (Bockle, Clements, McGregor, Muthig, & Schmid 2004). Avtorji raziskav so ugotovili, da se prednosti, ki izhajajo iz uporabe ogrodi, povečujejo s številom ponovnih uporab ogrodja.

Vpliv obsega uporabe ogrodi na njihovo uspešnost je razviden tudi iz ekonomskega modela produktnih linij (Enačba 1):

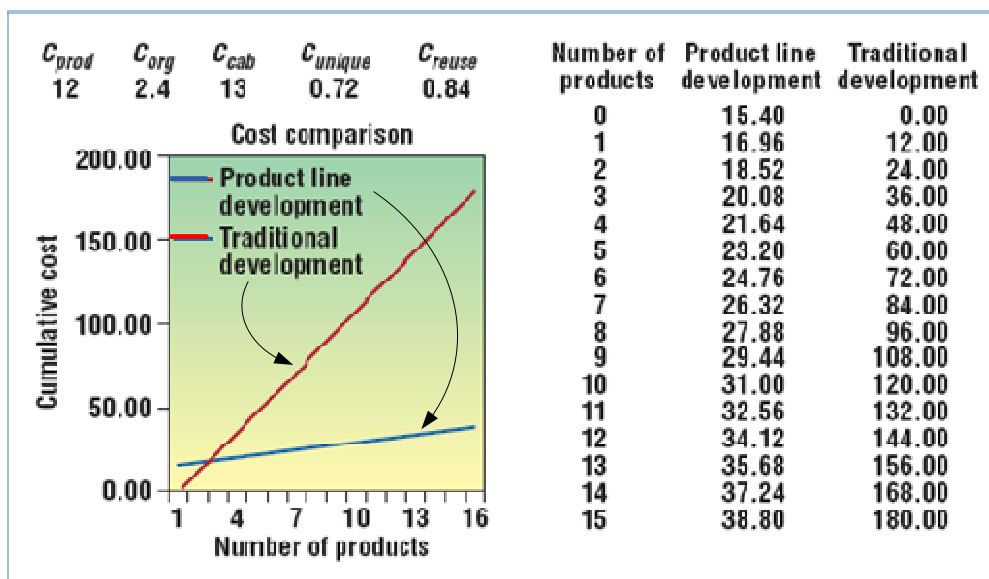
$$C = C_{org} + C_{cab} + \sum_{i=1}^{i=n} (C_{unique}(p_i) + C_{reuse}(p_i))$$

ENAČBA 1: STROŠKOVNI MODEL RAZVOJA NA SKUPNI PROGRAMSKI OSNOVI
(Bockle, Clements, McGregor, Muthig, & Schmid 2004; Bosch 2000; Northrop 1999)

Pomen posameznih faktorjev v navedenem modelu (Enačba 1) je naslednji:

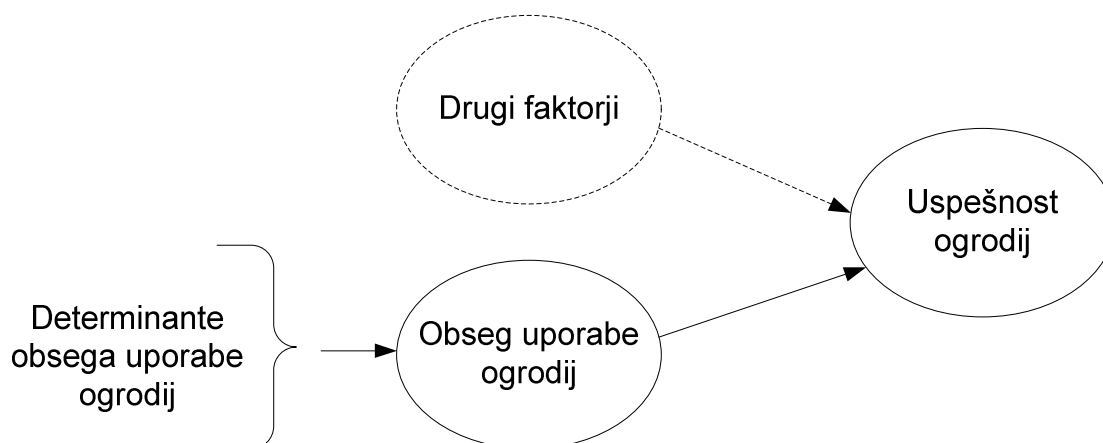
- C – predstavlja skupni strošek vzpostavitve in razvoja »n« izdelkov na osnovi produktne linije.
- C_{org} – predstavlja organizacijske stroške, ki so povezani s spremembo modela razvoja programske opreme (na primer: prehod od nesistematične ponovne uporabe k produktni liniji).
- C_{cab} – predstavlja stroške razvoja skupne osnove za »n« izdelkov.
- $C_{reuse}(p_i)$ – predstavlja stroške razvoja delov končnega izdelka » p_i «, ki nastanejo s ponovno uporabo skupne osnove.
- $C_{unique}(p_i)$ – predstavlja stroške razvoja delov končnega izdelka » p_i «, ki NE temeljijo na skupni osnovi.

Sodobne produktne linije so v veliki meri realizirane z ogrodji (glej poglavje 2.2.5.4), pri čemer predstavljajo ogrodja skupno osnovo (*core asset base*). Zato lahko iz navedene enačbe sklepamo, da ima število primerkov ogrodja (število »n«) pozitiven vpliv na produktivnost razvoja. Prav tako je iz enačbe razviden začetni ekonomski deficit, ki nastane zaradi spremembe procesov razvoja (C_{org}) in razvoja lastnega ali spoznavanja obstoječega ogrodja (C_{cab}) (Slika 35).



SLIKA 35: SKUPNI STROŠKI »OBIČAJNEGA RAZVOJA« IN RAZVOJA NA OSNOVI PRODUKTHNIH LINIJ
(BOCKLE, CLEMENTS, MCGREGOR, MUTHIG, & SCHMID 2004)

Na osnovi potrjenih empiričnih raziskav (Bockle, Clements, McGregor, Muthig, & Schmid 2004) in ekonomskega modela produktnih linij je možno trditi, da ima obseg uporabe ogrodij pozitiven vpliv na uspešnost ogrodij oziroma da je obseg uporabe ogrodij determinanta uspešnosti ogrodij. **S to ugotovitvijo smo potrdili predpostavko P4, ki je bila predstavljena v uvodnem poglavju.**



SLIKA 36: DETERMINANTE USPEŠNOSTI OGRODIJ

Fokus nadaljnjih raziskav disertacije bosta identifikacija in potrditev faktorjev, ki vplivajo na obseg uporabe ogrodij, pri čemer smo v disertaciji zavestno prezrli potencialne druge faktorje, ki vplivajo na uspešnost ogrodij (Slika 36).

3.4 TERENSKA RAZISKAVA: IDENTIFIKACIJA DETERMINANT SPREJETOSTI OGRODIJ

Osnovni cilj terenske raziskave, ki smo jo izvedli v okviru disertacije, je bil identificirati poglobitve faktorje, ki vplivajo na uporabnikovo odločitev, da bo izbral in uporabljal določeno ogrodje. S terensko raziskavo smo torej želeli podati odgovor na prvo raziskovalno vprašanje (RV₁), ki se glasi: »Kateri faktorji določajo sprejetost ogrodi in kako jih lahko merimo?«

Kvalitativne študije omogočajo alternativni pristop izvedbe kompleksnih raziskav in pomagajo argumentirati obstoječe raziskave na področju sprejetosti kompleksnih izdelkov in sistemov.

(Neuman 2005)

Terenska raziskava je bila načrtovana in izvedena skladno z metodo fokusne skupine (glej poglavje 3.1.2.1).

3.4.1 NAČRTOVANJE FOKUSNE SKUPINE

Fokusno skupino smo zasnovali v skladu s procesom izvedbe empiričnih raziskav⁵⁸, natančneje, s procesom fokusne skupine, ki vsebuje naslednje aktivnosti (Neuman 2005):

- opredelitev ciljev raziskave,
- opredelitev udeležencev raziskave,
- načrtovanje fokusnih skupin in vsebine vprašalnika in
- načrtovanje zapisovanja in analize rezultatov.

3.4.1.1 OPREDELITEV CILJEV RAZISKAVE

V okviru raziskave smo definirali primarne in sekundarne cilje. Primarna cilja sta naslednja:

- Na osnovi kvalitativnih podatkov, pridobljenih od uporabnikov ogrodi, identificirati faktorje, ki vplivajo na izbiro oziroma odločitev o uporabi določenega ogrodja.
- Faktorje, pridobljene z raziskavo, primerjati s faktorji obstoječih in relevantnih standardov in drugih empirično preizkušenih modelov.

⁵⁸ Postopek izvedbe empiričnih raziskav je sestavljen iz naslednjih korakov (Rai, Lang, & Welker 2002): (1) definicija raziskave, (2) načrtovanje raziskave, (3) realizacija raziskave, (4) izvedba raziskave, (5) analiza pridobljenih podatkov in (6) pisanje poročila o raziskavi.

3.4.1.2 OPREDELITEV UDELEŽENCEV IN FOKUSNE SKUPINE

Mesto terenske raziskave predstavlja eno ali več lokacij, kjer se izvaja terenska raziskava. Pri izbiri mesta terenske raziskave je treba upoštevati naslednje kriterije (Neuman 2005):

- možnost pridobitve zadostnega obsega podatkov,
- tujost mesta (raziskovalec in udeleženci se ne poznajo),
- ustreznost mesta (praktični razlogi, kot so: fizična razdalja, časovne omejitve ipd).

Gledano s sociološkega vidika, sestavlja skupino množica ljudi, ki imajo skupne interese. Ljudje v skupini med sabo komunicirajo, sprejemajo pravila oziroma obveznosti skupnosti in si delijo skupno identiteto.

Raziskava je bila zaradi navedenih kriterijev in značilnosti spletne fokusne skupine (Tabela 4) izvedena v okviru spletne oziroma virtualne skupnosti. Virtualna skupnost je skupina (skupina v sociološkem pomenu), katere člani se družijo z uporabo informacijskih tehnologij. Poglavitni medij spletne skupnosti je internet. Poglavitna značilnost, ki ločuje navidezno (virtualno) skupnost ljudi od »fizične« skupnosti ljudi, je v zamenjavi komunikacijskih kanalov. Medtem ko se pri »fizični« skupnosti za komunikacijo uporablja pretežno govor, se v virtualni skupnosti uporablja spletni medij v najrazličnejših oblikah. Implikacije zamenjave komunikacijskih medijev izničijo zahteve po prostorskih in časovnih omejitvah sodelovanja.

Spletne skupnosti oziroma mesta, ki so bila izbrana kot kandidati za izvedbo spletno zasnovane fokusne skupine, so se identificirala na osnovi skladnosti s ciljem raziskave. V ožji izbor so se uvrstila mesta, predstavljena v spodnji tabeli (Tabela 6):

TABELA 6: KANDIDATNE SPLETNE SKUPNOSTI ZA IZVEDBO RAZISKAVE

Spletna skupnost	Focus on Java technology	Slashdot	JCorporate
URL	http://java.about.com	www slashdot.org	www.jcorporate.com
Tema skupnosti	Tehnologija Java, specialne kategorije tem	Tehnologija na splošno, specialne kategorije tem	Programsko ogrodje Espresso
Število registriranih uporabnikov	Ni podatka	880.000+	34.000+
Pogostost objav	Srednja	Visoka	Srednja
Anonimnost raziskovalca⁵⁹	Da	Da	Ne

⁵⁹ Stopnja udeležnosti raziskovalca v terenski raziskavi se razteza od oddaljenega raziskovalca (*detached outsider*) do popolnoma vključenega raziskovalca (*intimately involved insider*). Nivo udeležnosti je odvisen od dejavnikov, kot so: specifične mesta raziskave in vloge raziskovalca. V okviru naše raziskave smo želeli minimizirati vpliv raziskovalca na udeležence raziskave, čemur ustreza vloga opazovalca.

Spletna skupnost, ki je najbolj ustrezala kriterijem za izvedbo terenske raziskave, je bila Slashdot⁶⁰. Slashdot (pogosto okrajšan kot »/.«) je priljubljena tehnološko usmerjena spletna skupnost, ki povzema in referencira aktualne teme z najrazličnejših tehnoloških področij. Slashdot ima več kot 880.000 uporabnikov, ki so na Slashdotu zbrali že preko 14,000.000 objav⁶¹. Slashdot zaseda prvo mesto med tehnološkimi forumi.

Slashdot vključuje storitev »Ask Slashdot«, ki omogoča postavljanje vprašanj skupnosti njegovih bralcev. Povzetke in tudi vprašanja storitve pregledajo uredniki skupnosti⁶². Omenjeno storitev smo uporabili v okviru naše raziskave.

3.4.1.3 NAČRTOVANJE FOKUSNIH SKUPIN IN VSEBINE VPRAŠALNIKA

Fokusne skupine, ki se izvajajo v fizičnem okolju, se običajno izvedejo v skupini, pri čemer posamezna raziskava vključuje od 6 do 12 članov. Razlogi za omejitve so predvsem v prostorskih in časovnih omejitvah fizičnih skupin. Prav tako je težavno obvladovati komunikacijo pri večjem številu udeležencev.

V okviru raziskave smo izvedli spletno različico fokusne skupine, zato velikost skupine ni povzročala tehničnih ali organizacijskih težav. Zaradi uporabe spletnega foruma, ki predstavlja asinhroni tip komunikacijskega medija, ni bilo treba določiti časovnih okvirov, ki veljajo za fizično fokusno skupino.

V okviru naše raziskave so bili udeleženci vnaprej nedefinirani, delno anonimni in popolnoma svobodni pri odločitvi o sodelovanju. Odgovor na zastavljeno raziskovalno vprašanje so podali člani spletne skupnosti Slashdot, ki so dojemali vprašanje kot koristno, in člani, ki so želeli podati lastne izkušnje. Slabosti takšne izvedbe spletne raziskave so bile (1) nepoznano število udeležencev raziskave in (2) nepoznana profil in vzorec udeležencev.

Vsebino in obliko vprašalnika, ki je bil objavljen v okviru storitve »Ask Slashdot«, so pogojevali naslednji dejavniki:

- **Raziskovalno vprašanje.** Na jedro vsebine vprašanja je v prvi vrsti vplivalo raziskovalno vprašanje.
- **Tehnološke zmožnosti in omejitve.** Skladno z zmožnostmi storitve »Ask Slashdot« je vsebina vprašalnika zajemala eno vprašanje odprtega tipa. Vprašanje je bilo oblikovano s

⁶⁰ www.slashdot.org

⁶¹ Statistika je povzeta po viru »Big boards«, dostopnem na naslovu <http://www.big-boards.com>.

⁶² Verjetnost objave vprašanja na skupnost Ask Slashdot je 20%.

hipertekstom (poudarjanje pomembnih delov vprašanja in hiperpovezovanje na druge vire).

- **Govorica spletne skupnosti.** S ciljem približati vprašanje skupnosti Slashdot se je vprašanje slovnično in oblikovno prilagodilo že sprejetim priljubljenim vprašanjem in govorici spletne skupnosti. Obliko vprašanja so pogojevala tudi priporočila urednikov skupnosti Slashdot. Vprašanje je bilo objavljeno v angleškem jeziku.
- **Klasifikacija interesnih področij.** Vprašanje je bilo klasificirano v področje »programiranje« (*programming*).

Na osnovi naštetih dejavnikov se je oblikovalo naslednje vprašanje: »Na podlagi česa se odločate o uporabi ogrodja?⁶³«.

V nadaljevanju je bilo k temu vprašanju podano pojasnilo, ki je vsebovalo⁶⁴:

- Kratko predstavitev priljubljenosti, pomembnosti in najvidnejših predstavnikov ogrodij.
- Predstavitev pomembnosti dolgoročne uporabe ogrodij in priljubljenosti ogrodij.
- Usmerjanje bralca k podajanju pravih in konstruktivnih odgovorov.

Na sprejeto in objavljeno vprašanje je bilo v enem tednu podanih 291 odgovorov⁶⁵. Pri tem jih je bila večina (248 odgovorov) podana na dan objave vprašanja.

⁶³ »How Do You Decide Which Framework to Use?«

⁶⁴ Vprašanje in odgovori so dostopni na <http://ask slashdot.org/article.pl?sid=06/02/24/0148201>.

⁶⁵ Vključno z odgovori na odgovore.

3.4.2 ANALIZA REZULTATOV FOKUSNE SKUPINE

Vhod v analizo podatkov fokusne skupine so predstavljali odgovori, ki so jih člani skupnosti *Slashdot* podali na zastavljeno vprašanje. Za potrebe analize podatkov so se odgovori s strani *Slashdot*⁶⁶ shranili v izvorni, hipertekstovni obliki. Pri tem je hipertekstovna datoteka obsegala 119 strani teksta v urejevalniku MS Word (Slika 37).

Statistika:	
Strani	119
Besede	35.343
Znaki (brez presledkov)	173.120
Znaki (s presledki)	208.203
Odstavki	1.616
Vrstice	4.377
<input type="checkbox"/> Vključi besedilna polja, sprotne in končne gumbice	
Zapri	

SLIKA 37: OPISNA STATISTIKA NEOBDELANIH PODATKOV

Število vseh podanih odgovorov je bilo 291, kar vključuje tudi vse odgovore na odgovore. Analiza števila odgovorov, ki so jih člani skupnosti podali na druga zastavljena vprašanja, je pokazala, da je bilo na zastavljeno vprašanje podanih nadpovprečno število odgovorov⁶⁷.

3.4.2.1 PODATKI IN METODA ANALIZE PODATKOV

Odgovori, ki so jih posredovali člani skupnosti *Slashdot*, so bili podani v kvalitativni obliki. Kvalitativni podatki so nenumerični in s tem drugačni od kvantitativno opisanih podatkov. Njihova vsebina navadno predstavlja neko notranjo strukturo oziroma sestavo, odnose ali povezave med posameznimi sestavinami in podobno. Kvalitativni podatki so lahko: (1) nesestavljeni, kot so: vozlišča grafa, posli stroja, ponudniki in porabniki storitve, ali (2) sestavljeni iz enostavnejših podatkov, kot so: urejeni pari, množice, permutacije in grafi.

V preteklosti je bila osnovna pomanjkljivost kvalitativnih raziskav pomanjkanje sistematičnega pristopa h kvalitativni analizi podatkov, saj se ti analizirajo drugače kot kvantitativni podatki. Kljub temu da še vedno ne obstaja enotna in uveljavljena metoda za kvalitativno analizo podatkov, postajajo kvalitativne metode vse bolj sistematične (Neuman 2005).

⁶⁶ Surovi podatki se nahajajo na lokaciji <http://ask.slashdot.org/article.pl?sid=06/02/24/0148201>.

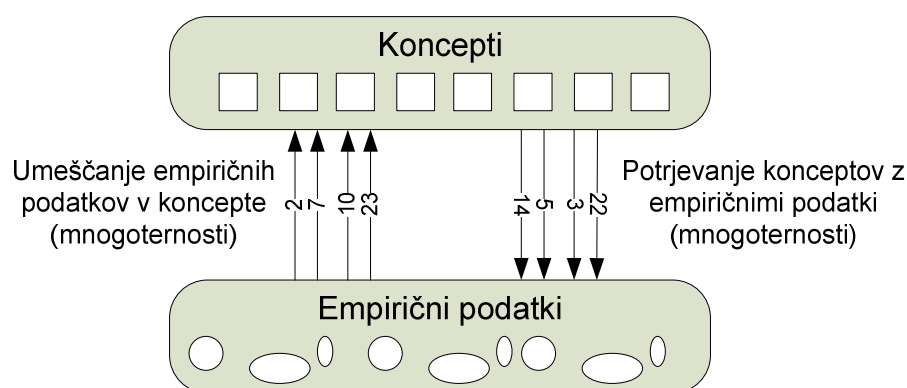
⁶⁷ Analiza je obsegala 30 vprašanj, ki so bila objavljena med 11. 2. 2006 in 25. 2. 2006. Na osnovi analiziranih vprašanj smo ugotovili, da je bilo v povprečju podanih 190 odgovorov na zastavljeno vprašanje. Standardna deviacija je znašala 161 odgovorov.

Osnovni namen kvalitativne analize podatkov je, na osnovi neobdelanih kvalitativnih podatkov (*raw data*) oblikovati nove ali izboljšati že obstoječe koncepte. Pri tem raziskovalec oblikuje definicije konceptov in raziskuje njihove medsebojne povezanosti. Med splošne tehnike kvalitativne analize podatkov spada »kodiranje«.

Neuman (2005) opisuje kodiranje kvalitativnih podatkov kot sestavni del analize podatkov, kjer razvijalec ureja neobdelane podatke v konceptualne kategorije in s tem oblikuje nove teme ali koncepte. Osnovni namen kodiranja podatkov je redukcija velikih količin podatkov v majhne in obvladljive podatkovne enote. Kodiranje podatkov je vodeno na osnovi raziskovalnega vprašanja in usmerja raziskovalca k oblikovanju novih raziskovalnih vprašanj, generalizacij in teorij. Kodiranje podatkov sestavljata dve sočasno izvajani aktivnosti: mehanična redukcija podatkov in analitična kategorizacija podatkov. Pri tem je granulacija kodiranja (na primer: posamezne besede, posamezne povedi ali odstavki) odvisna od raziskovalnega vprašanja, zgoščenosti zbranih podatkov in namena raziskave.

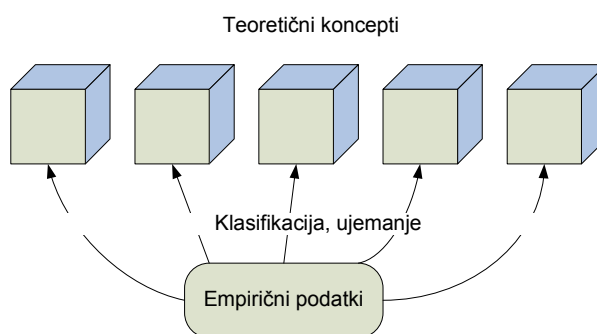
Tehnike kodiranja podatkov se uvrščajo med splošne tehnike analize kvalitativnih podatkov. Te raziskovalci pogosto dopolnjujejo s specialnimi, analitičnimi tehnikami, kot so: postopno ali sukcesivno približevanje (*successive approximation*), ponazorilna metoda (*illustrative method*), analiza domene (*domain analysis*), analitične primerjave (*analytic comparison*) in metoda negativnega primera (*negative case method*) (Neuman 2005). Za potrebe naše raziskave je bila poleg tehnike kodiranja uporabljena še kombinacija postopnega približevanja in ponazorilne metode.

Postopno približevanje je metoda analize kvalitativnih podatkov, v kateri se raziskovalec pomika med empiričnimi podatki in abstraktnimi koncepti, teorijami ali modeli z namenom izboljševanja teorije oziroma konceptov (Slika 38).



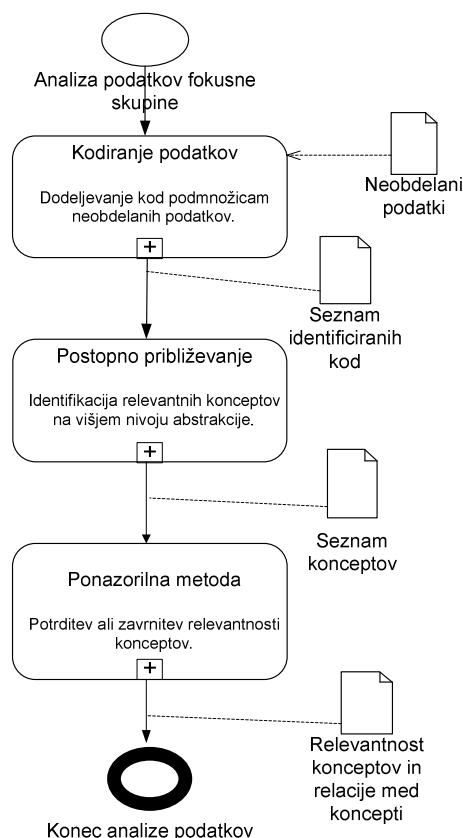
SLIKA 38: GRAFIČNA PONAZORITEV POSTOPNEGA PRIBLIŽEVANJA (NEUMAN 2005)

Pri ponazorilni metodi poskuša raziskovalec na osnovi empiričnih podatkov najti podporo za določeno teorijo oziroma koncepte. Pri tem je lahko obstoječa teorija ponazorjena s praznimi škatlami, ki jih poskuša raziskovalec zapolniti z empiričnimi podatki (Slika 39).



SLIKA 39: GRAFIČNA PREDSTAVITEV PONAZORILNE METODE (Meier 1998)

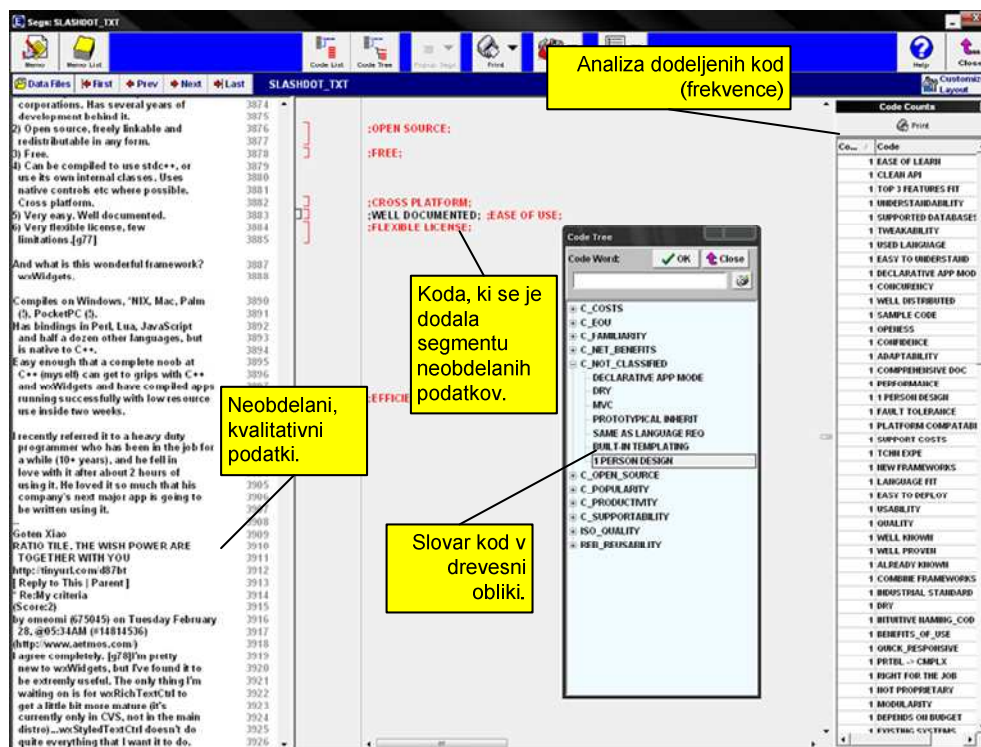
Na osnovi uporabljenih tehnik analize kvalitativnih podatkov, predmeta raziskave in metode raziskave je bil definiran postopek analize podatkov fokusne skupine, ki je v obliki notacije BPMN predstavljen na spodnji sliki (Slika 40).



SLIKA 40: PROCES ANALIZE PODATKOV FOKUSNE SKUPINE

Za potrebe kvalitativne analize podatkov smo uporabili beta različico orodja »Ethnograph 6.0«. Ethnograph je orodje, ki je namenjeno analizi kvalitativnih podatkov. Orodje omogoča urejanje,

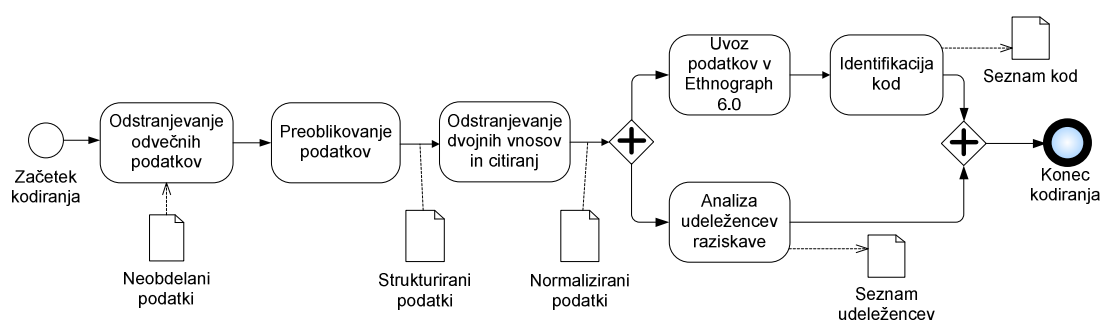
kodiranje in komentiranje podatkov, ki so zapisani v tekstovni obliki. Pridobljene kode se urejajo s slovarjem kod (*code book*), ki se lahko oblikuje v seznam ali drevo (Slika 41).



SLIKA 41: ZASLONSKA SLIKA ORODJA ETHNOGRAPH S POJASNILI

3.4.2.2 IZVEDBA IN REZULTATI KODIRANJA PODATKOV

Proces kodiranja podatkov je obsegal mehanično redukcijo neobdelanih podatkov in identifikacijo kod, kot je prikazano na spodnji sliki (Slika 42).



SLIKA 42: POSTOPEK REDUKCIJE PODATKOV IN ODPRTEGA KODIRANJA

Aktivnost odstranjevanja odvečnih podatkov je iz neobdelanih podatkov odstranila vse vključene objekte (slike, skripte, stilne predloge) in elemente, ki niso skladni z XHTML. Sledilo je preoblikovanje podatkov v strukturirano obliko neoznačenega besedila. V aktivnosti odstranjevanja dvojnih vnosov in citiranj so bili odstranjeni večkratni vnosi in citiranja

predhodnih odgovorov. Izhodni podatki aktivnosti so predstavljali vhod v sočasno identifikacijo kod in analizo udeležencev raziskave. Identifikacija kod je temeljila na raziskovalnem vprašanju in na vprašanju, zastavljenem skupnosti *Slashdot*. Rezultat odprtega kodiranja je predstavjal seznam identificiranih kod (Tabela 7) in seznam udeležencev raziskave.

TABELA 7: SEZNAM IDENTIFICIRANIH KOD IN PRIPADAJOČIH KONCEPTOV⁶⁸

Koda v slovenskem jeziku	Koda v angleškem jeziku	$m(a)$ ⁶⁹	Koncept v angleškem jeziku ⁷⁰
Enostaven za uporabo	<i>Ease of use</i>	12	<i>Ease of use</i>
Preprost	<i>Simplicity</i>	9	<i>Complexity</i>
Skladen z zahtevami	<i>Framework requirements fit</i>	8	<i>Suitability</i>
Prenosljiv	<i>Portability</i>	7	<i>Portability</i>
Razširljiv	<i>Extensibility</i>	5	<i>Adaptability</i>
Lahek	<i>Lightweight</i>	5	<i>Complexity</i>
Neodvisen od platforme	<i>Cross platform</i>	5	<i>Environment independence</i>
Zmožnost vzdrževanja	<i>Maintainability</i>	5	<i>Maintainability</i>
Zrel	<i>Maturity</i>	5	<i>Maturity</i>
Stabilen	<i>Stability</i>	5	<i>Stability</i>
Kakovost dokumentacije	<i>Documentation quality</i>	4	<i>Documentation quality</i>
Zmogljivosti	<i>Efficiency</i>	4	<i>Efficiency</i>
Poznan	<i>Familiarity</i>	4	<i>Familiarity</i>
Zmožnost učenja	<i>Learnability</i>	4	<i>Learnability</i>
Priljubljenost	<i>Popularity</i>	4	<i>Popularity</i>
Zmožnost hitrega razvoja	<i>Rapid Application Development</i>	4	<i>Productivity</i>
Medobratovalnost	<i>Interoperability</i>	3	<i>Interoperability</i>
Prednosti uporabe ogrodja	<i>Benefits of use</i>	3	<i>Net benefits</i>
Zmožnost zamenjave ogrodja	<i>Replaceability</i>	3	<i>Replaceability</i>
Odločitev nadrejenega	<i>Decided by boss</i>	2	Not classified
Uporaba vzorca MVC	<i>Model view controller</i>	2	Not classified
Brezplačen	<i>Free</i>	2	<i>Costs</i>
Dobra dokumentiranost	<i>Well documented</i>	2	<i>Documentation quality</i>
Robusten	<i>Robustness</i>	2	<i>Fault tolerance</i>
Zmožnost namestitve	<i>Installability</i>	2	<i>Installability</i>
Prilagodljiva licenca	<i>Flexible license</i>	2	<i>Legal</i>
Skladen z zahtevami za licenco	<i>License fit</i>	2	<i>Legal</i>

⁶⁸ Seznam je urejen po frekvenci kod, v padajočem vrstnem redu.

⁶⁹ Štetje enakih kod (f.) se je izvajalo na osnovi kode v angleškem jeziku.

⁷⁰ Klasifikacija v koncept se je izvajala na osnovi kode v angleškem jeziku.

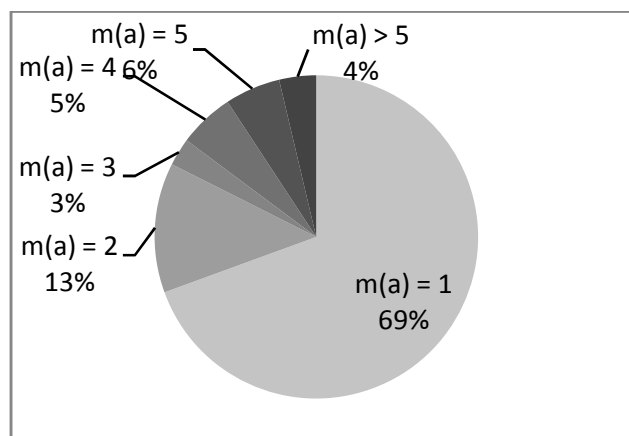
Koda v slovenskem jeziku	Koda v angleškem jeziku	$m(a)$ ⁶⁹	Koncept v angleškem jeziku ⁷⁰
Temelječ na odprti kodi	<i>Open source based</i>	2	<i>Legal</i>
Mnenja drugih	<i>Others opinions</i>	2	<i>Popularity</i>
Dober stil kodiranja	<i>Coding style</i>	2	<i>Selfdescriptiveness</i>
Skladen z uporabljenim programskim jezikom	<i>Fit to language used</i>	2	<i>Suitability</i>
Odvisno od podpore	<i>Support</i>	2	<i>Supportability</i>
Podporna orodja	<i>Tool support</i>	2	<i>Supportability</i>
Veliko faktorjev	<i>A lot of things to consider</i>	1	Not classified
Ima vgrajene predloge	<i>Built-in templating</i>	1	Not classified
Deklarativni model	<i>Declarative application model</i>	1	Not classified
Učenje na lastnih napakah	<i>Do not repeat yourself</i>	1	Not classified
Zasnovan od ene osebe	<i>One person design</i>	1	Not classified
Enake zahteve kot za izbiro programskega jezika	<i>Same as language requirements</i>	1	Not classified
Uporabi sodobna ogrodja	<i>Use new frameworks</i>	1	Not classified
Prilagodljiv	<i>Adaptability</i>	1	<i>Adaptability</i>
Fleksibilen	<i>Flexibility</i>	1	<i>Adaptability</i>
Zmožnost kombiniranja ogrodij	<i>Combine frameworks</i>	1	<i>Co-existence</i>
Minimalističen programski kod	<i>Minimalistic code</i>	1	<i>Complexity</i>
Ni kompleksen	<i>Not complex</i>	1	<i>Complexity</i>
Majhen	<i>Small size</i>	1	<i>Complexity</i>
Vreden zaupanja	<i>Confidence</i>	1	<i>Confidence</i>
Zanesljiv	<i>Dependability</i>	1	<i>Confidence</i>
Obstojen	<i>Sustainability</i>	1	<i>Confidence</i>
Odvisno od stroškov uporabe	<i>Costs</i>	1	<i>Costs</i>
Stroški podpore	<i>Support costs</i>	1	<i>Costs</i>
Odvisno od finančnih sredstev	<i>Depends on budget</i>	1	<i>Costs</i>
Obsežna dokumentacija	<i>Comprehensive documentation</i>	1	<i>Documentation quality</i>
Zmožen sočasnega izvajanja	<i>Concurrency</i>	1	<i>Efficiency</i>
Zmogljivosti	<i>Performance</i>	1	<i>Efficiency</i>
Hitra odzivnost	<i>Quick response times</i>	1	<i>Efficiency</i>
Hiter	<i>Speed</i>	1	<i>Efficiency</i>
Neodvisen od platforme	<i>Platform independence</i>	1	<i>Environment independence</i>
Podprte podatkovne baze	<i>Supported databases</i>	1	<i>Environment independence</i>
Poznan	<i>Already known</i>	1	<i>Familiarity</i>
Dobro poznan	<i>Well known</i>	1	<i>Familiarity</i>
Uporabi, kar poznaš	<i>What you know</i>	1	<i>Familiarity</i>

Koda v slovenskem jeziku	Koda v angleškem jeziku	$m(a)^{69}$	Koncept v angleškem jeziku ⁷⁰
Odvisno od tehničnih sposobnosti	<i>Technical experience with</i>	1	<i>Familiarity</i>
Odporen proti napakam	<i>Fault tolerance</i>	1	<i>Fault tolerance</i>
Ne previsoke zahteve	<i>Not steep requirements</i>	1	<i>Installability</i>
Enostaven za namestitev	<i>Easy to deploy</i>	1	<i>Installability</i>
Temelječ na standardih	<i>Standard based</i>	1	<i>Interoperability</i>
Enostaven za učenje	<i>Ease of learn</i>	1	<i>Learnability</i>
Intuitiven	<i>Intuitive</i>	1	<i>Learnability</i>
Vzdrževanje z majhnimi popravki ogrodja	<i>Tweakability</i>	1	<i>Maintainability</i>
Uveljavljen	<i>Established</i>	1	<i>Maturity</i>
Zrelost uporabljenih knjižnic	<i>Libraries maturity</i>	1	<i>Maturity</i>
Preizkušen	<i>Well proven</i>	1	<i>Maturity</i>
Odvisno od zrnatosti	<i>Granularity</i>	1	<i>Modularity</i>
Lahka sklopljenost	<i>Louse coupling</i>	1	<i>Modularity</i>
Modularen	<i>Modularity</i>	1	<i>Modularity</i>
Nelastniški	<i>Not proprietary</i>	1	<i>Legal</i>
Oprtokoden	<i>Open source</i>	1	<i>Legal</i>
Ima ekosistem odprte kode	<i>Open source ecosystem</i>	1	<i>Legal</i>
Odprt	<i>Openness</i>	1	<i>Legal</i>
Aktivna skupnost uporabnikov	<i>Community activity</i>	1	<i>Popularity</i>
Standard industrije	<i>Industrial standard</i>	1	<i>Popularity</i>
Razširjen	<i>Well distributed</i>	1	<i>Popularity</i>
Razširjen	<i>Widespread use</i>	1	<i>Popularity</i>
Skupna produktivnost	<i>Overall productivity</i>	1	<i>Productivity</i>
Konsistentna kakovost	<i>Consistent quality</i>	1	<i>Quality</i>
Kakovost	<i>Quality</i>	1	<i>Quality</i>
Skladen s starejšimi različicami	<i>Backward compatibility</i>	1	<i>Replaceability</i>
Jasen programski vmesnik	<i>Clean API</i>	1	<i>Selfdescriptiveness</i>
Intuitivni programski kod	<i>Intuitive code</i>	1	<i>Selfdescriptiveness</i>
Berljiv	<i>Readability</i>	1	<i>Selfdescriptiveness</i>
Odvisno od kompleksnosti aplikacije	<i>Depends on application complexity</i>	1	<i>Suitability</i>
Popoln nabor funkcij	<i>Feature complete</i>	1	<i>Suitability</i>
Skladen z obstoječim sistemom	<i>Existing system fit</i>	1	<i>Suitability</i>
Odvisno od programskega jezika	<i>First choose language</i>	1	<i>Suitability</i>
Odvisno od zasnove ogrodja	<i>Framework design</i>	1	<i>Suitability</i>
Skladen z zahtevami zasnove ogrodja	<i>Framework design fit</i>	1	<i>Suitability</i>

Koda v slovenskem jeziku	Koda v angleškem jeziku	$m(a)^{69}$	Koncept v angleškem jeziku ⁷⁰
Odvisno od programskega jezika	<i>Language dependent</i>	1	<i>Suitability</i>
Skladen z uporabljenim programskim jezikom	<i>Language fit</i>	1	<i>Suitability</i>
Skladnost s programsko platformo	<i>Platform compatibility</i>	1	<i>Suitability</i>
Podprte platforme	<i>Platforms supported</i>	1	<i>Suitability</i>
Primeren za nalogo	<i>Right for the task</i>	1	<i>Suitability</i>
Skladen s tremi najpomembnejšimi zahtevami	<i>Top 3 features fit</i>	1	<i>Suitability</i>
Odvisno od uporabljenega jezika	<i>Used language</i>	1	<i>Suitability</i>
Ima primere programskega koda	<i>Sample code</i>	1	<i>Supportability</i>
Enostavno ga je razumeti	<i>Easy to understand</i>	1	<i>Understandability</i>
Razumljiv	<i>Understandability</i>	1	<i>Understandability</i>
Uporaben	<i>Usability</i>	1	<i>Usability</i>
Skupaj kod		202	

V postopku identifikacije kod sta bili identificirani 202 kodi, ki sta predstavljali odgovor na zastavljeno vprašanje. Ugotovili smo, da je odgovore posredovalo 57 različnih uporabnikov, izmed katerih jih je bilo 6 anonimnih in 51 z vzpostavljenimi uporabniškimi računi. Iz navedenega je možno sklepati, da se uporabnik v povprečju odloča o izbiri ogrodja na osnovi 3.5 faktorja. Iz seznama kod je razvidno (Tabela 7), da so se posamezne kode identificirale večkrat. Najpogosteje identificirana koda je bila »enostaven za uporabo«.

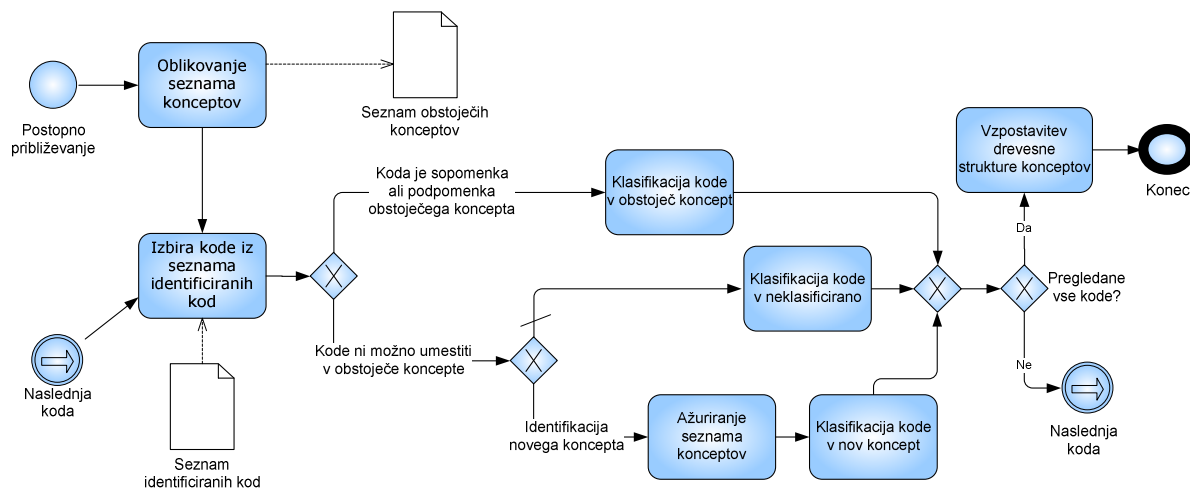
Na spodnjem tortnem grafikonu (Slika 43) so ponazorjeni deleži kod v odvisnosti od njihove frekvence. Iz grafikona je razvidno, da se je 69% kod pojavilo samo enkrat.



SLIKA 43: DELEŽ ENAKIH KOD V ODVISNOSTI OD NJIHOVE FREKVENCE

3.4.2.3 IZVEDBA IN REZULTATI POSTOPNEGA PRIBLIŽEVANJA

Cilj postopnega približevanja je bil, na osnovi podobnosti klasificirati kode v taksonomijo⁷¹ konceptov na višjem nivoju abstrakcije. Proces izvedbe postopnega približevanja je predstavljen na spodnji sliki (Slika 44).



SLIKA 44: PROCES IZVEDBE POSTOPNEGA PRIBLIŽEVANJA

Vhod v proces postopnega približevanja je predstavljal seznam identificiranih kod in začetni seznam konceptov (Slika 40). Ta se je oblikoval na osnovi: (1) karakteristik in podkarakteristik modela ISO/IEC 9126, (2) faktorjev in kriterijev modela ponovne uporabnosti REBOOT, (3) konceptov modela TAM in (4) konceptov modela uspešnosti IS. Vsaka posamezna koda se je nato poskušala umestiti v seznam začetnih konceptov na osnovi naslednjih semantičnih relacij (*R*):

- koda je enaka ali je sopomenka koncepta in
- koda je podpomenka koncepta.

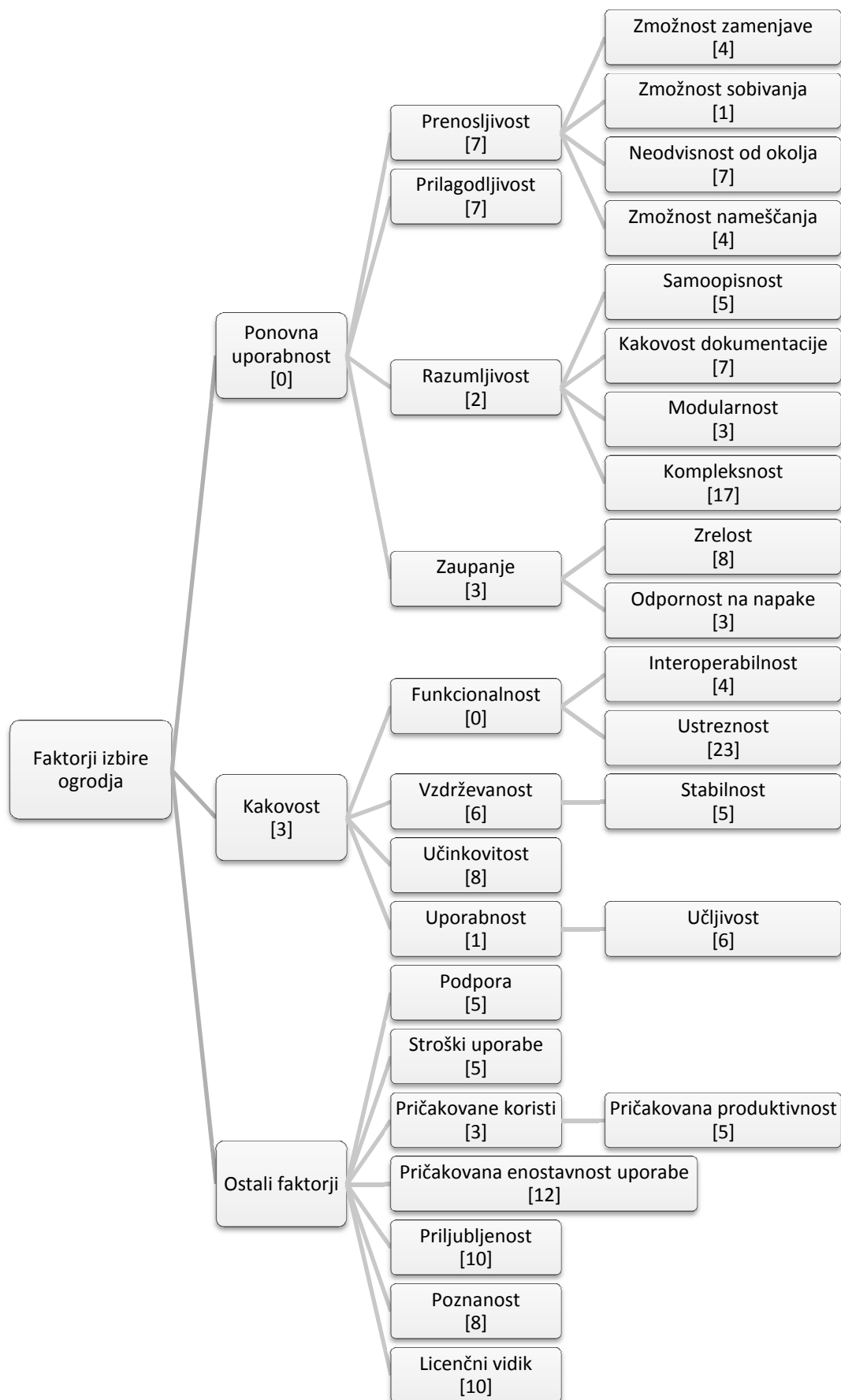
Če se koda ni mogla klasificirati v seznam konceptov, se je poskušal identificirati nov koncept oziroma nadpomenka, ki se je nato dodala v seznam konceptov. V nasprotnem primeru je ostala koda neklasificirana. Seznam konceptov in pripadajoči vir oziroma model sta predstavljena v naslednji tabeli (Tabela 8), pri čemer je povezava med koncepti in kodami zajeta v seznamu identificiranih kod (Tabela 7).

⁷¹ V matematiki je taksonomija drevesna zgradba za dano množico objektov.

TABELA 8: SEZNAM IDENTIFICIRANIH KONCEPTOV, UREJEN PO RELEVANTNOSTI

Koncept v slovenskem jeziku	Model	Koncept v angleškem jeziku	Relevantnost
Ustreznost	ISO	Suitability	23
Kompleksnost	REBOOT	Complexity	17
Dojeta enostavnost uporabe	TAM	Perceived ease of use	12
Neklasificirano	/	Not classified	11
Priljubljenost	/	Popularity	10
Licenčni vidik	/	Legal	10
Učinkovitost	ISO	Efficiency	8
Dobro poznavanje	/	Familiarity	8
Zrelost	REBOOT	Maturity	8
Prilagodljivost	REBOOT	Adaptability	7
Kakovost dokumentacije	REBOOT	Documentation quality	7
Neodvisnost od okolja	REBOOT	Environment independence	7
Prenosljivost	REBOOT	Portability	7
Zmožnost učenja	ISO	Learnability	6
Zmožnost vzdrževanja	ISO	Maintainability	6
Stroški uporabe	/	Costs	5
Produktivnost	IS Success	Productivity	5
Samoopisnost	REBOOT	Selfdescriptiveness	5
Stabilnost	ISO	Stability	5
Podpora	/	Supportability	5
Zmožnost namestitve	ISO	Installability	4
Medobratovalnost	ISO	Interoperability	4
Zmožnost zamenjave	ISO	Replaceability	4
Zaupanje	REBOOT	Confidence	3
Odpornost proti napakam	REBOOT	Fault tolerance	3
Modularnost	REBOOT	Modularity	3
Neto prednosti uporabe	IS Success	Net benefits	3
Kakovost	ISO	Quality	2
Razumljivost	REBOOT	Understandability	2
Zmožnost sobivanja	ISO	Co-existence	1
Uporabnost	ISO	Usability	1
Funkcionalnost	ISO	Functionality	0
Ponovna uporabnost	REBOOT	Reusability	0

Proces postopnega približevanja se je zaključil s preoblikovanjem seznama konceptov v drevesno obliko (Slika 45). Osnovo za vzpostavitev drevesne strukture so sestavljale povezave med koncepti pripadajočih modelov in struktura modela FCM (*factor-criteria-metrics*). Pri tem je najnižji nivo drevesa zajemal merila (*criteria*) v modelu FCM.



SLIKA 45: TAKSONOMIJA KONCEPTOV V DREVESNI OBLIKI

Iz prejšnje slike (Slika 45) je razvidno, da smo faktorje, po katerih se uporabniki odločajo o izbiri ogrodja, ločili na ponovno uporabo, kakovost in ostale faktorje. Osnovo za strukturo faktorjev so tvorile povezave v obstoječih modelih (poglavji 2.3 in 2.6).

3.4.2.4 IZVEDBA IN REZULTATI PONA ZORILNE METODE

Cilj ponazorilne metode je bil analizirati relevantnost konceptov na osnovi štetja frekvenc pripadajočih kod.

Relevantnost koncepta se je formalno definirala kot dvojica (C, e) , kjer C predstavlja množico konceptov in e funkcijo iz množice konceptov v množico naravnih števil ($e: C \rightarrow N$). Za vsak element c iz množice C se je definirala $e(c)$ kot vsota frekvenc vseh kod, ki so v semantični relaciji R s konceptom c :

$$\forall c \in C \wedge \forall a \in A: e(c) = \sum_{aRc} m(a)$$

Pri tem je A predstavljal množico kod ($a \in A$). Frekvenca kod je formalno definirana kot mnogoternost, ki označuje funkcijo $m(a)$. Posledično predstavlja dvojica (A, m) multimnožico (*multiset*), kjer je m funkcija iz množice A v množico naravnih števil ($m: A \rightarrow N$). Funkcija m je formalno definirana kot množica urejenih dvojic $\{(a, m(a)): a \in A\}$. Če je A končna množica, se velikost multimnožice (A, m) izračuna na naslednji način:

$$|(A, m)| = \sum_{a \in A} m(a)$$

Velikost množice A se ovrednoti brez mnogoternosti:

$$|A| = \sum_{a \in A} 1$$

Relevantnosti konceptov so podane v tabelarni (Tabela 8) in drevesni obliki (Slika 45). V spodnji tabeli (Tabela 9) je podana opisna statistika relevantnosti konceptov.

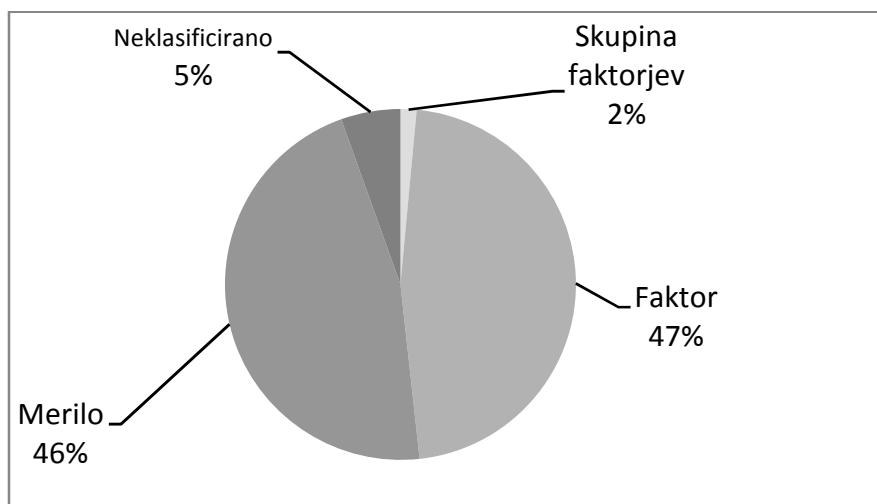
TABELA 9: OPISNA STATISTIKA RELEVANTNOSTI KONCEPTOV⁷²

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Relevantnost	34	,0000	18,0000	5,941176	4,1991426	17,633
Valid N (listwise)	34					

⁷² Tabela je prenesena iz orodja SPSS.

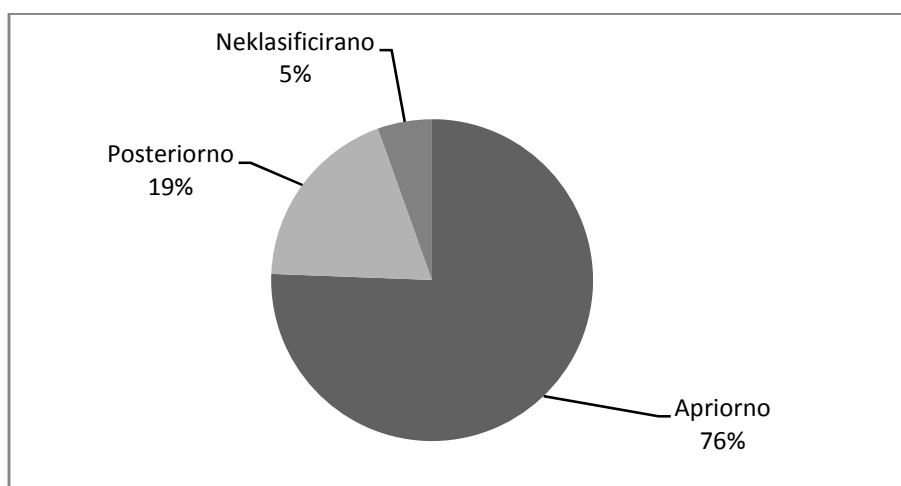
Izmed vseh identificiranih kod smo jih v koncepte klasificirali 191, pri tem jih je 11 ostalo neklasificiranih (Slika 46). Poglavitna razloga za nezmožnost klasificiranja kode v koncept sta bila: (1) prenizek nivo abstrakcije kode (na primer: deklarativni model razvoja, MVC) ali (2) previsok nivo abstrakcije kode (na primer: veliko faktorjev, enako kot zahteve za programski jezik).

Analiza klasifikacije kod glede na hierarhično strukturo konceptov je pokazala enakomerno porazdelitev kod na nivo faktorjev in meril (Slika 46).



SLIKA 46: ANALIZA KLASIFIKACIJE KOD GLEDE NA HIERARHIČNO STRUKTURO KONCEPTOV

Na spodnjem tortnem grafikonu je prikazano razmerje kod, ki so se klasificirale med apriorno definirane koncepte in posteriorno definirane koncepte.



SLIKA 47: ANALIZA KLASIFIKACIJE KOD GLEDE NA APRIORNO IN POSTERIORNO DEFINIRANJE KONCEPTOV

S ciljem zmanjševanja števila konceptov smo izvedli združevanje konceptov (*concept clustering*) tako, da smo glede na model FCM (*factor-criteria-metrics*) merila združili v faktorje. Pri tem smo faktorjem prišteli relevantnosti pripadajočih meril. Rezultati so podani v naslednji tabeli.

TABELA 10: RELEVANTNOSTI FAKTORJEV PO IZVEDBI ZDRUŽEVANJA KONCEPTOV

Faktor v slovenskem jeziku	Model	Faktor v angleškem jeziku	Relevantnost
Razumljivost	REBOOT	<i>Understandability</i>	34
Funkcionalnost	ISO	<i>Functionality</i>	27
Prenosljivost	REBOOT	<i>Portability</i>	23
Zaupanje	REBOOT	<i>Confidence</i>	14
Dojeta enostavnost uporabe	TAM	<i>Perceived ease of use</i>	12
Vzdrževanost	ISO	<i>Maintainability</i>	11
Priljubljenost	/	<i>Popularity</i>	10
Licenčni vidik	/	<i>Legal</i>	10
Učinkovitost	ISO	<i>Efficiency</i>	8
Dobro poznavanje	/	<i>Familiarity</i>	8
Pričakovane koristi	IS Success	<i>Net benefits</i>	8
Prilagodljivost	REBOOT	<i>Adaptability</i>	7
Uporabnost	ISO	<i>Usability</i>	7
Stroški uporabe	/	<i>Costs</i>	5
Podpora	/	<i>Supportability</i>	5

3.4.3 INTERPRETACIJA REZULTATOV FOKUSNE SKUPINE

Cilj izvedbe fokusne skupine je bil odgovoriti na raziskovalno vprašanje RV₁. V raziskavi je sodelovalo 57 uporabnikov ogrodij. Na osnovi njihovih odgovorov smo identificirali 108 različnih kod, ki vplivajo na njihove odločitve o izbiri ali nadaljnji uporabi ogrodja. Pri tem se je kot najbolj pogosta koda izkazala »enostavnost uporabe ogrodja«.

S ciljem zmanjšati število identificiranih kod na obvladljiv obseg in povezati identificirane kode z obstoječimi koncepti smo izvedli dve tehniki kvalitativne analize podatkov: postopno približevanje in ponazorilno metodo.

V okviru postopnega približevanja smo kode, ki smo jih identificirali iz pridobljenih podatkov, poskušali povezati z obstoječimi koncepti oziroma faktorji. Pri tem smo uporabili semantični relaciji »je sopomenka« in »je podpomenka«. Identificirane kode smo povezali s 30 faktorji, ki smo jih v 76% identificirali apriorno in v 19% posteriorno. Petih odstotkov kod nismo mogli povezati z bolj abstraktnimi faktorji. Na osnovi primerjave identificiranih konceptov s strukturo FCM (*factor-criteria-metrics*) smo ugotovili, da se uporabniki odločajo predvsem na nivoju faktorjev (*factor*) in meril (*criteria*). Drevesna struktura faktorjev, ki smo jo oblikovali na osnovi strukture modela FCM, temelji na treh skupinah faktorjev: ponovni uporabnosti ogrodja, kakovosti ogrodja in ostalih faktorjih, ki smo jih identificirali pretežno posteriorno.

S ponazorilno metodo smo poskušali ugotoviti relevantnost posameznih konceptov, in sicer s štetjem frekvenc pripadajočih kod. Ugotovili smo, da so najbolj relevantni koncepti povezani z ustreznostjo ogrodja in njegovo enostavnostjo uporabe oziroma kompleksnostjo. Za relevantne so se izkazali vsi faktorji in merila v modelu ponovne uporabnosti REBOOT. Z vidika modela kakovosti ISO/IEC 9126 velja izpostaviti faktorja učinkovitosti in vzdrževanosti, ki sta se izkazala s podobnim deležem relevantnosti kot faktorji iz modela ponovne uporabnosti.

S ciljem zmanjševanja števila konceptov smo izvedli še združevanje konceptov (*factor clustering*). Kot rezultat smo dobili 15 faktorjev iz modela ponovne uporabnosti, kakovosti in drugih faktorjev. Za najbolj relevantnega se je izkazal faktor razumljivosti.

Na osnovi raziskave je možno predpostavljati, da faktorji sprejetosti ogrodij vključujejo vse faktorje modela ponovne uporabnosti, podmnožico faktorjev modela kakovosti in druge, predvsem netehnološke faktorje, kot so: priljubljenost, pravni vidik in poznanost. Pri interpretaciji rezultatov je treba upoštevati omejitve raziskave, ki so predstavljene v naslednjem poglavju.

3.4.4 OMEJITVE RAZISKAVE

Pri interpretaciji in uporabi rezultatov raziskave je treba upoštevati naslednje omejitve: (1) omejitve metode raziskave, (2) omejitve vzorčnega okvira in (3) omejitve tehnik analize podatkov. Pri tem so bile slabosti fokusne skupine in značilnosti njene spletne različice predstavljene v poglavju 3.1.2.2. Poleg tega smo identificirali nekaj slabosti naše izpeljanke fokusne skupine.

Fokusno skupino smo izvedli na osnovi asinhronne komunikacije med udeleženci, kar je zaviralo hiter razvoj konstruktivnih diskusij. S ciljem minimalnega vpliva na udeležence raziskave je ostal moderator diskusije v vlogi pasivnega opazovalca (*detached outsider*), zato se je diskusija pogosto odmaknila od opredeljene tematike. Posledično je bil velik delež odgovorov nepovezan z definirano tematiko, kar je zmanjšalo delež identificiranih kod glede na obseg pridobljenih podatkov.

Vzorčni okvir naše populacije je bil namenski in nedeterminiran; v diskusijo so se ljudje vključili prostovoljno, na osnovi lastne motivacije. Iz tega razloga je rezultate naše raziskave nemogoče posplošiti na populacijo oziroma na vse uporabnike ogrodiv. Rezultati fokusne skupine bi pridobili zunanjo veljavnost, če bi se enake raziskave izvedle še v drugih spletnih skupnostih.

Kljub temu da smo se analize podatkov lotili sistematično, se subjektivnega vpliva ni dalo povsem odpraviti. V postopku identifikacije kod so se te identificirale neposredno ali posredno iz pridobljenih podatkov. Posredni pristop pomeni, da smo kodo identificirali na osnovi vsebine povedi ali odstavka, pri čemer se ta ni pojavila v besedilu. Rezultati kvalitativne analize podatkov so se pridobili analitično z uporabo »mehkih« semantičnih relacij med kodami in koncepti. Pri tem so se koncepti in povezave med koncepti (strukturiranje konceptov) povzeli iz modelov kakovosti in ponovne uporabnosti programske opreme. Zavedamo se, da v kolikor bi se uporabila drugačen nabor konceptov in obstoječih modelov, bi se lahko končni koncepti in njihova struktura signifikantno razlikovali.

Poleg vključevanja večjega števila ekspertov v kvalitativno analizo podatkov, bi lahko za izboljšanje natančnosti in ponovljivosti analize podatkov, izvedli postopke, ki jih je predlagal Trochim (1989):

- **Zanesljivost ponovnega testa** (*test-retest-reliability*). Izvedli bi analizo ponovljivosti raziskave, kjer bi podatke na enak način analizirali po preteku določenega časovnega obdobja. Pridobljene podatke prvega in drugega testa bi nato primerjali.

- **Zanesljivost razcepa podatkov** (*split-half-reliability*). Predstavlja naključen razcep pridobljenih podatkov na dva dela, neodvisno analizo obeh delov in primerjavo rezultatov.
- **Zanesljivost posameznih in skupnih rezultatov** (*individual-to-total-reliability*). Predstavlja primerjavo rezultatov posameznikov glede na skupne podatke.

3.5 PREGLED LITERATURE: IDENTIFIKACIJA DETERMINANT SPREJETOSTI OGRODIJ

Cilj raziskave je bil enak kot v predhodno izvedeni terenski raziskavi. Razlika je bila v podatkih raziskave, ki smo jih v tej raziskavi pridobili na osnovi pregleda literature. Raziskovalno vprašanje se je glasilo: »Katere faktorje, ki vplivajo na sprejetost in uspešnost ogrodi, je možno zaslediti v obstoječi literaturi?«

3.5.1 PROCES IN REZULTATI PREGLEDA LITERATURE

Identifikacija faktorjev je bila na osnovi pregleda literature izvedena na naslednji način. Definiran je bil iskalni prostor, ki je vključeval monografije, disertacije, tehnična poročila, prispevke, objavljene v znanstvenih revijah, in prispevke, predstavljene na znanstvenih konferencah. Iskanje je potekalo po elektronskih knjižnicah oziroma elektronskih zbirkah, kot so: »Citeseer«, »Science Direct« in »IEEE«. Termini, ki so predstavljali vhodni podatek v iskalnike, so bili oblikovani na osnovi⁷³: (1) predmeta raziskave (na primer: aplikacijska ogrodja, programska ogrodja, objektno orientirana ogrodja), (2) področja raziskave (na primer: uspešnost, kakovost, sprejetost) in (3) specifičnih terminov (na primer: determinante, smernice, razlogi, faktorji). Proces iskanja⁷⁴ se je definiriral kot nasičen, ko kombinacije iskalnih terminov niso podale novih rezultatov poizvedovanja. Rezultat procesa iskanja je bila množica identificiranih virov, ki so se analizirali s tehnikami kodiranja podatkov (glej poglavje 3.4.2.1). Pri tem so se kode v identificiranih virih pojavile neposredno ali so bile vključene v tekstovnih opisih.

TABELA 11: REZULTATI PREGLEDA LITERATURE

Naslov vira	Metoda raziskave	Odvisen koncept raziskave	Identificirane kode v angleškem jeziku
Critical success factors of object-oriented frameworks (van Gurp & Bosch 2001)	Lastne izkušnje (desetletne izkušnje pri uporabi trinajst in razvoju sedem ogrodi)	Uspešnost ogrodi	<i>Easy to learn</i> <i>Qualitative internal</i> <i>Good support</i> <i>Efficiency</i> <i>Appropriateness</i>
Design, implementation and evolution of object oriented frameworks:	Izkušnje raziskovalne skupine pri delu s	Izboljšave ogrodi	<i>Flexibility</i> <i>Reusability</i> <i>Usability</i>

⁷³ Termini so bili v iskalnik podani v angleškem jeziku.

⁷⁴ Proces iskanja je bil izveden 20. 11. 2007.

Naslov vira	Metoda raziskave	Odvisen koncept raziskave	Identificirane kode v angleškem jeziku
concepts and guidelines (Fontoura 1999)	projekti		
Object-Oriented Application Frameworks: the Untold Story (Fayad & Schmidt 1997)	Izkušnje pri delu s projekti	Faktorji, ki vplivajo na neuspeh razvoja in uporabe ogrodij	<i>Handling complexity</i> <i>Insufficient support</i> <i>Adequate documentation</i> <i>Support for maintenance</i>
Object-oriented application frameworks (Froehlich, Hoover, Liu, & Sorenson 1998)	Lastne izkušnje	Učinkovita uporaba ogrodij	<i>Development effort</i> <i>Learning curve</i> <i>Integratability</i> <i>Maintainability</i> <i>Efficiency</i> <i>Lack of standards</i>
Designing object-oriented frameworks (Shai 1997)	Lastne izkušnje, pregled literature	Želene lastnosti ogrodij	<i>Ease of use</i> <i>Extensibility</i> <i>Flexibility</i> <i>Completeness</i> <i>Consistent coding</i>
Pattern Language for Framework Construction (Guido, Francesco, & Andrea 1997)	Ni podatka	Faktorji kakovosti ogrodij	<i>Framework user effectiveness</i> <i>Extensibility</i> <i>Simplicity</i> <i>Understandability</i> <i>Framework integration</i>
The evaluation of framework reusability (Sindre, Conradi, & Karlsson 1995; Todd 1997)	Povzeto po modelu REBOOT	Faktorji, ki vplivajo na ponovno uporabnost ogrodij	<i>Portability</i> <i>Adaptability</i> <i>Understandability</i> <i>Confidence</i>
Development of Object-Oriented Frameworks (Freimut, Punter, Biffel, & Ciolkowski 2002)	Rezultati delavnice na OOPSLA'97	Faktorji, ki vplivajo na uspešnost ogrodij	<i>Sufficient documentation</i> <i>Minimalistic architecture</i> <i>Utilization because of advantages</i> <i>Easy to understand</i> <i>Easy to use</i> <i>Easy to extend</i>

Zgornja tabela (Tabela 11) prikazuje rezultate iskanja virov in identifikacije kod v relevantnih virih. Identificiranih je bilo 8 prispevkov, v katerih smo identificirali skupno 33 kod.

3.5.2 REZULTATI POSTOPNEGA PRIBLIŽEVANJA

Na osnovi postopka postopnega približevanja, ki je bil definiran v poglavju 3.4.2.3, so se identificirane kode klasificirale v taksonomijo konceptov na višjem nivoju abstrakcije. Pri tem je izhodiščne oziroma preliminarne koncepte predstavljal seznam konceptov, ki se je oblikoval v terenski raziskavi (glej Tabela 8).

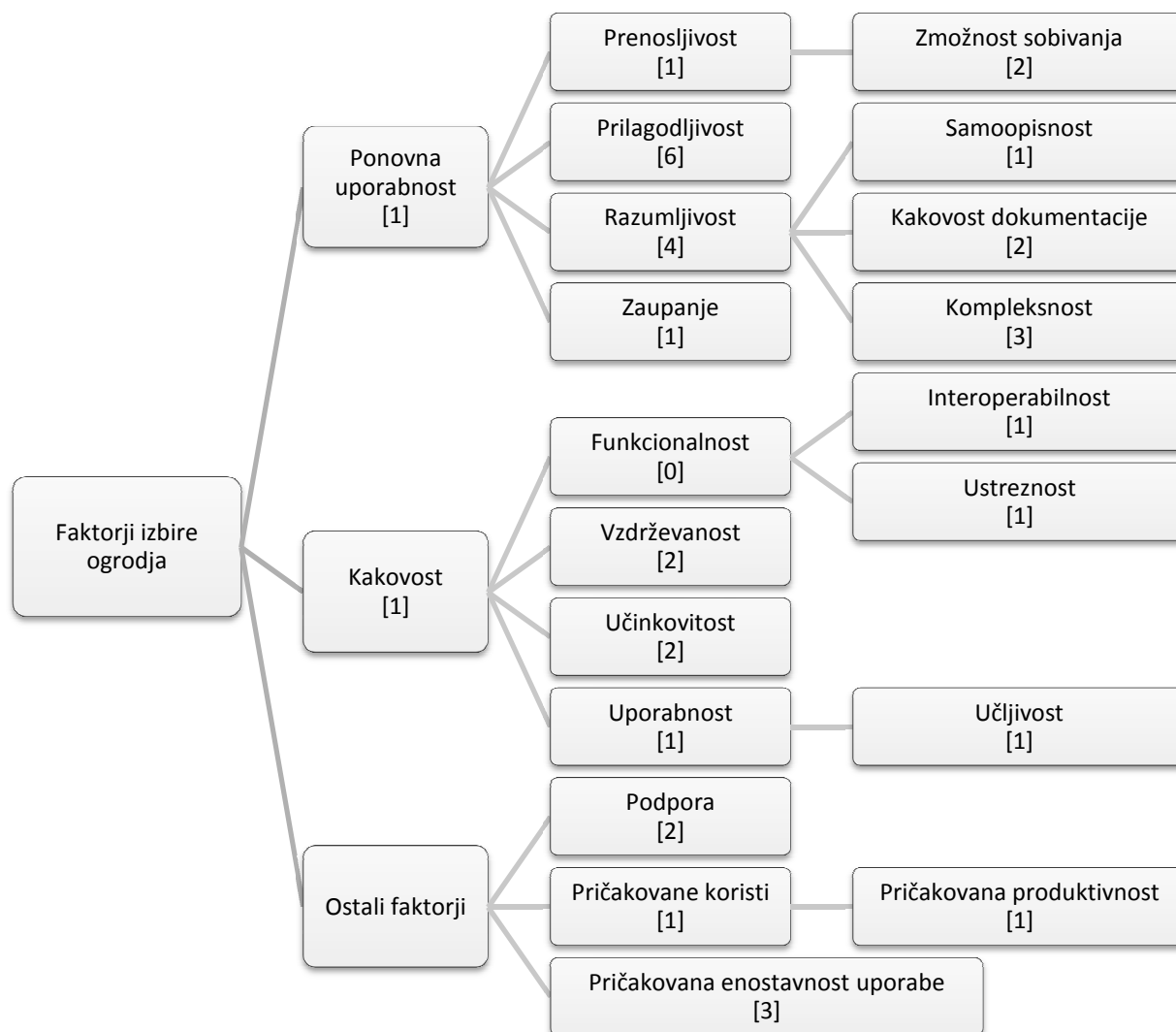
TABELA 12: SEZNAM IDENTIFICIRANIH KOD, MNOGOTERNOSTI IN PRIPADAJOČI KONCEPTI

Koda v slovenskem jeziku	Koda v angleškem jeziku	<i>m(a)</i>	Koncept v angleškem jeziku
Prilagodljivost	<i>Adaptability</i>	1	<i>Adaptability</i>
Primerna dokumentacija	<i>Adequate documentation</i>	1	<i>Documentation quality</i>
Primernost	<i>Appropriateness</i>	1	<i>Suitability</i>
Celovitost	<i>Completeness</i>	1	<i>Not classified</i>
Zaupanje	<i>Confidence</i>	1	<i>Confidence</i>
Konsistentni programski kod	<i>Consistent coding</i>	1	<i>Selfdescriptiveness</i>
Napor pri razvoju	<i>Development effort</i>	1	<i>Perceived ease of use</i>
Enostaven za uporabo	<i>Ease of use</i>	1	<i>Perceived ease of use</i>
Enostaven za razširjanje	<i>Easy to extend</i>	1	<i>Adaptability</i>
Enostaven za učenje	<i>Easy to learn</i>	1	<i>Learnability</i>
Enostaven za razumeti	<i>Easy to understand</i>	1	<i>Understandability</i>
Enostaven za učenje	<i>Easy to use</i>	1	<i>Perceived ease of use</i>
Učinkovitost	<i>Efficiency</i>	2	<i>Efficiency</i>
Razširljivost	<i>Extensibility</i>	2	<i>Adaptability</i>
Fleksibilnost	<i>Flexibility</i>	2	<i>Adaptability</i>
Integracija ogrodja	<i>Framework integration</i>	1	<i>Co-existence</i>
Učinkovitost uporabe ogrodja	<i>Framework user effectiveness</i>	1	<i>Productivity</i>
Dobra podpora	<i>Good support</i>	1	<i>Supportability</i>
Obvladovanje kompleksnosti	<i>Handling complexity</i>	1	<i>Complexity</i>
Nezadostna podpora	<i>Insufficient support</i>	1	<i>Supportability</i>
Zmožnost integriranja	<i>Integratability</i>	1	<i>Co-existence</i>
Pomanjkanje standardov	<i>Lack of standards</i>	1	<i>Interoperability</i>
Krivulja učenja	<i>Learning curve</i>	1	<i>Understandability</i>
Vzdrževanost	<i>Maintainability</i>	1	<i>Maintainability</i>
Minimalistična zasnova	<i>Minimalistic architecture</i>	1	<i>Complexity</i>
Prenosljivost	<i>Portability</i>	1	<i>Portability</i>
Kakovostna notranjost	<i>Qualitative internal</i>	1	<i>Quality</i>
Ponovna uporabnost	<i>Reusability</i>	1	<i>Reusability</i>
Enostavnost	<i>Simplicity</i>	1	<i>Complexity</i>
Zadostna dokumentacija	<i>Sufficient documentation</i>	1	<i>Documentation quality</i>
Podpora za vzdrževanje	<i>Support for maintenance</i>	1	<i>Maintainability</i>

Koda v slovenskem jeziku	Koda v angleškem jeziku	<i>m(a)</i>	Koncept v angleškem jeziku
Razumljivost	<i>Understandability</i>	2	<i>Understandability</i>
Uporabnost	<i>Usability</i>	1	<i>Usability</i>
Uporaba zaradi prednosti ogrodja	<i>Utilization because of advantages</i>	1	<i>Net benefits</i>
	Skupaj kod	38	

Iz zgornje tabele (Tabela 12) je razvidno, da smo 37 od 38 kod klasificirali v preliminarni seznam konceptov. Neklasificirana koda je ostala »celovitost« (*completeness*).

Na spodnji sliki je prikazana struktura konceptov v drevesni obliki. Izhodišče za strukturo je predstavljalo drevo konceptov, ki se je oblikovalo na osnovi terenske raziskave.



SLIKA 48: TAKSONOMIJA KONCEPTOV V DREVESNI OBLIKI

Iz slike (Slika 48) je razvidno, da sestavlja taksonomija konceptov pregleda literature podmnožico taksonomije konceptov, ki smo jih oblikovali na osnovi terenske raziskave.

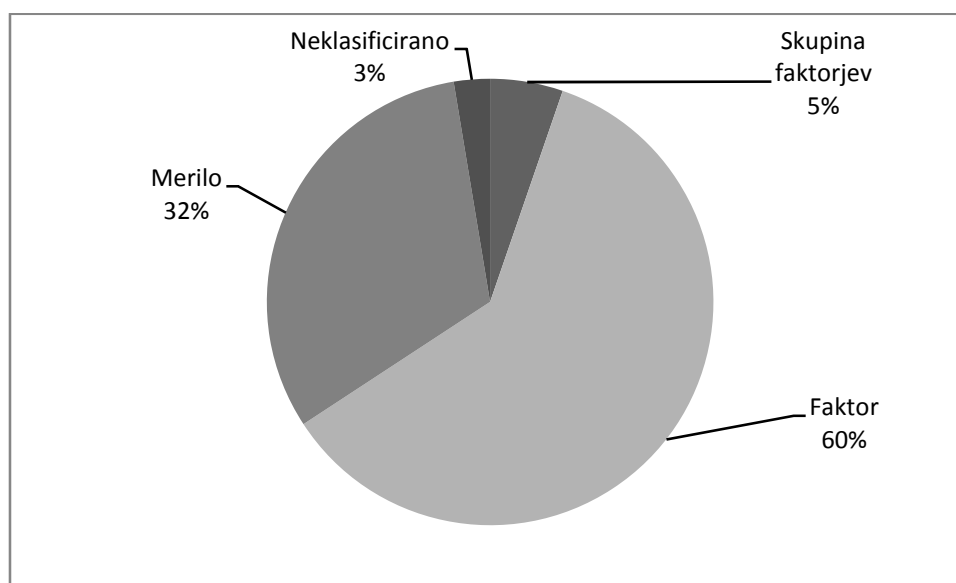
3.5.3 REZULTATI PONAZORILNE METODE

Relevantnost konceptov je bila izračunana po metodi, definirani v terenski raziskavi (glej poglavje 3.4.2.4). Rezultati (relevantnosti) so podani v drevesni obliki na prejšnji sliki (Slika 48)⁷⁵. Opisna statistika relevantnosti konceptov je podana v spodnji tabeli (Tabela 13).

TABELA 13: OPISNA STATISTIKA RELEVANTOSTI KONCEPTOV⁷⁶

Descriptive Statistics						
	N	Minimum	Maximum	Mean	Std. Deviation	Variance
Relevantnost	21	1	6	1,81	1,289	1,662
Valid N (listwise)	21					

Kode, ki smo jih identificirali v literaturi, so bile klasificirane v 21 konceptov. Pri tem so bili vsi koncepti definirani apriorno. V povprečju se je klasificiralo 1.81 kode na koncept. Na spodnji sliki (Slika 49) je prikazana analiza klasifikacije kod glede na hierarhično strukturo konceptov.



SLIKA 49: ANALIZA KLASIFIKACIJE KOD GLEDE NA HIERAHIČNO STRUKTURO KONCEPTOV

Iz tortnega grafikona (Slika 49) je možno razbrati, da sta bili, glede na strukturo FCM (*factor-criteria-metrics*), približno dve tretjini kod klasificirani med koncepte in ena tretjina kod med merila.

⁷⁵ Številke v oglatih oklepajih.

⁷⁶ Tabela je prenesena iz orodja SPSS.

3.5.4 OMEJITVE RAZISKAVE

Pri izvedbi pregleda literature smo analizirali omejitve, ki so povezane s: (1) procesom identifikacije literature, (2) veljavnostjo identificiranih virov in (3) analizo podatkov.

Ne razpolagamo s podatki, da smo v procesu iskanja identificirali vso relevantno literaturo. Prav tako ni poznan vzorec identificirane literature glede na celotno populacijo relevantne literature. Zato je tvegano posploševati rezultate pregleda literature na vso obstoječo relevantno literaturo.

Veljavnosti identificiranih virov nismo preverili, zato ne poznamo veljavnosti kod, ki smo jih identificirali v literaturi.

Kode, ki smo jih identificirali v literaturi, smo analizirali z enakimi tehnikami kot v terenski raziskavi, zato ima analiza podatkov tudi enake omejitve (glej poglavje 3.4.4). Pri oblikovanju taksonomije konceptov smo se zgledovali po rezultatih terenske raziskave. Drugačna vhodna taksonomija bi lahko podala drugačno taksonomijo konceptov.

V nadaljevanju je podana primerjava rezultatov terenske raziskave in pregleda literature.

3.6 PRIMERJAVA REZULTATOV PREGLEDA LITERATURE IN TERENSKE RAZISKAVE

Komparativna ali primerjalna metoda je definirana kot postopek primerjave enakih ali podobnih dejstev, pojavov, procesov in odnosov oziroma ugotavljanje njihovih podobnosti v obnašanju in intenzivnosti teh razlik med njimi (Ivanko 2007).

Komparativno metodo smo uporabili za primerjavo rezultatov predhodno izvedene terenske raziskave in pregleda literature. Cilj obeh raziskav je bil enak – identificirati faktorje, ki vplivajo na sprejetost ogrodiv. V spodnji tabeli (Tabela 14) so na osnovi metode GQM (Uebersax 2006) podane značilnosti izvedenih raziskav.

TABELA 14: PRIMERJAVA TERENSKE RAZISKAVE IN PREGLEDA LITERATURE Z UPORABO METODE GQM

Atribut	Terenska raziskava	Pregled literature
Predmet raziskave	ogrodja	ogrodja
Namen raziskave	identifikacija faktorjev	identifikacija faktorjev
Fokus raziskave	sprejetost	sprejetost, uspešnost, neuspeh, učinkovitost uporabe, kakovost, ponovna uporabnost
Udeleženci raziskave	Terenska raziskava: uporabniki ogrodiv, razvijalci ogrodiv	Pregled literature: raziskovalci
Kontekst raziskave	strokovnjaki z dejanskimi izkušnjami z ogrodivi	strokovna in znanstvena literatura

V okviru komparativne metode smo analizirali skupne značilnosti in razlike med rezultati obeh raziskav. Ker je bila taksonomija pregleda literature izvedena na osnovi taksonomije terenske raziskave, smo komparativno metodo izvedli tako, da smo opazovali rezultate pregleda literature na osnovi rezultatov terenske raziskave⁷⁷.

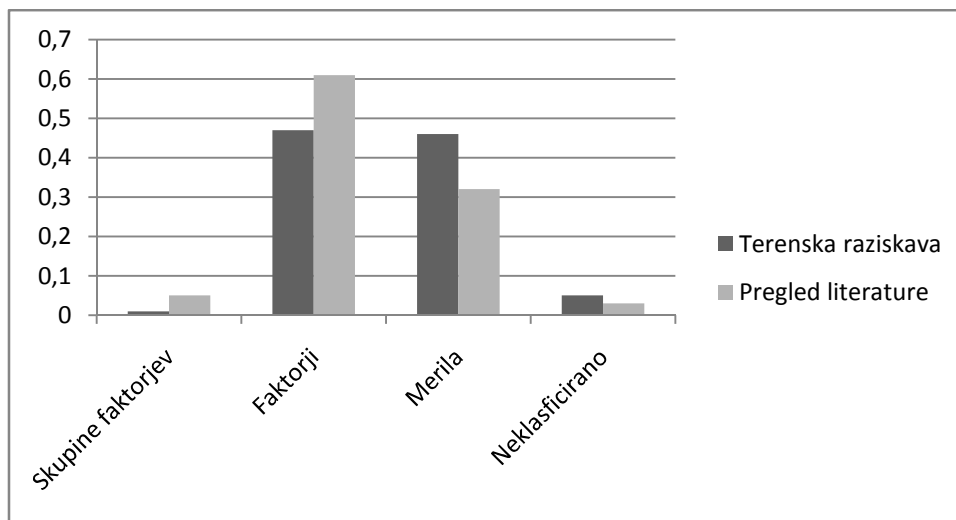
Rezultati pregleda literature se ujemajo z rezultati terenske raziskave v naslednjih vidikih:

- **Identificirani koncepti.** Koncepti, ki smo jih identificirali s pregledom literature, se v 37 od 38 primerov (97%) ujemajo s taksonomijo konceptov (Slika 45), ki smo jo oblikovali na osnovi terenske raziskave. Podatek je v podporo definirani taksonomiji konceptov.
- **Skupine identificiranih konceptov.** Koncepti, ki smo jih identificirali s pregledom literature, so se klasificirali v enake skupine konceptov kot koncepti, ki smo jih

⁷⁷ V primeru, da primerjamo A in B, kjer je $A < B$, se izvede komparativna metoda tako, da B opazujemo skozi A (*lens comparison*).

klasificirali s terensko raziskavo. Iz navedenega je možno predpostaviti, da koncepti obeh raziskav ustrezajo strukturi FCM (*factor-criteria-metrics*).

- **Število in relevantnosti identificiranih skupin konceptov.** Število in relevantnost konceptov, ki so se klasificirali v določeno skupino, so podobni v obeh raziskavah (Slika 50).



SLIKA 50: DELEŽI RELEVANTNOSTI KONCEPTOV V DOLOČENI SKUPINI

Tabela, ki je predstavljena v nadaljevanju (Tabela 15), podaja številčno primerjavo med različnimi skupinami konceptov v terenski raziskavi in pregledu literature.

TABELA 15: ŠTEVILČNA PRIMERJAVA IDENTIFICIRANIH KONCEPTOV IN NJIHOVIH RELEVANTNOSTI

Skupina identificiranih konceptov (A_x)	Terenska raziskava			Pregled literature		
	Štev. konceptov	$\sum_{a \in A_x} m(a)$	$\frac{\sum_{a \in A_x} m(a)}{ (A, m) }$	Štev. konceptov	$\sum_{a \in A_x} m(a)$	$\frac{\sum_{a \in A_x} m(a)}{ (A, m) }$
Skupine faktorjev (A_1)	1	3	0.01	2	2	0.05
Faktorji (A_2)	14	94	0.47	10	23	0.61
Merila (A_3)	15	93	0.46	8	12	0.32
Neklasfici. (A_0)	9	11	0.05	1	1	0.03
Skupaj (A)	30	201	1	20	38	1

Primerjavo skupin konceptov (A_x), ki smo jih identificirali s terensko raziskavo in pregledom literature, smo analizirali z Mann-Whitneyjevim U-testom. Mann-Whitneyjev U-test je neparametrična različica T-testa, namenjena manjšim vzorcem. Test je namenjen neodvisnim spremenljivkam in omogoča preverjanje hipotez; preverja enakost median dveh vzorcev in izračuna pripadajočo približno vrednost in interval zaupanja. V našem primeru je »tip raziskave« predstavljal neodvisno spremenljivko. Ničelna hipoteza (H_0), ki smo jo pri tem

zastavili, se je glasila: »Med razporeditvijo števila konceptov v terenski raziskavi in pregledu literature ni signifikantnih razlik.« Pri tem smo razporeditev konceptov obravnavali na osnovi njihovega števila in relevantnosti (Tabela 16).

TABELA 16: REZULTATI MANN-WHITNEYJEVEGA U-TESTA

Test Statistics ^b		
	Relevance	Number
Mann-Whitney U	4,000	4,500
Wilcoxon W	14,000	14,500
Z	-1,155	-1,016
Asymp. Sig. (2-tailed)	,248	,309
Exact Sig. [2*(1-tailed Sig.)]	,343 ^a	,343 ^a
a. Not corrected for ties.		
b. Grouping Variable: Group		

V zgornji tabeli (Tabela 16) so podani rezultati izvedbe Mann-Whitneyjevega U-testa. Ugotovljeno je bilo, da med skupinama (terenska raziskava in pregled literature) ne obstaja signifikantna razlika⁷⁸ niti v primeru relevantnosti konceptov ($p > 0.05$) niti v primeru števila konceptov ($p > 0.05$). Zato hipoteze H_0 nismo zavrgli.

Rezultati terenske raziskave in pregleda literature se razlikujejo v relevantnosti konceptov. V spodnji tabeli (Tabela 17) je prikazana ordinalna razvrstitev konceptov terenske raziskave in pregleda literature glede na deset najbolj relevantnih konceptov terenske raziskave.

TABELA 17: PRIMERJAVA RAZVRSTITVE KONCEPTOV GLEDE NA RELEVANTNOST

Koncept v angleškem jeziku	Razvrstitev konceptov po relevantnosti	
	v terenski raziskavi	v pregledu literature
Ustreznost	1	20
Kompleksnost	2	3
Dojeta enostavnost uporabe	3	4
Priljubljenost	4	/
Pravni vidik	5	/
Učinkovitost	6	7
Poznanost	7	/
Zrelost	8	/
Prilagodljivost	9	1
Kakovost dokumentacije	10	6

⁷⁸ Glej vrstico *Exact Sig. [2*(1-tailed Sig.)]*.

Iz tabele je razvidno, da se razvrstitev konceptov terenske raziskave in pregleda literature glede na relevantnost razlikuje. Možen razlog za razlike je, da so bile raziskave, ki smo jih identificirali v pregledu literature, predvsem tehnološko usmerjene (faktorji kakovosti in ponovne uporabnosti), medtem, ko naša raziskava vključuje še druge faktorje (na primer: ustreznost, poznanost, pravni vidik in priljubljenost).

Na osnovi skupnih lastnosti rezultatov terenske raziskave in pregleda literature je možno predpostavljati, da je taksonomija konceptov, ki se je oblikovala v terenski raziskavi, veljavna. Prav tako je možno predpostavljati, da uporabniki, v velikem deležu vrednotijo ogrodja na osnovi faktorjev in meril.

3.7 ANKETA: EMPIRIČNA VALIDACIJA DETERMINANT SPREJETOSTI IN USPEŠNOSTI OGRODIJ

Empirično validacijo faktorjev, ki smo jih identificirali s terensko raziskavo in s pregledom literature, smo izvedli z metodo ankete, izvedene v naslednjih fazah (Freimut, Punter, Biffi, & Ciolkowski 2002): opredelitev ankete, načrtovanje ankete, realizacija ankete, izvedba ankete in analiza rezultatov ankete.

3.7.1 OPREDELITEV ANKETE

V opredelitvi ankete se določijo cilji, raziskovalna vprašanja in tveganja izvedbe ankete (Freimut, Punter, Biffi, & Ciolkowski 2002). V našem primeru so bili cilji ankete določeni na osnovi raziskovalnih vprašanj, ki so bila predstavljena v uvodu disertacije (glej poglavje 3.1.1 in Tabela 2).

3.7.1.1 RAZISKOVALNO VPRAŠANJE RV₁

Odgovori na prvo raziskovalno vprašanje RV₁ so bili delno podani že v okviru predhodno izvedene terenske raziskave in pregleda literature.

Pregled literature na področju ogrodi je identificiral več virov (glej Tabela 11), ki so opredelili faktorje, ki imajo pomemben vpliv na zelene lastnosti ogrodi, kot so: kakovost, ponovna uporabnost, sprejetost in uspešnost uporabe. Ugotovljeno je bilo, da imajo obstoječe raziskave določene pomanjkljivosti (glej poglavje 3.5.4), kot sta nesistematična opredelitev faktorjev (pretežno na osnovi lastnih izkušenj) in njihova nepreverjena veljavnost.

Komplementarno k obstoječim raziskavam smo zato izvedli terensko raziskavo, ki je na osnovi odgovorov 57 strokovnjakov identificirala faktorje, ki vplivajo na njihovo sprejetje ogrodi. Na osnovi terenske raziskave smo pridobili rezultate, ki imajo širšo zunanjo veljavnost kot izkušnje posameznikov, vendar tudi ta raziskava ni preverila veljavnosti identificiranih faktorjev.

Pomanjkljivost obeh navedenih skupin raziskav (raziskave, ki so jih izvedli drugi in naša terenska raziskava), so torej nepreverjeni rezultati, zato smo kot prvi cilj raziskave opredelili analizo veljavnosti faktorjev sprejetosti ogrodi, ki smo jih identificirali v terenski raziskavi.

V terenski raziskavi smo identificirali 30 različnih faktorjev⁷⁹, za katere se lahko predpostavlja, da imajo signifikanten vpliv na sprejetost ogrodi (Tabela 8, Slika 45). Zaradi razmeroma velikega števila faktorjev smo izvedli analizo izvedljivosti raziskave, kjer smo ugotovili, da bi bil

⁷⁹ Glede na metodo FCM (*Factor-Criteria-Metrics*) smo identificirali skupine faktorjev, faktorje in kriterije.

obseg anketnega vprašalnika, če bi vključili vseh 30 faktorjev, prevelik. Takšen vprašalnik bi poleg vprašanj, povezanih z analiziranjem respondenta, in vprašanj za doseganje drugega in tretjega cilja ankete vključeval najmanj 90 Likertovih vprašanj⁸⁰ (Seddon, Staples, Patnayakuni, & Bowtell 1999), ki bi analizirala vpliv faktorjev na odvisne koncepte. Ker bi bila motivacija za prostovoljno sodelovanje v tako obsežni anketi dvomljiva, smo minimizirali število faktorjev na obseg, ki bi bil sprejemljiv za respondente.

3.7.1.2 RAZISKOVALNO VPRAŠANJE RV₂

Drugi cilj ankete je bil podati odgovor na raziskovalno vprašanje RV₂. Za pridobitev odgovora na RV₂ smo v raziskavo vključiti determinante sprejetosti, ki so se identificirale v (1) obstoječih modelih sprejetosti, in (2) determinante, ki so specifične za domeno ogrodiv.

Prvo skupino determinant smo pridobili iz modela TAM, ki je največkrat citiran model sprejetosti IT (Sharp 2006). Drugo skupino determinant smo pridobili na osnovi razrešenega prvega raziskovalnega vprašanja in drugih preverjenih modelov sprejetosti. Na osnovi obeh skupin faktorjev smo oblikovali dva modela sprejetosti:

- model sprejetosti ogrodiv, ki temelji na modelu TAM, in
- model sprejetosti ogrodiv, ki razširja model TAM s faktorji iz domene ogrodiv in s faktorji iz drugih preverjenih modelov sprejetosti.

Primerjava lastnosti obeh modelov je pokazala, kateri od modelov je natančnejši in primernejši v kontekstu ogrodiv.

V primeru RV₂ smo uporabili kombiniran potrditveno-raziskovalni pristop (*confirmatory-exploratory approach*) reševanja problema, saj smo poleg potrditve teoretičnih modelov oblikovali nov teoretični model – model sprejetosti ogrodiv. Model sprejetosti ogrodiv je temeljil na rezultatih predhodno analiziranih teoretičnih modelov.

3.7.1.3 RAZISKOVALNO VPRAŠANJE RV₃

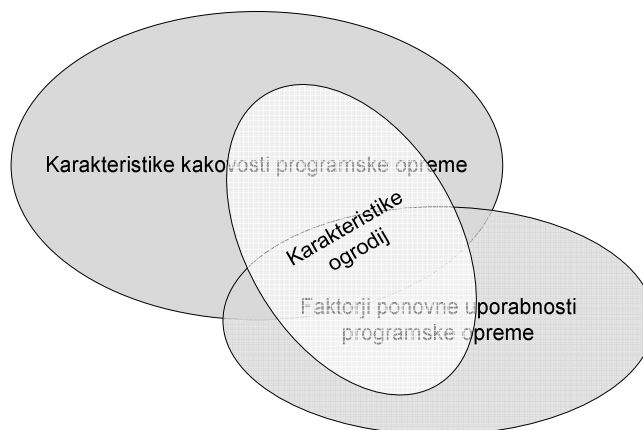
Tretje raziskovalno vprašanje (RV₃) je oblikovalo tretji cilj ankete - analizirati predpostavljeno povezavo med sprejetostjo in uspešnostjo ogrodiv. Za doseganje tretjega cilja smo v anketo vključili faktorje sprejetosti ogrodiv in faktorje uspešnosti ogrodiv. Faktorje sprejetosti ogrodiv smo pridobili na osnovi razrešenega drugega raziskovalnega vprašanja, medtem ko smo faktorje uspešnosti ogrodiv definirali na osnovi definicije uspešnosti programskih ogrodiv, ki smo jo podali v poglavju 3.2. S tretjim ciljem ankete smo poskušali demonstrirati veljavnost ugotovitev, ki smo jih navedli v poglavju 3.2.1.

⁸⁰ Vprašanje, ki temelji na Likertovi lestvici (*likert scale*), predvideva vsaj tri podvprašanja (*likert items*).

3.7.2 FAKTORJI IN HIPOTEZE RV₁

Model TAM ne določa zunanjih spremenljivk (*external variables*), ki imajo vpliv na uporabnikova dojetanja, ker so odvisne od vrste IT, ki se vrednoti (Nelson 1990). Zato predstavljajo zunanje spremenljivke eno od ključnih področij razširitev modela TAM (glej Slika 23). Na področju programske opreme je več raziskovalcev demonstriralo signifikanten vpliv dveh skupin zunanjih spremenljivk na uporabnikova dojetanja: karakteristike sistema (*system characteristics*) in posebnosti uporabnika (*individual differences*).

V okviru našega primera raziskave predstavljajo karakteristike ogrodja poseben primer karakteristik programske opreme in poseben primer ponovno uporabne programske opreme (Slika 51), kar je razvidno tudi iz taksonomije konceptov sprejetosti (Slika 45)⁸¹.



SLIKA 51: POVEZAVA MED KARAKTERISTIKAMI OGRODIJ, KAKOVOSTI IN PONOVNE UPORABNOSTI

Izmed 30 identificiranih faktorjev terenske raziskave se jih je 22 klasificiralo v obstoječe koncepte modela kakovosti in ponovne uporabnosti (glej Slika 45).

Druga skupina faktorjev s signifikantnim vplivom na sprejetost so posebnosti uporabnikov programske opreme (Nelson 1990). V terenski raziskavi smo identificirali dva faktorja, za katera lahko domnevamo, da sta odvisna od posameznika: ustreznost in poznanost.

⁸¹ Karakteristike programske opreme so definirane v modelu kakovosti programske opreme (glej poglavje 2.5). Karakteristike stopnje ponovne uporabnosti so definirane v modelu ponovne uporabnosti REBOOT (glej poglavje 0).

3.7.2.1 KARAKTERISTIKA OGRODJA: RAZUMLJIVOST

Na osnovi seštevka relevantnosti posameznih faktorjev in pripadajočih meril v taksonomiji faktorjev sprejetosti ogrodij je bilo ugotovljeno, da je razumljivost ogrodja najbolj relevanten koncept (Tabela 10). Definicijo razumljivosti programskega ogrodja (*framework understandability*) smo izpeljali iz definicije razumljivosti programske opreme:

Razumljivost ogrodja predstavlja napor uporabnika, da prepozna primernost, logično zasnovo in uporabo ogrodja za specifičen namen in pod specifičnimi pogoji.

Razumljivost ogrodja je koncept, ki je soroden faktorju modela REBOOT in podkarakteristiki uporabnosti programske opreme v modelu kakovosti. Splošno uveljavljeno mnenje o ogrodjih je, da so kompleksna in zato težavna za učenje, uporabo in vzdrževanje. Zato so raziskovalci predlagali izboljšave za povečanje razumljivosti ogrodij, kot so izboljšave dokumentiranja ogrodij, oblikovanje metod instanciranja ogrodij in preoblikovanje zasnove ogrodij (Leach 1996). Razumljivost, ki je karakteristika sistema, se pogosto zamenjuje z enostavnostjo uporabe, ki predstavlja uporabniško odvisen koncept. Iz navedenega lahko izpeljemo naslednjo hipotezo:

H1: Razumljivost ogrodja ima pozitiven vpliv na dojeto enostavnost uporabe ogrodja.

3.7.2.2 KARAKTERISTIKA OGRODJA: PRENOSLJIVOST

Prenosljivost ogrodja je soroden koncept faktorju modela REBOOT in karakteristiki kakovosti programske opreme. Definicija prenosljivosti je izpeljana iz modela ISO/IEC 9126 in je naslednja:

Prenosljivost ogrodja predstavlja sposobnost ogrodja, da se prenese iz enega v drugo okolje⁸².

Faktor prenosljivosti v modelu kakovosti vključuje merilo prilagodljivosti, medtem ko je prilagodljivost v modelu ponovne uporabnosti neodvisen faktor. V okviru naše raziskave smo prenosljivost modelirali skladno z modelom kakovosti iz naslednjih razlogov:

- Faktorja prenosljivosti in prilagodljivosti smo združili zaradi težnje po zmanjšanju obsega vprašalnika. Na osnovi seštevka relevantnosti faktorjev po izvedbi združevanja

⁸² Okolje je lahko organizacijsko, strojno ali programsko.

konceptov (Tabela 10) smo ugotovili, da je prenosljivost bolj relevantna od prilagodljivosti. Zato smo prilagodljivost vključili v prenosljivost in ne obratno.

- Za faktor prilagodljivosti nismo identificirali meril, zato obstaja možnost, da je prilagodljivost merilo prenosljivosti.

Prilagodljivost in prenosljivost veljata za pglavitni karakteristiki ogrodij, saj omogočata povečanje števila instanc ogrodja (Bockle, Clements, McGregor, Muthig, & Schmid 2004; Fayad & Hamu 2000). Hipoteza, ki jo postavljamo, je sledeča:

H2: Prenosljivost ogrodja ima pozitiven vpliv na dojeto uporabnost ogrodja.

Gledano z drugega zornega kota, je lahko previsoka prenosljivost tudi slabost, saj so raziskovalci ugotovili, da višanje prenosljivosti povečuje kompleksnost ogrodja (Froehlich, Hoover, Liu, & Sorenson 1998; Mattsson 1996). Iz tega sledi naslednja hipoteza:

H3: Prenosljivost ogrodja ima negativen vpliv na dojeto enostavnost uporabe ogrodja.

3.7.2.3 KARAKTERISTIKA OGRODJA: ZAUPANJE

Zaupanje v ogrodje je koncept, ki smo ga izpeljali iz faktorja modela ponovne uporabnosti, in je definiran naslednje:

Zaupanje v ogrodje predstavlja subjektivno oceno, da bo ogrodje v okvirih določenega časovnega obdobja in v spremenjenih okoliščinah delovalo po pričakovanjih.

Analiza relevantnosti faktorjev po izvedbi združevanja konceptov (Tabela 10) je pokazala, da je zaupanje izmed 15 faktorjev na četrtem mestu. Prav tako je možno zaslediti raziskave na področju ogrodij, ki so preučevale merili faktorja zaupanja: zrelost ogrodja in odpornost proti napakam (Froehlich, Hoover, Liu, & Sorenson 1998), (Mattsson & Bosch 2006), (Fayad & Hamu 2000).

Zaupanje v ogrodje je pomembno iz naslednjih razlogov. Prvi razlog je, da se ogrodje obrestuje šele na dolgi rok. Zato morajo uporabniki ogrodja imeti zaupanje v ogrodje, da bo to delovalo zadovoljivo daljše časovno obdobje. Drugi razlog je povezan z odpornostjo ogrodja proti napakam. Ker predstavljajo ogrodja pomemben del končnih aplikacij, vplivajo morebitne napake v ogrodju na vse njegove primerke. Iz obeh razlogov lahko predpostavljamo naslednje:

H4: Zaupanje v ogrodje ima pozitiven vpliv na dojeto uporabnost ogrodja.

3.7.2.4 KARAKTERISTIKA OGRODJA: UČINKOVITOST

Učinkovitost ogrodja predstavlja edino karakteristiko ogrodja, ki smo jo vključili v raziskavo in je ne najdemo v modelu ponovne uporabnosti. Razlog za to so visoka relevantnost faktorja (Slika 45) in mnenja drugih raziskovalcev, ki so omenili manjšo učinkovitost aplikacij, ki temeljijo na ogrodjih (Mattsson 1996). Primerjava domensko specifičnih jezikov (DSL) in ogrodij je prav tako pokazala slabšo učinkovitost ogrodij glede na DSL (Deursen 1997). Iz navedenega je možno predpostavljati, da ogrodja, ki zasedajo veliko računalniških virov, med uporabniki niso dobro sprejeta, in obratno:

H5: Učinkovitost ogrodja ima pozitiven vpliv na dojeto uporabnost ogrodja.

Učinkovitost predstavlja karakteristiko kakovosti programske opreme, na osnovi katere smo tudi izpeljali naslednjo definicijo:

Učinkovitost ogrodja predstavlja sposobnost ogrodja, da zagotovi primerne obratovalne zmogljivosti glede na obseg porabljenih računalniških virov.

3.7.2.5 POSEBNOSTI UPORABNIKA OGRODJA: UJEMANJE ZAHTEV IN OGRODJA

Ujemanje zahtev in ogrodja predstavlja koncept, ki spada v skupino posebnosti posameznikov (*individual differences*). Izpeljali smo ga iz: (1) koncepta ustreznosti, ki predstavlja podkarakteristiko modela kakovosti programske opreme, in (2) teoretičnega modela ujemanja zahtev in lastnosti IT ali krajše TTF (*Task-technology-fit*).



SLIKA 52: GRAFIČNA PONAŽORITEV KONCEPTA »UJEMANJE ZAHTEV IN OGRODJA«

Koncept ujemanja zahtev in ogrodja smo v raziskavo vključili iz naslednjih razlogov. Koncept ustreznosti, iz katerega smo izpeljali koncept »ujemanje zahtev in ogrodja«, se je v izvedeni terenski raziskavi izkazal kot najbolj relevanten. Prav tako so predhodne raziskave poudarile pomembnost ustreznosti ogrodja (*adequate fit*) (Fayad & Hamu 2000; Meier 1998). Ujemanje zahtev in ogrodja smo definirali na osnovi definicije koncepta TTF naslednje:

Ujemanje zahtev in ogrodja predstavlja razmerje med funkcijami ogrodja in zahtevami njegovih primerkov.

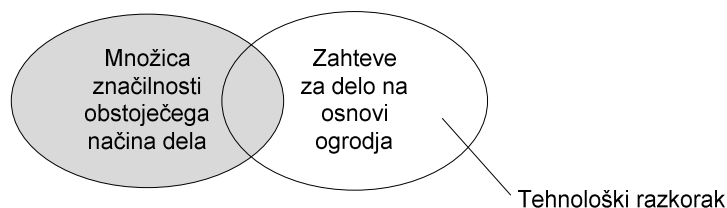
Predpostavljamo, da večji ko je presek med funkcionalnostmi ogrodja in zahtevami uporabnikov, večja bo sprejetost ogrodja (Slika 52):

H6: Ujemanje zahtev in ogrodja ima pozitiven vpliv na dojeto uporabnost ogrodja.

Vpliv TTF na dojeto uporabnost v modelu TAM se je v drugih kontekstih že demonstriral kot signifikanten (Dishaw & Strong 1999; Klopping & McKinney 2004).

3.7.2.6 POSEBNOSTI UPORABNIKA OGRODJA: TEHNOLOŠKI RAZKORAK

Vpeljava novega ogrodja zahteva spremembe na številnih področjih procesa razvoja programske opreme, nove zadolžitve in nova znanja razvijalcev, ki bodo uporabljali ogrodja. Večji ko je tehnološki razkorak med starim in novim načinom dela in potrebnim znanjem, daljše je obdobje vpeljave ogrodja in manjše so možnosti za njegovo uspešno vpeljavo (Slika 53).



SLIKA 53: GRAFIČNA PONAZORITEV KONCEPTA »TEHNOLOŠKI RAZKORAK«

Zaradi tega je možno predpostaviti, da ima tehnološki razkorak negativen vpliv na uporabnikova dojetanja enostavnosti uporabe ogrodja in posledično sprejetost ogrodij.

H7: Tehnološki razkorak ima negativen vpliv na dojeto enostavnost uporabe ogrodja.

Razloga za vključitev koncepta v raziskavo sta dva. Prvič, koncept smo izpeljali iz koncepta poznanosti, ki je v okviru terenske raziskave imel nadpovprečno relevantnost. Drugič, tehnološki razkorak se ujema s stroškovnim faktorjem C_{org} , ki je sestavni del stroškovnega modela razvoja na skupni programski osnovi (glej poglavje 3.3). Faktor C_{org} predstavlja pomemben del stroškov, ki so povezani s spremembo načina razvoja programske opreme, kar je analogno vpeljavi novega ogrodja. Tehnološki razkorak je že potrdil signifikanten vpliv na eksogena konstrukta modela TAM v kontekstu orodij CASE⁸³ (Chau 1996).

⁸³ Computer Aided Software Engineering.

3.7.3 FAKTORJI IN HIPOTEZE RV₂

Model TAM in njegove izpeljanke vključujejo dve determinanti sprejetosti oziroma vedenjskega namena: dojeta enostavnost uporabe (PEOU) in dojeta uporabnost (PU). Omenjeni determinanti sprejetosti sta potrdili veljavnost v najrazličnejših kontekstih in razširitvah modela TAM, kar je razvidno iz spodnje tabele (Tabela 18).

TABELA 18: PREDHODNE RAZISKAVE, KI TEMELJIJO NA TAM

Avtorji	Konstrukti	Kontekst IT	Metoda raziskave	Ugotovitve
Davis	PU, PEOU, U	PROFs, XEDIT, Chart-Master, Pendraw	Anketa, preizkus	PU→U, PEOU→U
Davis, Bagozzi in Warshaw	PU, PEOU, A, BI, U	WriteOne	Preizkus	PEOU→PU, PU→A, PEOU→A, A→BI, PU→BI, BI→U
Haynes in Thies	PU, PEOU, U	Avtomatski odzivnik	Anketa	PU→U, PEOU→U
Mathieson	EV, PU, PEOU, A, BI, U	Preglednica, kalkulator	Preizkus	PU→U, PEOU→U
Adams, Nelson in Todd	PU, PEOU, U	E-pošta, WordPerfect, 123, Harvard Graphics	Anketa	PEOU→U, PU→U, PEOU←→U
Bagozzi, Davis in Warshaw	U, PEOU, BI, U	WriteOne	Preizkus	PU→BI, PEOU→BI, BI→U
Igbaria, Guiraes in Davis	EV, PEOU, PU, U	Osebni računalnik	Anketa	EV→PEOU, EV→PU, PEOU→PU, PEOU→U, PU→U
Szajna	PU, PEOU, BI, U	E-pošta	Preizkus	PEOU→PU, PU→BI, BI→U
Hendrickson in Collins	PU, PEOU, U	1-2-3, WordPerfect	Preizkus	PEOU→PU, PEOU→U, PU→U
Chau	PEOU, kratkoročni PU, dolgoročni PU, BI	Word, Excel	Anketa	PEOU→ kratkoročni PU, PEOU→BI, kratkoročni PU → dolgoročni PU, kratkoročni PU →BI, dolgoročni PU→BI
Morris in Dillon	PEOU, PU, A, BI, U	Netscape	Anketa	PEOU→PU, PU→A, PEOU→A, PU→BI, A→BI, BI→U

Avtorji	Konstrukti	Kontekst IT	Metoda raziskave	Ugotovitve
Thompson	PU, PEOU, A, BI	Access, Orodja za razvoj spletnih strani	Anketa	PEOU→PU, PEOU→A, PU→A, PU→BI, A →BI, motivacija→BI, družbeni faktorji →A
Teo, Lim in Lai	PU, PEOU, U, PE	Internet	Anketa	PEOU→PU, PEOU→U, PEOU→PE, PU→U, PE→U
Dishaw, Strong	PU, PEOU, A, BI, U, TTF, Izkušnje z orodjem	Vodstveni IS	Anketa	TTF→PEOU, PEOU→P, PU→A, A→BI, Izkušnje z orodjem → PEOU, Izkušnje z orodjem, PU
Klopping in McKinney	PU, PEOU, BI, U, TTF	E-trgovine	Anketa	TTF→PU, TTF→PEOU, TTF→BI, BI→U, PU→U, PU→BI
Pojasnilo: Odnos do uporabe (A), Uporaba (U), Dojeto zadovoljstvo (PE), Vedenjski namen (BI), Zunanje spremenljivke (EV), Ujemanje zahtev in IT (TTF).				

Iz zgornje tabele je prav tako razvidno, da so številni avtorji razširili TAM z dodatnimi faktorji, ki so skladni z eno od štirih razširitvenih kategorij modela TAM (Slika 23). V našem primeru smo dodatne faktorje opredelili na osnovi rezultatov prvega raziskovalnega vprašanja. Za doseganje drugega cilja ankete smo definirali naslednjo hipotezo:

H8: Predlagan model sprejetosti ogrodir je natančnejši od modela TAM.

V primeru H8 smo uporabili kombiniran potrditveno-raziskovalni pristop (*confirmatory-exploratory approach*) reševanja problema. Pri tem so faktorji, ki sta jih vključevala predlagan model in model TAM, predstavljeni v nadaljevanju.

3.7.3.1 TAM: DOJETA UPORABNOST OGRODJA

Predhodne raziskave so demonstrirale, da je dojeta uporabnost (PU) najpomembnejša determinanta vedenjskega namena uporabe (BI), sprejetosti sistema in trajne uporabe (Hong, Thong, & Tam 2006; Legris, Ingham, & Colletette 2003). Zato predpostavljamo, da je tudi v kontekstu ogrodij dojeta uporabnost ogrodij glavni pogoj za sprejetost in uporabo ogrodja.

H9: Dojeta uporabnost ogrodja (FPU) ima pozitiven vpliv na sprejetost ogrodja (FA).

H10: Dojeta uporabnost ogrodja (FPU) ima pozitiven vpliv na namero o trajni uporabi ogrodja (CFUI).

Definicijo dojete uporabnost ogrodja smo izpeljali iz definicije PU (Davis 1989):

Dojeta uporabnost ogrodja je stopnja, do katere (potencialni) uporabnik ogrodja verjame, da bo uporaba ogrodja izboljšala njegovo delovno storilnost.

3.7.3.2 TAM: DOJETA ENOSTAVNOST UPORABE OGRODJA

Definicijo dojete enostavnosti uporabe smo izpeljali iz definicije PEOU (Davis 1989):

Dojeta enostavnost uporabe ogrodja je stopnja, do katere uporabnik ogrodja verjame, da bo uporaba ogrodja enostavna oziroma ne bo zahtevala dodatnih naporov.

Predhodne raziskave so potrdile signifikanten vpliv dojete enostavnosti uporabe na uporabnikovo vedenje, in sicer na dva načina: neposredno ali posredno, preko dojete uporabnosti (Davis, Bagozzi, & Warshaw 1989; Dishaw & Strong 1999; Lederer, Maupin, Sena, & Zhuang 2000). Na osnovi predhodnih raziskav in dejstva, da spadajo ogrodja med najbolj kompleksne ponovno uporabne tehnologije, predpostavljamo naslednje:

H11: Dojeta enostavnost uporabe ogrodja (FPEOU) ima pozitiven vpliv na dojeto uporabnost ogrodja (FPU).

H12: Dojeta enostavnost uporabe ogrodja (FPEOU) ima pozitiven vpliv na namero o trajni uporabi ogrodja (CFUI).

H13: Dojeta enostavnost uporabe ogrodja (FPEOU) ima pozitiven vpliv na sprejetost ogrodja (FA).

3.7.3.3 UPORABNIKOV VEDENJSKI ODZIV

V raziskavah, ki temeljijo na modelu TAM, se uporabljajo štiri odvisne spremenljivke: (1) namera uporabe (*intention to use*), (2) dejanska uporaba (*system use*), (3) sprejetost (*acceptance*) in (4) namera o trajni uporabi (*continuous usage intention*). V našo raziskavo smo vključili faktor sprejetosti in namero o trajni uporabi iz razlogov, ki so podani v nadaljevanju.

Sprejetost ogrodja smo izpeljali iz faktorja uporabniške sprejetosti. Uporabniška sprejetost je najpogosteje uporabljena odvisna spremenljivka v raziskavah, ki temeljijo na modelu TAM (Saga & Zmud 1994). Splošno uveljavljena predpostavka raziskav TAM je, da je uporabniška sprejetost kritičen faktor uspešne implementacije in uporabe sistema. Sprejetost ogrodja smo definirali na osnovi uporabniške sprejetosti (Dillon & Morris 1996):

Sprejetost ogrodja je nazorna pripravljenost uporabiti ogrodje za izvedbo opravila, za katero je to namenjeno.

Kljub navedenemu smo v raziskavo vključili še drugi odvisni konstrukt, ki se uporablja v sorodnih raziskavah (Tabela 19) – namero o trajni uporabi ogrodja. Razlog za vključitev drugega konstrukta so rezultati obstoječih raziskav na področju ogrodij (Michael 1999; Morisio, Romano, & Stamelos 2002; Moser & Nierstrasz 1996) in značilnosti stroškovnega modela produktivnih linij (Bockle, Clements, McGregor, Muthig, & Schmid 2004). Iz obeh vrst predhodnih raziskav je možno predpostavljati, da je sprejetost ogrodja potreben pogoj za uspešno uporabo ogrodij, pri čemer je zadosten pogoj trajna uporaba ogrodja. Skladno z navedenim predpostavljamo, da sta za uspešno uporabo ogrodja potrebni sprejetost in tudi tajna uporaba ogrodja. Pri tem je namera o trajni uporabi ogrodja dober približek trajne uporabe ogrodja, kar se ujema s temeljno idejo modelov, ki temeljijo na namerah (Hong, Thong, & Tam 2006).

Namera trajne uporabe ogrodja predstavlja interes posameznika uporabljati ogrodje pri bodočem delu.

TABELA 19: PREGLED RAZISKAV, KI SO UPORABILE MODEL TAM V OKOLJU IZKUŠENIH UPORABNIKOV

Avtorji	Kontekst IT	Respondenti	Izkušnje respondentov s predmetom raziskave
Adams, Nelson in Todd	E-pošta in glasovna pošta, WordPerfect, Lotus 1-2-3 in Harvard Graphics	Poslovni uporabniki in študenti	Povprečno 28 mesecev izkušenj z e-pošto, 21 mesecev izkušenj z glasovno pošto in 13-33 mesecev izkušenj s preostalo programsko opremo
Agarwal in Prasad	Osebni računalniki	Zaposleni v uveljavljenih IT podjetjih	Različne izkušnje
Chau	MS Word in MS Excel	Administratorji v neprofitnih organizacijah	50% respondentov je uporabljalo MS Word vsaj 1-krat dnevno. 33% respondentov je uporabljalo MS Excel vsaj 1-krat dnevno.
Davis	IBM e-poštni sistem (PROFS)	Zaposleni v IBM	Povprečno 6 mesecev izkušenj
Gefen in Straub	Aplikacije e-pošte	Znanstveni delavci	Večletne izkušnje
Gefen, Karahanna, Straub	Spletno nakupovanje	Študenti	Respondent je v povprečju opravil 7.18 spletnega nakupa
Hong, Thong, Wong, Tam	Digitalne knjižnice	Študenti	66% respondentov je bilo izkušenih uporabnikov.
Lederer, Maupin, Sena, Zhuang	Spletne strani	Uporabniki spletnih strani	Trenutni uporabniki novičarskih skupin
Yang in Yoo	Preglednice	Študenti	Izkušeni uporabniki preglednic

3.7.4 FAKTORJI IN HIPOTEZE RV₃

Za doseganje tretjega cilja ankete smo poleg faktorjev, ki so povezani z drugim ciljem raziskave, potrebovali še faktorje uspešnosti ogrođij. Faktorje uspešnosti ogrođij smo oblikovali na osnovi Seddonovega modela uspešnosti (Seddon 1997) in na njem temelječe definicije uspešnosti ogrođij, ki smo jo predstavili v poglavju 3.2.

Skladno z večdimenzionalnostjo uspešnosti IS smo definirali dva faktorja uspešnosti: neto prednosti uporabe ogrođja (NBF) in zadovoljstvo z uporabo ogrođja (SAT).

Neto prednosti uporabe ogrođja smo definirali kot faktor, katerega so ocenjevali uporabniki ogrođja (*self-reported*). V faktor smo vključili tri najpogostejše omenjene metrike uspešnosti ogrođij: (1) izboljšave na področju kakovosti rešitev, (2) izboljšave na področju produktivnosti razvoja in (3) izboljšave na področju časa izdelave (*lead time*). Kot drugi faktor uspešnosti ogrođij smo določili zadovoljstvo z uporabo ogrođja. Hipoteze tretjega cilja raziskave povezujejo faktorje drugega in tretjega cilja raziskave:

H14: Sprejetost ogrođja (FA) ima pozitiven vpliv na neto prednosti uporabe ogrođja (NBF).

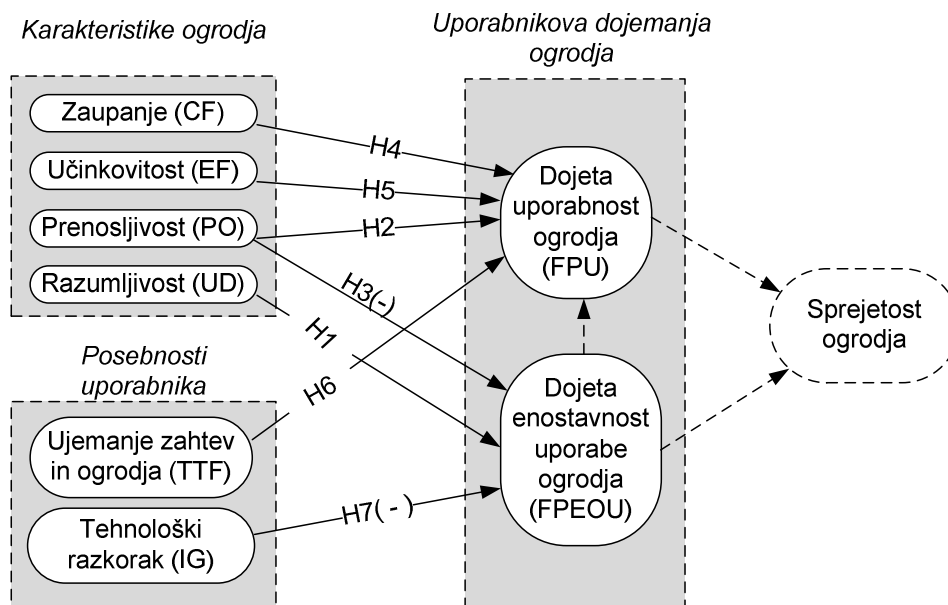
H15: Sprejetost ogrođja (FA) ima pozitiven vpliv na zadovoljstvo uporabe ogrođja (SAT).

H16: Namera trajne uporabe ogrođja (CFUI) ima pozitiven vpliv na neto prednosti uporabe ogrođja (NBF).

H17: Namera trajne uporabe ogrođja (CFUI) ima pozitiven vpliv na zadovoljstvo uporabe ogrođja (SAT).

3.7.5 TEORETIČNI MODELI

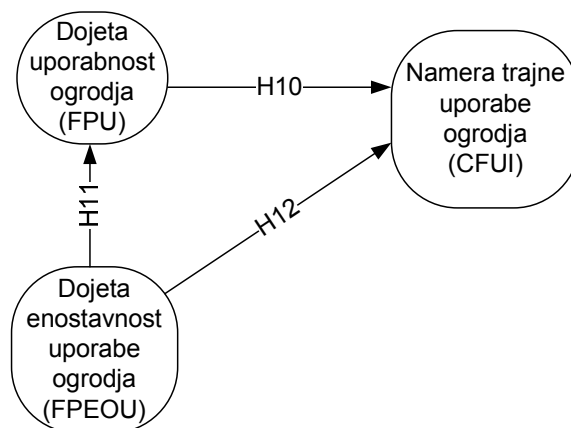
Na osnovi predhodno oblikovanih hipotez in faktorjev smo oblikovali konceptualne teoretične modele, ki so predstavljeni v nadaljevanju.



SLIKA 54: TEORETIČNI MODEL 1

Prvi teoretični model raziskave (Slika 54) sestavljajo šest neodvisnih faktorjev in dva odvisna faktorja. Iz modela je razvidno, da predstavljajo hipoteze H1 do H7 kavzalne povezave (polne puščice) med zunanjimi spremenljivkami modela TAM (karakteristike ogrodja in posebnosti uporabnika) in uporabnikovimi dojemaji ogrodij (dojeta uporabnost in dojeta enostavnost uporabe). Prekinjene puščice in črtkan zaobljen pravokotnik predstavljajo kavzalne povezave in faktor, ki ni bil vključen v analizo prvega teoretičnega modela, saj je bil analiziran že v predhodnih raziskavah (King & He 2006).

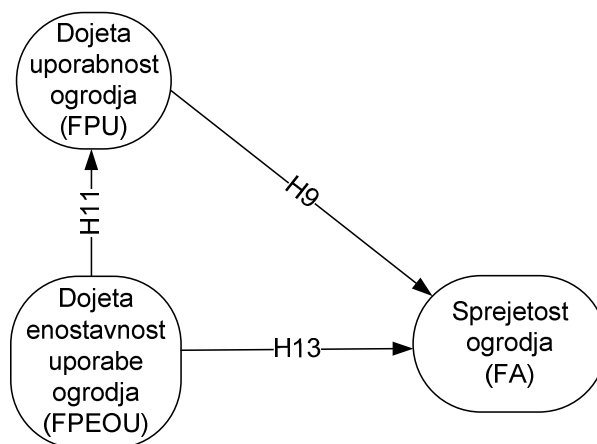
Z drugim raziskovalnim vprašanjem smo povezali štiri teoretične modele. Teoretični model 2A (Slika 55) predstavlja specializacijo poenostavljenega modela TAM, namenjenega trajni uporabi oziroma trajni nameri uporabe (*continued usage behavior*).



SLIKA 55: TEORETIČNI MODEL 2A

Zgornji model (Slika 55) temelji na konstruktih in povezavah, ki so se izkazale za veljavne v kontekstu mobilnega interneta (Hong, Thong, & Tam 2006). V omenjeni raziskavi, ki je primerjala različne modele, se je model TAM izkazal za najbolj generičnega, natančnega in enostavnega.

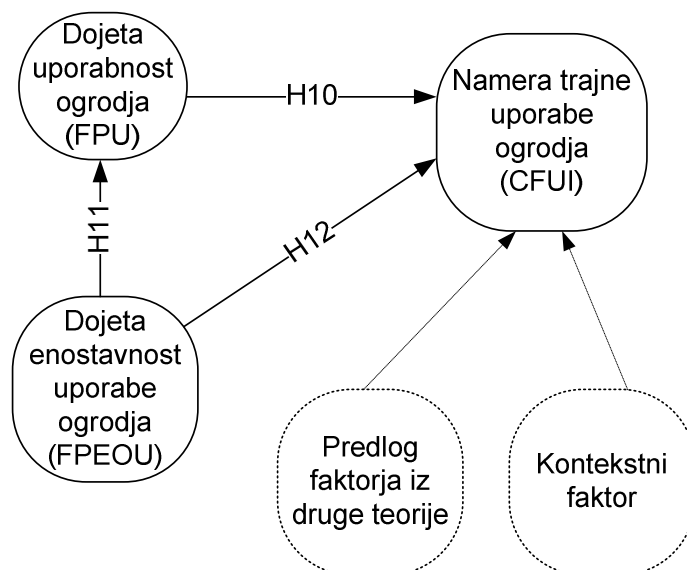
Model na spodnji sliki (Slika 68) predstavlja specializacijo poenostavljenega modela TAM s sprejetostjo ogrodja kot odvisno spremenljivko.



SLIKA 56: TEORETIČNI MODEL 2B

Za modela 2A in 2B torej predpostavljamo, da sta uporabna tudi v kontekstu ogrodij. Razlika med modeloma 2A in 2B je izključno v odvisni spremenljivki.

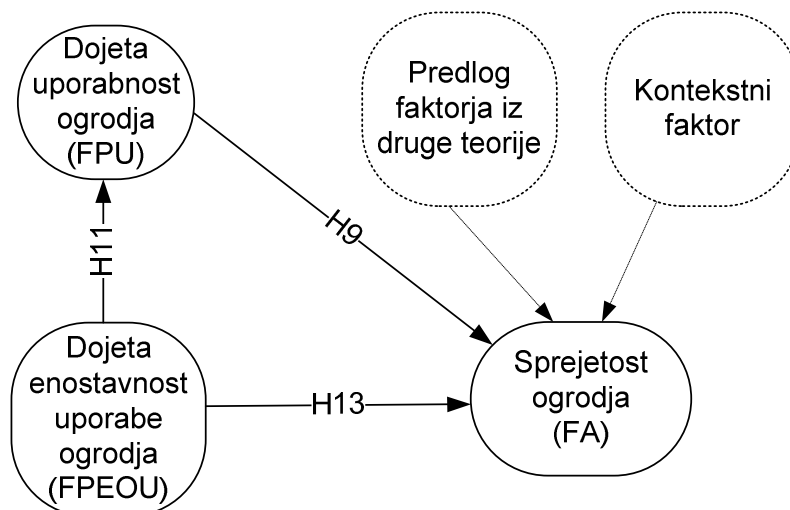
Na spodnji sliki (Slika 57) je predstavljena razširitev teoretičnega modela 2A s predlogom dveh faktorjev: faktorja iz druge teorije in kontekstnega faktorja.



SLIKA 57: TEORETIČNI MODEL 2C

Oba faktorja sta v času oblikovanja teoretičnih modelov še nedefinirana. Določena bosta na osnovi rezultatov analize teoretičnega modela 1. Naš cilj bo primerjati rezultate modelov 2A in 2C in tako ugotoviti, kateri zmed modelov je primernejši v kontekstu ogrodij.

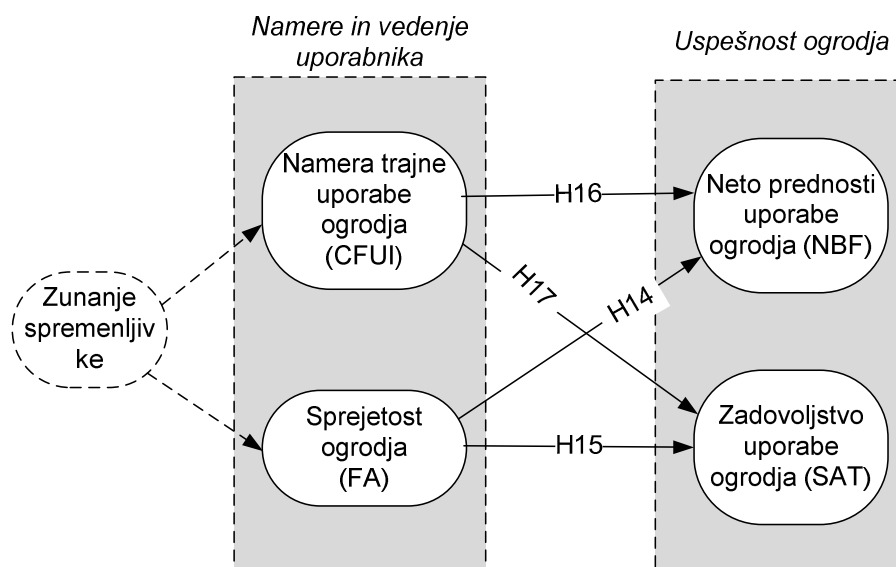
Na spodnji sliki (Slika 58) je predstavljena razširitev teoretičnega modela 2B s predlogom dveh faktorjev: faktorja iz druge teorije in kontekstnega faktorja.



SLIKA 58: TEORETIČNI MODEL 2D

Namen teoretičnega modela 2D je enak teoretičnemu modelu 2C, in sicer primerjati rezultate analize modelov 2B in 2D in tako ugotoviti, kateri zmed modelov je primernejši v kontekstu ogrodij. Enako kot pri modelu 2C bosta dodana faktorja določena na osnovi rezultatov analize teoretičnega modela 1.

Zadnji teoretični model (Slika 59) temelji na raziskovalnem vprašanju RV₃ in iz njega izhajajočih hipotezah.



SLIKA 59: TEORETIČNI MODEL 3

Model 3 vključuje dva eksogena, dva endogena faktorja in vse možne kombinacije povezav med eksogenimi in endogenimi faktorji. Namen teoretičnega modela 3 je demonstrirati posamezne povezave med odvisnimi konstrukti modela TAM in dimenzijami uspešnosti ogrodij. Operacionalizacija predstavljenih teoretičnih modelov je predstavljena v nadaljevanju.

3.7.6 VPRAŠALNIK

Operacionalizacijo konceptualnih modelov raziskave smo izvedli na osnovi obstoječih raziskovalnih vprašalnikov. Pri tem smo naša vprašanja izpeljali samo iz primernih, veljavnih in zanesljivih vprašanj (Tabela 21).

Večino konceptov smo operacionalizirali z uporabo Likertove lestvice⁸⁴ (*likert scale*). Likertova lestvica je posredna lestvica za merjenje stališč oziroma izjava, s katero vprašani izrazi stopnjo strinjanja ali nestrinjanja v zvezi z obravnavano zadevo (Uebersax 2006). Likertovo lestvico sestavlja množica trditev (*likert items*), ki merijo enak konstrukt. Pri vsaki trditvi mora anketiranec navesti, do kakšne mere se z njo strinja ali ne. Vrednosti, ki so definirane na ordinalni lestvici, so točno določene in definirane na lihem številu stopenj. Pri tem so opisne vrednosti simetrične glede na srednjo številčno vrednost (Tabela 20).

TABELA 20: LIKERTOVA LESTVICA S SEDMIMI STOPNJAMI

Opisna vrednost	Številska vrednost
Se v celoti ne strinjam	1
Se ne strinjam	2
Se delno ne strinjam	3
Sem neodločen	4
Se delno strinjam	5
Se strinjam	6
Se v celoti strinjam	7

Lestvica, ki ne izpolnjuje naštetih zahtev, ni Likertova lestvica, temveč kvečjemu ordinalna lestvica. Prednost Likertove lestvice pred ordinalno lestvico je, da jo lahko pri analizi podatkov obravnavamo kot intervalno lestvico.

Likertove lestvice nismo uporabili za merjenje faktorja SAT, pri katerem smo uporabili semantični diferencial (*semantic differential*). Semantični diferencial je posredna lestvica za merjenje stališč. Povezuje dve nasprotni besedi, respondent pa izbere točko, ki predstavlja njegovo mnenje.

⁸⁴ Lestvica se uporablja za zaprta vprašanja v anketi, s katerimi merimo respondentov odnos do raziskovanega pojava po posameznih atributih.

TABELA 21: VPRAŠANJA OPERACIONALIZIRANEGA TEORETIČNEGA MODELA⁸⁵

Konstrukti in Likertova vprašanja	Uporabljeni viri
CF: Zaupanje (Confidence)	
- CF1: <i>I believe that the framework is mature.</i>	(Sindre, Conradi, & Karlsson 1995)
- CF2: <i>The framework fails frequently.</i>	
- CF3: <i>The framework handles failures well if or when they occur.</i>	
UD: Razumljivost (Understandability)	
- UD1: <i>I believe the framework is self-descriptive.</i>	(Sindre, Conradi, & Karlsson 1995)
- UD2: <i>The accessibility, level of detail and quality of framework documentation is good.</i>	
- UD3: <i>The framework is easy to learn.</i>	(ISO 9126 2001)
PO: Prenosljivost (Portability)	
- AD1: <i>The framework can be easily adapted or extended to fulfill application requirements.</i>	(Sindre, Conradi, & Karlsson 1995)
- AD2: <i>The framework can be installed on different environments.</i>	(ISO 9126 2001)
- AD3: <i>The framework can be easily transferred from one environment to another.</i>	
EF: Učinkovitost (Efficiency)	
- EF1: <i>The framework requires too much of system resources.</i>	(ISO 9126 2001)
- EF2: <i>The framework provides appropriate response and processing times.</i>	
- EF3: <i>The framework slows down the system (computer).</i>	
IG: Tehnološki razkorak (Implementation gap)	
- IG1: <i>I felt that there existed a large gap between my existing skills and knowledge and those required by the framework.</i>	(Chau 1996)
- IG2: <i>Learning the framework was not a matter of building on what I knew already.</i>	
- IG3: <i>Major modification in our software development policies and procedures was necessary for the framework to truly fit in.</i>	
TTF: Ujemanje zahtev in ogrodja (Task technology fit)	
- TTF1: <i>The framework functions or services suited to application requirements in each individual case of its use.</i>	Nova trditev
- TTF1: <i>The framework provides suitable set of functions for my tasks and user objectives in each individual case of its use.</i>	(ISO 9126 2001)
FPEOU: Dojeta enostavnost uporabe ogrodja (Framework Perceived Ease of use)	
- FPEOU1: <i>The framework is rigid and inflexible to interact with.</i>	
- FPEOU2: <i>I find it is easy to get the framework to do what I want it</i>	

⁸⁵ Ker je bil anketni vprašalnik oblikovan in posredovan v angleškem jeziku, smo definirali operacionalizacijo le v angleškem jeziku.

Konstrukti in Likertova vprašanja	Uporabljeni viri
<i>to do.</i>	(Moore and Benbasat
- <i>FPEOU3: Overall, I believe that the framework is easy to use.</i>	1991)
- <i>FPEOU4: Learning to operate the framework is easy for me.</i>	
- <i>FPEOU5: I find it takes a lot of effort to become skillful at using the framework.</i>	
FPU: Dojeta uporabnost ogrodja (<i>Framework Perceived Usefulness</i>)	
- <i>FPU1: Using the framework increases my productivity.</i>	(Moore and Benbasat
- <i>FPU2: Using the framework increases my job performance.</i>	1991)
- <i>FPU3: Using the framework enables me to accomplish tasks more quickly.</i>	
- <i>FPU4: Overall, I find the framework useful in my job.</i>	
FA: Sprejetost ogrodja (<i>framework acceptance</i>)	
- <i>FA1: I have fully accepted the framework in my daily work.</i>	(Goodhue &
- <i>FA2: I feel that the framework constitutes an integral part of my daily work.</i>	Thompson 1995)
- <i>FA3: I consider myself a frequent user of the framework.</i>	
- <i>FA4: I fully use the capabilities of the framework.</i>	
CFUI: Namerna trajne uporabe ogrodja (<i>Continued framework usage intention</i>)	
- <i>CFUI1: I intend to increase my use of the framework in the future.</i>	(Hong, Thong, & Tam
- <i>CFUI2: I intend to continue my use of the framework in the future.</i>	2006)
- <i>CFUI3: I am not going to use the framework in the future</i>	
NBF: Neto prednosti uporabe ogrodja (<i>Net benefits of framework use</i>)	
- <i>NBF1: The framework increases the productivity of software development</i>	Nova trditev
- <i>NBF1: The framework increases the quality of software development</i>	
- <i>NBF3: The framework lowers the lead time for developing software</i>	
SAT: Zadovoljstvo z uporabo ogrodja (<i>Satisfaction with framework use</i>)	(Hong, Thong, & Tam
- <i>SAT1: How satisfied do you feel about your overall experience with the framework use?</i>	2006)
- <i>SAT2: How contented do you feel about your overall experience with the framework use?</i>	
- <i>SAT3: How delighted do you feel about your overall experience with the framework use?</i>	

Kot je razvidno iz tabele, smo v vprašalniku določena vprašanja negirali z namenom ugotavljanja napačnih odgovorov (*repeated questions*). Elektronska realizacija vprašalnika je prav tako omogočila naključno zaporedje vprašanj, kar je zmanjšalo medsebojne odvisnosti med podajanjem odgovorov (*ceiling effect*).

Poleg operacionaliziranih faktorjev je vprašalnik raziskave vključeval še naslednje elemente in (skupine) vprašanja:

- Zahvalo za pripravljenost sodelovati v anketi.
- Informacije o respondentu. Ta del vprašalnika je vseboval demografska vprašanja, povezana s spolom, izobrazbo in izkušnjami anketiranca. Vprašanja so bila zaprtega tipa.
- Informacije o vrednotenem ogrodju. Ta del vprašalnika je vseboval odprto vprašanje o nazivu ogrodja, ki ga anketiranec vrednoti v vprašalniku, o pogostosti uporabe ogrodja, vrsti uporabe ogrodja in vlogi, ki jo ima anketiranec pri uporabi ogrodja. Vprašalnik je torej respondenta usmeril v identifikacijo ogrodja, ki ga najpogosteje uporablja. Osredotočenje anketiranca na specifično ogrodje je skladno s Churchillovimi priporočili, ki temeljijo na ugotovitvi, da definiranje in analiziranje specifičnega objekta povečujeta zanesljivost in veljavnost rezultatov raziskave (Churchill 1979).
- Zaključek ankete, ki je vseboval odprto vprašanje za vnos komentarja in vprašanje o nadaljnjem obveščanju anketiranca o izvedbi raziskave.

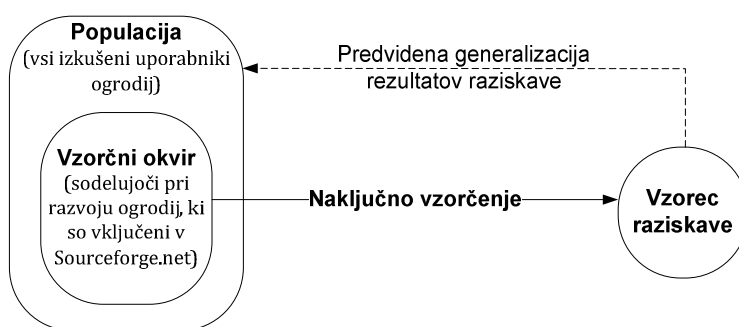
Vprašanja o respondentu in vrednotenem ogrodju smo, če je bilo mogoče, oblikovali zaprtega tipa. Izjema sta bili vprašanja o starosti respondenta in imenu vrednotenega ogrodja. Vprašanja, ki so povezana z izkušnostjo respondenta, smo oblikovali na lestvici Fibonaccijevega zaporedja z razlogom, ker smo menili, da se z večanjem izkušnosti respondenta, manjša natančnost določanja izkušnosti. Iz tega razloga so intervali ponujenih odgovorov vedno večji. Celotna vsebina vprašalnika je dostopna v prilogi disertacije.

3.7.7 VZORČENJE

Ankete se med sabo močno razlikujejo glede na vrsto izbranega vzorca⁸⁶ anketirancev (McBurney & White 2003). V okviru naše raziskave smo izvedli naključno vzorčenje⁸⁷ (*random sampling*). Naključno vzorčenje je proces, pri katerem ima vsak član iz vzorčnega okvira ali populacije enako in neodvisno možnost biti izbran v vzorec.

Idealno populacijo⁸⁸ (*population*) so v našem primeru predstavljali vsi izkušeni uporabniki ogrodij, ki imajo poleg večkratnih uporab določenega ogrodja še izkušnje z razvojem ogrodja. Za takšne uporabnike smo predpostavljali, da lahko dobro ovrednotijo faktorje, ki vplivajo na njihova dožemanja ogrodij.

Iz praktičnih razlogov procesa vzorčenja nismo izvedli na definirani populaciji, temveč na vzorčnem okviru⁸⁹ (*sample frame*). Vzorčni okvir so v našem primeru predstavljali člani odprtokodnih ogrodij skupnosti Sourceforge.net, do katerih nam je bil omogočen dostop. Sourceforge.net je največji repozitorij odprtokodnih projektov. V Sourceforge.net je vključenih preko 180.000 projektov, v katerih sodeluje več kot 1,880.000 članov⁹⁰. Sourceforge.net vsebuje obsežno klasifikacijo odprtokodnih projektov, v okviru katere je več kot 5000 projektov klasificiranih v projekte razvoja ogrodij. V projektih razvoja ogrodij sodeluje preko 11.000 članov.



SLIKA 60: GRAFIČNA PONAŽORITEV POSTOPKA VZORČENJA

Sourceforge.net predstavlja tudi raziskovalno okolje v smislu, da raziskovalcem omogoča dostop do posnetkov podatkovne baze. Za potrebe naše raziskave smo uporabili posnetek podatkovne

⁸⁶ (Statistični) vzorec zajema samo del populacije, na podlagi katerega dobimo oceno parametrov pri delnem opazovanju ali vzorčenju. Vzorci so lahko slučajni ali vnaprej določeni.

⁸⁷ Vzorčenje je proces opredelitve oziroma načrtovanja vzorca raziskave. Opredeliti je treba: vzorčno enoto (koga preučujemo), velikost vzorca (koliko oseb moramo preučiti) in postopek vzorčenja (kako izbrati preizkusne osebe). Osnova za vzorčenje so znani podatki o populaciji.

⁸⁸ Populacija predstavlja množico elementov (oseb), na katere se posplošijo rezultati raziskave oziroma vzorca.

⁸⁹ Vzorčni okvir predstavlja populacijo, v okviru katere bo izveden proces vzorčenja.

⁹⁰ Informacija je bila po www.sourceforge.net povzeta dne 11. 7. 2008.

baze, različice »december 2007«. Povpraševanje po posnetku je vrnilo 5216 projektov, ki so se klasificirali v kategorijo ogrodiv, in 11357 članov teh projektov. Ti člani so predstavljali vzorčni okvir naše raziskave.

Postopek naključnega vzorčenja smo izvedli z uporabo preglednice in makra za generiranje naključnih števil. Na osnovi tega procesa smo dobili vzorec, ki je vključeval 4000 naključno izbranih članov projektov razvoja odprtokodnih ogrodiv. Za vsakega člana smo pridobili naslednje podatke: ime, priimek, uporabniško ime, projekt in elektronski naslov.

3.7.8 VZPOSTAVITEV IN IZVEDBA ANKETE

Pred vzpostavitvijo dejanske ankete je bila izvedena še pilotna anketa. Namen izvedbe pilotne ankete je bil ugotoviti, ali anketiranci razumejo vsebino vprašalnika in ali imajo vsebinske in oblikovne pripombe glede podanega vprašalnika. V pilotni anketi je sodelovalo 22 študentov, ki so imeli izkušnje z uporabo ogrodiv. Na osnovi vrnjenih vprašalnikov, komentarjev študentov in opazovanja izvajalca pilotne ankete so se izvedle naslednje spremembe v anketnem vprašalniku:

- izboljšala so se navodila za izpolnjevanje ankete,
- določena vprašanja so se preoblikovala in
- dodatno so se pojasnili termini v določenih vprašanjih.

Modificiran anketni vprašalnik smo vzpostavili v spletnem anketnem okolju QuestionPro.com, za katero smo pridobili polletno prosto akademsko licenco. Poleg vzpostavitve vprašalnika je okolje omogočilo še oblikovanje poštnega seznama anketirancev, ki so po elektronski pošti prejeli poosebljena vabila za sodelovanje v anketi.

Članom vzorca raziskave, ki se na sodelovanje na anketi niso odzvali v 48 urah, je bil poslan vljuden opomnik. Odzivi anketirancev, glede na status prejetega vprašalnika so podani v spodnji tabeli (Tabela 22).

TABELA 22: STATUSI PREJETIH VPRAŠALNIKOV

Status vprašalnika	Število	Delež
Število poslanih vprašalnikov	4000	100.0%
Število odprtih vprašalnikov	934	23.4%
Število začetih vprašalnikov	710	17.8%
Število zaključenih vprašalnikov	447	11.2%
Število nepravilnih⁹¹ vprašalnikov	56	1.4%
Število uporabnih vprašalnikov	391	9.8%

Vhod v analizo podatkov je predstavljalo 391 uporabnik vprašalnikov, kar je 9.8% vseh poslanih vprašalnikov. Razmeroma majhen odziv (*response rate*) je v skladu s pričakovanji in v področju drugih spletnih anket. Velikost vzorca (N=391) je zadostovala priporočilom za statistično analizo podatkov z metodo SEM⁹².

3.7.9 METODA ANALIZE PODATKOV

Zajemu podatkov je sledila statistična analiza podatkov, v okviru katere smo želeli na osnovi pridobljenih izkustvenih podatkov demonstrirati veljavnost operacionaliziranega modela raziskave. Prvi korak metode analize podatkov je predstavljala izbira statistične tehnike, ki je bila izvedena na osnovi analize teoretičnega modela raziskave (Slika 54) in njegove operacionalizacije (Tabela 21). Pri tem smo upoštevali naslednje kriterije:

- **Vrsta empiričnih podatkov.** Statistična tehnika je odvisna od merskih lestvic empiričnih indikatorjev (Tabela 23).

TABELA 23: ODVISNOST STATISTIČNE TEHNIKE OD PODATKOV

Vrsta statistike	Merska lestvica
Neparametrična	Nominalna ali opisna
Neparametrična	Ordinalna
Neparametrična Parametrična⁹³	Približek intervalne
Parametrična	Intervalna
Parametrična	Razmernostna

⁹¹ Vprašalniki z nepopolnimi odgovori ali vprašalniki, ki so bili zavrženi na osnovi kontrolnih vprašanj.

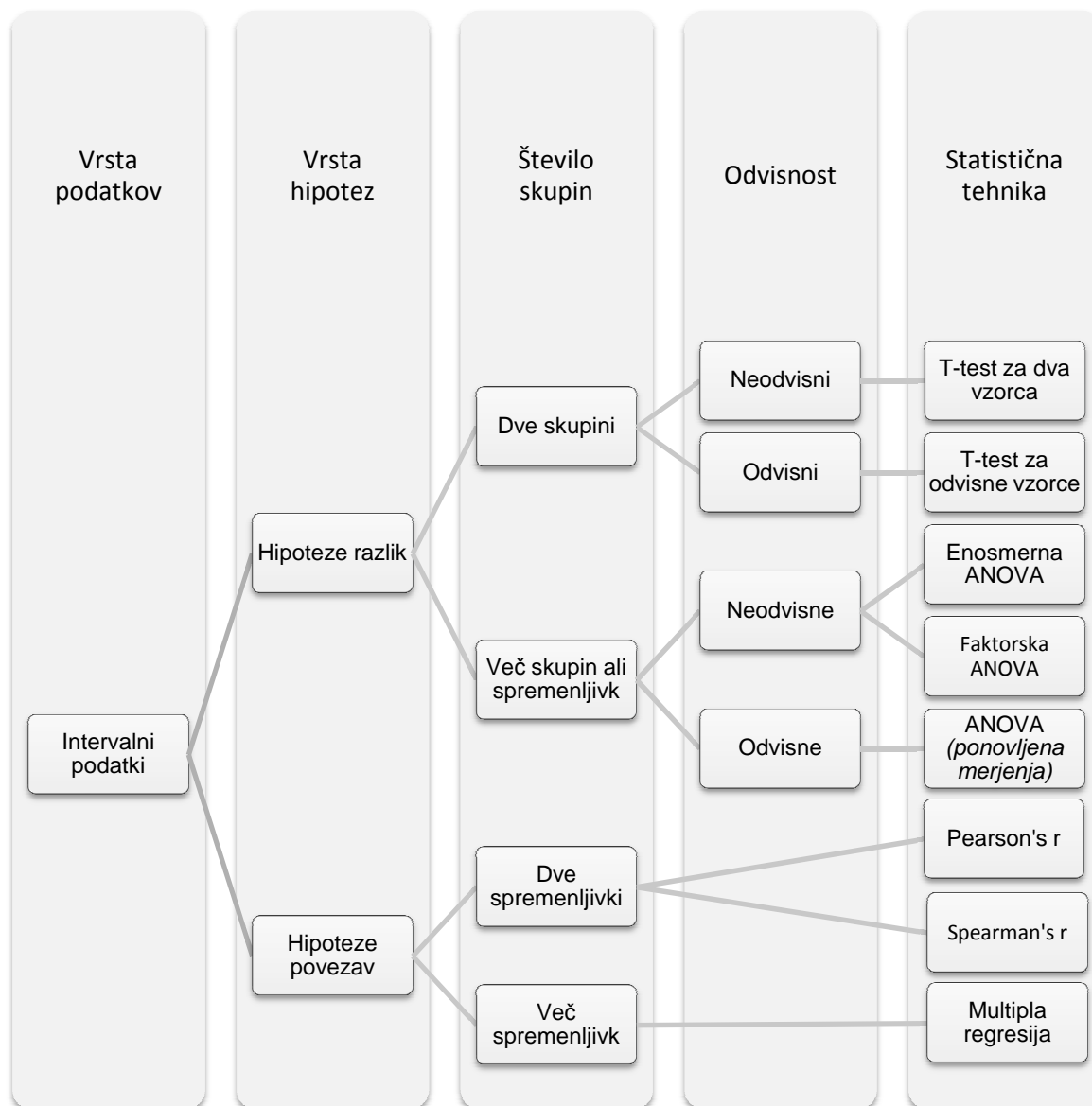
⁹² Stevens (1986) priporoča 15 respondentov na merjeno spremenljivko.

⁹³ V parametrično statistiko spadajo statistični testi, ki izhajajo iz predpostavke, da je distribucija populacije, iz katere izbiramo vzorce, parametrična (običajno normalna). Nasprotje je neparametrična statistika.

Indikatorji v našem teoretičnem modelu temeljijo na Likertovi lestvici, ki se uvršča med približke intervalne lestvice. Zato smo izbrali parametrično statistiko.

- **Vrsta hipotez.** V okviru raziskave nas zanimajo odvisnosti (hipoteze) med neodvisnimi in odvisnimi spremenljivkami.
- **Število vzorcev.** V okviru raziskave imamo en vzorec. Cilj statistike je posplošiti rezultate vzorca na celotno populacijo. Izbrali smo statistično tehniko, ki preučuje teste enega vzorca (*one sample tests*).
- **Število spremenljivk.** V okviru ankete imamo skupaj šest teoretičnih modelov, pri tem je vsak model sestavljen iz najmanj treh faktorjev. Iz tega razloga smo izbrali multivariatno statistično tehniko (*multivariate statistics*).

Na osnovi navedene analize raziskovalnega problema in odločitvenega drevesa za izbiro ustrezne statistične tehnike (Slika 61) ustreza naši raziskavi multipla regresija.



SLIKA 61: ODLOČITVENO DREVO ZA IZBIRO USTREZNE STATISTIČNE TEHNIKE

Multiple regresije je več vrst (glej prilogo C). Našemu teoretičnemu modelu ustreza analiza poti (*path analysis*), ki predstavlja razširitev regresije na način, ki omogoča analizo več odvisnih spremenljivk.

Dodatna lastnost teoretičnih modelov raziskave je, da vsebujejo latentne⁹⁴ in merljive spremenljivke⁹⁵. Za analizo pripadnosti empiričnih indikatorjev posameznemu faktorju je primerna faktorska analiza (*factor analysis*). Ker smo pripadnosti že opredelili oziroma so bile potrjene v okviru predhodnih raziskav, ustreza naši raziskavi poseben primer faktorske analize, ki se imenuje potrditvena faktorska analiza (*confirmatory factor analysis*).

⁹⁴ Latentna spremenljivka, koncept in faktor predstavljajo sinonime.

⁹⁵ Merljiva spremenljivka in empirični indikator sta sinonima.

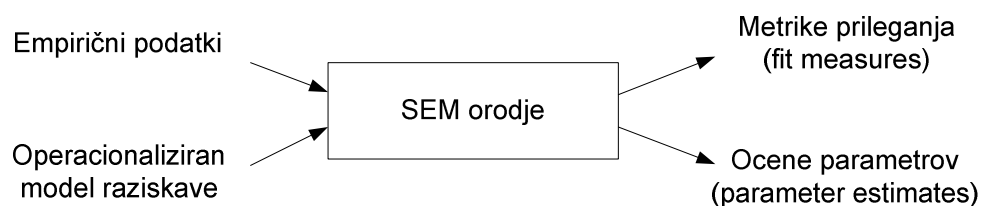
Naši teoretični modeli so torej sestavljeni iz dveh vrst modelov, ki se rešujejo z različnimi statističnimi tehnikami:

- **Merljivi modeli.** Merljivi modeli (*measurement models*) so modeli, ki so sestavljeni iz posameznega faktorja teoretičnega modela in pripadajočih empiričnih indikatorjev.
- **Strukturni modeli.** Strukturni modeli (*structural model*) predstavljajo teoretične modele brez empiričnih indikatorjev (od Slika 54 do Slika 59).

Statistična tehnika, ki omogoča sočasno analiziranje strukturnega in merljivih modelov, se imenuje modeliranje strukturnih enačb ali krajše SEM (*Structural Equation Modeling*). SEM predstavlja drugo generacijo statističnih tehnik. V nasprotju s prvo generacijo statističnih tehnik, kot je regresija, omogoča SEM podati odgovore na povezana raziskovalna vprašanja na enoten, sistematičen in celovit način (Slika 62). SEM smo uporabili še iz naslednjih razlogov:

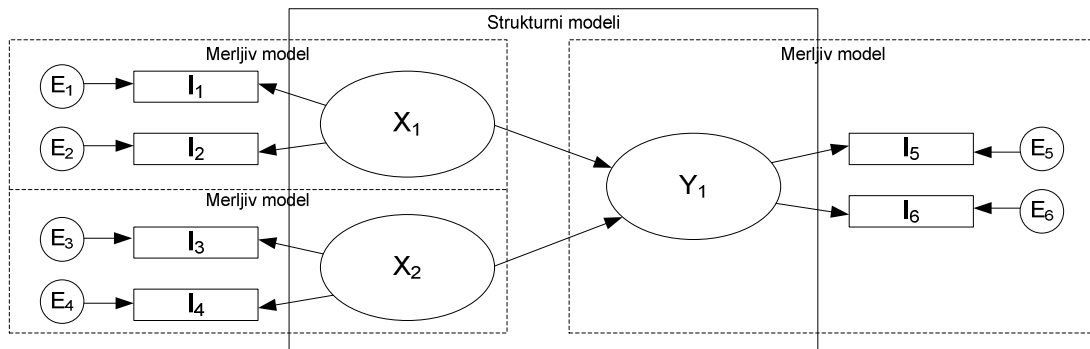
- **Potrjevanje teorij.** SEM je bolj primerna za potrjevanje teorij (*theory testing*) kot za oblikovanje teorij (*theory development*). SEM ustreza našemu cilju raziskave, saj imamo teoretične modele že oblikovane in jih želimo potrditi. SEM je prav tako primerna za kombiniran pristop, kjer se izmenjujeta potrditveni in raziskovalni pristop (*confirmatory – exploratory approach*). Takšen pristop je tudi najpogostejši (Garson 2008).
- **Analiza tehnik sorodnih raziskav.** Sorodne raziskave, ki so analizirale modele z latentnimi spremenljivkami, v vse večji meri uporabljajo SEM za potrjevanje njihovih modelov (Gefen, Straub, & Bordeau 2000).

SEM sestoji iz dveh delov (Anderson & Gerbing 1988): (1) merljivih modelov, ki prikazujejo povezave med latentnimi spremenljivkami in njihovimi indikatorji, in (2) strukturnega modela, ki ponazarja kavzalne povezave med eksogenimi in endogenimi faktorji.



SLIKA 62: ILUSTRACIJA DELOVANJA SEM

Merljivi modeli so sestavni deli modelov SEM, ki obravnavajo posamezen faktor in pripadajoče indikatorje (Slika 63).



SLIKA 63: MERLJIVI MODELI IN STRUKTURNI MODEL V MODELU SEM

Merljive modele smo določili s preoblikovanjem modelov SEM na naslednji način (Gefen, Straub, & Bordeau 2000):

- Določili smo enosmerne povezave (kavzalne povezave) med faktorji in pripadajočimi indikatorji.
- Določili smo enosmerne povezave med napakami oziroma motnjami indikatorjev in pripadajočimi indikatorji.
- Vsak možen par latentnih spremenljivk smo povezali z dvosmernimi povezavami (kovariance).
- Izločili smo vse kavzalne povezave med latentnimi spremenljivkami.

Za indikatorje končnega merljivega modela smo izvedli analizo konvergentne veljavnosti (*convergent validity*), diskriminacijske veljavnosti (*discriminant validity*) in notranje skladnosti (*internal consistency*).

Konvergentno veljavnost smo analizirali z metriko Cronbach's Alpha, ki se pogosto uporablja za ugotavljanje pripadnosti skupine indikatorjev skupni latentni spremenljivki. Faktorji, ki so preseglji mejno vrednost Cronbach's Alpha > 0.7, so bili sprejeti.

Drugo metriko, ki smo jo uporabili za potrebe konvergentne veljavnosti, je predstavljala metrika kompozitne zanesljivosti (*composite reliability*), ki se je izračunala naslednje:

$$Cr = \frac{(\sum \lambda_i)^2}{(\sum \lambda_i)^2 + \sum_i var(\varepsilon_i)}$$

Pri tem je λ_i predstavljala utež indikatorja (*factor loading*) x_i . $Var(\varepsilon_i)$ predstavlja varianco napake indikatorja x_i , ki se je izračunala kot $var(\varepsilon_i) = 1 - \lambda_i^2$.

Diskriminacijska veljavnost (*discriminant validity*) se je izračunala z metriko AVE (*average variance extracted*), naslednje:

$$AVE = \frac{\sum \lambda_i^2}{\sum \lambda_i^2 + \sum_i var(\varepsilon_i)}$$

Strukturne modele smo določili na naslednji način (Gefen, Straub, & Bordeau 2000):

- Določili smo enosmerne povezave (kavzalne povezave) med faktorji in pripadajočimi indikatorji.
- Določili smo enosmerne povezave med napakami oziroma motnjami indikatorjev in pripadajočimi indikatorji.
- Na osnovi hipotez H1-H8 in H10-17 smo določili enosmerne povezave (kavzalne povezave) med neodvisnimi in odvisnimi faktorji.
- Eksogene spremenljivke teoretičnih modelov smo povezali z dvosmernimi povezavami (kovariance).

Veljavnost merljivih in strukturnih modelov smo preverili z naslednjimi metrikami prileganja modelov (*goodness-of-fit*)⁹⁶: GFI (*goodness-of-fit*), AGFI (*adjusted goodness-of-fit*), CFI (*comparative fit index*), NFI (*normed fit index*), TLI (*Tucker Lewis Coefficient*), RMR (*root mean squared residual*) in RMSEA (*root mean squared error of approximation*). Mejne vrednosti so bile povzete po sorodnih raziskavah (Hong, Thong, & Tam 2006). Vrednosti metrike Hi-kvadrat (*chi squared*) pri analizi veljavnosti nismo upoštevali, saj so zaradi velikosti vzorca (n=391) primernejše predtem definirane metrike (glej prilogo C).

Za statistično analizo smo uporabili statistični paket SPSS 16.0 in njegov dodatek AMOS 16.0, ki je namenjen izključno obdelavi SEM. V okolju AMOS smo uporabili metodo največjega verjetja (*maximum likelihood estimation*), ki je privzeta funkcija nedoslednosti⁹⁷ v AMOS-u. Večino rezultatov v tabelarni in grafični obliki smo zaradi preglednosti preoblikovali in prevedli v slovenščino. Izvorni rezultati v tabelarni in grafični obliki so zbrani v prilogah.

⁹⁶ Metrike prileganja modelov so podrobneje predstavljene v prilogah.

⁹⁷ Funkcija nedoslednosti (*discrepancy function*) je matematična funkcija, ki pojasnjuje, kako dobro se strukturni model prilagaja empiričnim podatkom. Višje vrednosti nakazujejo slabo ujemanje strukturnega modela z empiričnimi podatki.

3.7.10 REZULTATI

V nadaljevanju so predstavljeni poglavitni rezultati statistične analize podatkov, ki so strukturirani naslednje:

- V podpoglavju Opisna statistika⁹⁸ so povzeta predvsem vprašanja, ki se nanašajo na profil respondenta in izkušnje respondenta z vrednotenim ogrođjem.
- V podpoglavjih, ki sledijo opisni statistiki, so podani poglavitni rezultati statistične analize merljivih in strukturnih modelov, ki temeljijo na predstavljenih teoretičnih modelih.

3.7.10.1 OPISNA STATISTIKA

V nadaljevanju so podani rezultati opisne statistike. V spodnji tabeli (Tabela 24) so zapisani vrednosti, frekvenca, odstotki in veljavni odstotki⁹⁹ spremenljivk, ki so poizvedovali po značilnostih in izkušnjah respondenta.

TABELA 24: OPISNA STATISTIKA RESPONDENTA - FREKVENCE

Spremenljivka	Koda	Vrednost	Frekvenca	Odstotek	Veljavni odstotek
Spol	1	Moški	378	2.3	2.3
	2	Ženski	9	96.7	97.7
		Manjkajoče	4	1.0	
Izobrazba	1	Srednja šola ¹⁰⁰	53	13.6	13.7
	2	Diploma	153	39.1	39.5
	3	Magisterij	125	32.0	32.3
	4	Doktorat	30	7.7	7.8
	5	Drugo	26	6.6	6.7
		Manjkajoče	4	1.0	-
Izkušnje z razvojem programske opreme	1	0–1 leto	1	0.3	0.3
	2	1–2 leti	12	3.1	3.1
	3	3–5 let	69	17.6	17.6
	4	5–8 let	94	24.0	24.0
	5	8–13 let	102	26.1	26.1
	6	13–21 let	69	17.6	17.6
	7	> 21 let	44	11.3	11.3
		Manjkajoče	0	0	-

⁹⁸ Opisna statistika je skupina statističnih metod, ki se ukvarjajo s povzemanjem pridobljenih podatkov. Te metode iščejo opisne (meta) podatke o populaciji in njenih sestavnih delih, da bi ustvarile pregledni opis.

⁹⁹ Veljavni odstotki izražajo delež posamezne vrednosti spremenljivke brez upoštevanja manjkajočih vrednosti.

¹⁰⁰ *High school*.

Spremenljivka	Koda	Vrednost	Frekvenca	Odstotek	Veljaven odstotek
Število uporabljenih ogrodij	1	1	13	3.3	3.4
	2	2	24	6.1	6.2
	3	3-5	118	30.2	30.5
	4	5-8	83	21.2	21.4
	5	8-13	56	14.3	14.5
	6	>13	93	23.8	24.0
		Manjkajoče		4	1.0
Izkušnje z uporabo ogrodij	1	<1 leto	12	3.1	3.1
	2	0-1 leto	31	7.9	7.9
	3	1-2 leti	60	15.3	15.4
	4	3-5 let	102	26.1	26.2
	5	5-8 let	91	23.3	23.3
	6	8-13 let	57	14.6	14.6
	7	>13 let	37	9.5	9.5
		Manjkajoče		1	0.3
Tip respondenta	1	Zgodnji	220	56.3	66.6
	2	Pozen	110	28.1	33.4
		Manjkajoče ¹⁰¹	61	15.6	-

V naslednji tabeli (Tabela 25) so podane naslednje opisne statistike: število prejetih odgovorov (N), najmanjša vrednost, največja vrednost, povprečje in standardna deviacija.

TABELA 25: OPISNA STATISTIKA - POVPREČJA IN ODKLONI

Spremenljivka	N	Min	Maks	Povprečje ¹⁰²	Standardna deviacija
Spol	387	1	2	1.0	0.2
Starost v letih	377	0	65	31.7	8.1
Izobrazba	387	1	5	2.5	1.0
Izkušnje z razvojem programske opreme	391	1	7	4.7	1.3
Število uporabljenih ogrodij	387	1	6	4.1	1.4
Izkušnje z uporabo ogrodij v letih	390	1	7	4.4	1.5
Vrsta uporabe vrednotenega ogrodja	387	1	2	1.2	0.4
Število zaključenih projektov z uporabo vrednotenega ogrodja	389	1	8	3.9	1.7
Izkušnje z uporabo vrednotenega ogrodja	390	1	6	2.8	1.3

¹⁰¹ Zaradi omejitev spletne ankete questionpro.com nekaterih respondentov ni bilo možno klasificirati med zgodnje ali pozne respondente.

¹⁰² Glej vrednosti posameznih številskih kod, ki se nahajajo v Tabela 24.

Spremenljivka	N	Min	Maks	Povprečje ¹⁰²	Standardna deviacija
Pogostost uporabe vrednotenega ogrodja	388	1	4	1.8	1.0

Iz zgornjih tabel (Tabela 24 in Tabela 25) je možno razbrati, da je bil tipičen respondent diplomant, moškega spola, star 32 let, z dolgoletnimi izkušnjami z razvojem programske opreme. Respondent je v okviru razvoja programske opreme v povprečju preizkusil od pet do osem različnih ogrodij in ima okoli petletne izkušnje z ogrodji.

Respondenti so v anketi ocenjevali 272 različnih ogrodij, pri čemer je seznam desetih najpogosteje vrednotenih ogrodij podan v spodnji tabeli (Tabela 26).

TABELA 26: SEZNAM DESETIH NAJPOGOSTEJE VREDNOTENIH OGRODIJ

Ime ogrodja	Frekvenca	Odstotek	Veljaven odstotek	Kumulativen odstotek
Spring	23	5.9	5.9	5.9
.NET	20	5.1	5.1	11.0
Struts	15	3.8	3.8	14.8
Eclipse	9	2.3	2.3	17.1
Django	7	1.8	1.8	18.9
Hibernate	6	1.5	1.5	20.5
Wicket	5	1.3	1.3	21.7
Qt	4	1.0	1.0	22.8
Ruby on Rails	4	1.0	1.0	23.8
Symfony	4	1.0	1.0	24.8

Iz seznama desetih najpogosteje vrednotenih ogrodij je razvidno, da so respondenti ocenjevali trenutno aktualna ogrodja. Pri tem so kljub dejstvu, da smo jih izbrali s seznama odprtokodnih ogrodij, ocenjevali tudi lastniška ogrodja (na primer: .NET).

V naslednji tabeli (Tabela 27) so podani vrednosti, frekvence, odstotki in veljavni odstotki spremenljivk, ki so poizvedovali po značilnostih in izkušnjah z uporabo ogrodja, ki ga je respondent vrednotil v okviru raziskave.

TABELA 27: OPISNA STATISTIKA IZKUŠENJ RESPONDENTA Z VREDNOTENIM OGRODJE

Spremenljivka	Vrednost	Frekvenca	Odstotek	Veljaven odstotek	
Tip uporabe ogrodja	1	Prostovoljen	302	77.2	78.0
	2	Obvezen	85	21.7	22.0
		Manjkajoče	4	1.0	-
Število zaključenih projektov z uporabo ogrodja	1	0	27	6.9	6.9
	2	1	53	13.6	13.6
	3	2	71	18.2	18.3
	4	3-5	118	30.2	30.3
	5	5-8	63	16.1	16.2
	6	8-13	26	6.6	6.7
	7	13-21	8	2.0	2.1
	8	>21	23	5.9	5.9
		Manjkajoče	2	0.5	-
Trajanje uporabe ogrodja	1	< 1 leto	66	16.9	16.9
	2	1-2 leti	102	26.1	26.2
	3	2-3 leta	100	25.6	25.6
	4	3-5 let	81	20.7	20.8
	5	5-8 let	30	7.7	7.7
	6	> 8 let	11	2.8	2.8
		Manjkajoče	1	0.3	-
Pogostost uporabe ogrodja	1	Dnevno	198	50.6	51.0
	2	Tedensko	120	30.7	30.9
	3	Mesečno	31	7.9	8.0
	4	Manj kot mesečno	39	10.0	10.1
		Manjkajoče	3	0.8	-

Iz prejšnje tabele je razvidno, da je tipičen respondent izbral ogrodje, ki ga uporablja, na prostovoljni osnovi. Z vrednotenim ogrođjem je respondent zaključil tri do pet projektov v obdobju med dvema in tremi leti. Večina respondentov uporablja vrednoteno ogrođje na dnevni ravni. Iz navedenega lahko sklepamo, da so respondenti izkušeni uporabniki vrednotenega ogrođja.

3.7.10.2 STATISTIČNA ANALIZA TEORETIČNEGA MODELA 1

REZULTATI MERLJIVEGA MODELA

Analiza rezultatov merljivega modela, ki se je izpeljal iz teoretičnega modela številka 1, je pokazala, da so bile tri od sedmih metrik prileganja zunaj priporočljivih vrednosti (glej uokvirjene vrednosti v Tabela 28), kar je pokazalo na nezadostno skladanje empiričnih podatkov in modela.

TABELA 28: METRIKE PRILEGANJA ZAČETNEGA IN KONČNEGA MERLJIVEGA MODELA

Metrika	Priporočljiva vrednost	Začetni merljiv model	Končni merljiv model
GFI	≥ 0.90	0.88	0.91
AGFI	≥ 0.80	0.85	0.88
NFI	≥ 0.90	0.88	0.92
TLI (NNFI)	≥ 0.90	0.91	0.94
CFI	≥ 0.90	0.93	0.95
RMSR	≤ 0.10	0.11	0.09
RMSEA	≤ 0.08	0.06	0.06
X²	-	732.8	451.1
(df,p)		(296,p < 0.001)	(202,p < 0.001)
X²/df	≤ 3	2.48	2.23

Izboljšanja merljivega modela smo se lotili z iterativnim procesom, kjer smo v posamezni iteraciji spremenili izbran indikator. Dobljene rezultate smo nato primerjali z začetnim modelom. Indikatorje, ki so bili predmet modifikacij, smo izbrali na osnovi uteži faktorjev (*factor loading*) in napotkov za modifikacije (*modification indices*), ki jih predlaga AMOS.

V končnem merljivem modelu so bile vse vrednosti metrik prileganja znotraj priporočljivih vrednosti. Iz končnega modela smo zaradi odstopanj izključili indikatorje TTF3, FPEOU1, FPEOU4 in PO1 (Slika 63). Opisna statistika indikatorjev končnega merljivega modela je prikazana v naslednji tabeli (Tabela 29).

TABELA 29: OPISNA STATISTIKA INDIKATORJEV KONČNEGA MERLJIVEGA MODELA

Indikator	N	Povprečje	Standardna deviacija	Factor loadings ¹⁰³	Squared multiple correlations ¹⁰⁴
PO2	391	2.00	1.39	0.89	0.79
PO3	391	2.27	1.45	0.96	0.92
CF1	391	2.45	1.45	0.70	0.48
CF2	391	2.20	1.17	0.73	0.54
CF3	391	2.59	1.27	0.70	0.49
EF1	391	2.79	1.54	0.88	0.78
EF2	391	2.27	1.09	0.65	0.42
EF3	391	2.55	1.50	0.82	0.67
IG1	391	5.22	1.58	0.77	0.59
IG2	391	4.76	1.59	0.68	0.46
IG3	391	5.01	1.65	0.69	0.47
FPEOU2	391	2.35	1.09	0.88	0.78
FPEOU3	391	2.46	1.21	0.82	0.67
FPEOU5	391	2.72	1.30	0.78	0.60
FPU1	391	1.87	1.07	0.89	0.80
FPU2	391	2.02	1.20	0.93	0.87
FPU3	391	1.99	1.18	0.95	0.90
FPU4	391	1.83	1.06	0.89	0.78
TTF1	391	2.06	1.32	0.83	0.69
TTF2	391	2.16	1.08	0.81	0.65
UD1	391	3.07	1.48	0.78	0.61
UD2	391	3.14	1.62	0.56	0.31
UD3	391	2.79	1.38	0.77	0.59

Za indikatorje končnega merljivega modela smo opravili še T-test za neodvisne skupine¹⁰⁵, da bi ugotovili, ali obstaja signifikantna razlika ($p < 0.05$) med zgodnjimi in poznimi¹⁰⁶ respondenti. Na osnovi izvedenega T-testa nismo našli statistične razlike med skupinama za nobenega od indikatorjev končnega merljivega modela¹⁰⁷.

Vrednosti AVE (*Average Variance Extracted*) za vse faktorje so presegle mejno vrednost 0.5, ki so jo predlagali (Fornell & Larcker 1981). Na osnovi AVE ugotavljamo, da je notranja konsistentnost faktorjev dobra.

¹⁰³ Vrednosti so povzete iz tabele »Standardized regression weights« v orodju AMOS.

¹⁰⁴ Vrednosti so povzete iz tabele »Squared multiple correlations« v orodju AMOS.

¹⁰⁵ Analize normalne porazdelitve podatkov (*skewness, kurtosis*) nismo potrebovali, ker je velikost vzorca presegala velikost $N > 100$.

¹⁰⁶ Respondenti, ki so pričeli izpoljevati anketo ob prejemu opomniku.

¹⁰⁷ Rezultati T-testa so v prilogi.

TABELA 30: ZANESLJIVOST KONSTRUKTOV

Faktor	Število indikatorjev	Povprečje	Standardna deviacija	AVE	Kompozitna zanesljivost (<i>composite reliability</i>)	Cronbach's alpha
PO	2	2.14	1.42	0.85	0.92	0.92
CF	3	2.41	1.30	0.50	0.75	0.75
EF	3	2.54	1.38	0.62	0.83	0.82
IG	3	5.00	1.61	0.51	0.75	0.75
FPPEOU	3	2.20	1.19	0.68	0.87	0.86
FPU	4	1.93	1.13	0.84	0.95	0.95
TTF	2	2.11	1.11	0.67	0.80	0.80
UD	3	3.00	1.49	0.51	0.75	0.74

Poleg zgornjih metrik zanesljivosti konstruktoev so (Fornell & Larcker 1981) predlagali način ugotavljanja diskriminacijske veljavnosti z uporabo AVE, in sicer ali vrednosti AVE presegajo kvadrirane korelacije med konstrukti ali ne.

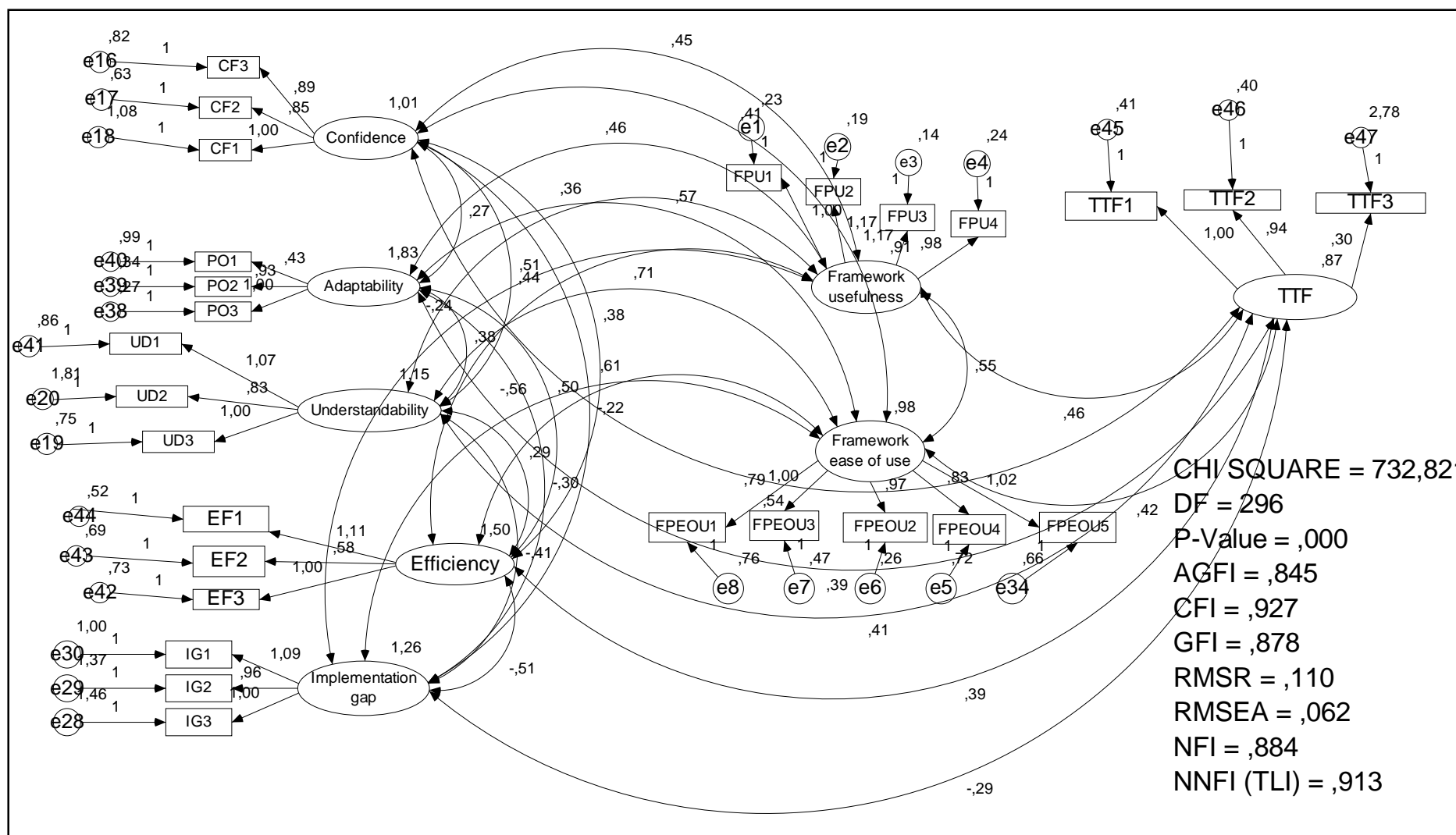
TABELA 31: DISKRIMINACIJSKA VELJAVNOST

	EF	PO	IG	TTF	UD	CF	FPEOU	FPU
EF	0.62							
PO	0.12	0.85						
IG	0.08	0.09	0.51					
TTF	0.08	0.14	0.04	0.80				
UD	0.17	0.05	0.06	0.12	0.75			
CF	0.33	0.09	0.04	0.04	0.22	0.50		
FPEOU	0.20	0.24	0.15	0.25	0.47	0.18	0.68	
FPU	0.27	0.14	0.12	0.05	0.12	0.22	0.33	0.95

Vse vrednosti AVE (diagonale v Tabela 31) so presegle kvadrirane korelacije med konstrukti (nediagonalne vrednosti v Tabela 31), kar kaže na dobro diskriminacijsko veljavnost. Rezultati merljivega modela vključno z metrikami prileganja so prikazani na naslednji strani¹⁰⁸ (Slika 63).

Na osnovi končnega merljivega modela sta bila ovrednotena začetni in končni strukturni model.

¹⁰⁸ Slike so prenesene neposredno iz orodja SPSS AMOS.



SLIKA 64: MERLJIV MODEL, IZPELJAN IZ TEORETIČNEGA MODELA 1

REZULTATI STRUKTURNIH MODELOV

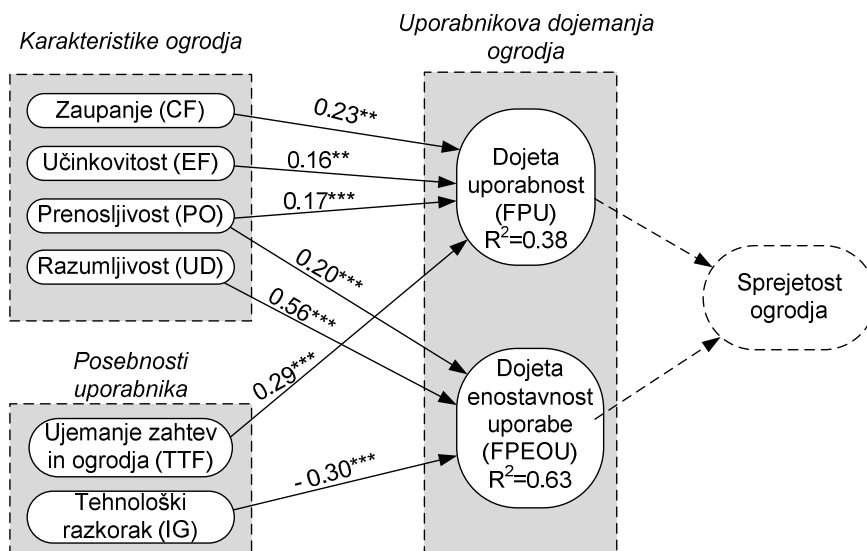
Za potrebe RV_1 sta bila analizirana dva strukturna modela: začetni in končni strukturni model. Začetni strukturni model je temeljil na hipotezah, ki so bile postavljene v okviru RV_1 . Končni strukturni model smo oblikovali na osnovi rezultatov začetnega strukturnega modela in napotkov za izboljšave modela, ki jih je podalo orodje AMOS.

Strukturna modela sta bila najprej analizirana z uporabo enakih metrik prileganja (*goodness of fit*), kot so se uporabile pri merljivih modelih. Rezultati metrik prileganj in priporočljive mejne vrednosti so podani v spodnji tabeli (Tabela 32).

TABELA 32: METRIKE PRILEGANJA ZAČETNEGA IN KONČNEGA STRUKTURNEGA MODELA

Metrika	Priporočljiva vrednost	Začetni strukturni model	Končni strukturni model
GFI	≥ 0.90	0.88	0.90
AGFI	≥ 0.80	0.84	0.87
NFI	≥ 0.90	0.88	0.91
TLI (NNFI)	≥ 0.90	0.92	0.95
CFI	≥ 0.90	0.94	0.96
RMSR	≤ 0.10	0.14	0.12
RMSEA	≤ 0.08	0.04	0.03
X² (df,p)	-	1277.34 (624, $p < 0.001$)	971.69 (567, $p < 0.001$)
X²/df	≤ 3	2.04	1.71
R²_{FPEOU}	-	0.63	0.51
R²_{FPU}	-	0.38	0.37
R²_{CF}	-	-	0.48

Na naslednji sliki (Slika 65) so prikazane uteži faktorjev (*factor loadings*), signifikantnosti kavzalnih povezav in stopnja pojasnjene variance v odvisnih spremenljivkah (R^2).



SLIKA 65: REZULTATI ZAČETNEGA STRUKTURNEGA MODELA

Iz rezultatov analize teoretičnega modela 1 (Slika 65) je razvidno, da so bile vse kavzalne povezave signifikantne. Pri tem je vpliv neodvisnih faktorjev na odvisne faktorje segal od 0.16 (UD→FPU) do 0.55 (UD→FPEOU). Odstopanje od zastavljene hipoteze smo našli pri H3 (PO→FPEOU), kjer smo predpostavljali negativen vpliv, medtem ko so rezultati podali pozitiven vpliv. Kot smo predpostavljali, je bil negativen vpliv ugotovljen pri (IG→FPEOU). Stopnja pojasnjene variance endogenih faktorjev se je nahajala med 0.38 (FPU) in 0.63 (FPEOU).

V nadaljevanju so podani rezultati strukturnega modela v tabelarni obliki (Tabela 33). Podani so rezultati za vse uporabnike in posebej za tiste, ki uporabljajo vrednoteno ogrodje prostovoljno in obvezno.

TABELA 33: VPLIV TIPA UPORABE OGRODJA NA ZAČETNI STRUKTURNI MODEL

	Prostovoljna uporaba	Obvezujoča uporaba	Vsi uporabniki
CF → FPU	0.33***	-0.04 ^{NS}	0.23**
EF → FPU	0.12 ^{NS}	0.19 ^{NS}	0.16**
PO → FPU	0.16 ^{NS}	0.15 ^{NS}	0.17***
TTF → FPU	0.21**	0.47*	0.29***
PO → FPEOU (-)	0.25***	0.09 ^{NS}	0.20***
UD → FPEOU	0.47***	0.77***	0.56***
IG → FPEOU (-)	-0.32***	-0.17 ^{NS}	-0.30***
R^2_{FPU}	0.37	0.40	0.38
R^2_{FPEOU}	0.54	0.79	0.63
*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, ^{NS} $p \geq 0.05$ (ni signifikantno)			

Iz zgornje tabele (Tabela 33) so razvidne višje vrednosti vpliva faktorjev, signifikantnosti in stopnje pojasnjene variance za prostovoljne uporabnike v primerjavi z uporabniki, katerih uporaba ogrodja je obvezna.

Končni strukturni model (Slika 66) smo oblikovali na osnovi rezultatov teoretičnega modela 1 in napotkov za izboljšave modela, ki jih je podalo orodje AMOS. Modifikacij modela smo se lotili postopno, pri tem smo testirali povezave z visokim indeksom modifikacij (*modification index-M.I.*).

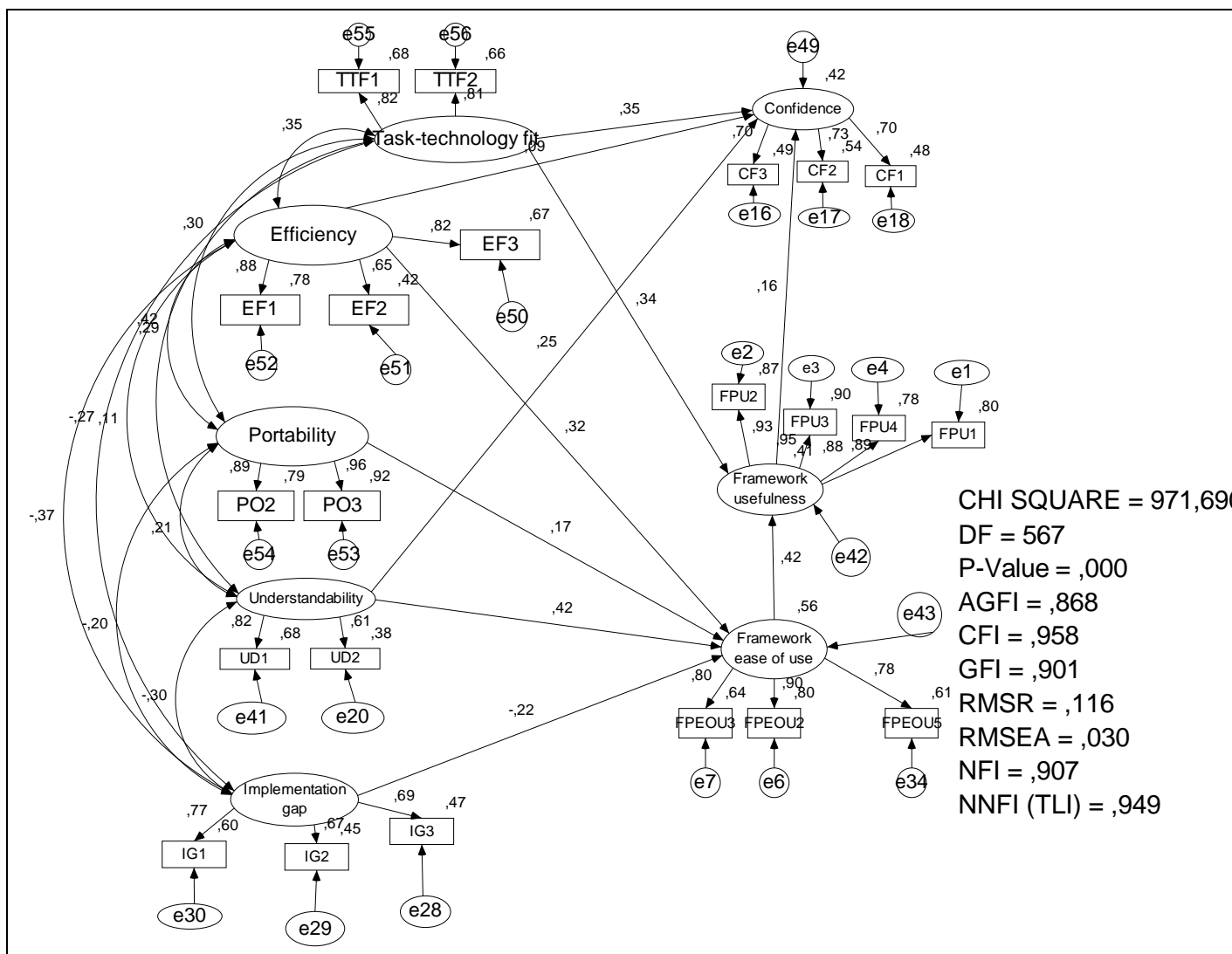
Dobljen končni model je demonstriral boljše prileganje empiričnih podatkov, kar je razvidno iz metrik prileganja (Slika 66 in Tabela 32). Zunaj priporočljivih vrednosti je tako ostala le metrika RMSR.

Poglavitna razlika med začetnim in končnim strukturnim modelom je v modeliranju faktorja CF, ki smo ga v končnem strukturnem modelu modelirali kot odvisno spremenljivko. V končnem modelu smo prav tako izvzeli indikator UD3, saj je indeks modifikacij pokazal previsoko stopnjo korelacije s faktorjem FPEOU. Stopnja pojasnjene variance končnega strukturnega modela se nahaja med 0.41 (FPU) in 0.56 (FPEOU).

TABELA 34 VPLIV TIPA UPORABE OGRODJA NA KONČNI STRUKTURNI MODEL

	Prostovoljna uporaba	Obvezujoča uporaba	Vsi uporabniki
TTF → FPU	0.33***	0.32*	0.34***
IG → FPEOU	-0.24***	-0.02 ^{NS}	-0.22***
UD → FPEOU	0.36***	0.71**	0.42***
PO → FPEOU	0.21***	0.09 ^{NS}	0.17***
UD → CF	0.26**	0.38*	0.25***
EF → CF	0.22**	-0.22 ^{NS}	0.09 ^{NS}
EF → FPEOU	0.31***	0.39***	0.32***
TTF → CF	0.25**	0.51**	0.35***
PU → CF	0.26***	-0.15 ^{NS}	0.16*
R²_{CF}	0.48	0.57	0.42
R²_{FPU}	0.37	0.48	0.41
R²_{FPEOU}	0.51	0.73	0.56
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)			

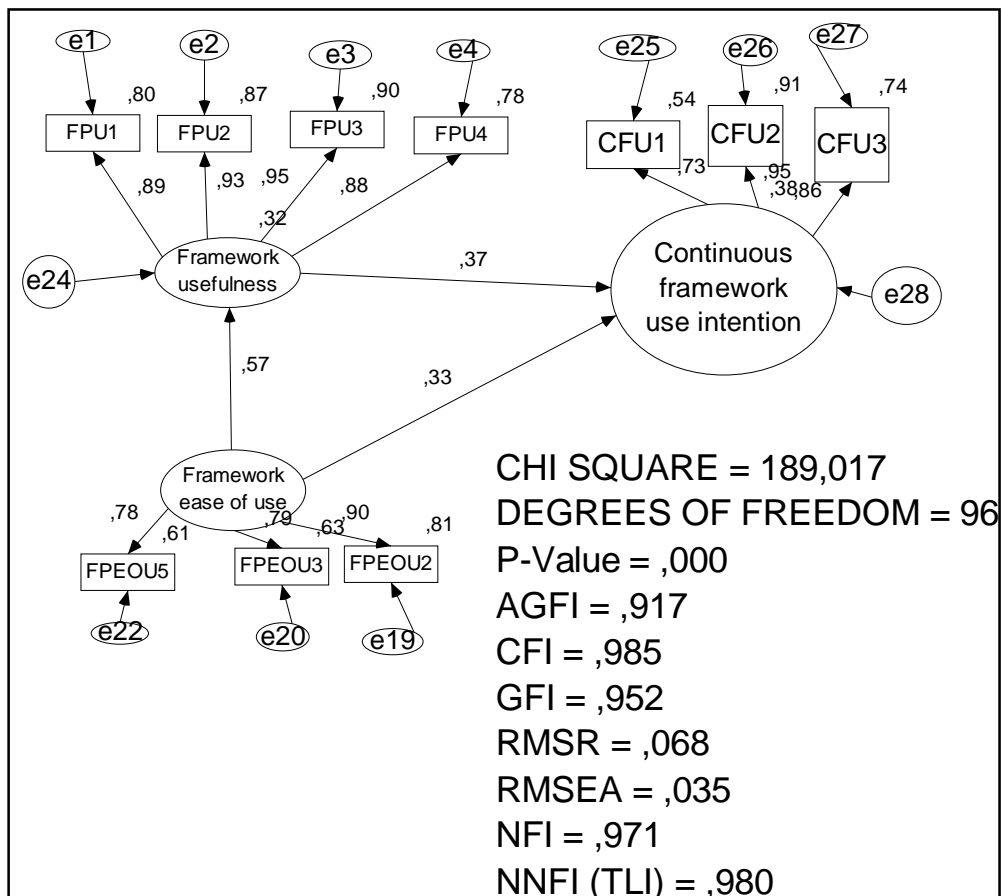
Iz zgornje tabele (Tabela 34) je razvidno, da je končni strukturni model primeren za prostovoljno uporabo ogrodja, medtem ko so se štiri od devetih kavzalnih povezav v primeru obvezne uporabe izkazale za nesignifikantne. Dve povezavi (PU→CF in EF→CF) sta imeli različno predznačene vplive v odvisnosti od tipa uporabnika.



SLIKA 66: REZULTATI KONČNEGA STRUKTURNEGA MODELA, IZPELJANEGA IZ TEORETIČNEGA MODELA 1

3.7.10.3 STATISTIČNA ANALIZA TEORETIČNIH MODELOV RV₂

V okviru statistične analize teoretičnih modelov RV₂ smo analizirali modele, ki smo jih oblikovali v okviru drugega raziskovalnega vprašanja. Analize merljivih modelov ni bilo treba izvesti, ker faktorji v teoretičnih modelih RV₂ predstavljajo podmnožico faktorjev preostalih teoretičnih modelov raziskave, pri katerih so se merljivi modeli izkazali za ustrezne.

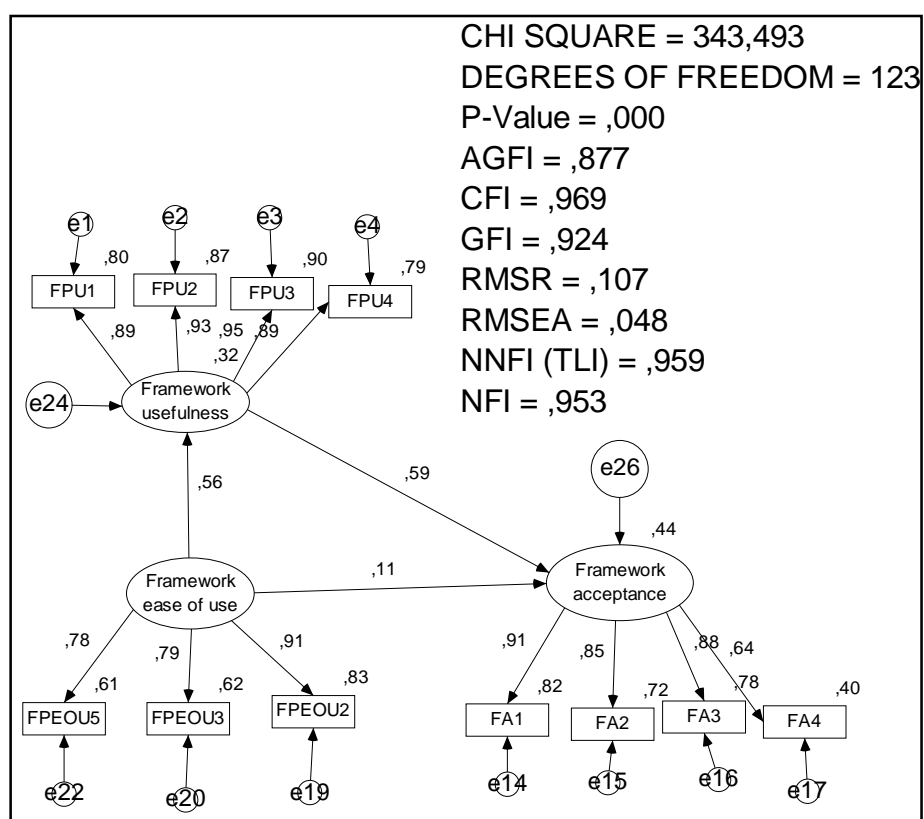


SLIKA 67: REZULTATI ANALIZE TEORETIČNEGA MODELA 2A (VSI UPORABNIKI)

Analiza teoretičnega modela 2A je demonstrirala skladnost modela z empiričnimi podatki, saj so bile vse metrike prileganja znotraj mejnih vrednosti (Tabela 37). Vse kavzalne povezave v modelu so se izkazale za signifikantne ($p < 0.001$) z vplivom (*factor loading*), ki je podoben predhodnim raziskavam. Podobnost s predhodno raziskavo se je potrdila tudi v metrikah prileganja modela empiričnim podatkom (Tabela 35).

TABELA 35: PRIMERJAVA REZULTATOV MODELA 2A S SORODNO RAZISKAVO

	Rezultati modela 2A	Rezultati sorodne raziskave 109
Kavzalne povezave		
(F)PEOU → (F)PU	0.57***	0.64***
(F)PEOU → C(F)UI	0.33***	0.49***
(F)PU → C(F)UI	0.37***	0.40***
Delež pojasnjene variance v endogenih faktorjih (R²)		
R ² _{(F)PU}	0.32	0.35
R ² _{C(F)UI}	0.38	0.63
Metrike prilaganja		
GFI	0.95	0.99
AGFI	0.92	0.98
NFI	0.97	0.98
NNFI	0.98	0.97
CFI	0.99	0.98
RMSR	0.07	0.06
RMSEA	0.04	0.06
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)		



SLIKA 68: REZULTATI ANALIZE TEORETIČNEGA MODELA 2B (VSI UPORABNIKI)

¹⁰⁹ (Hong, Thong, & Tam 2006).

Analiza teoretičnega modela 2B (Slika 68) je prav tako potrdila dobro ujemanje modela z empiričnimi podatki, z eno metriko prileganja malo nad priporočljivo vrednostjo (Tabela 37). Signifikantnost vseh kavzalnih povezav je bila visoka ($p < 0.001$), z vplivom (*factor loading*), ki se ujema z rezultati predhodnih raziskav poenostavljenega modela TAM (King & He 2006).

TABELA 36: VPLIV TIPA UPORABE OGRODJA NA TEORETIČNA MODELA 2A IN 2B

	Prostovoljna uporaba		Obvezujoča uporaba		Vsi uporabniki	
	Model 2A	Model 2B	Model 2A	Model 2B	Model 2A	Model 2B
FPEOU → FPU	0.52***	0.52***	0.64***	0.64***	0.57***	0.56***
FPEOU → FA	-	0.18**	-	-0.08 ^{NS}	-	0.11*
FPU → FA	-	0.62***	-	0.67***	-	0.59***
FPEOU → CFUI	0.35***	-	0.24*	-	0.33***	
FPU → CFUI	0.39***	-	0.33 ^{NS}	-	0.37***	
R²_{FPU}	0.27	0.27	0.41	0.42	0.32	0.32
R²_{FA}	-	0.53	-	0.38	-	0.44
R²_{CFUI}	0.42	-	0.27	-	0.38	-

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, ^{NS} $p \geq 0.05$ (ni signifikantno)

TABELA 37: PRILEGANJE STRUKTURNEGA MODELA 2A IN 2B EMPIRIČNIM PODATKOM

Metrika	Priporočljiva vrednost	Model 2A	Model 2B
GFI	≥ 0.90	0.99	0.92
AGFI	≥ 0.80	0.92	0.88
NFI	≥ 0.90	0.97	0.95
TLI (NNFI)	≥ 0.90	0.98	0.96
CFI	≥ 0.90	0.99	0.97
RMSR	≤ 0.10	0.07	0.11
RMSEA	≤ 0.08	0.04	0.05
X ²	-	189.02	343.49
(df,p)		(96, $p < 0.001$)	(123, $p < 0.001$)
X ² /df	≤ 3	1.97	2.79

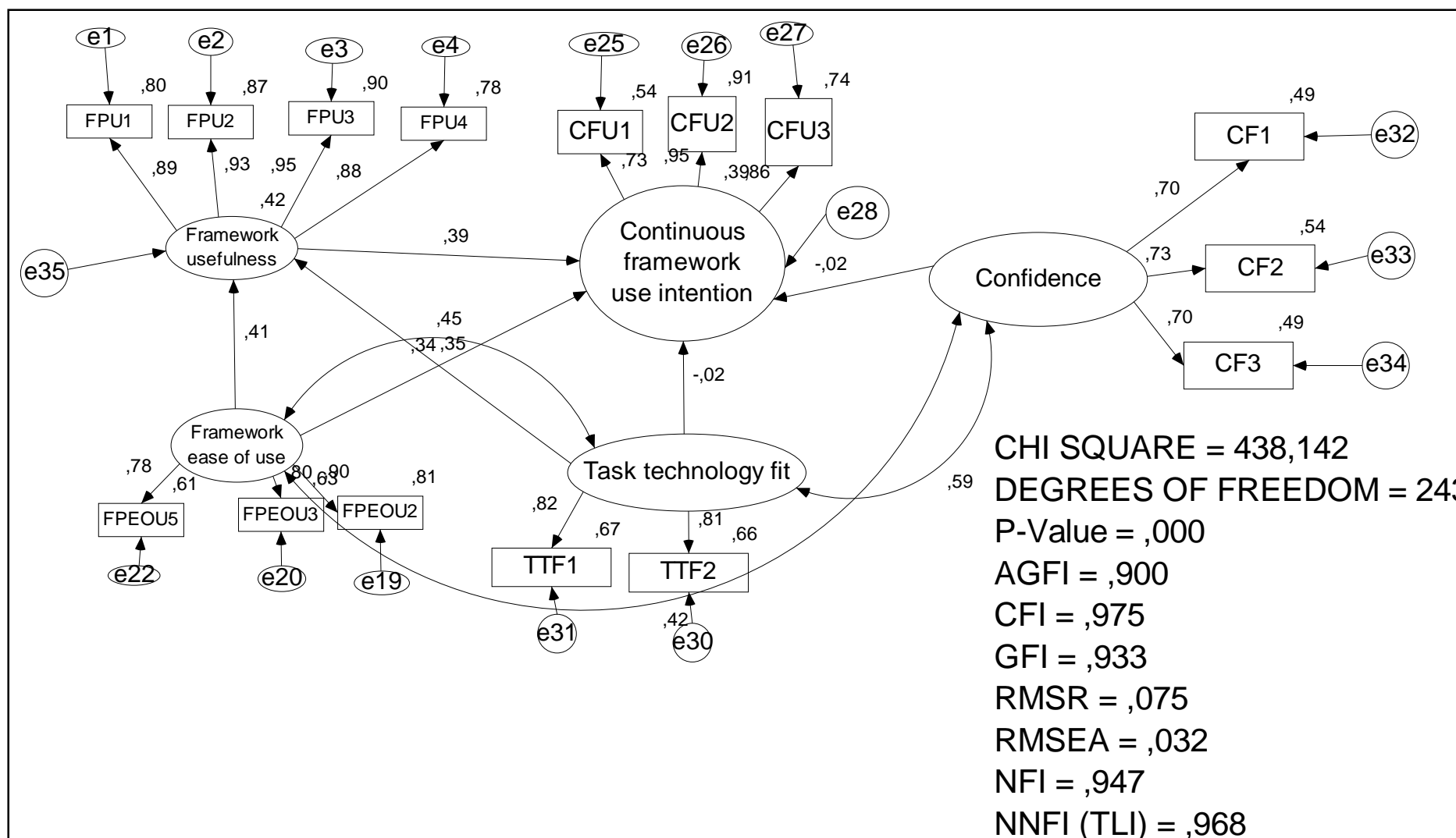
Modela 2A in 2B smo nato razširili v modela 2C in 2D z vključitvijo faktorja iz druge teorije in z vključitvijo kontekstnega faktorja.

Faktor, ki smo ga prenesli iz druge teorije, je ujemanje zahtev in ogrodja (TTF). Faktor je sestavni del modela TTF (*Task-technology fit model*)(Goodhue & Thompson 1995). Model TTF predstavlja drugo skupino modelov sprejetosti (glej poglavje 2.4.2). Razloga za vključitev TTF sta naslednja:

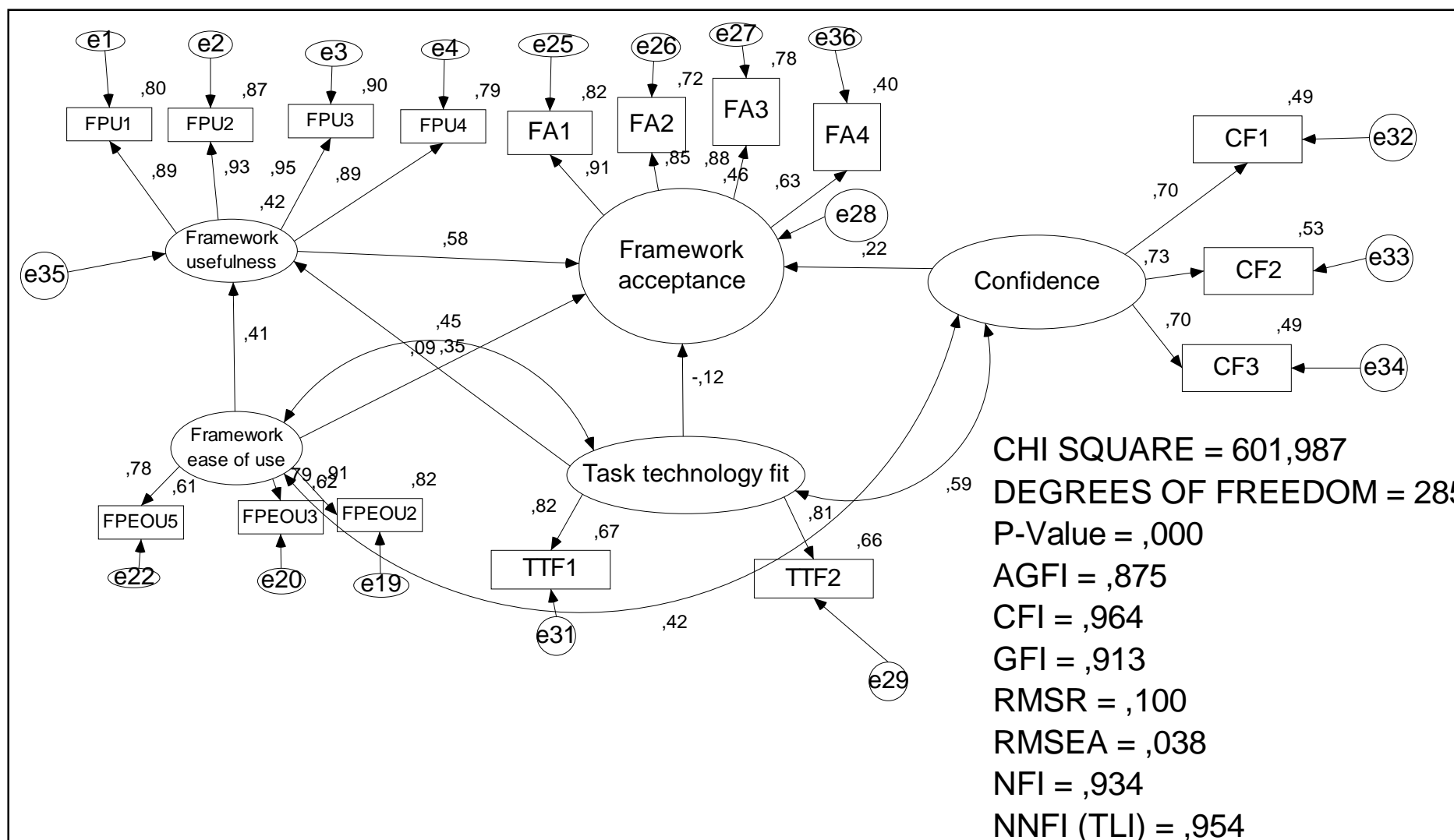
- Predhodne raziskave so potrdile signifikanten vpliv TTF na namero uporabe (*behavioral intention to use*)(Klopping & McKinney 2004).
- Rezultati raziskave¹¹⁰ so potrdili signifikanten in velik vpliv TTF na uporabnost ogrodja (FPU) (Slika 66).

Za kontekstni faktor smo izbrali »zaupanje« (CF), ki smo ga v okviru končnega strukturnega modela 1 definirali kot endogeni faktor. Rezultati so pokazali veljavnost takšnega modela (Slika 66). V nadaljevanju so podani rezultati modelov 2C in 2D (Slika 69 in Slika 70).

¹¹⁰ Glej rezultate končnega strukturnega modela, ki smo ga oblikovali na osnovi teoretičnega modela 1.



SLIKA 69: REZULTATI ANALIZE TEORETIČNEGA MODELA 2C (VSI UPORABNIKI)



SLIKA 70: REZULTATI ANALIZE TEORETIČNEGA MODELA 2D (VSI UPORABNIKI)

Iz statistične analize modelov 2C in 2D (Slika 69 in Slika 70) je razvidno, da imata oba vključena faktorja (TTF in CF) v obeh modelih nesignifikanten in zanemarljiv vpliv na odvisni spremenljivki CFUI in FA.

V spodnjih tabelah so podane ključne vrednosti modelov, ki smo jih izpeljali iz poenostavljenega modela TAM in modelov, ki predstavljajo njihovo razširitev.

TABELA 38: PRIMERJAVA REZULTATOV MODELA 2A IN MODELA 2C

	Model 2A	Model 2C
FPEOU → FPU	0.57***	0.63***
FPEOU → CFUI	0.33***	0.37***
FPU → CFUI	0.37***	0.36***
TTF → CFUI	-	-0.02 ^{NS}
CF → CFUI	-	-0.02 ^{NS}
TTF → FPU	-	0.35***
R²_{CFUI}	0.42	0.39
R²_{FPU}	0.27	0.35
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)		

TABELA 39: PRIMERJAVA METRIK PRILEGANJA ZA MODEL 2A IN MODEL 2C

Metrika	Priporočljiva vrednost	Model 2A	Model 2C
GFI	≥ 0.90	0.99	0.93
AGFI	≥ 0.80	0.92	0.90
NFI	≥ 0.90	0.97	0.95
TLI (NNFI)	≥ 0.90	0.98	0.97
CFI	≥ 0.90	0.99	0.98
RMSR	≤ 0.10	0.07	0.08
RMSEA	≤ 0.08	0.04	0.03
X²	-	189.02	438.14
(df,p)		(96, p < 0.001)	(243, p < 0.001)
X²/df	≤ 3	1.97	1.80

Iz zgornjih tabel je razvidno, da konstrukta, ki smo ju vključili v osnovni model (TTF in CF), nista prispevala k izboljšanju pojasnjene variance odvisnega faktorja CFUI. Prav tako imata dodana faktorja TTF in CF nesignifikanten vpliv na CFUI. Iz tega je možno sklepati, da je model TAM za trajno uporabo primernejši od njegove razširitve. FPEOU in FPU ostajata edini poznani determinanti namere o trajni uporabi ogrodij.

V naslednjih tabelah je podana primerjava modelov 2B in 2D (Tabela 40 in Tabela 41).

TABELA 40: PRIMERJAVA REZULTATOV MODELA 2B IN MODELA 2D

	Model 2B	Model 2D
FPEOU → FPU	0.56***	0.41***
FPEOU → FA	0.11*	0.09 ^{NS}
FPU → FA	0.59***	0.58***
TTF → FA	-	-0.12 ^{NS}
CF → FA	-	0.22**
TTF → FPU	-	0.36***
R²_{FPU}	0.32	0.42
R²_{FA}	0.44	0.46
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)		

TABELA 41: PRIMERJAVA METRIK PRILEGANJA ZA MODEL 2B IN MODEL 2D

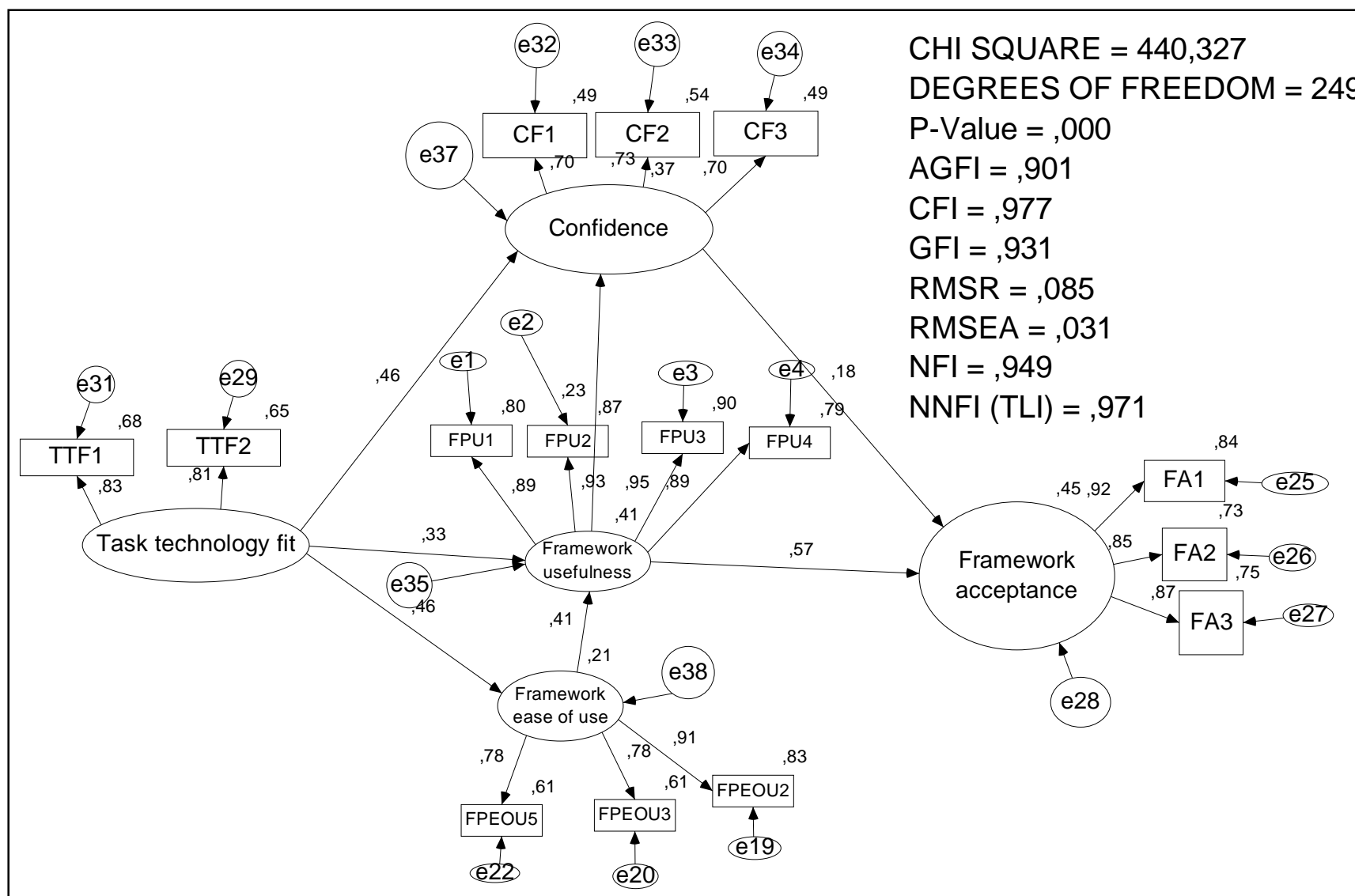
Metrika	Priporočljiva vrednost	Model 2B	Model 2D
GFI	≥ 0.90	0.92	0.91
AGFI	≥ 0.80	0.88	0.88
NFI	≥ 0.90	0.95	0.93
TLI (NNFI)	≥ 0.90	0.96	0.95
CFI	≥ 0.90	0.97	0.96
RMSR	≤ 0.10	0.11	0.10
RMSEA	≤ 0.08	0.05	0.04
X²	-	343.49	601.98
(df,p)		(123, p < 0.001)	(285, p < 0.001)
X²/df	≤ 3	2.79	2.11

Na osnovi primerjave modelov, ki določata sprejetost ogrodij (modela 2B in 2D), smo ugotovili, da ima model 2D nekaj prednosti pred modelom 2B:

- Model 2D ima boljše rezultate metrik prileganja kot model 2B, saj ima vse metrike znotraj priporočljivih vrednosti.
- Stopnja pojasnjene variance sprejetosti je v modelu 2D višja od modela 2B.
- Dodan konstrukt zaupanje (CF) je demonstriral signifikanten vpliv na sprejetost (FA).

Iz navedenega je možno sklepati, da je model TAM, ki smo ga razširili s konstruktom zaupanja, primernejši od njegove osnovne različice (modela 2B). Na osnovi naših ugotovitev sta poglavitni determinanti sprejetosti ogrodij FPU in CF.

Na osnovi rezultatov teoretičnih modelov 1, 2A, 2B, 2C in 2D smo v okviru RV_2 izvedli še eksplorativen del raziskave, v okviru katerega smo oblikovali končni model sprejetosti ogrodij ali krajše FAM (*Framework Acceptance Model*). Rezultat je podan na naslednji sliki (Slika 71).



SLIKA 71: MODEL SPREJETOSTI OGRODIJ - REZULTATI STRUKTURNEGA MODELA

Na osnovi statistične analize smo ugotovili dobro prileganje modela FAM z empiričnimi podatki, saj se je izkazalo, da so bile vse metrike ujemanja znotraj priporočljivih vrednosti (glej Slika 71). V spodnji tabeli so podani poglavitni rezultati modela FAM (Tabela 42).

TABELA 42: POGLAVITNI REZULTATI MODELA SPREJETOSTI OGRODIJ

Model sprejetosti ogrodiij			
	Vsi uporabniki	Obvezna uporaba	Prostovoljna uporaba
TTF → CF	0.46***	0.70***	0.39***
TTF → FPEOU	0.46***	0.58***	0.38***
TTF → FPU	0.33***	0.31*	0.32***
FPEOU → FPU	0.41***	0.46***	0.40***
FPU → CF	0.23***	-0.09 ^{NS}	0.35***
FPU → FA	0.57***	0.52***	0.65***
CF → FA	0.18**	0.26*	0.12 ^{NS}
R²_{FPU}	0.41	0.48	0.36
R²_{FA}	0.45	0.43	0.52
R²_{FPEOU}	0.21	0.34	0.15
R²_{CF}	0.37	0.43	0.40
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)			

FAM specializira in razširja model TAM na naslednje načine:

- Vključuje konstrukt TTF, ki je povzet iz druge teorije (Goodhue & Thompson 1995). Pokazalo se je, da je TTF prvi pogoj za CF, FPEOU in FPU. Takšna ugotovitev ustreza predhodnim študijam (Dishaw & Strong 1999; Klopping & McKinney 2004).
- Vključuje konstrukt CF, ki predstavlja kontekstni faktor, specifičen za ogrodja. Pokazalo se je, da ima v kontekstu ogrodiij CF večji vpliv na sprejetost kot FPEOU. FPEOU ima le posreden vpliv na sprejetost ogrodiij, in sicer preko FPU.

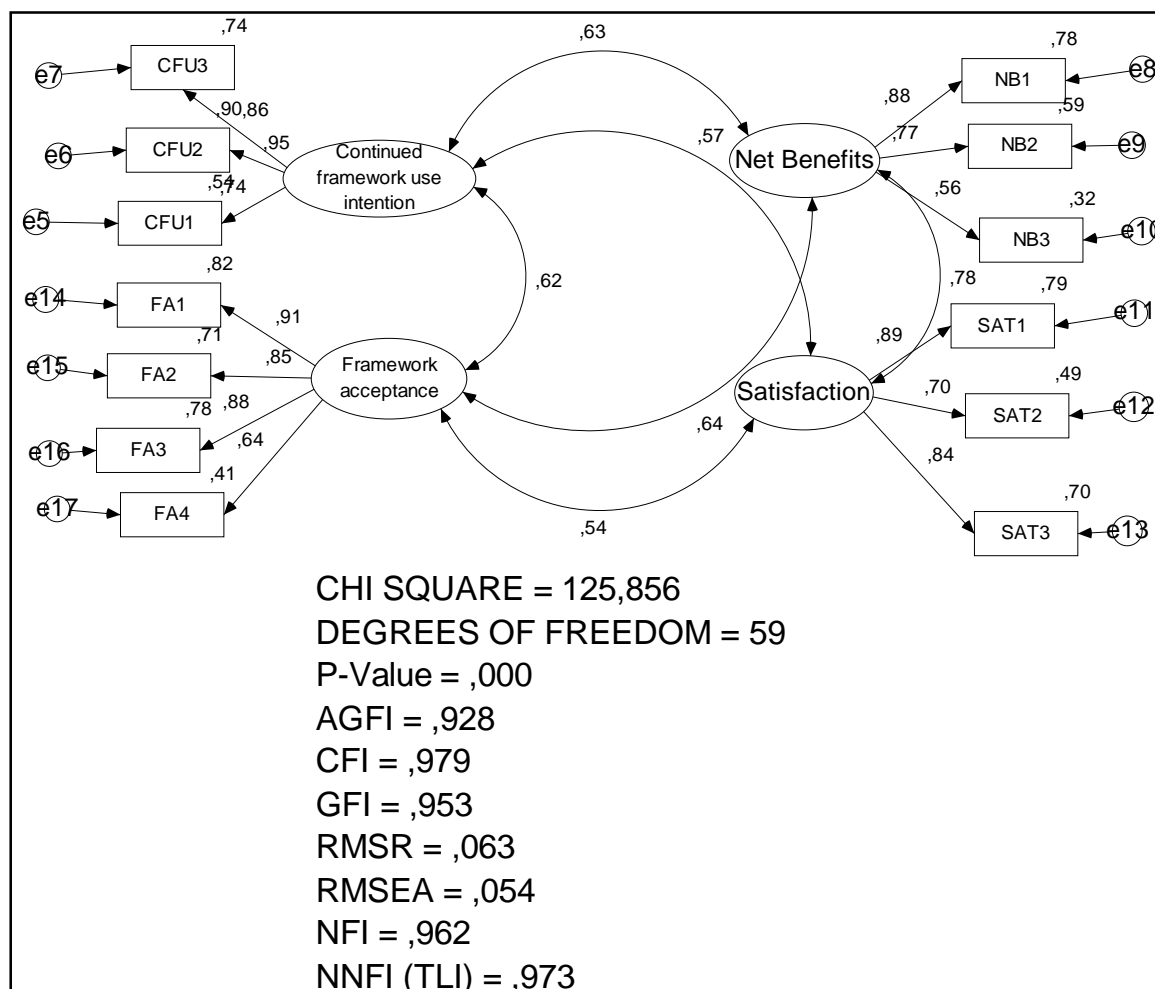
Pokazalo se je, da je v kontekstu ogrodiij model FAM popolnejši in natančnejši od modela TAM iz naslednjih razlogov:

- Vse metrike prileganja so v modelu FAM boljše v primerjavi z modelom TAM (glej Slika 68 in Slika 71).
- Delež pojasnjene variance (R²) odvisne spremenljivke je v modelu FAM višji kot v modelu TAM.
- Model FAM je zaradi večjega števila faktorjev natančnejši od modela TAM.

3.7.10.4 STATISTIČNA ANALIZA TEORETIČNEGA MODELA 3

REZULTATI MERLJIVEGA MODELA

Analiza rezultatov merljivega modela, ki se je izpeljal iz teoretičnega modela številka 3, je pokazala, da so bile vse metrike prilaganja znotraj priporočljivih vrednosti (Slika 72).



SLIKA 72: MERLJIV MODEL TEORETIČNEGA MODELA 3

V naslednji tabeli (Tabela 43) je podana opisna statistika indikatorjev merljivega modela. Za indikatorje merljivega modela smo opravili še T-test za neodvisne skupine¹¹¹ z namenom ugotoviti, ali obstaja signifikantna razlika ($p < 0.05$) med zgodnjimi in poznimi¹¹² respondenti. Na osnovi izvedenega T-testa nismo našli statistične razlike med skupinama za nobenega od indikatorjev končnega merljivega modela¹¹³.

¹¹¹ Analize normalne porazdelitve podatkov (*skewness, kurtosis*) nismo potrebovali, ker je velikost vzorca presegala velikost $N > 100$.

¹¹² Respondenti ki so pričeli izpolnjevati anketo ob prejemu opomniku.

¹¹³ Rezultati T-testa so v prilogi.

TABELA 43: OPISNA STATISTIKA INDIKATORJEV TEORETIČNEGA MODELA 3

Indikator	N	Min.	Maks.	Povprečje	Standardna deviacija	<i>Factor loadings</i> ¹¹⁴	<i>Squared multiple correlations</i> ¹¹⁵
FA1	391	1	7	2.39	1.45	0.91	0.82
FA2	391	1	7	2.73	1.57	0.85	0.72
FA3	391	1	7	2.35	1.47	0.89	0.78
FA4	391	1	7	2.83	1.47	0.64	0.41
CFUI1	391	1	7	2.64	1.48	0.74	0.54
CFUI2	391	1	7	2.04	1.25	0.95	0.90
CFUI3	391	1	7	1.99	1.37	0.86	0.75
NB1	391	1	7	1.97	1.10	0.88	0.78
NB2	391	1	7	2.01	1.12	0.78	0.59
NB3	391	1	7	2.31	1.49	0.56	0.32
SAT1	391	1	7	1.85	0.96	0.89	0.79
SAT2	391	1	7	2.26	1.26	0.7	0.49
SAT3	391	1	7	2.25	1.06	0.84	0.70

Vrednosti AVE, za vse faktorje, so presegle mejno vrednost 0.5, ki so jo predlagali Fornell in Larcker (1981). Pomeni, da je notranja konsistentnost faktorjev dobra.

TABELA 44: ZANESLJIVOST KONSTRUKTOV

Faktor	Število indikatorjev	Povprečje	Standardna deviacija	AVE	Kompozitna zanesljivost (<i>composite reliability</i>)	Cronbach's alpha
FA	4	2.57	1.49	0.68	0.89	0.89
CFUI	3	2.22	1.37	0.73	0.89	0.88
NBF	3	2.10	1.24	0.56	0.79	0.76
SAT	3	2.12	1.09	0.66	0.85	0.84

Vse vrednosti AVE (diagonale v Tabela 45) so presegle kvadrirane korelacije med faktorji (nediagonalne vrednosti v Tabela 45), kar kaže na dobro diskriminacijsko veljavnost.

¹¹⁴ Vrednosti so povzete iz tabele »Standardized regression weights« v orodju AMOS.

¹¹⁵ Vrednosti so povzete iz tabele »Squared multiple correlations« v orodju AMOS.

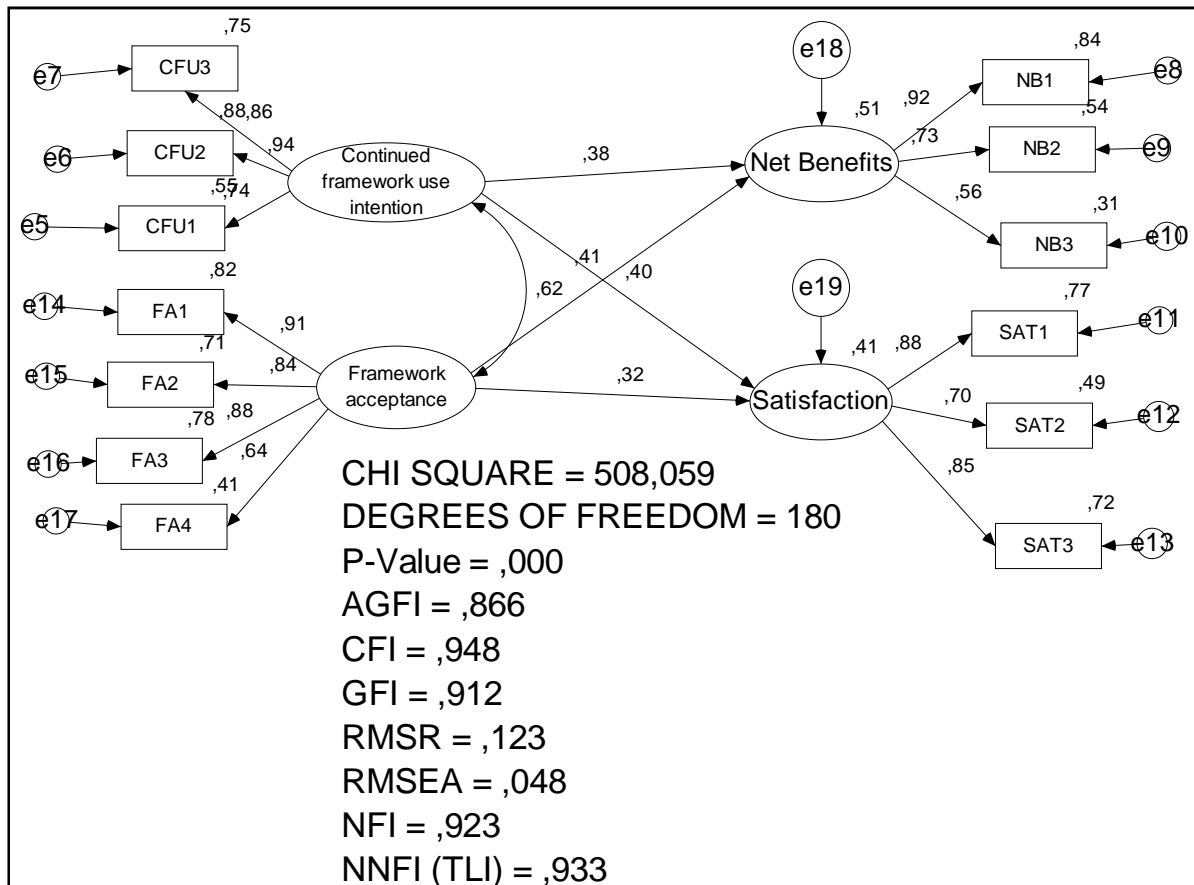
TABELA 45: DISKRIMINACIJSKA VELJAVNOST TEORETIČNEGA MODELA 3

	SAT	NBF	FA	CFUI
SAT	0.66			
NBF	0.61	0.56		
FA	0.29	0.41	0.68	
CFUI	0.33	0.40	0.38	0.73

Na osnovi navedenih ugotovitev smo sprejeli merljiv model, kot smo ga določili z vprašalnikom.

REZULTATI STRUKTURNEGA MODELA

V okviru raziskovalnega vprašanja RV₃ smo analizirali strukturni model, ki je temeljil na teoretičnem modelu 3. Poglavitni rezultati strukturnega modela, vključno z metrikami prileganja, so podani na spodnji sliki (Slika 73).



SLIKA 73: REZULTATI ANALIZE TEORETIČNEGA MODELA 3 (VSI UPORABNIKI)

Na osnovi metrik prileganja je možno sklepati, da predlagan strukturni model ustreza empiričnim podatkom. Iz rezultatov strukturnega modela lahko razberemo, da imajo vse povezave med eksogenimi in endogenimi faktorji signifikanten in razmeroma velik vpliv na odvisne faktorje, ki sega od 0,32 (FA → SAT) do 0,41 (CFUI → SAT). Vpliv vrste uporabe na strukturni model je podan v spodnji tabeli (Tabela 46).

TABELA 46: VPLIV TIPA UPORABE OGRODJA NA TEORETIČNI MODEL 3

	Prostovoljna uporaba	Obvezujoča uporaba	Vsi uporabniki
FA → SAT	0.40***	0.37**	0.32***
FA → NBF	0.52***	0.38**	0.41***
CFUI → SAT	0.32***	0.31*	0.40***
CFUI → NBF	0.30***	0.33*	0.38***
R²_{SAT}	0.42	0.39	0.41
R²_{NBF}	0.56	0.42	0.51
*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno)			

Iz tabele lahko razberemo, da so signifikantnosti povezav, uteži faktorjev (*factor loadings*) in stopnje pojasnjene variance v endogenih konstruktih večje v primeru prostovoljne uporabe ogrodja.

3.7.11 ANALIZA VELJAVNOSTI REZULTATOV

Kot vsaka empirična študija ima tudi naša anketa omejitve in tveganja, ki so povezani z veljavnostjo njenih rezultatov (Freimut, Punter, Biffel, & Ciolkowski 2002). Analizirali smo naslednje tipe veljavnosti (McBurney & White 2003): notranjo veljavnost (*internal validity*), zunanjo veljavnost (*external validity*), eksperimentalno veljavnost (*experimental validity*), veljavnost konstruktov (*construct validity*) in statistično veljavnost (*statistical validity*).

Notranja veljavnost ankete je povezana predvsem z nadzorovanjem spremenljivk, ki se najpogosteje izvede z izločitvijo, fiksiranjem in slučajnostjo (Freimut, Punter, Biffel, & Ciolkowski 2002). Z nezmožnostjo fiksiranja so v našem primeru povezana tveganja dveh različnih skupin respondentov (hitri in pozni respondenti), vendar smo z izvedenima T-testoma demonstrirali, da med omenjenima skupinama ni signifikantnih razlik (glej prilogo D). Z uporabo slučajnosti smo se lotili naključnega vrstnega reda zastavljenih vprašanj. S tem smo se izognili neželenim medsebojnim vplivom (*ceiling effects*) med vprašanji.

Zunanja veljavnost ankete je povezana z zmožnostjo posploševanja rezultatov vzorca na celotno populacijo. V našem primeru smo oblikovali vzorec iz vzorčnega okvira in ne iz celotne populacije. Vzorčni okvir smo določili iz praktičnih razlogov, kar lahko predstavlja tveganja za zunanjo veljavnost. V prid zunanji veljavnosti govorijo naslednja dejstva: (1) vzorčni okvir je bil razmeroma velik ($N > 11.000$), (2) vzorec smo iz vzorčnega okvira izbrali naključno in (3) respondenti so ocenjevali tudi ogrodja zunaj vzorčnega okvira (na primer: ogrodje .NET).

Eksperimentalna veljavnost je povezana z zmožnostjo ponovitve ankete (*survey replication*). Tveganja, ki so povezana z eksperimentalno veljavnostjo, smo poskušali zmanjšati z natančno definicijo profila udeležencev in njihovih izkušenj z vrednotenim ogrođjem. V vprašalnik smo vključili ponovljiva vprašanja (*repeated questions*), ki so omogočila kontrolo in izločanje neustreznih odgovorov. Veljavnost konstruktov smo zagotovili z operacionalizacijo, ki je temeljila na predhodnih študijah (Tabela 21).

Statistična veljavnost se ukvarja z vprašanjem, ali so rezultati raziskave pravilni ali so posledica nepravilnosti statistične tehnike. Z uporabljenimi statističnimi tehnikami smo demonstrirali majhne verjetnosti, da so relacije v modelih predmet naključij, nismo pa dokazali dejanskih kavzalnih povezav. Tveganje nepravilnih statističnih rezultatov smo zmanjšali z velikim statističnim vzorcem ($N=391$). Drugo tveganje statistične veljavnosti je povezano s strukturnimi modeli. Zavedamo se, da so rezultati odvisni od zasnove definiranih strukturnih modelov in da lahko obstajajo drugi primerljivi ali celo boljši strukturni modeli.

3.7.12 INTERPRETACIJA REZULTATOV

V spodnji tabeli (Tabela 47) je zajet povzetek rezultatov hipotez, ki smo jih podali v disertaciji. Podrobnejša interpretacija rezultatov ankete je predstavljena v naslednjih podpoglavjih.

TABELA 47: REZULTATI HIPOTEZ

Hipoteza	Povezava	Rezultat	Komentar
H1	UD → FPEOU	Potrjena	Pozitiven vpliv razumljivosti na enostavnost uporabe ogrodja se je izkazal za signifikantnega v vseh primerih.
H2	PO → FPU	Zavržena	Pozitiven vpliv prenosljivosti na doje to uporabnost ogrodja se je izkazal za nesignifikantnega v primeru prostovoljne in obvezne uporabe. Vpliv PO→FPU (β) je v vseh primerih znašal pod 0.20.
H3	PO → FPEOU(-)	Zavržena	Hipotezo smo zavrgli, ker se je izkazalo, da obstaja med PO in FPEOU majhen ($\beta < 0.25$) pozitiven vpliv (predpostavljali smo negativen vpliv).
H4	CF → FPU	Zavržena	Pozitiven in majhen ($\beta < 0.33$) vpliv CF→FPU se je izkazal le v primeru začetnega strukturnega modela.
H5	EF → FPU	Zavržena	Hipotezo smo zavrgli, ker sta se signifikatnost in majhen vpliv izkazala le v primeru začetnega strukturnega modela in vseh uporabnikov.
H6	TTF → FPU	Potrjena	Hipotezo smo sprejeli, ker se je izkazala za signifikantno v vseh primerih.
H7	IG → FPEOU (-)	Potrjena	Hipotezo smo sprejeli, ker se je izkazala za signifikantno v primerih vseh uporabnikov in prostovoljne uporabe ogrodja.
H8		Potrjena	Podrobnosti so podane v poglavju 3.7.12.2.
H9	FPU → FA	Potrjena	Hipoteze smo sprejeli, ker so se izkazale za signifikantne v vseh teoretičnih modelih in pri vseh vrstah uporabnikov.
H10	FPU → CFUI	Potrjena	
H11	FPEOU → FPU	Potrjena	
H12	FPEOU → CFUI	Potrjena	
H13	FPEOU → FA	Zavržena	Hipotezo smo zavrnili, ker je izkazovala signifikatnost $p < 0.05$ le v modelu 2B. Pri tem je bil vpliv FPEOU na FA majhen ($\beta < 0.20$).
H14	FA → NBF	Potrjena	Hipoteze smo sprejeli, ker so se izkazale za signifikantne v vseh primerih (obvezna uporaba, prostovoljna uporaba, vsi uporabniki).
H15	FA → SAT	Potrjena	
H16	CFUI → NBF	Potrjena	
H17	CFUI → SAT	Potrjena	

V naslednji tabeli (Tabela 48) je podan pregled analize hipotez glede na signifikantnost kavzalnih povezav v strukturnih modelih disertacije.

TABELA 48: PREGLED ANALIZE HIPOTEZ PO STRUTURNIH MODELIH

Strukturni model		1-Začetni	1-Končni	Model 2A	Model 2B	Model 2C	Model 2D	FAM	Model 3
Prostovoljna	H1	***	***						
Obvezna		***	**						
Vsi uporabniki		***	***						
Prostovoljna	H2	ns	***						
Obvezna		ns	ns						
Vsi uporabniki		***	***						
Prostovoljna	H3	***	***						
Obvezna	(-)	ns	ns						
Vsi uporabniki		***	***						
Prostovoljna	H4	***							
Obvezna		ns							
Vsi uporabniki		**							
Prostovoljna	H5	ns							
Obvezna		ns							
Vsi uporabniki		**							
Prostovoljna	H6	**	***					***	
Obvezna		*	*					*	
Vsi uporabniki		***	***					***	
Prostovoljna	H7	_* **	_ * **						
Obvezna	(-)	- ns	- ns						
Vsi uporabniki		_ * **	_ * **						
Prostovoljna	H9				***			***	
Obvezna					***			***	
Vsi uporabniki					***		***	***	
Prostovoljna	H10			***					
Obvezna				ns					
Vsi uporabniki				***		***			
Prostovoljna	H11			***	***			***	
Obvezna				***	***			***	
Vsi uporabniki				***	***	***	***	***	
Prostovoljna	H12			***					
Obvezna				*					
Vsi uporabniki				***		***			
Prostovoljna	H13				**				

Strukturni model		1-Začetni	1-Končni	Model 2A	Model 2B	Model 2C	Model 2D	FAM	Model 3
Obvezna					ns				
Vsi uporabniki					*		ns		
Prostovoljna	H14								***
Obvezna									**
Vsi uporabniki									***
Prostovoljna	H15								***
Obvezna									**
Vsi uporabniki									***
Prostovoljna	H16								***
Obvezna									*
Vsi uporabniki									***
Prostovoljna	H17								***
Obvezna									*
Vsi uporabniki									***

*** p < 0.001, ** p < 0.01, * p < 0.05, ^{NS} p ≥ 0.05 (ni signifikantno), - negativni vpliv

3.7.12.1 INTERPRETACIJA REZULTATOV RV₁

V okviru raziskovalnega vprašanja RV₁ smo analizirali vpliv dveh skupin faktorjev na uporabnikova dojetanja ogrodiv: karakteristike ogrodja in posebnosti uporabnika. Na osnovi rezultatov ankete lahko podamo naslednje ugotovitve.

- Za obe skupini faktorjev (karakteristike ogrodja in posebnosti uporabnika) se je izkazalo, da vplivata na uporabnikova dojetanja ogrodiv, ki smo jih modelirali s faktorjema dojete enostavnosti uporabe ogrodja in dojete uporabnosti ogrodja. Rezultati so skladni z rezultati predhodnih raziskav, ki so bile izvedene v drugačnem kontekstu (Hong, Thong, Wong, & Tam 2001).
- V okviru karakteristik ogrodja smo analizirali vpliv štirih faktorjev, ki smo jih definirali na osnovi predhodno izvedene terenske raziskave, modela kakovosti programske opreme in modela ponovne uporabnosti programske opreme. Za vse faktorje se je v okviru zastavljenega teoretičnega modela potrdil pozitiven in signifikanten vpliv na uporabnikova dojetanja ogrodiv. Iz tega lahko sklepamo, da so karakteristike ogrodiv sestavljene iz faktorjev kakovosti programske opreme in faktorjev ponovne uporabnosti programske opreme.
- Za faktor »prenosljivost« se je izkazalo, da ima pozitiven vpliv na dojeto enostavnost uporabe ogrodja, čeprav smo predpostavljali nasprotno. Morebiten razlog je v

razvijalčevem dojetanju ogrodja, da bo z dobro prenosljivostjo zmanjšal bodoči napor pri prilagajanju ogrodja, čeprav je praviloma ogrodje zaradi boljše prenosljivosti tudi kompleksnejše.

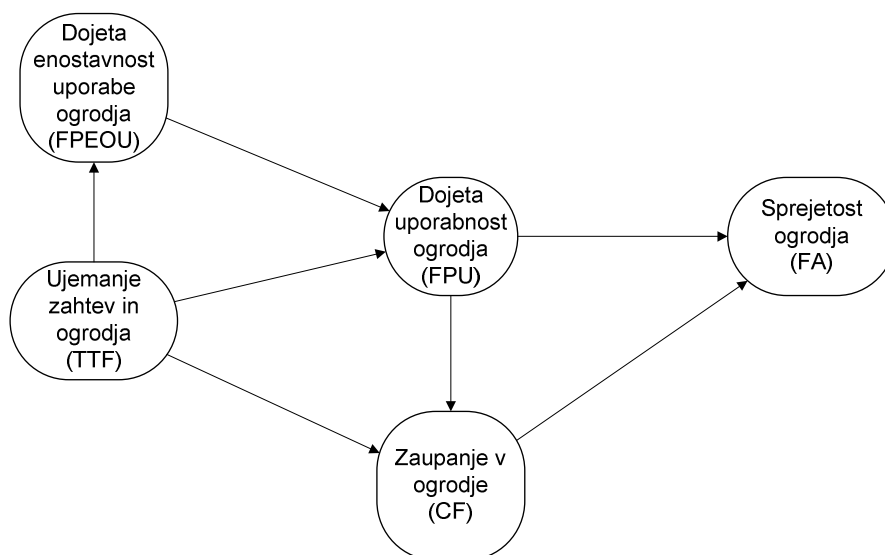
- Za večino kavzalnih povezav v končnem strukturnem modelu se je izkazalo, da vplivajo predvsem na dojeto enostavnost uporabe ogrodja. Edini faktor v končnem strukturnem modelu, ki vpliva na dojeto uporabnost ogrodja, je ujemanje zahtev z ogrođjem (TTF). Morebiten razlog je v kompleksnosti ogrođij in v to usmerjenih uporabnikovih dojetanjih ogrođij. Cilj uporabnikov je torej med ustreznimi ogrođji izbrati takšno, ki je najbolj enostavno za uporabo.
- Za zaupanje, ki je sestavni del modela ponovne uporabnosti, se je izkazalo, da ga je primerneje modelirati kot endogeno spremenljivko, na katero vplivajo: TTF, učinkovitost in razumljivost. Iz tega je možno sklepati, da ima v kontekstu ogrođij zaupanje podoben pomen kot dojeta enostavnost uporabe in dojeta uporabnost.
- Za tehnološki razkorak se je potrdila naša hipoteza, da ima negativen in signifikanten vpliv na dojeto enostavnost uporabe. Iz navedenega lahko sklepamo, da uporabnik ne bo uporabil ogrodja, ki bo preveč odstopalo od utečenih in preizkušenih tehnologij.

3.7.12.2 INTERPRETACIJA REZULTATOV RV₂

V okviru raziskovalnega vprašanja RV₂ smo primerjali štiri modele sprejetosti, da bi ugotovili, kateri je najprimernejši v kontekstu ogrođij. Oblikovali smo dva modela, ki sta se oblikovala na osnovi predhodnih raziskav, in dva modela, ki sta predstavljala nadgradnjo predhodnikov. Na osnovi rezultatov ankete lahko podamo naslednje ugotovitve.

- Za poenostavljeni model TAM se je izkazalo, da je primeren tudi v kontekstu ogrođij. S to ugotovitvijo smo razširili področje uporabe modela TAM na doslej neraziskano področje ogrođij. Dosedanje študije TAM so temeljile izključno na programski opremi, ki je namenjena končnim uporabnikom. Iz navedenega lahko sklepamo, da je TAM primeren tudi za uporabnike »polizdelkov«.
- Podobne rezultate kot za poenostavljeni model TAM smo dobili tudi za TAM, ki je namenjen trajni uporabi programske opreme.
- Za oba modela TAM se je izkazalo, da sta primerna le v primeru prostovoljne uporabe ogrodja, kar prav tako ustreza predhodnim raziskavam.
- Primerjava obeh modelov TAM s predhodnimi raziskavami se ujema tudi v povezavi med dojeto enostavnostjo uporabe in odvisno spremenljivko. Omenjena povezava je pomembnejša v primeru napovedovanja trajne uporabe ogrodja.

- Razširitev modela TAM za napovedovanje trajne uporabe ogrodja je podala naslednje ugotovitve. Konstrukta, s katerima smo razširili osnovni model (zaupanje in ujemanje zahtev z ogrođjem), nista povečala natančnosti napovedovanja odvisne spremenljivke. Zaupanje in ujemanje zahtev z ogrođjem sta zato le zunanji spremenljivki modela TAM za napovedovanje trajne uporabe.
- Poenostavljeni model TAM, ki smo ga nadgradili z enakima konstruktoma, je podal drugačno ugotovitev od predhodne. Ugotovili smo, da zaupanje vpliva na sprejetost ogrođja, medtem ko TTF ostaja zunanja spremenljivka. Do podobne ugotovitve so prišli (Yu et al. 2005) v kontekstu trgovine na osnovi interaktivne televizije (*T-Commerce*). Avtorji so v primeru izkušenih uporabnikov ugotovili, da na namero o uporabi vpliva faktor zaupanje (*trust*).
- Model sprejetosti, ki smo ga oblikovali na eksplorativen način na osnovi predhodnih ugotovitev predstavlja prilagoditev in nadgradnjo splošno uporabnega modela TAM za kontekst ogrođij. V modelu sta poleg faktorjev modela TAM vključena kontekstni faktor zaupanje (CF) in faktor ujemanje zahtev z ogrođjem (TTF). Izkazalo se je, da je predlagani model sprejetosti ogrođja (Slika 74) v kontekstu ogrođij natančnejši in popolnejši od poenostavljenega modela TAM.



SLIKA 74: MODEL SPREJETOSTI OGROĐIJ

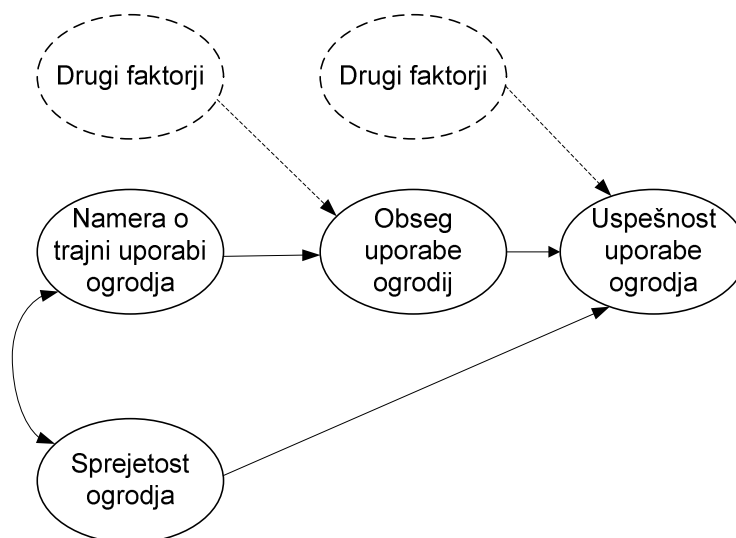
Z oblikovanjem modela sprejetosti ogrođij (Slika 74) smo potrdili tudi tezo doktorske disertacije, saj model integrira dva predhodna modela (model TAM in model TTF) in vključuje faktor iz domene ogrođij (CF). Poleg tega je statistična analiza potrdila, da je model sprejetosti ogrođij natančnejši od obeh modelov, ki jih vključuje.

3.7.12.3 INTERPRETACIJA REZULTATOV RV₃

Cilj oblikovanja RV₃ je bil identificirati in pojasniti povezavo med sprejetostjo in uspešnostjo ogrodiv. Pri raziskavi RV₃ smo se zgledovali Seddonovem modelu uspešnosti in njegovi interpretaciji povezave med sprejetostjo, uporabo in uspešnostjo (uporabe). Na osnovi tega smo oblikovali teoretični model, ki je vseboval dva eksogena faktorja iz modela TAM in dva endogena faktorja, ki sta predstavljala dimenziji uspešnosti. Na osnovi statistične analize teoretičnega modela lahko podamo naslednje ugotovitve.

- Merljiv model je potrdil konvergentno in divergentno veljavnost faktorjev, iz česar lahko sklepamo, da smo ustrezno operacionalizirali faktorja neto prednosti uporabe ogrodiv (NBF) in zadovoljstvo uporabe ogrodiv (SAT).
- Kot smo predpostavljali, je raziskava strukture teoretičnega modela 3 potrdila povezanost faktorjev sprejetosti s faktorji uspešnosti ogrodiv. Iz tega lahko sklepamo, da v kontekstu ogrodiv pomeni več uporabe več prednosti in s tem večjo stopnjo uspešnosti. Skladno s tem je uspešnost uporabe ogrodiva odvisna od obsega njegove uporabe.
- Med sprejetostjo ogrodiva in namero o njegovi trajni uporabi obstaja visoka korelacija, kar pomeni, da sta si konstrukta zelo podobna.

Iz naštetih ugotovitev lahko oblikujemo naslednji kavzalni model uspešnosti ogrodiv (Slika 75):



SLIKA 75: MODEL USPEŠNOSTI OGRODIJ

Iz modela (Slika 25) je razvidno, da smo demonstrirali povezanost konceptov, ki smo jih analizirali v disertaciji (polne črte). S tem smo podali odgovor na tretje raziskovalno vprašanje (RV₃). Vpliv faktorjev, ki jih v okviru disertacije nismo analizirali (prekinjene črte), ostaja nepojasnen.

3.8 POVZETEK RAZISKAV

Raziskave, ki smo jih izvedli v okviru disertacije, so podale odgovore na tezo disertacije in iz nje izhajajoča raziskovalna vprašanja. Poglavitne ugotovitve raziskav ob upoštevanju navedenih omejitev raziskav so naslednje:

3.8.1 ODGOVORI NA RAZISKOVALNO VPRAŠANJE RV₁

Odgovor na zastavljeno raziskovalno vprašanje RV₁ je naslednji. Na osnovi terenske raziskave smo oblikovali taksonomijo faktorjev, ki vplivajo na uporabnikove odločitve o izbiri, sprejetju ali nadaljnji uporabi ogrodja. Faktorje smo strukturirali v drevesno obliko na osnovi sorodnih modelov kakovosti in ponovne uporabnosti. Ugotovili smo, da na uporabnikove odločitve vplivajo faktorji kakovosti, ponovne uporabnosti in ostali netehnološki faktorji, med katere spadajo: priljubljenost, poznanost, pravni vidik uporabe ogrodja in podpora ogrodju. Iz tega je možno sklepati, da sprejetost ogrodja ni odvisna le od »tehnologije« ogrodja, temveč še od drugih vidikov in posebnosti posameznikov oziroma organizacij. Do navedenih ugotovitev smo prišli na osnovi kvalitativnih raziskav.

Demonstracijo signifikantnosti faktorjev, ki vplivajo na sprejetost ogrodij, smo zaradi velikega števila faktorjev izvedli na podmnožici faktorjev, ki so izkazovali največjo relevantnost. Pri tem smo upoštevali rezultate predhodnih raziskav. Ugotovili smo, da ima izbrana podmnožica faktorjev signifikanten vpliv na uporabnikova dojetanja ogrodij in s tem na sprejetost ogrodij. Faktorji, za katere smo potrdili signifikantnost vpliva na uporabnikova dojetanja, so: zaupanje, učinkovitost, prenosljivost, razumljivost, ujetanje zahtev in ogrodja in tehnološki razkorak. Definicije in empirične indikatorje vseh faktorjev smo izpeljali iz pripadajočih virov (na primer: ISO/IEC 9126 in REBOOT).

3.8.2 ODGOVORI NA RAZISKOVALNO VPRAŠANJE RV₂

Odgovor na zastavljeno raziskovalno vprašanje RV₂ smo dobili na osnovi primerjave kvantitativnih statističnih podatkov. Za potrebe odgovora na RV₂ smo oblikovali štiri teoretične modele: dva splošna modela, ki smo ju povzeli po obstoječih virih, in dva specifična modela, ki sta predstavljal nadgradnjo splošnih modelov. Po izvedeni statistični analizi strukturnih modelov smo primerjali primerljiva splošna in specifična modela. Pri tem so ugotovitve naslednje:

- Oblikovali smo model sprejetosti ogrodij, ki je natančnejši od primerljivega modela TAM. Model vsebuje dve determinanti sprejetosti ogrodij: dojeta uporabnost ogrodja in zaupanje v ogrodje.
- Specialen model napovedovanja namere trajne uporabe ogrodja se ni izkazal za natančnejšega od primerljivega modela TAM.

Na osnovi navedenih modelov in izbranega potrditveno-raziskovalnega pristopa smo oblikovali model sprejetosti ogrodij, ki se je v kontekstu ogrodij izkazal za popolnejšega in natančnejšega od modela TAM.

3.8.3 ODGOVORI NA RAZISKOVALNO VPRAŠANJE RV₃

Odgovor na zastavljeno raziskovalno vprašanje RV₃ smo dobili na osnovi oblikovanja in analize strukturnih modelov, ki povezujejo faktorje sprejetosti ogrodij in faktorje uspešnosti ogrodij. Ugotovili smo, da obstaja povezava med faktorjema sprejetost in namera o trajni uporabi ogrodja ter faktorji uspešnosti ogrodij. Višja sprejetost, večja namera o trajni uporabi ogrodja in večji obseg uporabe ogrodja vplivajo na uspešno uporabo ogrodja.

3.8.4 ODGOVOR NA ZASTAVLJENO TEZO DISERTACIJE

Ugotovili smo, da z integracijo modelov, ki določajo sprejetost IT, uspešnost IS in specifičnih faktorjev iz domene ogrodij, lahko oblikujemo kavzalni model, ki integrira faktorje sprejetosti ogrodij in faktorje uspešnosti ogrodij in ki je natančnejši od splošnih modelov. V disertaciji smo za splošni model sprejetosti uporabili model TAM, uspešnost ogrodja smo povzeli po Seddonovem modelu uspešnosti IS, specifične faktorje iz domene ogrodij pa smo določili na osnovi terenske raziskave. V nadaljevanju so prikazane aplikacije raziskav, ki temeljijo na potrjeni tezi disertacije.

4 APLIKACIJE RAZISKAV

V tem poglavju so predstavljeni rešitve in osnutki rešitev, ki temeljijo na rezultatih raziskav disertacije.

"Stvari sploh ne obstajajo, so samo procesi."

David Bohm

4.1 MODEL ZA OCENJEVANJE SPREJETOSTI PROGRAMSKIH OGRODIJ

Vprašalnik za ocenjevanje sprejetosti programskih ogrodi (Tabela 49) temelji na rezultatih merljivega in strukturnega modela, ki smo jih oblikovali na osnovi teoretičnega modela 2D (Slika 58). V vprašalnik so vključena Likertova vprašanja za določanje dveh neposrednih determinant sprejetosti ogrodi: zaupanja v ogrodje in dojete uporabnosti ogrodja (glej tudi Slika 71).

TABELA 49: VPRAŠALNIK ZA OCENJEVANJE SPREJETOSTI OGRODIJ

Vprašanje	Se močno strinjam	Se strinjam	Se delno strinjam	Sem neopredeljen	Se delno ne strinjam	Se ne strinjam	Se močno ne strinjam
CF: Zaupanje v ogrodje							
CF1: Menim, da je ogrodje zrelo.	7	6	5	4	3	2	1
CF2: Ogrodje odpoveduje pogosto.(-) ¹¹⁶	1	2	3	4	5	6	7
CF3: V primeru odpovedi jih ogrodje dobro obvladuje.	7	6	5	4	3	2	1
FPU: Dojeta uporabnost ogrodja							
FPU1: Uporaba ogrodja povečuje mojo produktivnost.	7	6	5	4	3	2	1
FPU2: Uporaba ogrodja povečuje učinkovitost na delovnem mestu.	7	6	5	4	3	2	1
FPU3: Uporaba ogrodja mi omogoča, da opravi delo hitreje.	7	6	5	4	3	2	1
FPU4: V celoti gledano, menim, da je ogrodje uporabno za delo, ki ga opravljam.	7	6	5	4	3	2	1

Na osnovi navedenega vprašalnika in uteži faktorjev (β), ki jih je podala statistična analiza teoretičnega modela 2D (Slika 70), je možno določiti model za ocenitev sprejetosti ogrodja na naslednji način:

$$FA = \sum_{i=1}^{i=n} \frac{CF_i}{n} \times \frac{\beta_{FACF}}{\beta_{FACF} + \beta_{FAFPU}} + \sum_{j=1}^{j=m} \frac{FPU_j}{m} \times \frac{\beta_{FAFPU}}{\beta_{FACF} + \beta_{FAFPU}}$$

¹¹⁶ Vprašanja, ki so označena z (-), so inverzna.

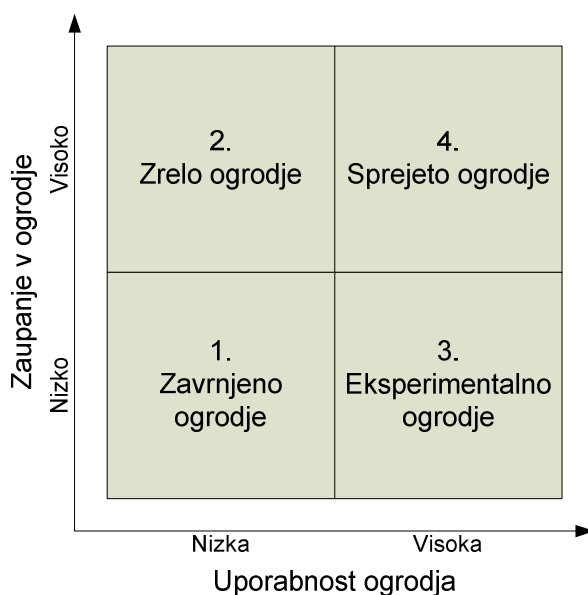
Pri tem je vrednost $n=3$, $m=4$, $\beta_{FACF} = 0.22$ in $\beta_{FAFPU}=0.58$. Rezultati modela so normalizirani na interval Likertove lestvice [1,7], pri čemer višje vrednosti pomenijo boljši rezultat. Model je primeren za ocenjevanje in napovedovanje sprejetosti ogrodja. Delež pojasnjene variance (R^2) faktorja sprejetosti je enak 0.42 (Slika 70). Ker je obseg vprašalnika razmeroma majhen, je vprašalnik primeren za pridobivanje povratnih informacij uporabnikov, ki se pogosto realizirajo s spletnimi anketami.

Ker model ocenjuje sprejetost (FA) posredno, z merjenjem stališč uporabnikov ogrodij (Likertova lestvica), se natančnost modela povečuje s številom ocenitev. V tem primeru se povprečje sprejetosti (\overline{FA}) vrednoti na naslednji način:

$$\overline{FA} = \sum_{i=1}^{i=n} \frac{FA(i)}{n}$$

Pri tem predstavlja $FA(i)$ oceno sprejetosti uporabnika »i«, število »n« pa skupno število vrednotenj sprejetost. Poleg večje natančnosti modela, ki izhaja iz večjega števila ocenitev ogrodja, je takšen model primeren tudi za analizo sprejetosti ogrodja v odvisnosti od sprememb oziroma časa. Tako lahko na podlagi modela ugotovljamo, ali se sprejetost ogrodja povečuje ali zmanjšuje.

Vpliv posameznih faktorjev (CF in FPU) na sprejetost ogrodja se lahko ponazori z mrežo determinant ogrodij (Slika 76). Podobno raziskavo so v primeru modela TAM izvedli Keil et al. (1995).



SLIKA 76: MREŽA DETERMINANT OGRODIJ

Pomen posameznih kvadrantov (Slika 76) je naslednji:

- **Kvadrant 1** predstavlja ogrodja z nizko uporabnostjo in nizkim zaupanjem v ogrodje. Z vidika modela sprejetosti ogrodij uporabniki takšna ogrodja zavračajo, saj niso niti zanesljiva niti uporabna.
- **Kvadrant 2** predstavlja zrela ogrodja, ki imajo nizko stopnjo uporabnosti. Takšna ogrodja so sicer preizkušena in običajno že dalj časa na trgu, vendar uporabnikom bistveno ne povečujejo storilnosti. V to skupino se lahko uvrščajo tudi ogrodja, ki po eni strani povečujejo storilnost, vendar so po drugi strani kompleksna in zapletena za uporabo, kar zmanjšuje storilnost.
- **Kvadrant 3** predstavlja ogrodja, ki imajo velik potencial. To so ogrodja, ki sicer povečujejo storilnost, vendar so še nezrela. V to skupino lahko uvrstimo ogrodja, ki imajo po eni strani odlično filozofijo in temeljijo na zahtevah uporabnikov, vendar so še v fazi preizkušanja.
- **Kvadrant 4** predstavlja uporabna in zrela ogrodja. Ogrodja, ki se uvrščajo v to skupino, povečujejo storilnost in so zanesljiva. Zato jih uporabniki sprejemajo.

Z vidika vodij projektov razvoja ogrodij predstavljata model za ocenjevanje sprejetosti ogrodij in mreža determinant ogrodij osnovo za izvedbo ustreznih izboljšav na ogrodjih.

4.2 OSNOVE METODE ZA OCENJEVANJE IN IZBOLJŠANJE SPREJETOSTI PROGRAMSKIH OGRODIJ

V okviru raziskav disertacije smo ugotovili, da je sprejetost ogrodja faktor, ki vpliva na uporabo ogrodja in skupaj z namero o trajni uporabi na uspešnost uporabe ogrodja (glej poglavje 3.7.10.4). Slednjo ugotovitev lahko uporabimo za oblikovanje naslednje predpostavke:

Pr6: Sprejetost in uspešnost ogrodja lahko povečujemo z modifikacijami ogrodja, ki imajo pozitiven vpliv na faktorje sprejetosti ogrodij.

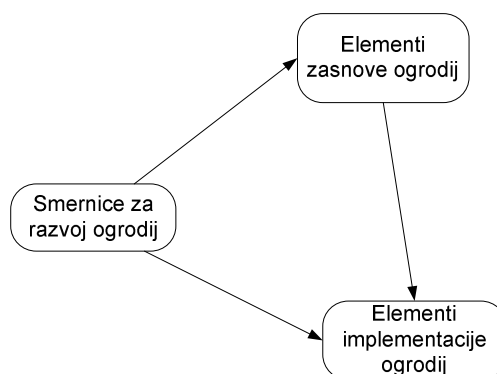
Faktorje, ki vplivajo na sprejetost ogrodij, smo podrobneje raziskali v okviru terenske raziskave in ankete, ki je preučevala raziskovalno vprašanje RV₁. Pri tem smatramo, da je smiselno izvajati takšne modifikacije ogrodij, ki pozitivno vplivajo na karakteristike ogrodij, za katere se je izkazalo, da pozitivno vplivajo na uporabnikova dojetanja ogrodij.

Realizacija navedene predpostavke je problematična v primeru, ko obravnavamo ogrodja kot celoto. Ogradja so označena kot najkompleksnejša tehnika ponovne uporabe, ki sestoji iz množice načrtovalskih in implementacijskih rešitev. Prav tako se ogrodja med sabo močno razlikujejo (glej poglavje 2.2.4), zato je vprašanje, koliko so modifikacije specifičnega ogrodja primerne za modifikacije drugega ogrodja.

Ta problema kompleksnosti in različnosti ogrodij sta rešljiva z definiranjem generične zasnove ogrodja, ki razgradi ogrodje na njegove najpogostejše oziroma abstraktne sestavne dele.

4.2.1 IDENTIFIKACIJA SESTAVNIH DELOV OGRODJA

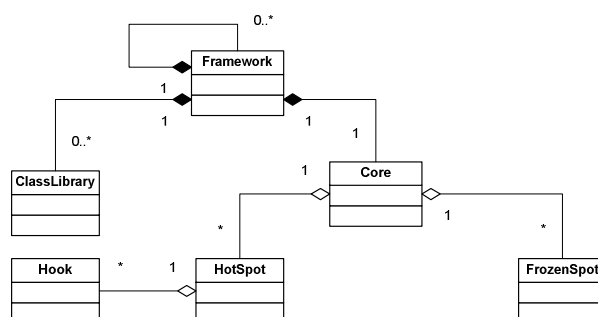
Na osnovi pregleda literature smo identificirali tri vrste elementov, ki imajo vpliv na ogrodje oziroma njegovo implementacijo: elemente implementacije ogrodij, elemente zasnove ogrodij in smernice razvoja ogrodij. Pri tem so tri vrste elementov med seboj povezane, kot prikazujejo kavzalne povezave na spodnji sliki (Slika 77).



SLIKA 77: POVEZAVA MED SESTAVNIMI DELI OGRODJA

4.2.1.1 ELEMENTI IMPLEMENTACIJE

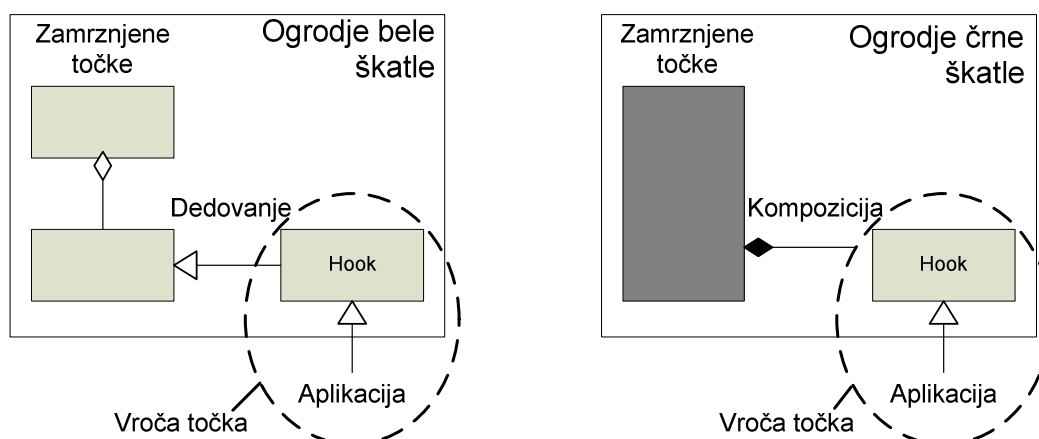
Programski kod oziroma implementacijo ogrodij je analiziralo več avtorjev (Froehlich, Hoover, Liu, & Sorenson 1998; Sangdon, Hansuk, Youngjong, & Sangduck 1999). Na osnovi njihovih ugotovitev smo oblikovali razredni diagram najpogostejših elementov implementacije.



SLIKA 78: NAJPOGOSTEJŠI ELEMENTI NA NIVOJU IMPLEMENTACIJE OGRODIJ

Ogrodja so najpogosteje zgrajena iz vmesnikov, dejanskih in abstraktnih razredov, ki tvorijo jedro ogrodja (*core*), in knjižnice ogrodja (*class library*). Jedro zagotavlja ogrodju poglavitne funkcionalnosti in obnašanje. Knjižnice razširjajo jedro ogrodja s konkretnimi komponentami, ki so uporabne z manjšimi modifikacijami ali brez modifikacij. Ogrodja pogosto vključujejo tudi druga ogrodja (na primer: ogrodje JCorporate Expresso).

Jedro ogrodja se pogosto deli na vroče točke (*hot spots*) in zamrznjene točke (*frozen spots*). Vroče točke predstavljajo dele ogrodja, ki so namenjeni razširitvam. Vroče točke lahko vsebujejo več razširitvenih metod (*hooks*), ko so namenjene dejanskim prilagoditvam in razširitvam. Zamrznjene točke predstavljajo dele ogrodja, ki se ne spreminjajo in tako ostajajo enaki pri različnih instancah ogrodja.

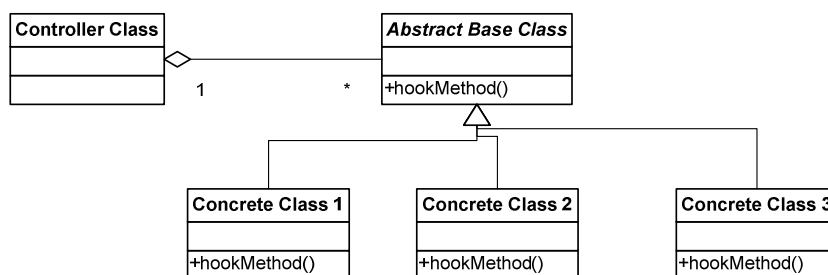


SLIKA 79: IMPLEMENTACIJA VROČIH TOČK V OGRODJU BELE IN ČRNE ŠKATLE (PARSONS ET AL. 2006)

4.2.1.2 ELEMENTI ZASNOVE OGRODJA

Predhodne raziskave so analizirale različne elemente zasnove ogrodij (Froehlich, Hoover, Liu, & Sorenson 1998; Mattsson 2000; Parsons, Rashid, Telea, & Speck 2006), pri čemer sta se kot najbolj pogosti identificirali dve skupini elementov zasnove: dinamično povezovanje (*dynamic binding*) in vzorec načrtovanja (*design patterns*).

Dinamično povezovanje predstavlja tehniko, ki omogoča, da se koda, ki se bo izvedla, določi šele v času izvajanja (*run-time*), kar je v nasprotju z izvedbo, ki se določi v času prevajanja (*compile-time*). Dinamično povezovanje predstavlja ključno lastnost, ki loči ogrodja od drugih tehnik ponovne uporabe, kot so knjižnice ali komponente. Dinamično povezovanje omogoča, da ogrodje kliče primerke in ne obratno, zato je poznano tudi kot »obrat nadzora« ali »Hollywoodsko načelo« (Johnson 2005). Dinamično povezovanje se lahko implementira s podrazredi (*subclassing*), vgradnjo odvisnosti (*dependency injection*), šablonami metod (*template methods*) in zaključki (*closures*).



SLIKA 80: KONCEPT DINAMIČNEGA POVEZOVANJA, KI SE UPORABLJA V OGRODJIH
(CUNNINGHAM, LIU, & ZHANG 2006)

Drugo skupino elementov zasnove ogrodij sestavljajo vzorci načrtovanja, ki so se z ogrodji povezovali vse od pojava ogrodij (Froehlich, Hoover, Liu, & Sorenson 1998). Vzorci načrtovanja predstavljajo opis sodelujočih objektov ali razredov, ki so prilagojeni za reševanje specifičnega načrtovalskega problema (Gamma, Helm, Johnson, & Vlissides 1995). Glede na namen se vzorci načrtovanja delijo na: temeljne (*fundamental*), ustvarjalske (*creational*), strukturne (*structural*), vedenjske (*behavioral*) in sočasne (*concurrent*). Vzorci načrtovanja se uporabljajo za reševanje različnih načrtovalskih problemov ogrodij, pri čemer so najpogosteje uporabljeni za oblikovanje vročih točk z namenom povečevanja prilagodljivosti in razširljivosti. Vzorce načrtovanja, ki se pogosto uporabljajo v ogrodjih, je analiziralo več avtorjev (Cunningham, Liu, & Zhang 2006; Guéhéneuc et al. 2006; Johnson 1992; Lajoie & Keller 1994; Srinivasan 1999). Vzorci načrtovanja, ki se najpogosteje uporabljajo v ogrodjih, so predstavljeni v naslednji tabeli (Tabela 50).

TABELA 50: VZORCI NAČRTOVANJA, KI SE POGOSTO UPORABLJAJO V OGRODIJ

Vzorec načrtovanja	Opis vzorca načrtovanja
Adapter (<i>Adapter</i>)	Prilagodi vmesnik razreda vmesniku odjemalca.
Fasada (<i>Facade</i>)	Zagotavlja poenostavljen vmesnik do obsežnejših razredov oziroma knjižnic.
Opazovalec (<i>Observer</i>)	Omogoča registracijo objektov na dogodke, ki jih prožijo drugi objekti.
Edinec (<i>Singleton</i>)	Omeji instanciranje razreda na en objekt.
Strategija (<i>Strategy</i>)	Vzorec je uporaben v primerih, ko je treba algoritme v aplikaciji zamenjati v času izvajanja.
Kompozit (<i>Composite</i>)	Definira kompozitni objekt, ki sestoji iz več podobnih objektov. Kompozit izpostavi lastnosti in metode podobnih objektov v obliki enovitega objekta.

4.2.1.3 SMERNICE RAZVOJA OGRODIJ

Smernice razvoja ogrodij ali filozofija ogrodij imajo vpliv na elemente načrtovanja in na elemente implementacije ogrodij. Poglavitni namen smernic na področju ogrodij je izboljšanje interakcije med ogrodji in njihovimi uporabniki (Mattsson 2000), kar sovпада s cilji disertacije. Smernice ogrodij so predlagali različni avtorji (Tabela 51), pri čemer sta najbolj celovit katalog smernic oblikovala Landin in Niklasson (Landin & Niklasson 1995). Njun katalog sestavlja 71 smernic ogrodij.

TABELA 51: SMERNICE OGRODIJ, KI SO JIH PREDLAGALI RAZLIČNI AVTORJI

Avtor(ji)	Smernica
(Taligent Inc. 1995)	Zmanjšaj število razredov in metod, ki se morajo preobložiti.
	Poenostavi interakcijo med ogrodjem in razširitvami aplikacije.
	Izoliraj programski kod, ki je odvisen od okolja.
	Izkoristi ogrodje v največjem možnem obsegu.
	Posreduj izvorni kod, da lahko uporabniki zaobidejo omejitve ogrodja.
	Zagotovi »hook metode«, ki bodo aplikacije obveščale o spremembah v ogrodju.
(Birrer & Eggenschwiler 1993)	Združi podobne funkcionalnosti v enovito abstrakcijo.
	Razbij velike abstrakcije v manjše, ki so bolj fleksibilne.
	Implementiraj vse ključne variacije abstraktnih razredov.
	Namesto dedovanja uporabljaj kompozicijo.
(van Gorp & Bosch 2001)	Vmesnik komponente loči od implementacije.
	Vmesniki naj temeljijo na vlogah.
	Za kombiniranje različnih vlog uporabi dedovanje vlog.
	Lahko sklopljenost uporabi prej kot delegacijo.
	Lahko delegacijo uporabi prej kot dedovanje.

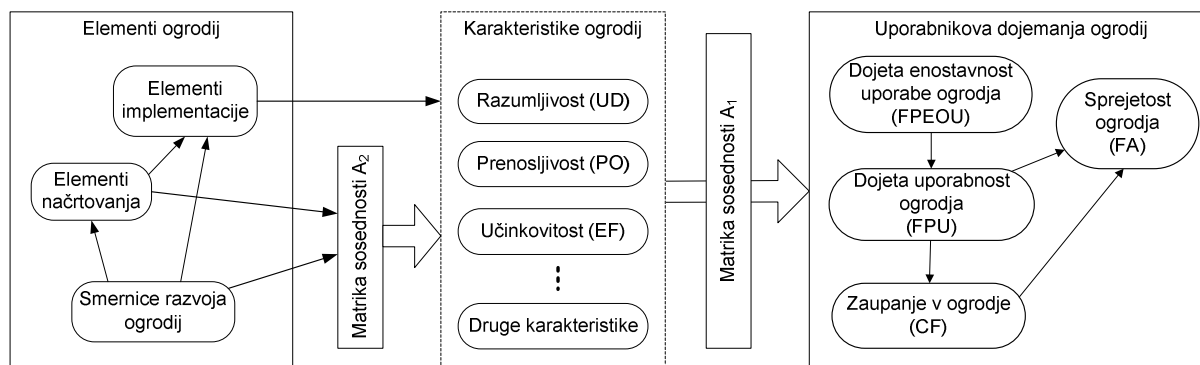
Avtor(ji)	Smernica
	Uporabljaljaj majhne komponente. Uporabi standardizirane rešitve. Avtomatiziraj konfiguracijo. Avtomatiziraj dokumentacijo.
(Manolescu, Noble, & Voelter 2006)	Argumentiraj odločitev razvoja ogrodja. Lepota enostavnosti. Prepreči, da postane ogrodje preobsežno in neobvladljivo. Razvijaj ogrodje jasno in konsistentno. Razvijaj pilotne aplikacije. Uporabljaljaj majhne razrede z namenom povečevanja fleksibilnosti in omejevanja kompleksnosti. Izvajaj teste, ki temeljijo na pilotnih aplikacijah. V razvoj ogrodja vključuj uporabnike ogrodij. V ogrodje dodaj spremembe le, če prihajajo zahteve od več uporabnikov.
(Roberts & Johnson 1997)	Razvijaj ogrodje postopoma.

Naštete smernice so neodvisne od ogrodja, pri čemer obstajajo tudi smernice, ki jih avtorji povezujejo z določenim ogrodjem. Primera takšnih smernic sta »Uporaba konvencij namesto konfiguracije« (*convention over configuration*) in DRY (*Don't Repeat Yourself*), ki predstavljata osnovo razvoja ogrodja »Ruby on Rails« (RoR). Neodvisno od vezanosti na določeno ogrodje sta obe smernici lahko uporabljani tudi v drugih ogrodjih.

Poleg smernic obstajajo tudi »protivzorci« (*anti-patterns*) ogrodij, ki so pogoste slabe rešitve, zato naj bi se jih razvijalci ogrodij izogibali. Primer so protivzorci, ki so jih na osnovi izkušenj identificirali avtorji ogrodja »Keel«.

4.2.2 INTEGRIRAN KONCEPTUALNI MODEL OGRODIJ

V nadaljevanju je predstavljen konceptualni model ogrodij, ki temelji na predpostavki Pr₆. Model predstavlja integracijo poglavitnih elementov ogrodij, karakteristik ogrodij in uporabnikovih dojemanj ogrodij.



SLIKA 81: INTEGRIRAN KONCEPTUALNI MODEL OGRODIJ

Iz zgornjega modela (Slika 81) je razvidno, da imajo elementi ogrodij vpliv na karakteristike ogrodij, ki vplivajo na uporabnikova dojemanja ogrodij. Vplivi med karakteristikami ogrodij in uporabnikovimi dojemanjem ogrodij so bili raziskani v okviru disertacije, medtem ko so vplivi med elementi ogrodij in karakteristikami ogrodij neraziskani. Matriki sosednosti (A_1 in A_2) sta namenjeni ponazoritvi kompleksnih vplivov (mного proti mnogo) med posameznimi skupinami.

Matrika sosednosti izhaja iz teorije grafov. Graf G je množica točk v prostoru in povezav med temi točkami. Označimo ga z $G=(V, P)$, kjer je $V(G)$ množica točk in $P(G)$ množica parov vozlišč, katerih elemente imenujemo povezave.

V našem primeru so točke $V(G)$ kar elementi ogrodij oziroma faktorji in $P(G)$ kavzalne povezave. Ker so kavzalne povezave usmerjene (ponazorjene s puščico), je naš graf usmerjen graf (*directed graph, digraph*).

Graf lahko predstavimo z matriko sosednosti. Matrika sosednosti grafa je kvadratna matrika $A(G) = [a_{ij}]$, v kateri velja naslednje:

$$a(i, j) = \begin{cases} 1, & (i, j) \in P \\ 0, & (i, j) \notin P \end{cases}$$

V primeru uteženih grafov hranimo utež $w: P(G) \rightarrow \mathbb{R}$ v matriki sosednosti na naslednji način:

$$a(i, j) = \begin{cases} w, & (i, j) \in P \\ 0 \text{ ali } \infty, & (i, j) \notin P \end{cases}$$

V našem primeru imamo utežen in usmerjen graf, kjer so uteži ($w: P(G) \rightarrow (-1,1) \subset \mathbb{R}$) enake beta koeficientu (β) oziroma uteži faktorjev (*factor loading*). Utež w na povezavi med aroma vozlišč (i,j) je torej definirana naslednje:

$$a(i,j) = \begin{cases} \beta, & (i,j) \in P \\ 0, & (i,j) \notin P \end{cases}$$

Na osnovi rezultatov raziskav disertacije¹¹⁷ lahko definiramo matriko sosednosti A_1 naslednje (Tabela 52):

TABELA 52: MATRIKA SOSEDNOSTI A_1

	Razumljivost	Prenosljivost	Učinkovitost	Zaupanje v ogrodje	Enostavnost uporabe ogrodja	Uporabnost ogrodja
Razumljivost	0	0	0	0.25	0.42	0
Prenosljivost	0	0	0	0	0.17	0
Učinkovitost	0	0	0	0.09	0.32	0
Zaupanje v ogrodje	0	0	0	0	0	0
Enostavnost uporabe ogrodja	0	0	0	0	0	0.42
Uporabnost ogrodja	0	0	0	0.16	0	0

Povezave med elementi ogrodij in karakteristikami ogrodij predstavljajo vpliv elementov ogrodij na karakteristike ogrodij. Vpliv je možno ovrednotiti na dva načina: (1) neposredno z merjenjem in (2) posredno z analizo vplivov elementov ogrodij na karakteristike ogrodij.

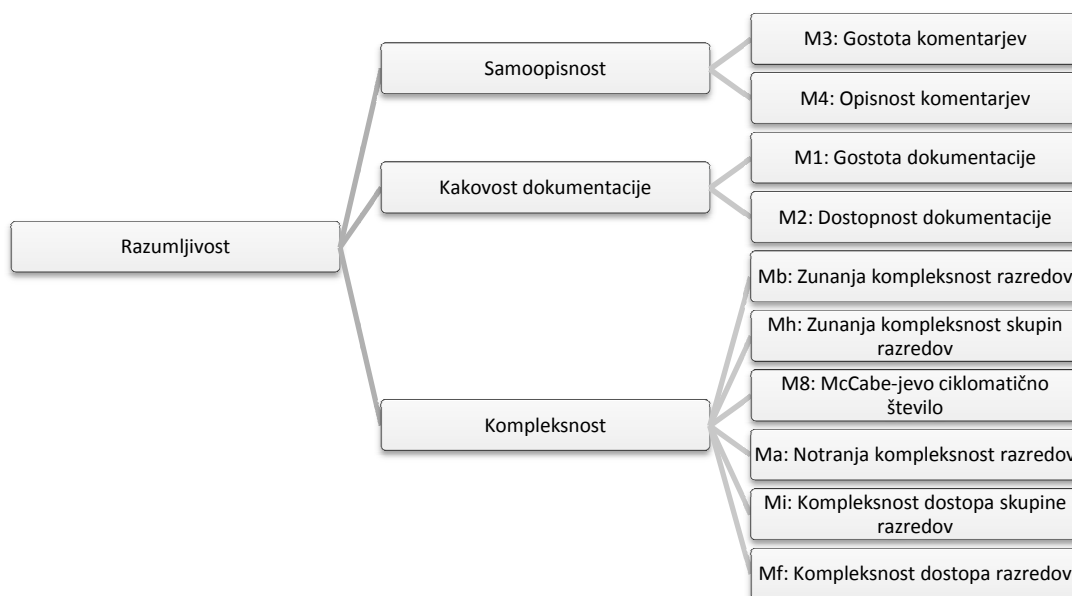
¹¹⁷ Uporabljene so bile vrednosti na kavzalnih povezavah (usmerjene puščice) rezultatov končnega strukturnega modela, izpeljanega iz teoretičnega modela 1.

4.2.2.1 NEPOSREDNO VREDNOTENJE ELEMENTOV OGRODIJ

Neposredno vrednotenje elementov ogrodij je primerno za elemente implementacije oziroma programski kod in ostale artefakte ogrodja. Metrike, ki omogočajo vrednotenje elementov implementacije, so opredeljene v modelih, kot sta model REBOOT in model ISO/IEC 9126. REBOOT in ISO/IEC 9126 določata metrike artefaktov programske opreme (*product metrics*), ki se združujejo v merila in nazadnje v faktorje. Metrike se določajo na objektivni (merjenje programskega koda) in subjektivni način (kontrolni sezname). V nadaljevanju je podan način merjenja razumljivosti, prenosljivosti, učinkovitosti in zaupanja, kot ga opredeljujeta model REBOOT (Slika 82) in model ISO/IEC 9126. Faktor razumljivost je predstavljen v podrobnosti (faktor, merila in metrike), za ostale faktorje pa so za nivo metrik navedene le reference.

RAZUMLJIVOST

Na predhodni sliki je (Slika 82) prikazana razgradnja faktorja »razumljivost« na merila in metrike. Posamezne metrike (M) so kvalitativne (označene s črkami) in kvantitativne (označene s številkami).



SLIKA 82: MODEL FCM FAKTORJA RAZUMLJIVOSTI

Kot primer vzemimo merilo »kompleksnost«, ki je sestavni del razumljivosti in ki jo definirajo naslednje metrike:

M8: McCabejevo ciklomatično število (CC):

$$CC = \text{število odločitev} + 1$$

Pri tem predstavlja odločitev alternativo oziroma odločitven stavek znotraj funkcije oziroma metode. Vrednosti CC pomenijo kompleksnost oziroma tveganja v programskem kodu.

TABELA 53: POMEN VREDNOSTI CC

CC	Kompleksnost in tveganje
<10	Enostaven program brez tveganj
11-20	Kompleksnejši program z obvladljivimi tveganji
21-50	Kompleksen program s tveganji
>50	Program, ki ga ni mogoče obvladovati in ima zato visoka tveganja

Ma: Notranja kompleksnost razreda v(G):

$$v(G) = e - n + 2$$

Pri tem je:

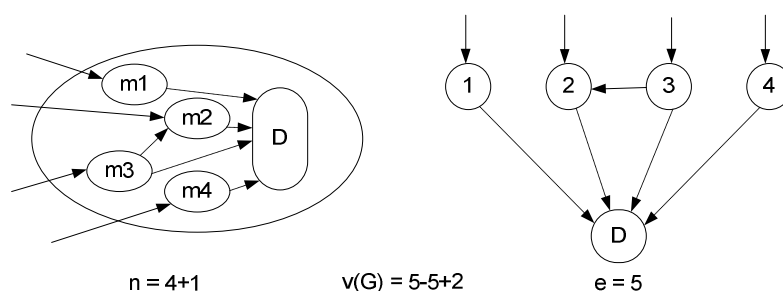
$$n = \|\{i|m_i \text{ uporablja podatke v } D\} \cup \{j|m_j \text{ kliče } m_k, \forall j, k \neq j\}\| + 1$$

$$e = \|\{i|m_i \text{ uporablja podatke v } D\} + \{j|m_j \text{ kliče } m_k, \forall j, k \neq j\}\|$$

Pri tem je:

- n ... število vozlišč + 1 (vse metode, ki kličejo podatke ali druge metode),
- e ... število robov (vse uporabe podatkov od določene metode plus klici drugih metod),
- m... metode razreda in
- D ... podatki (instancirane spremenljivke).

Na naslednji sliki (Slika 83) je prikazano merjenje kompleksnosti razreda, kjer predstavljajo: »mi« = metode, črtkane puščice = dostop do podatkov in polne puščice = klice metod.



SLIKA 83: PONAŽORITEV MERJENJA KOMPLEKSNOSTI RAZREDA

Mb: Zunanja kompleksnost razreda:

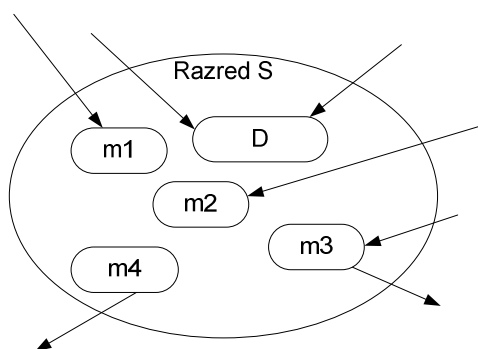
Zunanja kompleksnost razreda (S) se ovrednoti naslednje:

$$FF = Fan_in + \sum_x Fan_out_x$$

Pri tem se fan_in in fan_out_x ovrednotita naslednje:

$$Fan_in = \|\{i|m_i \text{ je javna metoda}\}\| + \|\{j|d_j \text{ je javen podatek}\}\|$$

$$Fan_out_x = \|\{x|m_x \text{ kliče } m_y, x \in S, y \notin S\} \cup \{x|m_x \text{ uporablja } D_y, x \in S, y \notin S\}\|$$



SLIKA 84: PONAZORITEV ZUNANJE KOMPLEKSNOSTI RAZREDA

Mf: Kompleksnost dostopa razredov:

$$Mf = \frac{\#parametrov + \#globalnih\ spremenljivk}{\#parametrov + \#globalnih\ spremenljivk \times W}$$

Pri tem predstavlja:

- # ... število
- W utež (predlagana vrednost je 3)

Mi: Kompleksnost dostopa skupine razredov:

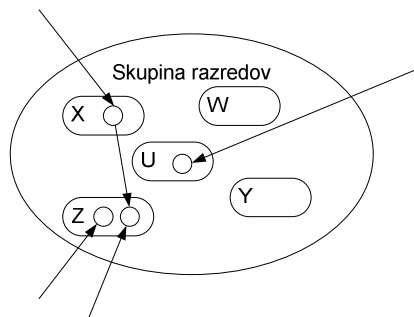
Kompleksnost dostopa skupine razredov je definirana naslednje:

$$Mi = \frac{\sum Mf_j}{\#razredov}$$

Pri tem predstavlja Mf_j metriko Mf od j-tega razreda.

Mh: Zunanja kompleksnost skupine razredov:

Zunanja kompleksnost skupine razredov predstavlja število od zunaj dostopnih metod in podatkov, ki se nahajajo v skupini razredov (v komponenti ali v knjižnici).



SLIKA 85: PONAŽORITEV METRIKE ZUNANJE KOMPLEKSNOSTI SKUPINE RAZREDOV

Na zgornji sliki (Slika 85) so razredi prikazani z ovali X,Y,W,U,Z, metode in podatki pa s krogi znotraj ovalov. Puščice označujejo klice metod oziroma podatkov. Metrika M_h je definirana naslednje:

$$M_h = \sum_i (w_i \times Mb_i)$$

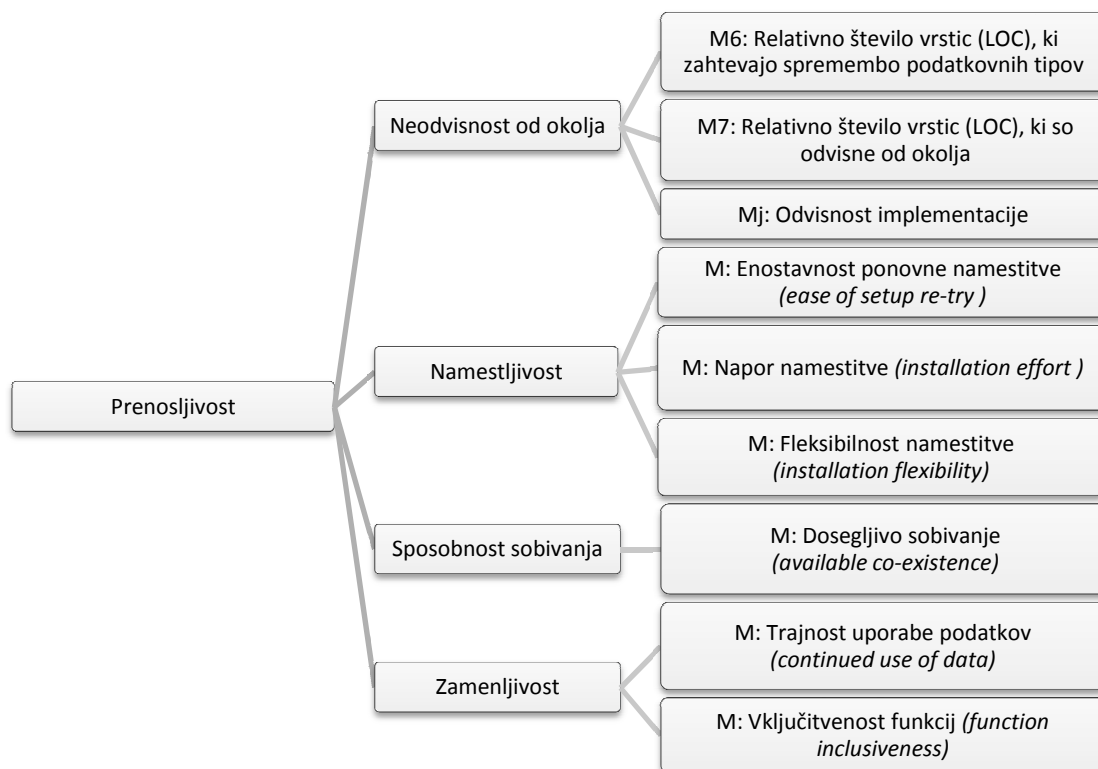
Pri tem predstavlja Mb_i zunanjo kompleksnost i -tega razreda in w_i naslednje:

$$w_i = \frac{ELOC_i}{\sum_j ELOC_j}$$

ELOC (*Effective Lines Of Code*) predstavlja izpeljanko metrike LOC (*Lines Of Code*), ki ne upošteva komentarjev, samostojnih oklepajev in praznih vrstic.

PRENOSLJIVOST

Metrike, ki so sestavni del faktorja prenosljivosti, so definirane v modelu REBOOT in modelu ISO/IEC 9126-3. Na spodnji sliki (Slika 86) je prikazan FCM model prenosljivosti, pri katerem so metrike za merilo "neodvisnost od okolja" povzete po modelu REBOOT in metrike preostalih meril (namestljivost, sposobnost sobivanja in zamenljivost) po modelu ISO/IEC 9126-3.

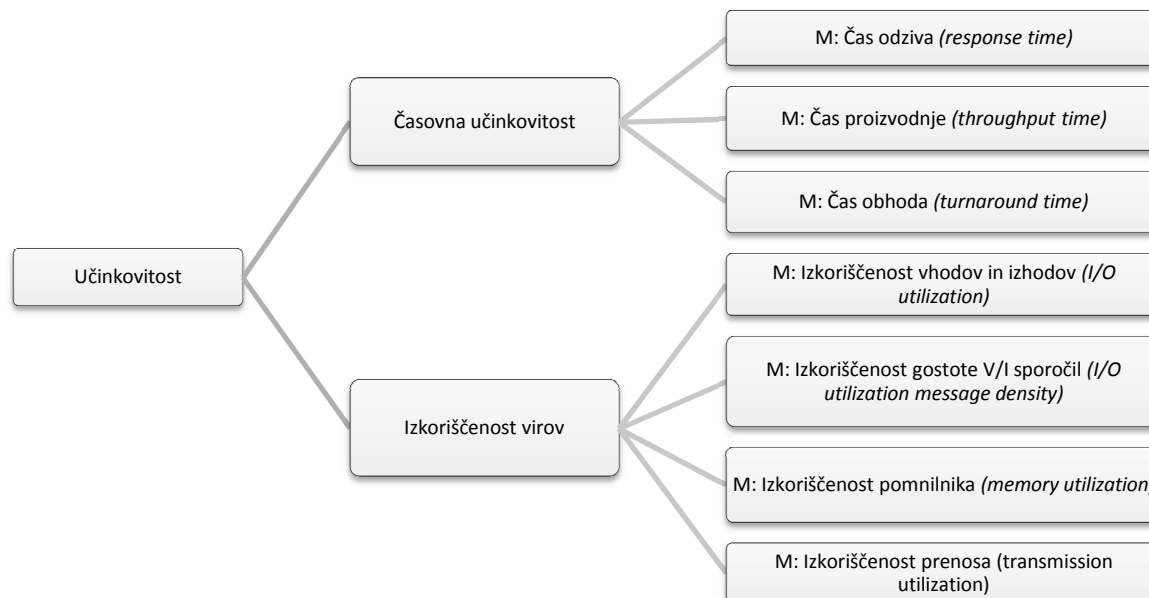


SLIKA 86: MODEL FCM ZA FAKTOR PRENOSLJIVOSTI

Iz zgornje slike (Slika 86) je razvidno, da je faktor razumljivosti mogoče ovrednotiti na osnovi devetih metrik, ki so združene v štiri merila.

UČINKOVITOST

Na spodnji sliki (Slika 87) je prikazan FCM faktorja učinkovitost, ki temelji na metrikah modela ISO/IEC 9126-3.

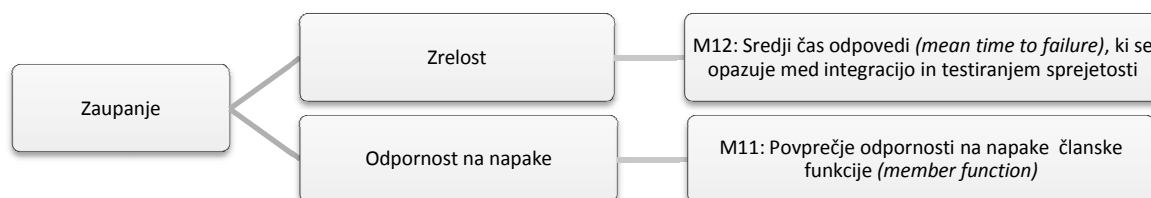


SLIKA 87: MODEL FCM FAKTORJA UČINKOVITOST

Merjenje učinkovitosti je možno izvesti preko dveh meril: časovne učinkovitosti in izkoriščenosti virov. Metrike za obe merili so podrobneje predstavljene v modelu ISO/IEC 9126-3.

ZAUPANJE

Faktor zaupanje je definiran v modelu ponovne uporabnosti REBOOT, kot je prikazano na spodnji sliki (Slika 88).



SLIKA 88: MODEL FCM FAKTORJA ZAUPANJE

Definicije metrik M11 in M12 in pripadajočih meril, so podrobneje predstavljene v modelu REBOOT.

4.2.2.2 POSREDNO VREDNOTENJE ELEMENTOV OGRODIJ

Drugi način vrednotenja elementov ogrodij je posreden, kar je primerno za elemente načrtovanja in smernice razvoja ogrodij. Posredno merjenje se lahko izvede na dva načina: (1) preko vplivov elementov načrtovanja in smernic razvoja ogrodij na elemente implementacije (Slika 81) ali (2) preko matrike sosednosti A_2 .

Matriko sosednosti A_2 je možno oblikovati z analiziranjem vplivov elementov načrtovanja in smernic razvoja ogrodij na karakteristike ogrodij in druge faktorje. Vplive vzorcev načrtovanja je možno razbrati iz katalogov vzorcev, ki podajajo tudi implikacije posameznega vzorca (Gamma, Helm, Johnson, & Vlissides 1995). Rešitev na področju vzorcev načrtovanja so podali Guéhéneuc et al. (2006), in sicer so analizirali vplive vzorcev na karakteristike kakovosti programske opreme (Tabela 54).

TABELA 54: VPLIV VZORCEV NAČRTOVANJA NA KARAKTERISTIKE PROGRAMSKE OPREME
(GUÉHÉNEUC, GUYOMARC'H, KHOSRAVI, & SAHRAOUI 2006)

Vzorec načrtovanja		Razširljivost	Enostavnost	Splošnost	Modularnost	Učljivost	Razumljivost	Operabilnost	Skalabilnost	Robustnost
Abstraktna tovarna	<i>Abs. Fact.</i>	E	E	G	G	G	G	G	G	G
Graditelj	<i>Builder</i>	G	G	F	F	F	G	F	G	G
Tovarna	<i>Fact. Met.</i>	P	P	F	G	G	G	G	G	G
Prototip	<i>Prototype</i>	E	G	F	G	F	G	F	E	G
Edinec	<i>Singleton</i>	P	B	F	E	F	F	F	G	G
Adapter	<i>Adapter</i>	F	F	P	G	G	F	F	G	F
Most	<i>Bridge</i>	G	F	G	G	F	F	G	G	G
Kompozicija	<i>Composite</i>	F	F	F	F	F	G	F	F	G
Dekorater	<i>Decorator</i>	E	E	G	F	G	G	G	G	F
Fasada	<i>Facade</i>	G	G	G	G	F	G	F	F	F
Zrno	<i>Flyweight</i>	P	P	F	G	G	P	F	G	G
Namestnik	<i>Proxy</i>	G	P	F	G	F	P	G	G	F
Veriga odgovornosti	<i>Chain of Res.</i>	G	G	G	P	F	F	G	P	F
Ukaz	<i>Command</i>	G	P	F	F	P	B	G	G	G
Interpreter	<i>Interpreter</i>	G	F	G	F	F	F	G	G	F
Iterator	<i>Iterator</i>	E	E	G	F	G	F	F	G	G
Posredovalec	<i>Mediator</i>	G	F	G	G	F	F	G	G	F
Spomin	<i>Memento</i>	G	F	F	B	P	F	G	F	P
Opazovalec	<i>Observer</i>	E	G	E	F	F	G	G	G	G
Stanje	<i>State</i>	G	G	F	P	F	B	G	G	F
Strategija	<i>Strategy</i>	G	F	P	F	P	P	F	P	F

Vzorec načrtovanja		Razširljivost	Enostavnost	Splošnost	Modularnost	Učljivost	Razumljivost	Operabilnost	Skalabilnost	Robustnost
Šablonska metoda	<i>Tem. Met.</i>	E	G	F	F	G	G	G	G	G
Obiskovalec	<i>Visitor</i>	E	G	G	F	G	P	F	G	F

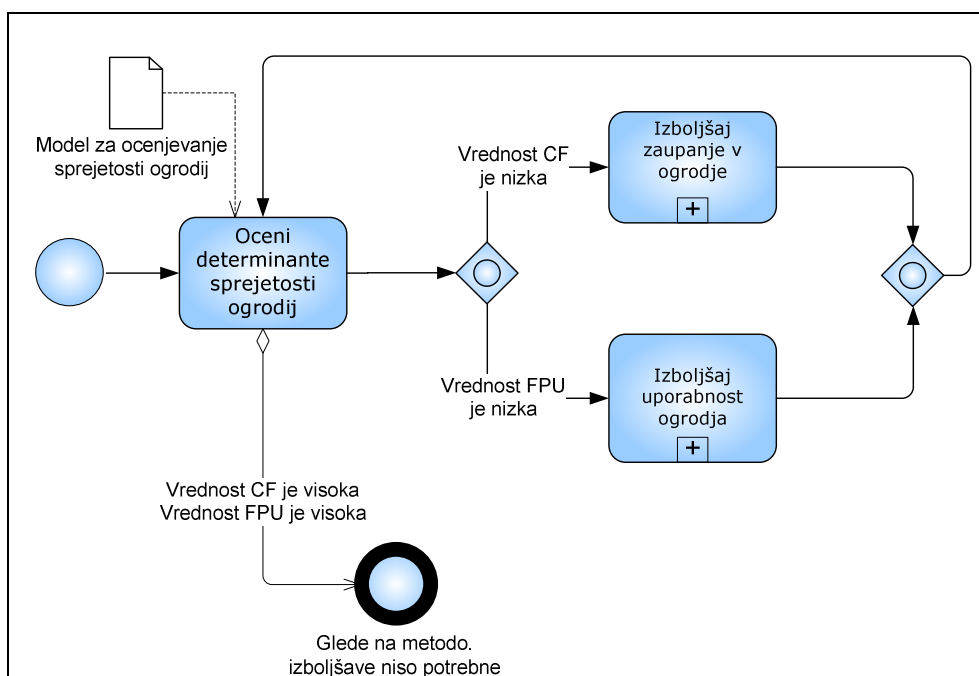
E = odlično (excellent), G =dobro (good), F = zadovoljivo (fair), P = skromno (poor), B = slabo (bad)

Na področju smernic razvoja ogrodi se je zgornjemu delu (Tabela 54) najbolj približal Landin (1995), ki je katalogiziral 71 smernic za razvoj ogrodi in podal implikacije posameznih smernic. Rešitev na področju smernic razvoja zrelih ogrodi sta podala Roberts in Johnson (1997), ki sta oblikovala proces zorenja ogrodi, temelječ na smernicah razvoja ogrodi (Slika 9).

Drugi način oblikovanja matrike sosednosti A_2 se lahko izvede z empiričnimi raziskavami, ki analizirajo razliko med karakteristikami ogrodi pred uvedbo določene spremembe in po njej.

4.2.3 PROCES IZBOLJŠEVANJA SPREJETOSTI OGRODIJ

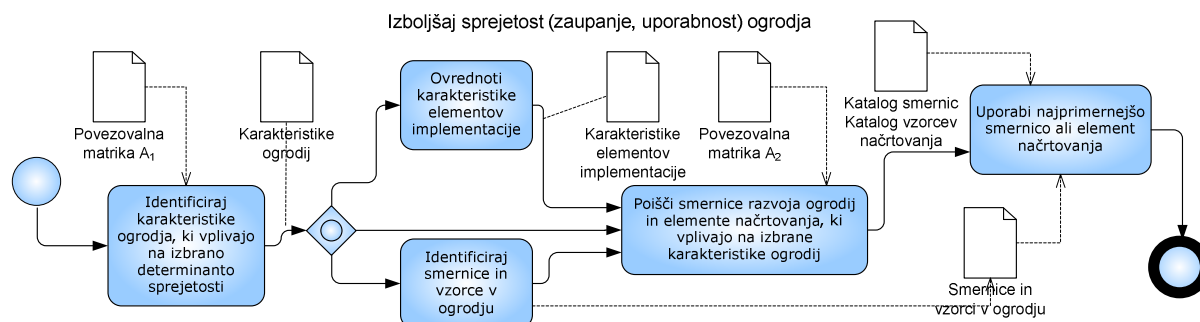
Na osnovi modela za ocenjevanje sprejetosti in integriranega konceptualnega modela ogrodi je možno oblikovati proces izboljševanja sprejetosti ogrodi (Slika 89).



SLIKA 89: PROCES IZBOLJŠEVANJA SPREJETOSTI OGRODIJ

Osnovna ideja procesa izboljševanja sprejetosti ogrođij (Slika 89) temelji na ocenjevanju determinant sprejetosti ogrođij in izboljšavi determinante, ki izkazuje slabše vrednosti. Izboljšave na področju posamezne determinante sprejetosti omogočajo, da se ogrođje razvija v smeri večje sprejetosti (glej Slika 76). Celoten proces je iterativen, kar pomeni, da se po vpeljavi izboljšave ponovno ovrednotijo determinante sprejetosti ogrođij in proces se ponovi.

Izbrana (slabša) determinanta sprejetosti (CF ali FPU) se lahko izboljša po naslednjem podprocesu (Slika 90):



SLIKA 90: PODPROCES IZBOLJŠAVE DETERMINANT SPREJETOSTI OGRODJA

Podproces se prične z izbiro faktorjev, ki vplivajo na determinanto sprejetosti (CF ali FPU). Pri tem se lahko faktorji identificirajo na osnovi matrike sodelovanja A_1 . Ko so identificirane karakteristike ogrođij, ki jih je smiselno izboljšati, se izvedejo (opcijsko) naslednje aktivnosti: (1) preko metrik programskega koda se ovrednotijo karakteristike elementov implementacije in (2) identificirajo se smernice in elementi (vzorci) načrtovanja, ki so že vključeni v ogrođje.

Na osnovi rezultatov prejšnje aktivnosti se izberejo karakteristike, ki jih je najbolj smiselno izboljšati. Izboljšave se nato izvedejo z apliciranjem smernic in elementov načrtovanja ali z modifikacijami elementov implementacije (glej poglavje 4.2.1).

4.2.4 POVZETEK IN NADALJNI KORAKI RAZVOJA METODE

V poglavju 4.2 so bile predstavljene osnove metode za izboljšanje sprejetosti ogrodij. Metoda temelji na iterativnem in parcialnem izboljševanju najpomembnejših faktorjev, ki vplivajo na sprejetost ogrodja: uporabnosti ogrodja in zaupanja v ogrodje. Postopek metode je naslednji. Na osnovi vrednotenja sprejetosti ogrodij se identificirajo karakteristike ogrodij, ki imajo vpliv na faktorje sprejetosti. Karakteristike ogrodij se nato poskušajo izboljšati s smernicami razvoja ogrodij in z vzorci načrtovanja, za katere se je že izkazalo, da pozitivno vplivajo na izbrane lastnosti ogrodij. Predstavljena so bila le izhodišča metode. Aktivnosti, ki jih je treba izvesti, da bo metoda dejansko uporabna v praksi, so naslednje:

- **Dokončanje metode.** Preden bo metoda uporabna, jo je treba dokončati v več pogledih. Prvič, predstavili smo le primere metrik, ki se lahko uporabijo za merjenje karakteristik artefaktov ogrodja. Treba je izvesti pregled vseh metrik ter izbiro in prilagoditev najprimernejših metrik za kontekst ogrodij. Drugič, treba je definirati matriko sosednosti A2, kar pomeni, da je treba identificirati vse elemente načrtovanja in smernice razvoja ogrodij in identificirati njihov vpliv na karakteristike ogrodij. Za elemente ogrodij, katerih vplivi na karakteristike ogrodij niso poznani, je treba izvesti empirične raziskave. Primerna metoda raziskave je laboratorijski eksperiment, ki temelji na eni skupini in vključuje predtest in posttest (*single group pretest posttest experimental design*).
- **Poenostavitev uporabe metode.** Za praktično uporabnost je treba metodo avtomatizirati. V celoti avtomatiziran proces bi zahteval: (1) avtomatizacijo modela za ocenjevanje sprejetosti ogrodij, (2) identifikacijo karakteristik ogrodij, ki jih je smiselno izboljšati, (3) identifikacijo vzorcev in smernic v ogrodju, (4) avtomatizacijo metrik artefaktov ogrodij in (5) ekspertni sistem, ki bi predlagal zamenjavo in/ali uvedbo smernic oziroma elementov načrtovanja ogrodja.
- **Preizkus metode v praksi.** Uporabnost metode je treba preveriti še v praksi, kar zahteva izvedbo ene ali več študij primerov (*case study*) ali akcijskih raziskav (*action research*).

4.3 METODA ZA IZBIRO OGRODJA

4.3.1 OPREDELITEV POBLEMA

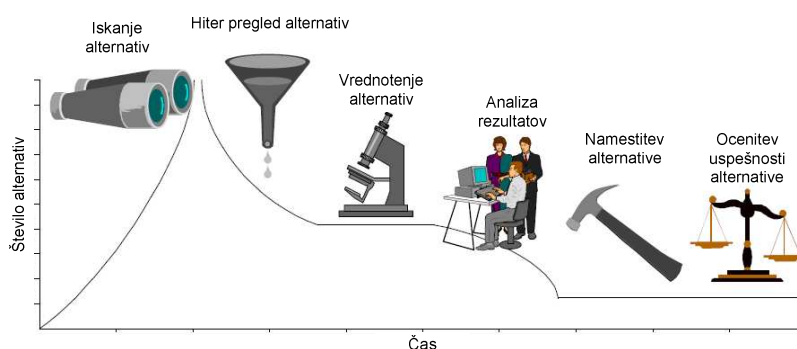
Uporabniki ogrodij oziroma razvijalci končnih rešitev lahko izbirajo med tremi osnovnimi pristopi razvoja:

- **Razvoj na osnovi lastnega ogrodja** oziroma ogrodja, ki je nastalo v okviru organizacije (*in-house framework*).
- **Razvoj na osnovi lastniškega ogrodja** (*proprietary framework*). Lastniška ogrodja razvijajo organizacije z namenom zaračunavanja licenčnine (podobno kot programska oprema COTS¹¹⁸).
- **Razvoj na osnovi odprtokodnega ogrodja** (*open source framework*). Odprtokodna ogrodja se razvijajo v okviru odprtokodnih projektov, ki temeljijo na odprtokodnem modelu razvoja programske opreme.

Zaradi težavnega razvoja lastnega ogrodja in velikega števila javno dostopnih ogrodij se pogosto odločamo za uporabo obstoječega ogrodja. Ker je prvi pogoj za uspešno uporabo ogrodja njegova trajna uporaba, pomeni izbira ogrodja kritičen korak v fazi razvoja družin izdelkov. V nadaljevanju je predstavljen sistematičen pristop k izbiri najprimernejšega ogrodja.

4.3.2 PREDLAGANA REŠITEV

Rešitev, ki jo predlagamo, temelji na raziskavah disertacije in na rešitvah, ki so se na področju drugih vrst programske opreme izkazale za uporabne v praksi. Na področju programske opreme COTS (*Commercial Off-The-Shelf*) so Kontio et al. (1995) razvili metodo OTSO (*Off-The-Shelf Option*), za katero se je v praksi izkazalo, da povečuje učinkovitost in sistematičnost ter pri tem ne povečuje stroškov odločanja (Slika 91).



SLIKA 91: ILUSTRACIJA METODE OTSO (Kontio et al. 1995)

¹¹⁸ *Commercial off-the-shelf*.

Koraki metode OTSO so naslednji (Slika 91):

- **Iskanje alternativ.** Začetek metode predstavlja aktivnost iskanja alternativ. Pri tem je v iskanje smiselno vključiti različne vire (na primer: spletne vire, publikacije in proizvajalce). Pogoj za začetek iskanja je jasna definicija poglobitnih zahtev za alternative.
- **Hiter pregled alternativ.** Cilj aktivnosti je omejitev velikega števila alternativ, ki so se identificirale v fazi iskanja, na manjši obseg, ki ga bo mogoče podrobneje analizirati.
- **Vrednotenje alternativ.** Aktivnost je namenjena vrednotenju posameznih alternativ glede na vnaprej definirane kriterije. Ker je vrednotenje velikega števila kriterijev lahko časovno in finančno preobsežno, je treba določiti pomembnost posameznih kriterijev. Vrednotenje alternativ vsebuje dve podaktivnosti: finančno vrednotenje in vrednotenje kakovosti.
- **Analiza rezultatov.** Aktivnost je namenjena ovrednotenju podatkov, ki so nastali v predhodni aktivnosti. V tej aktivnosti se za odločitveni model priporoča uporaba metode AHP (*Analytic Hierarchy Process*).
- **Namestitev alternative.** Predstavlja dejansko uporabo izbrane alternative.
- **Ocenitev uspešnosti alternative.** Predstavlja aktivnost, ki omogoča ocenitev dejanske uspešnosti uporabe izbrane alternative in povratne informacije v proces izbire alternative v smislu spreminjanja kriterijev odločanja ali uteži v odločitvenem modelu. Kriteriji odločanja se določijo v aktivnosti iskanja alternativ.

Rešitev, ki jo predlagamo, temelji na metodi OTSO, prilagojeni za področje ogrođij. Pri tem temelji prilagoditev na rezultatih raziskav disertacije, in sicer: na faktorjih izbire ogrođij (Slika 45) in rezultatih teoretičnega modela 1 (Slika 66). Poleg tega smo prilagoditev metode zasnovali na naslednjih predpostavkah, ki izhajata iz citatov priznanih raziskovalcev na področju ogrođij:

Pr₇: Razvijalci programskih rešitev bodo za razvoj programskih rešitev izbrali ogrođje, za katero domnevajo, da jim bo prineslo največje neto prednosti^{119 120}.

Skladno s Pr₇ bomo v model odločanja vključili stroškovni model, ki bo ocenil stroške uporabe določenega ogrođja in te primerjal s stroški razvoja družine aplikacij brez uporabe ogrođja.

Pr₈: Prvi pogoj za uspešno uporabo ogrođja je ustreznost ogrođja za izbrano nalogo¹²¹.

¹¹⁹ Koncept neto prednosti je predstavljen v poglavju 2.3.

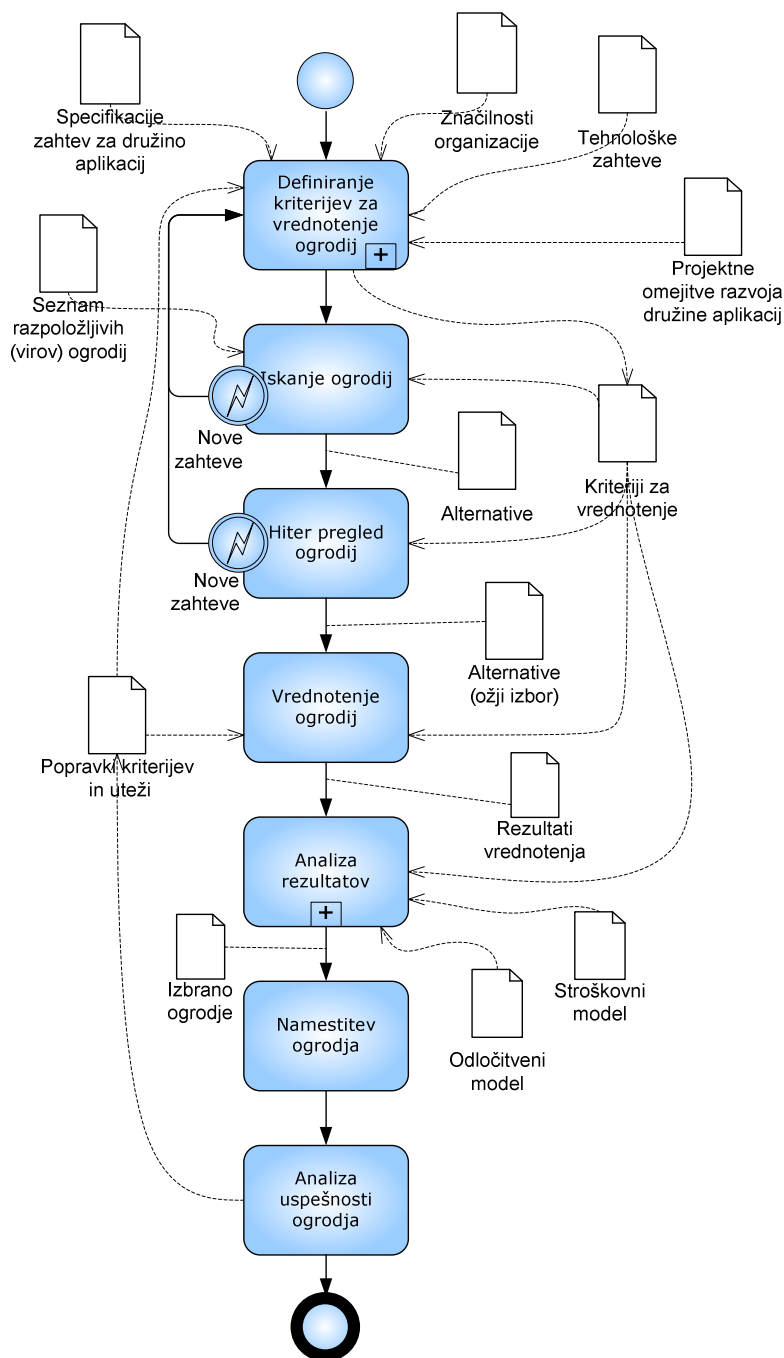
¹²⁰ "The most profoundly elegant framework will never be reused unless the cost of understanding in and then using its abstractions is lower than the programmer's perceived cost of writing them from scratch" (Booch 1994)

¹²¹ "The most important to framework success is the question of appropriateness." (Meier 1998)

Pr₈ se ujema z modelom TTF. Zato bomo v model odločanja vključili primerjavo med zahtevami in lastnostmi oziroma funkcionalnostmi ogrodja.

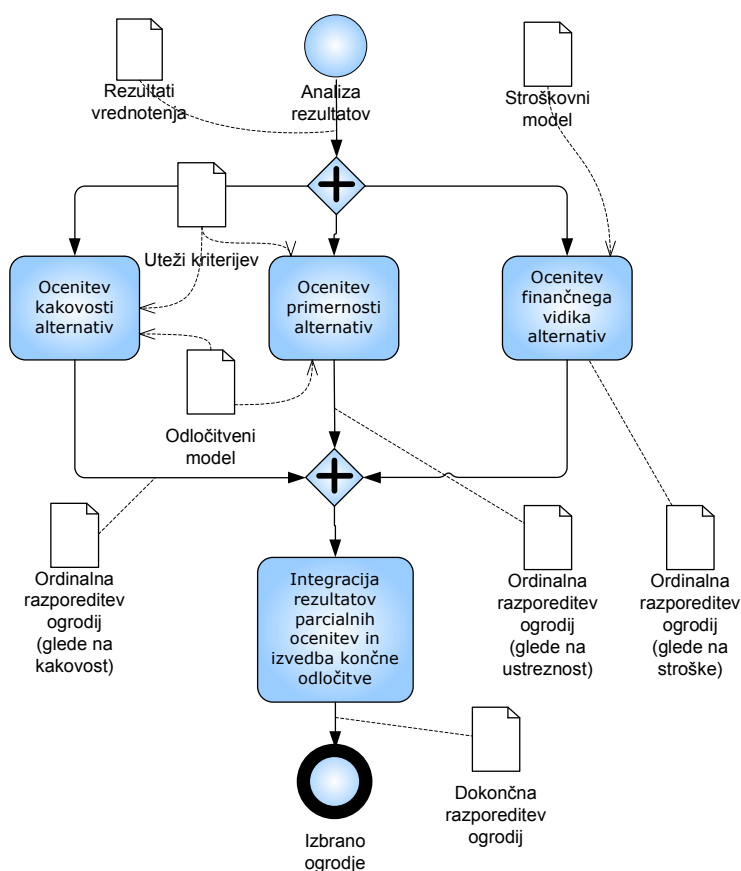
4.3.3 PROCES IZBIRE OGRODJA

Na spodnji sliki (Slika 92) je v notaciji BPMN prikazan proces izbire ogrodja, ki temelji na metodi OTSO.



SLIKA 92: PROCES IZBIRE OGRODJA

Iz predhodne slike (Slika 92) je razvidno, da se proces izbire ogrodja prične z definiranjem kriterijev za vrednotenje ogrodij, ki se oblikujejo na osnovi različnih virov: (1) specifikacij zahtev za družino aplikacij, (2) zahtev in omejitev projekta razvoja družin aplikacij, (3) značilnosti organizacije, ki bo razvijala družino aplikacij (obstoječa infrastruktura, razpoložljivost virov, pravna usmerjenost) in (4) tehnoloških zahtev družine aplikacij (arhitektura, tehnologija, zasnova). Definiranju kriterijev sledijo koraki, kot jih definira metoda OTSO. Pri tem se lahko kriteriji za izbiro ogrodij spremenijo, če se v aktivnostih iskanja in hitrega pregleda alternativ identificirajo novi, pomembni kriteriji. Poglavitno aktivnost odločanja predstavlja analiza rezultatov vrednotenja ogrodij, ki se izvede na naslednji način (Slika 93):



SLIKA 93: PODPROCES ANALIZE REZULTATOV VREDNOTENJA ALTERNATIV

Osnovna ideja analize rezultatov sta neodvisno vrednotenje in ordinalna razporeditev ogrodij z vidika kakovosti, vidika primernosti in finančnega vidika.

- Vidik kakovosti se lahko ovrednoti na osnovi kriterijev, ki so se opredelili v taksonomiji faktorjev izbire ogrodij (Slika 45). Pri tem je smiselno upoštevati faktorje ponovne uporabnosti ogrodij in faktorje kakovosti ogrodij. Za utežitev kriterijev se priporoča uporaba metode AHP.

- Primernost alternativ se lahko ovrednoti na osnovi specifik posameznega odločanja in kod, ki so se v kvalitativni analizi terenske raziskave umestile v merilo »ustreznost« (glej Tabela 12). Za utežitev kriterijev se priporoča uporaba metode AHP.
- Ocenitev stroškovnega vidika alternativ se lahko izvede na osnovi stroškovnega modela razvoja produktnih linij, ki so ga predstavili Bockle et al. (2004).

Vhod v zadnji korak analize rezultatov predstavljajo trije neodvisni sezname ordinalnih razporeditev ogrodi: (1) ordinalna razporeditev ogrodi glede na kakovost, (2) ordinalna razporeditev ogrodi glede na ustreznost in (3) ordinalna razporeditev ogrodi glede stroške. V zadnjem koraku se na osnovi odločitvenega modela ali znanja eksperta izvede dokončna izbira ogrodja.

Proces izbire ogrodja predvideva povratne informacije (glej aktivnost "analiza uspešnosti ogrodja"), ki se oblikujejo v fazi uporabe ogrodja.

4.3.4 NADALJNI KORAKI RAZVOJA METODE

Predstavili smo le osnutek metode za izbiro ogrodja. Nadaljnji koraki razvoja metode bodo usmerjeni predvsem v naslednja področja:

- **Avtomatizacija procesa izbire ogrodja.** Parcialne rešitve, ki avtomatizirajo aktivnosti procesa izbire ogrodja, že obstajajo. Na primer, programska rešitev ExpertChoice™¹²² omogoča obtežitev kriterijev in identifikacijo najprimernejše alternative po metodi AHP. Kljub temu menimo, da je za praktično uporabo metode treba avtomatizirati in medsebojno povezati posamezne aktivnosti procesa izbire ogrodja. Prav tako je pomembno oblikovati repozitorije rezultatov ocenjevanja in omogočiti povratne informacije za optimiziranje uteži in kriterijev ocenjevanja.
- **Preizkus metode v praksi.** Metodo in njeno avtomatizacijo je treba preizkusiti v praksi, in sicer z vidika njene uporabnosti, enostavnosti uporabe, časovnega in stroškovnega vidika. Raziskovalna metoda, ki je primerna za preizkus metode v praksi, je študija primera (*case study*).

¹²² Rešitev je dostopna na www.expertchoice.com.

4.4 POVZETEK APLIKACIJ RAZISKAV

V poglavju Aplikacije raziskav so bili predstavljeni rešitve in osnutki rešitev, ki temeljijo na raziskavah disertacije in imajo praktično vrednost. Predstavili smo tri medsebojno povezane (osnutke) rešitve: model za ocenjevanje sprejetosti ogrodij, metodo za izboljšanje sprejetosti ogrodij in metodo za izbiro ogrodja.

Model za ocenjevanje sprejetosti ogrodij temelji na rezultatih teoretičnega modela 2D (Slika 70). Njegov temeljni namen je napovedovanje sprejetosti ogrodij na osnovi ocenitve dveh neposrednih determinant sprejetosti ogrodij: dojete uporabnosti ogrodij in zaupanja v ogrodje. Na osnovi modela sprejetosti ogrodij smo oblikovali mrežo determinant ogrodij, ki opredeljuje štiri tipe sprejetosti ogrodij: zavrnjeno ogrodje, zrelo ogrodje, eksperimentalno ogrodje in sprejeto ogrodje.

Na osnovi modela sprejetosti ogrodij smo oblikovali osnutek metode za ocenjevanje in izboljšanje sprejetosti ogrodij. Metoda temelji na predpostavki, da je sprejetost in uspešnost ogrodja mogoče povečati z modifikacijami ogrodja, ki imajo pozitiven vpliv na faktorje sprejetosti ogrodij. Metoda upošteva problem kompleksnosti in raznolikosti ogrodij, zato identificira poglobitve abstraktne sestavne dele ogrodja: elemente implementacije ogrodja (programski kod), elemente načrtovanja ogrodja (vzorci načrtovanja) in smernice razvoja ogrodja. V sklopu metode za ocenjevanje in izboljšanje sprejetosti ogrodij smo oblikovali integriran konceptualni model ogrodij, ki v enotnem kavzalnem modelu ponazori povezavo med sestavnimi deli ogrodja in uporabnikovimi dojemami ogrodij. Osnovna ideja metode je, da se z vpeljavo ali modifikacijami poglobitvenih sestavnih delov ogrodij poskušajo izboljšati karakteristike ogrodij, za katere je ugotovljeno, da vplivajo na sprejetost ogrodij. Predstavljena so bila izhodišča metode za izboljševanje sprejetosti ogrodij, prav tako tudi smernice nadaljnega razvoja metode.

Zadnjo rešitev, ki smo jo predstavili v poglavju Aplikacije raziskav, ponazarja metoda za izbiro ogrodja. Metodo smo oblikovali po metodi OTSO, ki se je izkazala za učinkovito v kontekstu programske opreme tipa COTS. Na osnovi rezultatov naših raziskav smo metodo OTSO prilagodili za področje ogrodij. Predstavili smo proces izbire ogrodja, ki vključuje vse faze izbire, in sicer vse od oblikovanja zahtev za ogrodje do vključitve izbranega ogrodja v uporabo. Metoda temelji na postopnem zmanjševanju potencialnih kandidatov oziroma alternativ vse do končne izbire. Končna izbira se izvede na osnovi neodvisne evalvacije treh pomembnih vidikov ogrodij:

- ocene stroškovnih prednosti in slabosti vpeljave ogrodja glede na razvoj družine izdelkov brez uporabe ogrodja,

- ocene kakovosti posameznih ogrodij glede na faktorje ponovne uporabnosti in faktorje kakovosti ogrodij, ki vplivajo na izbiro ogrodja in ki smo jih identificirali v terenski raziskavi disertacije, in
- ocene ustreznosti posameznih ogrodij glede na zahteve razvoja družine aplikacij, omejitve organizacije, projektne zahteve in tehnološke zahteve.

Metoda ni dokončana, zato so predstavljene tudi smernice nadaljnjega razvoja metode in njenega testiranja v praksi.

5 ZAKLJUČKI

Ponovna uporaba je temelj vseh inženirskih disciplin, zato ima pomembno vlogo tudi v programskem inženirstvu. Programska ogrodja (okrajšano ogrodja) predstavljajo eno izmed ključnih tehnologij ponovne uporabe programske opreme in napoveduje se, da bodo odigrala osrednjo vlogo v razvoju programske opreme prihodnosti.

"Znanost je zgrajena iz dejstev, tako kot je hiša zgrajena iz kamna. Vendar samo spisek dejstev ni nič bolj znanost, kot je kup kamenja hiša."

Henri Poincare

Čeprav lahko, glede na relativno mladost programskega inženirstva, programska ogrodja opredelimo kot zrelo in preizkušeno tehnologijo, na omenjenem področju do naše disertacije nismo zasledili raziskav, ki so ključne za vpeljavo in učinkovito uporabo ogrodij.

V disertaciji smo se bolj kot na tehnološko zasnovo ogrodij usmerili v efekte, ki jih imajo ogrodja in lastnosti ogrodij, na njihove uporabnike in na uspešnost projektov, v katerih so uporabljena ogrodja. Pri tem smo se zgledovali po preizkušenih teorijah, ki so se izkazale za verodostojne v drugih tehničnih disciplinah. Osrednjo teoretično osnovo disertacije je tako predstavljal teoretični model TAM (*Technology Acceptance Model*), ki velja za najbolj razširjen, enostaven in preizkušen model za ocenjevanje in napovedovanje sprejetosti informacijskih tehnologij. Za TAM je vse do naše disertacije veljalo, da je uporaben izključno za končne uporabnike, vendar smo z rezultati naših raziskav razširili njegovo uporabnost tudi na področje ogrodij oziroma na področje razvijalcev programskih rešitev.

Poleg slednje ugotovitve smo v disertaciji podali odgovore na zastavljeno tezo disertacije in iz nje izhajajoča raziskovalna vprašanja. Tako smo ugotovili, da je možno razširiti model TAM s faktorji iz drugih teorij in s faktorji iz domene ogrodij ter s tem povečati razumevanje in natančnost ocenjevanja sprejetosti ogrodij. S tem smo povečali razumljivost področja sprejetosti ogrodij.

Identificirali in ovrednotili smo množico zunanjih faktorjev, ki vplivajo na odločitve o izbiri ogrodja, in te preoblikovali v lažje obvladljivo drevesno obliko. Najbolj relevantne faktorje smo analizirali s statistično analizo strukturnih modelov in s tem demonstrirali povezanost karakteristik ogrodij in posebnosti uporabnikov ogrodij s faktorji modela sprejetosti ogrodij. S tem smo razširili dosedanje znanje na področju sprejetosti informacijskih tehnologij, ki je karakteristike sistema in posebnosti posameznikov obravnavalo le kot slepe spremenljivke (*dummy variable*).

Model sprejetosti ogrodij smo razširili tudi na področje faktorjev posledic sprejetosti. Pri tem smo se zgledovali po Seddonovem modelu uspešnosti informacijskih sistemov, ki področje sprejetosti povezuje s področjem uspešnosti uporabe tehnologij. Za področje ogrodij smo potrdili Seddonov pogoj za povezanost področja sprejetosti in uspešnosti, ki pravi, da sta področji kavzalno povezljivi le v primeru tehnologij, katerih neto prednosti uporabe se povečujejo z obsegom uporabe. Povedano drugače, smo s statistično analizo strukturnih modelov ugotovili, da sprejetost ogrodja in namera o trajni uporabi ogrodja vplivata na obseg uporabe ogrodij, ki pa je prvi pogoj za uspešno uporabo ogrodij.

Izvirne znanstvene ugotovitve, ki smo jih predstavili zgoraj, so povečale sistematičnost, discipliniranost in merljivost področja ogrodij. S tem smo področje ogrodij prenesli na višjo raven na lestvici zrelosti inženirskih disciplin.

Izvirni znanstveni prispevki sami ne prinašajo neposrednih prednosti za strokovnjake, ki razvijajo in uporabljajo ogrodja, zato smo v okviru aplikacij raziskav predstavili tri rešitve, ki izsledke raziskav disertacije približajo omenjenim strokovnjakom.

V okviru aplikacij raziskav disertacije smo predstavili formalni model za ocenjevanje sprejetosti ogrodij, katerega osnovni namen je napovedovanje sprejetosti ogrodij na osnovi ocenitve dveh neposrednih determinant sprejetosti ogrodij: dojete uporabnosti ogrodij in zaupanja v ogrodje. Model je namenjen razvijalcem ogrodij, saj lahko z uporabo modela ogrodje razvijajo v smeri večje sprejetosti.

Na osnovi modela sprejetosti ogrodij smo oblikovali osnutek metode za ocenjevanje in izboljšanje sprejetosti ogrodij. Metoda temelji na predpostavki, da je sprejetost in uspešnost ogrodja mogoče povečati z modifikacijami ogrodja, ki imajo pozitiven vpliv na faktorje sprejetosti ogrodij. Pri oblikovanju metode smo upoštevali kompleksnost in raznolikost ogrodij. Osnovna ideja metode je, da se z vpeljavo ali modifikacijami poglavitnih sestavnih delov ogrodij poskušajo izboljšati karakteristike ogrodij, za katere je ugotovljeno, da vplivajo na sprejetost ogrodij. Metoda je namenjena uporabniško usmerjenemu razvoju ogrodij, za katerega so drugi avtorji ugotovili, da predstavlja prvi pogoj za uspešno ponovno uporabo.

Tretja rešitev, ki jo predlagamo, je namenjena uporabnikom ogrodij. Metodo za izbiro ogrodja smo oblikovali iz dveh razlogov: (1) dragega in težavnega razvoja novih ogrodij in (2) velike množice lastniških in odprtokodnih ogrodij, ki so dostopna razvijalcem aplikacij. Metodo smo oblikovali na osnovi že preizkušene metode OTSO, ki je potrdila uporabnost v kontekstu programske opreme tipa COTS. Metodo OTSO smo prilagodili za področje ogrodij z vključitvijo rezultatov terenske raziskave disertacije (drevesnega modela faktorjev izbire ogrodij) in z

upoštevanjem oblikovanega modela sprejetosti ogrodij. Osnovna ideja metode je v postopnem zmanjševanju potencialnih kandidatov oziroma alternativ vse do končne izbire ogrodja, ki se izvede na osnovi neodvisne evalvacije treh pomembnih vidikov ocene ogrodij: stroškovnega, vidika kakovosti in vidika ustreznosti za nalogo in organizacijo.

Nadaljevanje disertacije je možno na področju znanstvenih izsledkov disertacije in tudi na področju aplikacij raziskav. Kakor vse znanstvene raziskave imajo tudi raziskave naše disertacije omejitve, ki so povezane z metodami raziskav, modeli raziskav, metodami analize rezultatov, subjektivnostjo raziskovalcev, vzorci raziskav in drugim. Iz tega razloga je smiselno izvesti komplementarne raziskave, ki bi uporabile drugačne pristope in s tem povečale ali zavrnilo veljavnost naših izsledkov. Druga smer nadaljnjih raziskav je usmerjena v druga področja programskega inženirstva in informatike, v katerih bi lahko ponovno uporabili rezultate in izkušnje, ki smo jih pridobili z raziskavami disertacije, strogo omejenimi na področje ogrodij.

Na področju aplikacij raziskav je treba nadaljevati zastavljeno delo in tako dokončati rešitve, ki smo jih predlagali v okviru disertacije. Za praktično uporabnost predlaganih rešitev menimo, da jih je treba kar najbolj avtomatizirati in izboljševati s preizkušanjem v praksi.

Če povzamemo, je disertacija primer kumulativnega in empiričnega pristopa k raziskovanju in oblikovanju novih znanj na področju ogrodij. Rezultati disertacije so v večji meri netehnični in s tem komplementarni obstoječemu tehničnemu znanju na področju ogrodij. Menimo, da je z disertacijo programsko inženirstvo pridobilo dodaten ugled v družini inženirskih disciplin.

6 PRILOGE

V poglavju Priloge so predstavljeni podatki in dokumenti, ki so sestavni del raziskav disertacije, vendar smo jih zaradi obsežnosti in lažje berljivosti izpustili iz osnovne vsebine disertacije.

PRILOGA A: VPRAŠALNIKA

V tej prilogi sta predstavljena vprašalnika, ki smo ju uporabili v okviru raziskav disertacije.

 VPRAŠALNIK FOKUSNE SKUPINE

Vprašanje, ki je bilo zastavljeno na Slashdot.org, se je glasilo:

»How Do You Decide Which Framework to Use? «

Vprašanje je vključevalo še pojasnilo, ki je bilo oblikovano naslednje:

Software frameworks are increasingly popular software reuse technique, because they provide infrastructure functionalities to an application, or a layer of an application and therefore reduce the work of a software developer. *Numerous* complementary (for example: *Struts* and *Hibernate*) and competitive (for example: *JSF vs. Struts* or *JSF vs. ASP.Net*) software frameworks are available as both *proprietary* and *open source* software. A major precondition for the success of a software framework is their acceptance, which is related to market share or community size. On the other side, application developers need to review and select the best available software framework for their needs. Which factors do you evaluate before you decide to use a specific software framework?"

"Our presumption is that software developers mostly evaluate following software framework characteristics based on:

perceived ease of use (e.g. easy to learn, easy to adapt)

perceived usability (e.g. improving developer performances, reducing work, faster development),

perceived sustainability (e.g. perceived long term support, supporting standards, clear project directions) and

perceived fit to specific developer requirements (e.g. suited language, suited functions, suited architecture).

What are your criteria? Do you support the factors listed above? I am not asking for a preference on a specific software framework, but rather an explanation on the non-trivial task of framework selection, which might be very usable for both frameworks developers and framework users.

VPRAŠALNIK ANKETE

Vprašalnik ankete sta sestavljala dva dela: e-poštno vabilo za sodelovanje v anketi in spletni vprašalnik. Zaslonska slika e-poštnega vabila je predstavljena spodaj (Slika 94: Predloga e-poštnega vabila za sodelovanje v anketi (Slika 94)).

Dear "<name, surname>"

This is to ask a few minutes of your time to help my research group and me with a project that we are carrying out at the University of Maribor (<http://www.uni-mb.si>).


Our study would benefit greatly from your input, and we would therefore like to invite you to complete a short survey. Our research aims to investigate framework developers and users' experiences and opinions about the factors which influence the success of **software frameworks**. As you have been involved in the "<project_name>" project at **Sourceforge.net**, we feel that you have experiences with software frameworks and could so provide valuable information. **The survey will take less than 10 minutes to complete.**

We assure you that the information you provide in the survey will remain strictly confidential, will be reported only in aggregate form and will not be used for any purpose other than for this study.

[CLICK HERE TO START SURVEY](#)

Should you have any questions or concerns please do not hesitate to contact us at gregor.polancic@uni-mb.si. We thank you in advance for your time and support.

Sincerely,
Gregor Polančič

Powered By:
QuestionPro  [Privacy](#) | [Security](#)
[Surveys](#) | [Email Marketing](#) | [Web Polls](#)

University of Maribor | Smetanova 17 | Maribor | Ma | 2000 | Slovenia [Unsubscribe](#) | [Report Abuse](#)

SLIKA 94: PREDLOGA E-POŠTNEGA VABILA ZA SODELOVANJE V ANKETI

V nadaljevanju je podana vsebina spletnega vprašalnika ankete.

Thank you very much for participating in our survey about software frameworks. We are grateful for the time and effort you have accepted to spend for us. Keep in mind that what we are mainly interested in your feelings and opinions, so please feel free to answer a question even if you are unsure about the accuracy of the information in your possess.

1/4: Information about yourself

Gender

1. Male
2. Female

Education

1. High school
2. Bachelor
3. Master
4. Doctorate
5. Other _____

Software development experience in years

1. 0-1 year
2. 1-2 years
3. 3-5 years
4. 5-8 years
5. 8-13 years
6. 13-21 years
7. More than 21 years

Number of frameworks explored (used, tried out) so far

1. 1
2. 2
3. 3-5
4. 5-8
5. 8-13
6. More than 13

Frameworks experience in years

1. Less than 1 year
2. 1-2 years
3. 2-3 years
4. 3-5 years
5. 5-8 years
6. 8-13 years
7. More than 13 years

2/4: General questions about the evaluated framework

Please concentrate on a single (most experienced or most frequently used) framework in following questions. You will relate to this framework in following questions.

Name of the framework you will evaluate in this study.

My framework use is

1. Voluntary
2. Mandatory

Framework related function / position

1. Domain analyst
2. Framework developer
3. Framework user (application developer)
4. Project manager
5. Project leader
6. Consultant
7. Other _____

How many projects did you complete so far using the framework?

1. 0
2. 1
3. 2
4. 3-5
5. 5-8
6. 8-13
7. 13-21
8. More than 21

In how much of current software projects do you use the framework?

1. 0% - 25% of current projects
2. 25% - 50% of current projects
3. 50% - 75% of current projects
4. 75% - 100% of current projects

For how long do you already use the framework?

1. Less than 1 year
2. 1 - 2 years
3. 2 - 3 years
4. 3 - 5 years
5. 5 - 8 years
6. More than 8 years

On average, how frequently do you use the framework?

1. Daily
2. Weekly
3. Monthly
4. Less than monthly

3/4: Questions about your experiences with the evaluated framework

Please indicate your agreement with the next set of statements using the following rating scale.

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
CF1: I believe that the framework is mature.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CF2: The framework fails frequently.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CF3: The framework handles failures well if (when) they occur.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
UD1: The framework is self-descriptive.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UD2: The accessibility, level of detail and quality of documentation is good.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UD3: The framework is easy to learn.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
AD1: The framework can be easily adapted or extended to fulfill application requirements.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AD2: The framework can be installed on different environments.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
AD3: The framework can be easily transferred from one environment to another.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
EF1: The framework requires too much of system resources.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EF2: The framework provides appropriate response and processing times.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EF3: The framework slows down the system (computer).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
IG1: I felt there existed a large gap between my existing skill and knowledge and those required by the framework.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IG2: Learning the framework was not a matter of building on what I knew already.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IG3: Major modification in our software development policies and procedures was necessary for the framework to truly fit in.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
TTF1: The framework functions or services DO NOT suit to my/our application requirements.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TTF2: The framework provides suitable set of functions for specified tasks and user objectives.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

TTF3: In each case I use the framework only if it suits to specified requirements.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

4/4: Questions about your experiences with the evaluated framework

Almost done. Please indicate your agreement with the next set of statements using the following rating scale.

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
PEOU1: The framework is rigid and inflexible to interact with.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEOU2: I find it is easy to get the framework to do what I want it to do.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEOU3: Overall, I believe that the framework is easy to use.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEOU4: Learning to operate the framework is easy for me.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PEOU5: I find it takes a lot of effort to become skillful at using the framework.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
PU1: Using the framework increases my productivity.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU2: Using the framework increases my job performance.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU3: Using the framework enables me to accomplish tasks more quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PU4: Overall, I find the framework useful in my job.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
FA1: I have fully accepted the framework in my daily work.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FA2: I feel that the framework constitutes an integral part of my daily work.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FA3: I consider myself a frequent user of the framework.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FA4: I fully use the capabilities of the framework.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
CFU1: I intend to increase my use of the framework in the future.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CFU2: I intend to continue my use of the framework in the future.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CFU3: I am not going to use the framework in the future.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
NB1: In my case the framework increases the productivity of software development.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NB2: In my case the framework increases the quality of software development.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NB3: In my case the framework lowers the time for finishing software (lead time).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	Strongly agree	Agree	Partial agree	Neutral	Partial disagree	Disagree	Strongly disagree
PO1: I perceive the framework as popular.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PO2: Overall, I believe that the framework is popular.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PO3: Other developers prefer this framework.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please enter any comments or concerns you may have about this survey.

Would you like to be informed about survey results?

1. Yes, please
2. No, thanks

PRILOGA B: PODATKOVNA BAZA SOURCEFORGE

V tej prilogi so predstavljene slike dokumenta oziroma prošnje za dostop do slike podatkovne baze Sourceforge.net. Zadnja slika ponazarja entitetno-relacijski model baze Sourceforge.net, na osnovi katerega smo pridobili seznam sodelujočih pri projektih razvoja ogrodivj.

Request for SourceForge.net Data for a Scholarly Research Study of the Free/Open Source Software Development Phenomenon

OSTG has licensed the University of Notre Dame (UND) the right to share SourceForge.net data with researchers studying the Free/Open Source Software Development phenomenon. Under the terms of this license, we need a description of the research project, the names and contact information of the researchers who will have access to the data, and a copy of the attached sublicense to be signed by the principle investigator on the project who will be responsible for ensuring compliance with the terms of the sublicense.

Please provide the following information. Send this page and a signed copy of the attached sublicense to UND Principal Investigator: Greg Madey, Computer Science and Engineering, 384 Fitzpatrick Hall, University of Notre Dame, Notre Dame, IN 46556 USA; Fax: 574-631-9260; E-Mail: oss@nd.edu

Description of research project (identify sources of external funding support if any):

1. PhD research which investigates major factors which influence open source frameworks success.
2. PhD research which will investigate attitudes of opensource development project professionals at every stage in their project and from every type of team member of virtual community (project manager, developer, content manager etc.).


Names, affiliations, mail addresses, and email addresses of researchers who will have access to the SourceForge.net data:

Gregor Polančič, teaching asistant, research engineer, Katja H.Pulko, teaching asistant, research engineer
University of Maribor, Faculty of electrical engineering and computer science, Institute of Informatics
Addr: Smetanova ulica 17, 2000 Maribor, Slovenia
Tel.: +386 2 220 74 21
Fax: +386 2 220 72 72
URL: <http://lisa.uni-mb.si>
Email: gregor.polancic@uni-mb.si

Katja Harej Pulko, teaching asistant, research engineer
University of Maribor, Faculty of electrical engineering and computer science, Institute of Informatics
Addr: Smetanova ulica 17, 2000 Maribor, Slovenia
Tel.: +386 2 220 74 21
Fax: +386 2 220 72 72
URL: <http://lisa.uni-mb.si>
Email: katja.harej@uni-mb.si

Person responsible to ensure compliance with the terms of the sublicense:

prof. Marjan Heričko, deputy of the head of the Institute of Informatics and head of the Laboratory of information systems



SLIKA 95: PROŠNJA ZA DOSTOP DO BAZE SOURCEFORGE 1/3

SourceForge.net Database Sublicense Terms and Conditions

Introduction

SourceForge.net (<http://sourceforge.net/>) is the world's largest Open Source software development website, with the largest repository of Open Source code and applications available on the Internet. Owned and operated by OSTG, Inc. ("OSTG"), SourceForge.net provides free services to Open Source developers. This website is database driven and the supporting database includes historic and status statistics on projects and user activities at the website. OSTG has shared and intends to continue sharing certain SourceForge.net data (the "Data") with the University of Notre Dame ("UND") for the sole purpose of supporting academic and scholarly research on the Open Source Software phenomenon. OSTG has given UND permission to in turn share Data with other academic researchers studying the Open Source software phenomenon; provided, however, that UND may not provide all of the Data to any one requesting party without OSTG's prior written consent.

Release of Data

To advance the understanding of Open Source software phenomenon, all or portions of the Data will be made available to researchers ("Researchers") on a case-by-case basis in accordance with these provisions and an underlying agreement between OSTG and UND. All requests for all or portions of the Data must be submitted in writing to the UND Principal Investigator by the Researcher. Only parties recognized by OSTG and the UND Principal Investigator as academic and scholarly researchers are eligible to receive the Data. To receive all or a portion of the Data in accordance with these provisions, Researcher must sign this document, agree to abide by SourceForge.net's Terms and Conditions of Use (the "Terms of Use"), available at http://sourceforge.net/docman/display_doc.php?docid=6048&group_id=1, and observe the additional restrictions listed below (collectively, "Restrictions"). In addition to other possible remedies, failure to observe these Restrictions may result in revocation of permission to use the Data as well as denial of access to additional Data distributed by OSTG and UND. Data may be made available to Researcher via an Internet site or on CD or other media. There will be no charge for Data made available and downloaded via the Internet.

Consent

In addition to agreeing to abide by SourceForge.net's Terms of Use, Researcher agrees to the following additional restrictions (the "Additional Restrictions"):

1. Redistribution: Without prior approval from the UND Principal Investigator, Data, in whole or in part, will not be further distributed, published, copied, or disseminated in any way or form whatsoever, whether for profit or not. This includes further distributing, copying or disseminating to a different facility or organizational unit within the Researcher's university, organization, and/or company, as applicable.
2. Modification and Commercial Use: Data, in whole or in part, may not be modified or used for commercial purposes. The license granted herein is specifically for the internal research purposes of Researcher, and Researcher shall not duplicate or use Data, or any seal, logo, mark, or phrase associated with or owned by UND and/or OSTG, to manufacture, promote, or sell products or technologies (or portions thereof) either directly or indirectly for commercialization or any other direct for-profit purpose without the prior written permission of UND.
3. Requests for Data from the SourceForge.net database: All requests for Data from the SourceForge.net database will be forwarded to the UND Principal Investigator.
4. Publication Requirements: Those seeking to include more than 2 kilobytes of actual Data from the SourceForge.net database in reports, papers, and other documents to be published or released must first obtain approval in writing from the UND Principal Investigator. Data shall not be used in any way that could cause SourceForge.net users, UND, or OSTG embarrassment, mental anguish, or other harm. In all cases, project ID, user ID data, and other identifying attributes shall be anonymized to preserve privacy of the SourceForge.net users.

5. Citation: All documents and papers that report on research that uses SourceForge.net Data must acknowledge the use of the database by including an appropriate citation that will be provided upon request by the UND Principal Investigator.

6. Publications to UND: A copy of all reports and papers that are for public or general release that use the SourceForge.net database must be forwarded immediately upon release or publication to the UND Principal Investigator.

7. Conflicts. Any conflict between these Additional Restrictions and SourceForge.net's Terms of Use shall be resolved in favor of these Additional Restrictions.

8. Indemnification: Researcher and/or researcher's employing institution agree to indemnify, defend, and hold harmless SourceForge.net, OSTG, Inc., the University of Notre Dame du Lac and its Board of Trustees, officers, employees and agents, individually and collectively, from any and all losses, expenses, damages, demands and/or claims based upon any such injury or damage (real or alleged) and shall pay all damages, claims, judgments or expenses resulting from Researcher's use of the SourceForge.net database.

NAME (in capitals)

GREGOR POLANČIČ, KATJA H. PULKO

RESEARCH ADVISOR (if you are a student)

ROMANA VAJDE HORVAT, PhD

SIGNATURE





DATE

13th December 2007

ORGANIZATION, MAIL ADDRESS, and EMAIL ADDRESS (in capitals)

UNIVERSITY OF MARIBOR

Dean

Faculty of electrical engineering and

prof. Igor Tičar

computer science

SMETANOVA 17

2000 MARIBOR

SLOVENIJA, EUROPE



Send to UND Principal Investigator: Greg Madey, Computer Science and Engineering, 384 Fitzpatrick Hall, University of Notre Dame, Notre Dame, IN 46556 USA; Fax: 574-631-9260; E-Mail: oss@nd.edu (if sending a fax, please send follow-up hard copy by mail)

PRILOGA C: PREDSTAVITEV POMEMBNEJŠIH POSTOPKOV STATISTIČNE ANALIZE PODATKOV

V tej prilogi so predstavljeni pomembnejši postopki statistične analize podatkov, ki so se uporabljali v raziskavah disertacije oziroma so bili osnova za kompleksnejše statistične tehnike raziskav disertacije.

Statistika je znanost in veščina o razvoju znanja z uporabo izkustvenih podatkov. Njena osnova je matematična statistika, ki je veja uporabne matematike. V statistični teoriji naključnost in negotovost opišemo v okviru teorije verjetnosti. Ker je naloga statistike izluščiti »najboljšo« informacijo iz razpoložljivih podatkov, jo nekateri uvrščajo v teorijo odločanja.

Statistične metode temeljijo na teoriji verjetnosti. Osnovni koncepti so populacija, vzorci, vzorčenje in verjetnost domnev. Izhodišče za statistično delo je zbiranje podatkov bodisi z anketiranjem bodisi s sistematskim opazovanjem, temu pa sledi povzemanje na osnovi opisne statistike. Statistika se ukvarja z obdelavo, analizo, predstavitvijo množičnih pojavov v znanosti, gospodarstvu, industriji, kmetijstvu, zdravstvu, socialnem življenju, šolstvu, biologiji, prometu, trgovini, varstvu okolja in vseh geografskih vedah.

Statistična obdelava podatkov oblikuje podatke na način, po katerem je mogoče priti do predstave o večji populaciji, ki jo predstavlja vzorec. Taka obdelava vključuje preizkuse domnev (npr. anketna vprašanja z dvema možnima odgovoroma), določanje lastnosti zbranih kvantitativnih podatkov, časovne vrste (npr. napovedovanje prihodnjih trendov), opisovanje povezanosti (korelacija), oblikovanje odnosov med spremenljivkami (regresija) in drugo.

POVPREČJE, VARIANCA IN STANDARDNI ODKLON

Povprečje slučajnega vzorca x_1, x_2, \dots, x_n , ki predstavlja realizacijo slučajne spremenljivke X , izračunamo na naslednji način:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Pogosto nas ne zanima samo, katero vrednost bo slučajna spremenljivka X zavzela v povprečju, ampak tudi, koliko je v povprečju od te srednje vrednosti oddaljena. Za izračun tega nam služita dodatni številski karakteristiki: varianca ali disperzija in standardni odklon ali standardna deviacija.

Varianca meri, koliko je kvadrat slučajne spremenljivke v povprečju oddaljen od svoje pričakovane vrednosti. Večja ko je varianca, bolj razpršena je, v povprečju, slučajna spremenljivka.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Slabost variance je, da se pri njenem izračunu kvadrirajo enote, v katerih se meri slučajna spremenljivka, zaradi tega si je včasih težko predstavljati, kaj varianca pomeni. Zato pogosto podamo razpršenost slučajne spremenljivke z njenim standardnim odklonom, ki se izračuna kot kvadratni koren variance $\sigma = \sqrt{\sigma^2}$.

Na standardni deviaciji temelji tudi ena najbolj znanih transformacij testnega rezultata, z-vrednost, ki je definirana naslednje:

$$Z_x = \frac{x_i - \bar{x}}{\sigma_x}$$

Velikost z-vrednosti pove, kako oddaljen, v enotah standardne deviacije, je položaj posameznega rezultata od srednje vrednosti (torej od ničelne točke). Skupaj s krivuljo normalne porazdelitve nam z-vrednosti služijo za neposredno in preprosto razlago testnih rezultatov.

KORELACIJA IN KOVARIANCA

Lastnosti so korelacijsko (stohastično) povezane, kadar ni (direktne) odvisnosti Y od X oziroma je ne poznamo. Korelacija ali korelacijski koeficient je številska mera, ki predstavlja moč linearne povezanosti dveh spremenljivk. Statistična veda s korelacijo v splošnem označuje odvisnost dveh spremenljivk v statistični populaciji ali populacijah.



SLIKA 99: ILUSTRACIJA KORELACIJSKE ZVEZE MED SPREMENLJIVKAMA X IN Y

Korelacijo je mogoče meriti z več koeficienti, prilagojenimi za različne vrste podatkov, ki so na voljo. Izmed korelacijskih koeficientov je najbolj znan Pearsonov koeficient korelacije.

Pearsonov koeficient korelacije je matematična in statistična številska mera, ki izraža velikost linearne povezanosti spremenljivk X in Y, merjenih na istem predmetu preučevanja. Koeficient je definiran kot vsota vseh produktov standardnih odklonov obeh vrednosti v razmerju s

stopnjami prostosti oziroma kot razmerje med kovarianco in produktom obeh standardnih odklonov:

$$r_{xy} = \frac{\sum z_x z_y}{N - 1}$$

pri tem je: z_x z-vednost spremenljivke X, z_y z-vednost spremenljivke Y in N število vseh statističnih enot

$$r_{xy} = \frac{C_{xy}}{\sigma_x \sigma_y}$$

ali pa je: C_{xy} kovarianca, σ_x standardni odklon spremenljivke X in σ_y standardni odklon spremenljivke Y.

Dobljeni rezultat je eden izmed kvadratnih korenov (bodisi pozitiven bodisi negativen) koeficienta determinacije r_{xy}^2 , ki je razmerje med pojasnjeno varianco in skupno varianco:

$$r_{xy}^2 = \frac{\sum (Y' - \bar{Y})^2}{\sum (Y - \bar{Y})^2}$$

Pri tem je: Y dejanska vrednost dane spremenljivke Y in Y' predvidena vrednost iste spremenljivke Y ob znani korelaciji med X in Y ter vrednosti X.

Vrednost Pearsonovega koeficienta korelacije se lahko nahaja med vrednostma -1 in 1. Tako vrednost -1 predstavlja popolno negativno povezanost spremenljivk, pri čemer je na grafu odvisnosti videti le ravno črto, ki z naraščajočo neodvisno spremenljivko potuje navzdol. Vrednost 1 pomeni popolno pozitivno povezanost in navzgor usmerjeno črto na grafu. V praktičnem preizkušanju odvisnosti in uporabni statistiki skoraj ni mogoče izračunati popolne (funkcijske) odvisnosti -1 ali 1, saj na posamezno odvisno spremenljivko vpliva praviloma več dejavnikov, med njimi tudi slučajni vplivi. Pearsonov koeficient 0 označuje ničelni vpliv ene spremenljivke na drugo.

Pogoj za računanje tega koeficienta je linearna odvisnost obeh vpletenih spremenljivk. Za določanje povezanosti spremenljivk, ki niso povezane linearno, se uporablja Spearmanov koeficient korelacije.

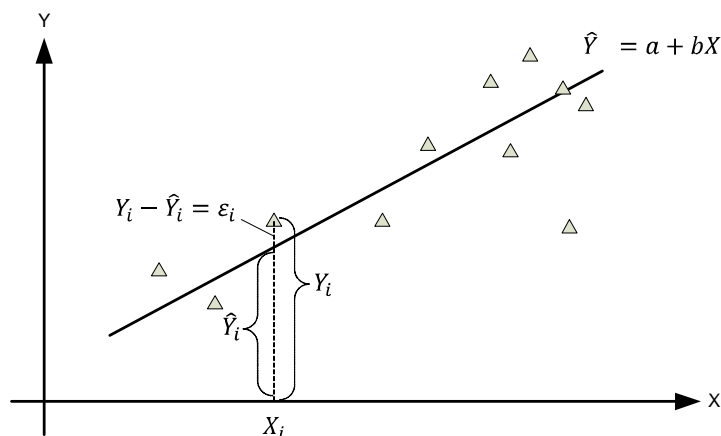
 REGRESIJSKA ANALIZA

Regresijska analiza je statistična metoda oziroma skupina metod, ki nam pomaga analizirati odnos med odvisno spremenljivko in eno ali več neodvisnimi spremenljivkami. Pri regresijski analizi ravnamo tako, da najprej postavimo teoretične predpostavke o odnosih med spremenljivkami oziroma postavimo regresijski model, ki ga nato testiramo na določenem vzorcu. Z regresijsko analizo nato ocenimo parametre modela in statistični pomen tega modela. Poleg opisne vloge ima regresijska analiza tudi napovedovalno vlogo, ko iz sprejetega modela in ocen njegovih parametrov lahko iz vrednosti neodvisnih spremenljivk napovemo vrednost odvisne spremenljivke.

Regresijske analize je več vrst, pri čemer je najpreprostejša bivariatna regresija, pri kateri je odvisnost med odvisno in neodvisno spremenljivko linearna.

$$Y = a + bX + \varepsilon$$

Linearna regresija predstavlja poskus, da bi empirično ugotovljeno funkcijsko odvisnost približno opisali s preprosto linearno funkcijo (Slika 100).



SLIKA 100: ILUSTRACIJA LINEARNE REGRESIJE

Izraz $\varepsilon_i = Y_i - \hat{Y}_i$ je razlika med merjeno vrednostjo Y_i in regresijsko vrednostjo $\hat{Y}_i = a + bX_i$ odvisne spremenljivke Y . Najpomembnejši kriterij za ujemanje regresijskega modela $\hat{Y} = a + bX$ in merskih podatkov (X_i, Y_i) je kriterij najmanjših kvadratov, ki zahteva, da je vsota kvadratnih odstopanj $\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$ najmanjša možna.

Kazalec kvalitete regresijske zveze se imenuje determinacijski koeficient R^2 in podaja vrednost, kako dobro se regresijska zveza $\hat{Y} = a + bX$ prilega eksperimentalnim podatkom. Izračuna se na naslednji način:

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = \frac{\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = \frac{\text{pojasnjena variacija}}{\text{skupna variacija}}$$

Pri tem je:

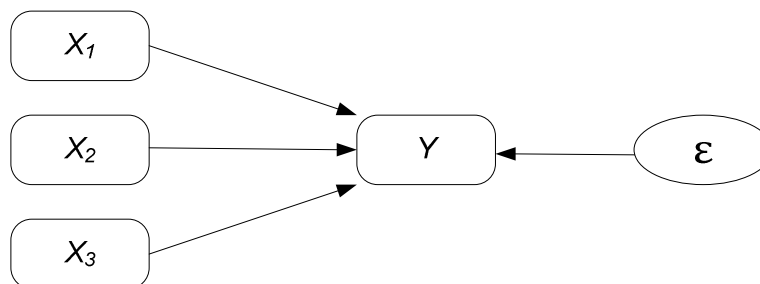
- $\sum_{i=1}^n (Y_i - \bar{Y})^2$ skupna variacija spremenljivke Y,
- $\sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$ pojasnjena variacija spremenljivke Y in
- $\sum_{i=1}^n (Y_i - \hat{Y}_i)^2 = \sum_{i=1}^n \varepsilon_i^2$ nepojasnjena variacija spremenljivke Y.

Vrednost R^2 je definirana na intervalu $[0,1]$. Pri tem na primer vrednost $R^2 = 0.9$ pomeni, da dana regresijska funkcija pojasnjuje 90% skupne variacije spremenljivke Y.

Multipla linearna regresija (*multiple linear regression*) predstavlja nadgradnjo bivariatne regresije, v kateri več neodvisnih spremenljivk napoveduje odvisno spremenljivko.

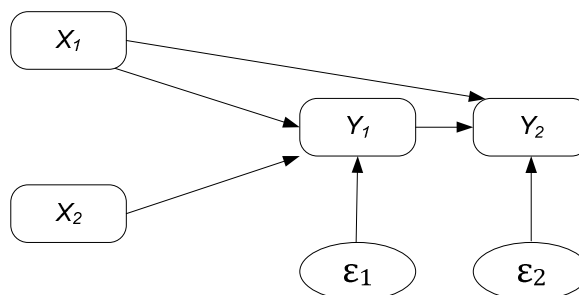
$$Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \varepsilon$$

V navedenih enačbah predstavljajo b_i regresijski koeficient, ki kaže, koliko je pričakovano povečanje odvisne spremenljivke Y, če se X_i poveča za enoto. Spremenljivka ε predstavlja ostanek oziroma napako (Slika 101).



SLIKA 101: GRAFIČNA PONAŽORITEV MULTIPLE REGRESIJE

Nadgradnjo multiple linearne regresije predstavlja analiza poti (*path analysis*), ki omogoča sočasno pojasnjevanje več odvisnih spremenljivk (Slika 102).



SLIKA 102: GRAFIČNA PONAŽORITEV ANALIZE POTI

 FAKTORSKA ANALIZA

Faktorska analiza je študija povezav med spremenljivkami, pri kateri poskušamo najti novo množico spremenljivk oziroma faktorjev (ki jih je manj kot merjenih spremenljivk), ki predstavljajo to, kar je skupnega opazovanim spremenljivkam. Faktorska analiza poskuša poenostaviti kompleksnost povezav med množico opazovanih spremenljivk z razkritjem skupnih razsežnosti ali faktorjev, ki omogočajo vpogled v osnovno strukturo podatkov. Splošni faktorski model lahko predstavimo kot zvezo med spremenljivkami $X_i(i=1,\dots,m)$, $F_r(r=1,\dots,k)$ in $E_i(i=1,\dots,m)$, kjer je:

$$X_i = \sum_{r=1}^k a_{ir}F_r + E_i$$

Pri tem so:

- X_i : merjene spremenljivke (empirični indikatorji),
- F_r : skupni faktorji,
- E_i : specifičen faktor, ki vpliva samo na X_i in
- a_{ir} je faktorska utež, ki kaže vpliv faktorja F_r na X_i .

Zgornji model lahko opišemo tudi v matrični obliki:

$$X = FA' + E$$

Na podlagi te enačbe in običajnih predpostavk faktorskega modela lahko izpeljemo naslednjo faktorsko enačbo:

$$\Sigma = AA' + \psi$$

V splošnem predstavlja Σ matriko varianc in kovarianc. Če so merjene spremenljivke standardizirane, je Σ korelacijska matrika.

Naloga faktorske analize je, da iz znanih elementov matrike varianc in kovarianc (ali korelacijske matrike) Σ , izračunamo neznane parametre faktorskega modela A (faktorske uteži) in specifične variance ψ . Za izvedbo faktorske analize se lahko uporabljajo različne metode. Pogosto uporabljana je metoda glavnih osi, ki rešuje problem faktorske analize iterativno.

Za uspešno izvedbo faktorske analize morajo biti izpolnjeni naslednji pogoji: (1) med spremenljivkami morajo obstajati povezave, (2) spremenljivke so med sabo povezane linearno, (3) normalnost porazdelitve spremenljivk, če želimo izvesti teste statistične značilnosti, in (4) število spremenljivk mora biti manjše od velikosti vzorca.

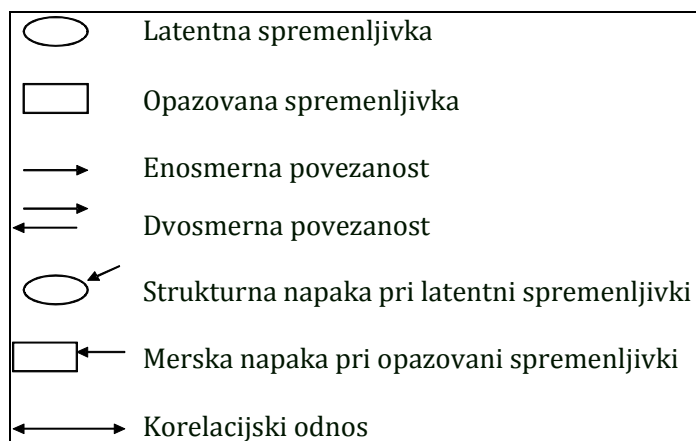
Faktorska analiza je uporabna za: (1) identifikacijo vzorcev v povezavah med spremenljivkami, (2) skrčenje večjega števila spremenljivk na manjše število faktorjev in (3) potrjevanje hipotez o predefinirani strukturi faktorjev in pripadajočih indikatorjev (*confirmatory factor analysis*).

MODELI STRUKTURNIH ENAČB

Modeliranje strukturalnih enačb (*structural equation models*) predstavlja družino statističnih tehnik, ki vključujejo analizo poti in faktorsko analizo. Modeli strukturalnih enačb se lahko uporabljajo za naslednje namene:

- Strikten potrditveni pristop, kjer ugotavljamo, ali se vzorec varianc in kovarianc v empiričnih podatkih ujema s strukturalnim modelom, ki ga je definiral raziskovalec.
- Pristop alternativnih modelov, kjer raziskovalec ugotavlja, kateri izmed alternativnih modelov izkazuje najboljše ujemanje s podatki.
- Pristop razvoja modelov je najbolj pogost pristop. V takšnem pristopu raziskovalec analizira strukturalni model in na osnovi rezultatov predlaga izboljššan model.

Modeli strukturalnih enačb so najpogosteje ponazorjeni v grafični obliki. Simboli, ki jih pri tem uporabljamo, so naslednji (Slika 103):



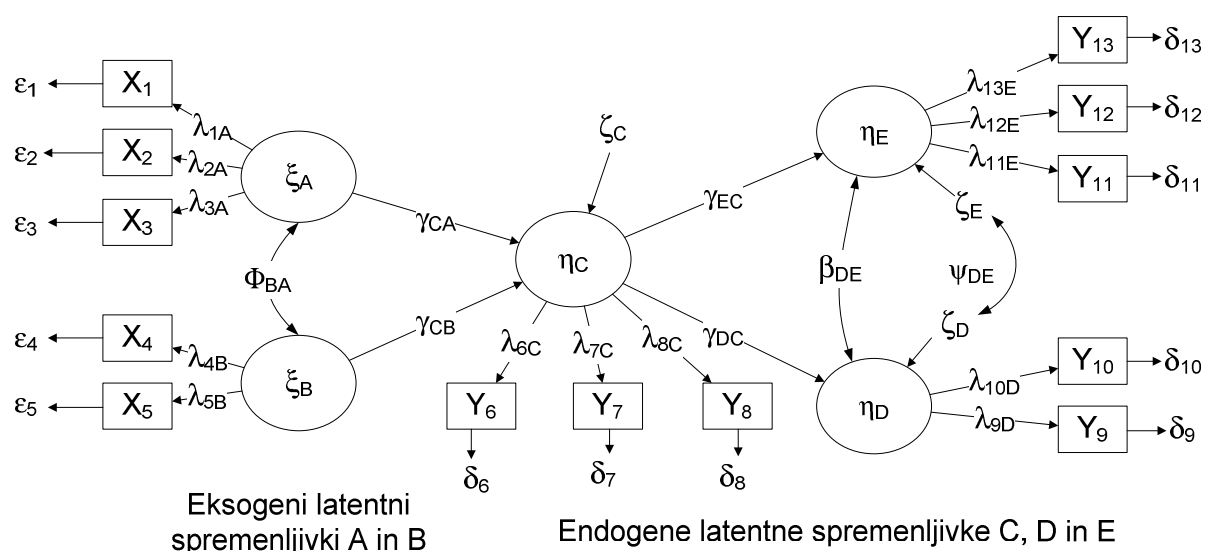
SLIKA 103: SIMBOLI, KI SE UPORABLJAJO V MODELIH STRUKTURNIH ENAČB

Pri tem je notacija označevanja spremenljivk in zvez v modelu naslednja (Tabela 55)¹²³:

¹²³ Pri označevanju strukturalnih modelov obstajajo, med avtorji manjša odstopanja.

TABELA 55: OZNAČEVANJE STRUKTURNIH MODELOV

Simbol	Pomen
X_i	Neodvisne merske spremenljivke (indikatorji)
Y_i	Odvisne merske spremenljivke (indikatorji)
ξ (Ksi)	Eksogeni latentni konstrukti (spremenljivke)
η (eta)	Endogeni latentni konstrukti (spremenljivke)
γ (gama)	Zveze med latentnimi konstrukti
λ (lambda)	Zveze med merskimi spremenljivkami in latentnimi konstrukti (<i>item loading</i>)
Φ (fi)	Korelacije med eksogenimi latentnimi konstrukti
β (beta)	Korelacije med endogenimi latentnimi konstrukti
δ (delta)	Napake merjenja pri odvisnih spremenljivkah
ε (epsilon)	Napake merjenja pri neodvisnih spremenljivkah
ζ (zeta)	Rezidualna varianca v strukturi
Ψ (psi)	Korelacija med rezidualnimi variancami v strukturi



SLIKA 104: PRIKAZ SINTAKSE STRUKTURNEGA MODELA (GEFEN, STRAUB, & BORDEAU 2000)

Proces modeliranja strukturnih enačb je sestavljen iz dveh osnovnih korakov: validacije merljivega modela in validacije strukturnega modela.

Merljivi model je sestavni del strukturnega modela, ki preučuje latentne spremenljivke in pripadajoče indikatorje. Merljivi model oblikujemo iz strukturnega modela na naslednji način:

- Vse latentne spremenljivke med sabo povežemo s kovariancami (obojesmerne povezave).

- Latentne spremenljivke in pripadajoče indikatorje povežemo s kavzalnimi povezavami (enosmerne povezave).
- Z enosmernimi povezavami prav tako povežemo napake merjenja in pripadajoče spremenljivke.

Merljivi model se pogosto uporablja kot ničelni model. Validacija merljivega modela temelji na potrditveni faktorski analizi, medtem ko je validacija strukturnega modela izvedena z analizo poti na latentnih spremenljivkah. Strukturni model analiziramo, ko dobimo zadovoljive rezultate metrik prileganja merljivega modela.

METRIKE PRILEGANJA

V okviru raziskav disertacije (analiziranje merljivih in strukturnih modelov) smo uporabili naslednje metrike prileganja.

HI-KVADRAT

Hi-kvadrat (*chi-square*) se imenuje tudi diskrepanca ali funkcija diskrepance in se izračuna kot razmerje vsote kvadratov razlik med opazovanimi (O) in pričakovanimi izidi (E) ter pričakovanimi izidi (kot je predpostavljeno v ničelni hipotezi):

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Vrednost hi-kvadrat ne sme biti signifikantna, če želimo dobro ujemanje modela s podatki. Zato se hi-kvadrat imenuje tudi »metrika slabega ujemanja«, pri kateri majhne vrednosti kažejo na signifikantno slabo ujemanje strukture kovarianc v modelu in matrike kovarianc v opazovanih podatkih. Če je hi-kvadrat modela manjši od 0.05, naj bi se model zavrgel.

Metrika hi-kvadrat ima tudi pomanjkljivosti, ki lahko vodijo k neutemeljenemu zavračanju modela. Zato lahko ob velikih vzorcih ($n > 200$) in dobrih drugih metrikah prileganja model sprejmemo kljub signifikantnosti hi-kvadrata.

Izboljšava hi-kvadrat je metrika relativen hi-kvadrat (χ^2/n), ki hi-kvadrat deli s stopnjami svobode (*degrees of freedom*). Model, ki ga sprejmemo, mora imeti relativen hi-kvadrat manjši od 3.

GOODNESS-OF-FIT INDEX

Goodness-of-fit index (GFI) se izračuna naslednje:

$$GFI = 1 - \frac{(\chi^2 \text{ za privzet model})}{(\chi^2 \text{ za ničelni model})}$$

Vrednosti GFI se nahajajo na intervali [0,1], pri tem naj bo vrednost GFI za sprejeti model vsaj 0.9. Vrednost GFI je občutljiva za velikost vzorca, saj se z večanjem vzorca večja tudi GFI.

ADJUSTED GOODNESS-OF-FIT

Adjusted goodness-of-fit (AGFI) je izpeljan iz GFI na naslednji način:

$$AGFI = 1 - \left[(1 - GFI) \frac{p(p + 1)}{2df} \right]$$

Pri tem pomenita »p« število parametrov in »df« stopnjo svobode¹²⁴. Vrednost AGFI za sprejete modela naj bo večja od 0.80. Podobno kot GFI se tudi AGFI povečuje z večanjem velikosti vzorca.

COMPARATIVE FIT INDEX

Comparative fit index (CFI) primerja ujemanje danega modela z ujemanjem ničelnega modela.

$$CFI = \frac{1 - \max(chisq - df, 0)}{(\max(chisq - df), (chisqn - dfn), 0)}$$

Pri tem pomenijo:

- *chisq* = hi-kvadrat danega modela,
- *chisqn* = hi-kvadrat ničelnega modela,
- *df* = stopnje svobode danega modela in
- *dfn* = stopnje svobode ničelnega modela.

Vrednosti CFI so definirane na intervalu [0,1], pri čemer so preferirane višje vrednosti. Tako so vrednosti CFI, ki so večje od 0.90, zadostne, da se model sprejme. Dobra lastnost metrike CFI je, da je zelo neobčutljiva za velikost vzorca.

¹²⁴ Število stopenj svobode (df) se izračuna kot število opazovanih primerov minus 1 (N-1).

 NORMED FIT INDEX

Normed fit index (NFI) se je razvil kot alternativa CFI:

$$NFI = \frac{chisqn - chisq}{chisqn}$$

Vrednosti NFI so definirane na intervalu [0,1], preferirane so višje vrednosti. NFI predstavlja delež izboljšanja ujemanja danega modela glede na ničelni model. Na primer, NFI = 0.5, pomeni, da je ujemanje danega modela za 50% boljše od ničelnega modela. Za sprejetje danega modela se zahtevajo vrednosti, ki so večje od 0.90.

 TUCKER LEWIS INDEX

Tucker lewis coefficient (TLI) ali *Non-normed fit index* (NNFI) je podoben NFI, ampak »kaznuje« kompleksnost modela. TLI se izračuna naslednje:

$$TLI = \frac{\frac{chisqn}{dfn} - \frac{chisq}{df}}{\frac{chisq}{df} - 1}$$

Vrednosti TLI lahko presežejo interval [0,1]. TLI blizu 1 pomeni dobro ujemanje danega modela. Vrednosti TLI pod 0.90 zahtevajo respecifikacijo modela. Dobra lastnost TLI je majhna občutljivost za velikost vzorca.

 ROOT MEAN SQUARE RESIDUALS

Root mean square residuals (RMR ali RMSR) predstavlja srednjo absolutno vrednost rezidualov (ostankov) matrike kovarianc. Metrika je omejena na spodnjem intervalu, kjer pomenijo vrednosti, bližje 0, boljše ujemanje danega modela. Priporočljiva vrednost RMSR za sprejetje modela je manjša od 0.10.

ROOT MEAN SQUARE ERROR OF APPROXIMATION

Root mean square error of approximation (RMSEA) je priljubljena metrika ujemanja, ki ne zahteva ničelnega modela. Izračuna se na naslednji način:

$$RMSEA = \frac{chisq}{(n-1)df} - \frac{df}{(n-1)df} \times 0.5$$

Ujemanje danega modela je zadostno, če so vrednosti RMSEA manjše od 0.8. Dobra lastnost metrike je relativna neobčutljivost za velikost vzorca. Slabost RMSEA je, da vrednoti kompleksnost modela na osnovi stopenj svobode (df).

PRILOGA D: REZULTATI STATISTIČNE ANALIZE PODATKOV

V tej prilogi so podani podrobnejši rezultati statistične analize podatkov. Tabele in preostali izpisi so povzeti iz okolij SPSS in SPSS/AMOS.

SEZNAM VREDNOTENIH OGRODIJ

V spodnji tabeli (Tabela 56) so podana ogrodja, ki so jih respondenti vrednotili v anketi disertacije. Številčne vrednosti predstavljajo, koliko respondentov je vrednotilo dano ogrodje.

TABELA 56: SEZNAM VREDNOTENIH OGRODIJ

	Frequency	Percent	Valid Percent	Cumulative Percent
Valid Spring	23	5,9	5,9	5,9
.NET	20	5,1	5,1	11,0
Struts	15	3,8	3,8	14,8
Eclipse	9	2,3	2,3	17,1
Django	7	1,8	1,8	18,9
Hibernate	6	1,5	1,5	20,5
Wicket	5	1,3	1,3	21,7
Qt	4	1,0	1,0	22,8
Ruby on Rails	4	1,0	1,0	23,8
Symfony	4	1,0	1,0	24,8
TurboGears	4	1,0	1,0	25,8
/	3	,8	,8	26,6
ARéVi	3	,8	,8	27,4
Code Igniter	3	,8	,8	28,1
elml	3	,8	,8	28,9
Stripes	3	,8	,8	29,7
Zend Framework	3	,8	,8	30,4
ACE - Adaptive Communication Environment	2	,5	,5	30,9
boost	2	,5	,5	31,5
Boost C++ Libraries	2	,5	,5	32,0
coid	2	,5	,5	32,5
CS-Framework	2	,5	,5	33,0
Domingo	2	,5	,5	33,5
EL4Net	2	,5	,5	34,0
JBoss	2	,5	,5	34,5
jpos	2	,5	,5	35,0
JUnit	2	,5	,5	35,5
Lazarus LCL	2	,5	,5	36,1
OSGi	2	,5	,5	36,6
PowerPlant	2	,5	,5	37,1

Python Wikipedia Robot Framework	2	,5	,5	37,6
QuickFrame	2	,5	,5	38,1
Spago	2	,5	,5	38,6
wxWidgets	2	,5	,5	39,1
'edusoft' framework planned and developed mainly for educational management purpose	1	,3	,3	39,4
AjaxCore	1	,3	,3	39,6
AjaxWS	1	,3	,3	39,9
Alias Wavefront Maya User Interface	1	,3	,3	40,2
Ammentos	1	,3	,3	40,4
Amplify MDA	1	,3	,3	40,7
animeka framework	1	,3	,3	40,9
ANTLR	1	,3	,3	41,2
Aranea	1	,3	,3	41,4
Ark Web Application Framework	1	,3	,3	41,7
Ascape	1	,3	,3	41,9
AutoPatch	1	,3	,3	42,2
BananaKernel	1	,3	,3	42,5
bitweaver	1	,3	,3	42,7
BloCxx	1	,3	,3	43,0
Bloom: rule-based web loom	1	,3	,3	43,2
Blue Havic	1	,3	,3	43,5
BNMAPI	1	,3	,3	43,7
Booguu	1	,3	,3	44,0
Brico	1	,3	,3	44,2
C++ Portable Components (POCO)	1	,3	,3	44,5
C++ STL	1	,3	,3	44,8
CakePHP, Zend, seagull	1	,3	,3	45,0
canoe	1	,3	,3	45,3
Castle	1	,3	,3	45,5
Cinco framework	1	,3	,3	45,8
Clairv	1	,3	,3	46,0
ClearWork	1	,3	,3	46,3
Clutch	1	,3	,3	46,5
CMV Framework	1	,3	,3	46,8
Cocuyo Compiler Framework	1	,3	,3	47,1
ColdFrame	1	,3	,3	47,3
COM	1	,3	,3	47,6
Componentes ACBr	1	,3	,3	47,8
COMPOST	1	,3	,3	48,1
CoreLibrary	1	,3	,3	48,3
cppCommon (internal to my company)	1	,3	,3	48,6
CPPGPGPU Library	1	,3	,3	48,8
Credit Card Processing Framework	1	,3	,3	49,1
crGUI PHP Framework	1	,3	,3	49,4

CrystalSpace	1	,3	,3	49,6
cs-arrayToPath	1	,3	,3	49,9
Cuttle MVC Framework	1	,3	,3	50,1
CyberCore	1	,3	,3	50,4
CyberLink for C	1	,3	,3	50,6
dbscript	1	,3	,3	50,9
Dimensional Analysis	1	,3	,3	51,2
DimensioneX	1	,3	,3	51,4
drupal	1	,3	,3	51,7
ECForms	1	,3	,3	51,9
EIS Integration Server	1	,3	,3	52,2
EMF	1	,3	,3	52,4
EnerGINE	1	,3	,3	52,7
Enkel	1	,3	,3	52,9
ennea	1	,3	,3	53,2
EnRiche	1	,3	,3	53,5
Enunciate	1	,3	,3	53,7
Enveria Development IDE	1	,3	,3	54,0
ePlatform	1	,3	,3	54,2
ERP Application Framework for Delphi	1	,3	,3	54,5
Esperantus	1	,3	,3	54,7
Eurocenter Database Framework	1	,3	,3	55,0
Exception Management Framework	1	,3	,3	55,2
Fisterra	1	,3	,3	55,5
Fit	1	,3	,3	55,8
flex	1	,3	,3	56,0
Freevo	1	,3	,3	56,3
GAIA - Genetic Algorithm Infrastructure for Applications	1	,3	,3	56,5
glfw	1	,3	,3	56,8
Globus Toolkit	1	,3	,3	57,0
GPLAB	1	,3	,3	57,3
GPMI	1	,3	,3	57,5
Graphical Eclipse Framework	1	,3	,3	57,8
grinda	1	,3	,3	58,1
Grizzly (grizzly.dev.java.net)	1	,3	,3	58,3
GWT Pleso Framework	1	,3	,3	58,6
harmoni application framework	1	,3	,3	58,8
Heavenize	1	,3	,3	59,1
http://sourceforge.net/projects/omodel	1	,3	,3	59,3
Input Validation Framework for .NET	1	,3	,3	59,6
Irrlicht Engine	1	,3	,3	59,8
J2EE and custom	1	,3	,3	60,1
J2SDK	1	,3	,3	60,4
JAT Java Application Template	1	,3	,3	60,6
JAVA	1	,3	,3	60,9

Java Collections	1	,3	,3	61,1
Java standard API / class library	1	,3	,3	61,4
javassist	1	,3	,3	61,6
JBJF	1	,3	,3	61,9
JGenesis	1	,3	,3	62,1
JOMM	1	,3	,3	62,4
Joss-Seam	1	,3	,3	62,7
JProactor	1	,3	,3	62,9
JPS	1	,3	,3	63,2
JSambuca	1	,3	,3	63,4
JSBSim	1	,3	,3	63,7
JSDC	1	,3	,3	63,9
JSF	1	,3	,3	64,2
JSSF	1	,3	,3	64,5
jump	1	,3	,3	64,7
jWebApp	1	,3	,3	65,0
JX Application Framework	1	,3	,3	65,2
jxta	1	,3	,3	65,5
jzeno	1	,3	,3	65,7
KAGEfx	1	,3	,3	66,0
LainSoft Forge	1	,3	,3	66,2
libblackboard	1	,3	,3	66,5
libd20	1	,3	,3	66,8
libodbc++	1	,3	,3	67,0
libs11n	1	,3	,3	67,3
LIMB	1	,3	,3	67,5
linux os	1	,3	,3	67,8
Lusca	1	,3	,3	68,0
LWJGL	1	,3	,3	68,3
Maven 2.x	1	,3	,3	68,5
Mentawai	1	,3	,3	68,8
MFC	1	,3	,3	69,1
Microsoft Enterprise Library	1	,3	,3	69,3
Middleheaven	1	,3	,3	69,6
Milos Solution Platform	1	,3	,3	69,8
Modular Audio Recognition Framework	1	,3	,3	70,1
moltools	1	,3	,3	70,3
mootools	1	,3	,3	70,6
MPENV	1	,3	,3	70,8
ms FRAMEWORK 1.1	1	,3	,3	71,1
Mule	1	,3	,3	71,4
Multiform	1	,3	,3	71,6
MWT	1	,3	,3	71,9
myfuses	1	,3	,3	72,1
MyGWT	1	,3	,3	72,4
Nebula Device 2	1	,3	,3	72,6
Neobase PHP Framework	1	,3	,3	72,9
NetBeans RCP	1	,3	,3	73,1
NetJsWire	1	,3	,3	73,4
NEW - New Easy Webpaging	1	,3	,3	73,7

NextStep	1	,3	,3	73,9
NMEA Library	1	,3	,3	74,2
NuSphere PHPED Professional	1	,3	,3	74,4
OAProject	1	,3	,3	74,7
Object Relational Mapper (ORM)	1	,3	,3	74,9
OFBiz framework	1	,3	,3	75,2
Ogre	1	,3	,3	75,4
oolime	1	,3	,3	75,7
Open WS-Eventing	1	,3	,3	76,0
OpenCV	1	,3	,3	76,2
OpenTranquera	1	,3	,3	76,5
OpenXava	1	,3	,3	76,7
Oracle JDeveloper	1	,3	,3	77,0
OSICS	1	,3	,3	77,2
oxygen io	1	,3	,3	77,5
Panda3D	1	,3	,3	77,7
PHP RAD Controls	1	,3	,3	78,0
phpbike	1	,3	,3	78,3
phpCore	1	,3	,3	78,5
Picocontainer	1	,3	,3	78,8
PLCS/DEX	1	,3	,3	79,0
PocketGraphics	1	,3	,3	79,3
POCO C++ Libraries	1	,3	,3	79,5
POG	1	,3	,3	79,8
portalServices (advanced rainbow variant)	1	,3	,3	80,1
powermops	1	,3	,3	80,3
protojax	1	,3	,3	80,6
pure data	1	,3	,3	80,8
PyDADL	1	,3	,3	81,1
pysces	1	,3	,3	81,3
python twisted	1	,3	,3	81,6
pywikipediabot	1	,3	,3	81,8
Qcodo	1	,3	,3	82,1
qooxdoo	1	,3	,3	82,4
Query Object Framework	1	,3	,3	82,6
Quick View	1	,3	,3	82,9
Quinoa Framework	1	,3	,3	83,1
RADFrame for PHP5	1	,3	,3	83,4
rails	1	,3	,3	83,6
RaPiD Framework	1	,3	,3	83,9
Relational Analysis Toolkit	1	,3	,3	84,1
RiCPP	1	,3	,3	84,4
Riser	1	,3	,3	84,7
Roma Framework	1	,3	,3	84,9
rt-scheduler	1	,3	,3	85,2
RTSJ	1	,3	,3	85,4
RUNES	1	,3	,3	85,7
SAGA GIS	1	,3	,3	85,9
SAM Kernel	1	,3	,3	86,2
SAXml with JSAX	1	,3	,3	86,4

screen + vi + cscope	1	,3	,3	86,7
Scrolling Game Development Kit 2	1	,3	,3	87,0
Seagull	1	,3	,3	87,2
Seam	1	,3	,3	87,5
SenseRelate	1	,3	,3	87,7
sesame	1	,3	,3	88,0
SIL: Standard Interface Library	1	,3	,3	88,2
sil++	1	,3	,3	88,5
SimpleORM	1	,3	,3	88,7
SiteVerse	1	,3	,3	89,0
smith	1	,3	,3	89,3
SoEasyPHP	1	,3	,3	89,5
SOOGL	1	,3	,3	89,8
Sourceforge	1	,3	,3	90,0
Spellbook	1	,3	,3	90,3
Spirit	1	,3	,3	90,5
Sprit 92	1	,3	,3	90,8
Starthrower Utilities	1	,3	,3	91,0
Storyboard Framework - A Workflow design for WebObjects	1	,3	,3	91,3
Stratos Framework	1	,3	,3	91,6
SUnit	1	,3	,3	91,8
Swing	1	,3	,3	92,1
SwingBean	1	,3	,3	92,3
SWT	1	,3	,3	92,6
Tangra framewrok for PHP	1	,3	,3	92,8
Tank Battles	1	,3	,3	93,1
the boost c++ libraries	1	,3	,3	93,4
The evolution framework for finacial products	1	,3	,3	93,6
tmk	1	,3	,3	93,9
TopEngine	1	,3	,3	94,1
Torque Game Builder	1	,3	,3	94,4
Trolltech Qt	1	,3	,3	94,6
TuxFrw	1	,3	,3	94,9
Unidraw	1	,3	,3	95,1
unixODBC	1	,3	,3	95,4
View providers driven applications	1	,3	,3	95,7
Vigo:3D	1	,3	,3	95,9
web-based applications	1	,3	,3	96,2
wingS	1	,3	,3	96,4
wsmo4j	1	,3	,3	96,7
wxHarbour	1	,3	,3	96,9
wxPython	1	,3	,3	97,2
xajax	1	,3	,3	97,4
XAL Accelerator Framework	1	,3	,3	97,7
Xargon Library	1	,3	,3	98,0
Xcc XWeb	1	,3	,3	98,2
xholon	1	,3	,3	98,5

XML Variant Configuration	1	,3	,3	98,7
Language				
ZCA	1	,3	,3	99,0
zenphp	1	,3	,3	99,2
Zeus-Framework	1	,3	,3	99,5
ZNF	1	,3	,3	99,7
zope	1	,3	,3	100,0
Total	391	100,0	100,0	

PRIMERJAVA ZGODNJIH IN POZNIH RESPONDENTOV

V spodnji tabeli (Tabela 57) so prikazani rezultati T-testov za posamezna vprašanja (indikatorje) glede na zgodnje in pozne respondente. Pri nobeni izmed spremenljivk signifikantnost T-testa ni padla pod vrednost $p < 0.05$. Iz tega razloga smo pri vseh spremenljivkah sprejeli ničelno hipotezo (ni signifikantnih razlik med zgodnjimi in poznimi respondenti). Levenov test enakosti varianc je pokazal, da se pri štirih od skupno 40 spremenljivk varianca med zgodnjimi in poznimi respondenti signifikantno razlikuje ($p > 0.05$). V teh štirih primerih je izvedba T-testa tvegana.

TABELA 57: REZULTATI T-TESTA, KI JE BIL IZVEDEN NA NEODVISNIH VZORCIH

		Levene's Test for Equality of Variances		T-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
									Lower	Upper
FA1	Equal variances assumed	6,325	,012	1,122	328	,263	,191	,170	-,144	,526
	Equal variances not assumed			1,206	264,205	,229	,191	,158	-,121	,503
FA2	Equal variances assumed	1,451	,229	,444	328	,657	,082	,184	-,281	,444
	Equal variances not assumed			,456	234,356	,649	,082	,179	-,272	,435
FA3	Equal variances assumed	,089	,766	-,105	328	,916	-,018	,173	-,358	,322
	Equal variances not assumed			-,108	235,794	,914	-,018	,168	-,349	,312
FA4	Equal variances assumed	3,431	,065	-,027	328	,979	-,005	,171	-,340	,331
	Equal variances not assumed			-,028	245,233	,978	-,005	,163	-,326	,317
CFU1	Equal variances assumed	3,188	,075	,938	328	,349	,164	,174	-,179	,507
	Equal variances not assumed			,991	252,271	,323	,164	,165	-,162	,489
CFU2	Equal variances assumed	3,903	,049	,867	328	,386	,127	,147	-,161	,416
	Equal variances not assumed			,939	269,070	,349	,127	,136	-,140	,394
CFU3	Equal variances assumed	1,593	,208	1,161	328	,246	,186	,160	-,129	,502
	Equal variances not assumed			1,234	256,462	,218	,186	,151	-,111	,484
NB1	Equal variances assumed	,043	,835	,815	328	,416	,105	,128	-,148	,357
	Equal variances not assumed			,849	243,243	,397	,105	,123	-,138	,347

NB2	Equal variances assumed	2,884	,090	-,796	328	,427	-,105	,131	-,363	,154
	Equal variances not assumed			-,799	220,217	,425	-,105	,131	-,362	,153
NB3	Equal variances assumed	7,565	,006	-1,369	328	,172	-,241	,176	-,587	,105
	Equal variances not assumed			-1,291	187,220	,198	-,241	,187	-,609	,127
SAT1	Equal variances assumed	,003	,959	,800	328	,424	,091	,114	-,133	,314
	Equal variances not assumed			,846	252,612	,399	,091	,108	-,121	,303
SAT2	Equal variances assumed	5,511	,019	,943	328	,347	,136	,145	-,148	,421
	Equal variances not assumed			1,005	258,985	,316	,136	,136	-,131	,404
SAT3	Equal variances assumed	,934	,335	-,257	328	,798	-,032	,124	-,276	,212
	Equal variances not assumed			-,265	237,826	,791	-,032	,120	-,268	,205
CF1	Equal variances assumed	,080	,777	,244	328	,807	,041	,168	-,289	,371
	Equal variances not assumed			,239	207,363	,811	,041	,171	-,296	,378
CF2	Equal variances assumed	,098	,755	,368	328	,713	,050	,136	-,217	,317
	Equal variances not assumed			,367	216,124	,714	,050	,136	-,219	,319
CF3	Equal variances assumed	,240	,624	1,020	328	,309	,150	,147	-,139	,439
	Equal variances not assumed			1,019	217,736	,309	,150	,147	-,140	,440
UD1	Equal variances assumed	,035	,852	-1,033	328	,302	-,182	,176	-,528	,164
	Equal variances not assumed			-1,040	221,824	,300	-,182	,175	-,526	,163
UD2	Equal variances assumed	1,017	,314	,165	328	,869	,032	,193	-,348	,411
	Equal variances not assumed			,169	232,614	,866	,032	,188	-,339	,403
UD3	Equal variances assumed	,044	,835	-,167	328	,868	-,027	,164	-,349	,295
	Equal variances not assumed			-,166	215,360	,868	-,027	,165	-,352	,297
PO1	Equal variances assumed	,295	,587	,732	328	,465	,100	,137	-,169	,369
	Equal variances not assumed			,750	232,938	,454	,100	,133	-,163	,363
PO2	Equal variances assumed	,133	,716	,636	328	,525	,109	,172	-,229	,447
	Equal variances not assumed			,625	208,354	,533	,109	,175	-,235	,453
PO3	Equal variances assumed	,314	,576	,635	328	,526	,114	,179	-,239	,466
	Equal variances not assumed			,641	223,884	,522	,114	,177	-,236	,463
EF1	Equal variances assumed	,530	,467	-1,207	328	,228	-,218	,181	-,574	,137
	Equal variances not assumed			-1,186	208,174	,237	-,218	,184	-,581	,144
EF2	Equal variances assumed	,236	,628	-,107	328	,915	-,014	,127	-,264	,237
	Equal variances not assumed			-,106	212,799	,916	-,014	,128	-,267	,240
EF3	Equal variances assumed	1,808	,180	-1,344	328	,180	-,241	,179	-,593	,112
	Equal variances not assumed			-1,328	211,153	,186	-,241	,181	-,598	,117
IG1	Equal variances assumed	,405	,525	-,346	328	,730	-,064	,184	-,426	,298

	Equal variances not assumed			-,342	211,811	,733	-,064	,186	-,430	,303
IG2	Equal variances assumed	,001	,976	,074	328	,941	,014	,183	-,346	,374
	Equal variances not assumed			,074	215,306	,941	,014	,184	-,349	,376
IG3	Equal variances assumed	,438	,508	-,394	328	,694	-,077	,196	-,463	,309
	Equal variances not assumed			-,397	222,473	,692	-,077	,195	-,461	,307
TTF1	Equal variances assumed	,956	,329	-,310	328	,756	-,041	,132	-,300	,218
	Equal variances not assumed			-,307	211,669	,759	-,041	,133	-,304	,222
TTF2	Equal variances assumed	,001	,980	,634	328	,526	,082	,129	-,172	,336
	Equal variances not assumed			,614	200,256	,540	,082	,133	-,181	,345
TTF3	Equal variances assumed	2,018	,156	-1,226	328	,221	-,245	,200	-,639	,148
	Equal variances not assumed			-1,276	242,925	,203	-,245	,192	-,624	,133
FPEOU1	Equal variances assumed	2,009	,157	-,862	328	,389	-,118	,137	-,388	,151
	Equal variances not assumed			-,845	207,055	,399	-,118	,140	-,394	,157
FPEOU5	Equal variances assumed	,606	,437	-,294	328	,769	-,045	,154	-,349	,258
	Equal variances not assumed			-,298	226,451	,766	-,045	,152	-,346	,255
FPEOU2	Equal variances assumed	,822	,365	,451	328	,652	,059	,131	-,198	,317
	Equal variances not assumed			,441	204,993	,660	,059	,134	-,205	,323
FPEOU3	Equal variances assumed	,752	,387	-,247	328	,805	-,036	,147	-,326	,253
	Equal variances not assumed			-,255	237,674	,799	-,036	,142	-,317	,244
FPEOU4	Equal variances assumed	1,716	,191	-,850	328	,396	-,118	,139	-,392	,155
	Equal variances not assumed			-,837	209,938	,403	-,118	,141	-,396	,160
FPU1	Equal variances assumed	1,015	,314	,970	328	,333	,118	,122	-,121	,358
	Equal variances not assumed			1,027	253,618	,306	,118	,115	-,109	,345
FPU2	Equal variances assumed	2,235	,136	1,051	328	,294	,145	,138	-,127	,418
	Equal variances not assumed			1,119	257,798	,264	,145	,130	-,110	,401
FPU3	Equal variances assumed	,096	,757	1,021	328	,308	,136	,134	-,126	,399
	Equal variances not assumed			1,065	244,463	,288	,136	,128	-,116	,389
FPU4	Equal variances assumed	,571	,450	,730	328	,466	,091	,125	-,154	,336
	Equal variances not assumed			,776	257,172	,438	,091	,117	-,140	,322

LITERATURA

- Adele, G. 1984, SMALLTALK-80: the interactive programming environment Addison-Wesley Longman Publishing Co. Inc..
- Al-Gahtani, S. S. "System Characteristics, User Perceptions and Attitudes in the Prediction of Information Technology Acceptance: A Structural Equation Model", in Diffusion Interest Group in Information Technology 1998.
- Amar, L. & Coffey, J. 2005, "Measuring the benefits of software reuse - Examining three different approaches to software reuse", DR DOBBS JOURNAL, vol. 30, no. 6, pp. 73-76.
- Anderson, J. C. & Gerbing, D. W. 1988, "Structural Equation Modeling in Practice - A Review and Recommended 2-Step Approach", Psychological Bulletin, vol. 103, no. 3, pp. 411-423.
- Arevalo, G. & Mens, T. 2002, "Analysing object-oriented application frameworks using concept analysis," in Advances in Object-Oriented Information Systems, Springer, pp. 51-58.
- Batory, D., Cardone, R., & Smaragdakis, Y. 2000, "Object-Oriented Frameworks and Product Lines", in First Software Product Line Conference, P.Donohoe, ed., pp. 227-247.ce
- Behrens, S., Jamieson, K., Jones, D., & Cranston, M. 2005, "Predicting System Success using the Technology Acceptance Model: A Case Study ", in 16th Australasian Conference on Information Systems.
- Birrer, A. & Eggenschwiler, T. 1993, "Frameworks in the Financial Engineering Domain: An Experience Report", p. 21.
- Blake, I., Margrethe, H. O., & Jack, J. B. 1983, "The measurement of user information satisfaction", Communications of the Acm, vol. 26, no. 10, pp. 785-793.
- Bockle, G., Clements, P., McGregor, J. D., Muthig, D., & Schmid, K. 2004, "Calculating ROI for software product lines", Ieee Software, vol. 21, no. 3, p. 23-+.
- Boehm, B. 1999, "Managing software productivity and reuse", Computer, vol. 32, no. 9, pp. 111-113.
- Boehm, B. 1993, "Economic Analysis of Software Technology Investments," in Analytical Methods in Software Engineering Economics, T. R. Gullede & W. P. Hutzler, eds., Springer-Verlag, New York, pp. 1-37.
- Boehm, B. W., Brown, J. R., Lipow, M. L., Kaspar, J. R., McCleod, G. J., & Merrit, M. J. 1978, Characteristics of Software Quality, 1 edn, Elsevier.
- Booch, G. 1994, "Design an application framework", DR DOBBS JOURNAL, vol. 19, no. 2, pp. 24-32.

- Bosch, J. 2000, Design and use of software architectures: adopting and evolving a product-line approach ACM Press/Addison-Wesley Publishing Co..
- Bosch, J., Molin, P., Mattsson, M., & Bengtsson, P. 1997, "Object-Oriented Frameworks - Problems & Experiences," in Building Application Frameworks: Object Oriented Foundations of Framework Design, Fayad, Schmidt, & Johnson, eds., Wiley & Sons.
- Chase, L. & Alvarez, J. 2000, "Internet research: The role of the focus group", Library & Information Science Research, vol. 22, no. 4, pp. 357-369.
- Chau, P. Y. K. 1996, "An empirical investigation on factors affecting the acceptance of CASE by systems developers", Information & Management, vol. 30, no. 6, pp. 269-280.
- Churchill, G. A. 1979, "Paradigm for Developing Better Measures of Marketing Constructs", Journal of Marketing Research, vol. 16, no. 1, pp. 64-73.
- Cote, M.-A., Suryn, W., & Georgiadou, E. 2006, "Software Quality Model Requirements for Software Quality Engineering", in SOFTWARE QUALITY MANAGEMENT - INTERNATIONAL CONFERENCE, Southampton Solent University, pp. 31-50.
- Cunningham, H. C., Liu, Y., & Zhang, C. H. 2006, "Using classic problems to teach Java framework design", Science of Computer Programming, vol. 59, no. 1-2, pp. 147-169.
- Davis, F. D. 1989, "Perceived Usefulness, Perceived Ease Of Use, And User Acceptance Of Information Technology", MIS Quarterly, vol. 13, no. 3, pp. 318-331.
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. 1989, "User Acceptance of Computer-Technology - A Comparison of 2 Theoretical-Models", Management Science, vol. 35, no. 8, pp. 982-1003.
- DeLone, W. H. & Mclean, E. R. 2002, "Information Systems Success Revisited", in 35th Hawaii International Conference on System Sciences, IEEE Computer Society.
- DeLone, W. H. & Mclean, E. R. 1992, "Information Systems Success: The Quest for the Dependent Variable", Information Systems research, vol. 3, no. 1, pp. 60-96.
- Deursen, A. V. 1997, "Domain - specific languages versus object-oriented frameworks: A financial engineering case study", in STJA' 97, Ilmenau Technical University, pp. 35-39.
- Dillon, A. & Morris, M. G. 1996, "User acceptance of information technology: Theories and models", Annual Review of Information Science and Technology, vol. 31, pp. 3-32.
- Dishaw, M. T. & Strong, D. M. 1999, "Extending the technology acceptance model with task-technology fit constructs", Information & Management, vol. 36, no. 1, pp. 9-21.
- Čagran, B. & Pšunder, M. 2003, Priročnik za izdelavo diplomskega dela Pedagoška fakulteta.
- Fayad, M. & Hamu, D. 2000, "Enterprise frameworks: guidelines for selection", ACM Computing Survey, vol. 32, no. 1, p. 4.

- Fayad, M. E. & Schmidt, D. C. 1997, "Object-oriented application frameworks", *Communications of the Acm*, vol. 40, no. 10, pp. 32-38.
- Fenton, N. E. & Pfleeger, S. L. 1998, *Software Metrics: A Rigorous and Practical Approach* PWS Publishing Co., Boston, MA, USA.
- Fishbein, M. & Ajzen, I. 1975, *Belief, Attitude, Intention, and Behavior. An Introduction to Theory and Research* Addison-Wesley.
- Fontoura, M. F. 1999, "Object-Oriented Application Frameworks: the Untold Story", in *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'99)*.
- Fornell, C. & Larcker, D. F. 1981, "Evaluating Structural Equation Models with Unobservable Variables and Measurement Error", *Journal of Marketing Research*, vol. 18, no. 1, pp. 39-50.
- Frakes, W. B. & Kang, K. 2005, "Software reuse research: Status and future", *Ieee Transactions on Software Engineering*, vol. 31, no. 7, pp. 529-536.
- Freimut, B., Punter, T., Biffel, S., & Ciolkowski, M. 2002, *State-of-the-Art in Empirical Studies*, Virtuelles Software Engineering Kompetenzzentrum.
- Froehlich, G., Hoover, J., Liu, L., & Sorenson, P. 1998, "Designing object-oriented frameworks," in *CRC Handbook of Object Technology*, CRC Press.
- Fujiwara, H., Kusumoto, S., Inoue, K., Suzuki, A., Ootsubo, T., & Yuura, K. 2003, "Case studies to evaluate a domain specific application framework based on complexity and functionality metrics", *Information and Software Technology*, vol. 45, no. 1, pp. 43-49.
- Gallego, M. D., Luna, P., & Bueno, S. 2008, "User acceptance model of open source software", *Computers in Human Behavior*, vol. 24, no. 5, pp. 2199-2216.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. 1995, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1 edn, Addison-Wesley Professional.
- Garrity, E. J., Glassberg, B., Kim, Y. J., Sanders, G. L., & Shin, S. K. 2005, "An experimental investigation of web-based information systems success in the context of electronic commerce", *Decision Support Systems*, vol. 39, no. 3, pp. 485-503.
- Garson, D. G. *Structural Equation Modeling, Statnotes: Topics in Multivariate Analysis*. <http://www2.chass.ncsu.edu/garson/pa765/structur.htm> . 2008.
Ref Type: Electronic Citation
- Gefen, D., Straub, D. W., & Bordeau, M.-C. 2000, "Structural Equation Modeling and Regression: Guidelines for Research and Practice", *Communications of AIS*, vol. 4, no. 7, pp. 1-70.
- Glass, R. L., Ramesh, V., & Vessey, I. 2004, "An analysis of research in computing disciplines", *Communications of the Acm*, vol. 47, no. 6, pp. 89-94.

- Goodhue, D. L. & Thompson, R. L. 1995, "Task-Technology Fit and Individual-Performance", *MIS Quarterly*, vol. 19, no. 2, pp. 213-236.
- Grimm, L. G. & Yarnold, P. R. 2004, *Reading and understanding multivariate statistics*, 9th printing edn, Washington (D.C.) : American Psychological Association.
- Guéhéneuc, Y.-G., Guyomarc'h, J.-Y., Khosravi, K., & Sahraoui, H. 2006, "Design Patterns as Laws of Quality," in *Object-Oriented Design Knowledge: Principles, Heuristics, and Best Practices*, IGI Publishing, pp. 105-142.
- Guido, C., Francesco, B., & Andrea, V. 1997, "The evaluation of framework reusability", *SIGAPP Appl.Comput.Rev.*, vol. 5, no. 2, pp. 21-27.
- Hallsteinsen, S. & Paci, M. 1997, *Experiences in software evolution and reuse: twelve real world projects* Springer, Berlin.
- Hong, S. J., Thong, J. Y. L., & Tam, K. Y. 2006, "Understanding continued information technology usage behavior: A comparison of three models in the context of mobile internet", *Decision Support Systems*, vol. 42, no. 3, pp. 1819-1834.
- Hong, W. Y., Thong, J. Y. L., Wong, W. M., & Tam, K. Y. 2001, "Determinants of user acceptance of digital libraries: An empirical examination of individual differences and system characteristics", *Journal of Management Information Systems*, vol. 18, no. 3, pp. 97-124.
- IEEE. IEEE standard glossary of software engineering terminology. 10-12-1990. 1990.
Ref Type: Generic
- ISO 9126. ISO/IEC TR 9126-Software engineering, product quality, quality model. 2001.
International Organization for Standardization.
Ref Type: Generic
- Ivanko, Š. 2007, *Raziskovanje in pisanje del : metodologija in tehnologija raziskovanja in pisanja strokovnih in znanstvenih del*, 1 edn, Kamnik : Cubus image, 2007.
- Johnson, R. E. 1992, "Documenting Frameworks Using Patterns", *Sigplan Notices*, vol. 27, no. 10, pp. 63-76.
- Johnson, R. E. 2005, "J2EE development frameworks", *Computer*, vol. 38, no. 1, pp. 107-110.
- Johnson, R. E. 1997, "Frameworks = (components + patterns)", *Commun.ACM*, vol. 40, no. 10, pp. 39-42.
- Johnson, R. E. & Foote, B. 1988, "Designing Reusable Classes", *Journal of Object-Oriented Programming*, vol. 1, no. 2, p. 22-&.
- Johnston, R. & Shanks, G. *Research Methods in Information Systems*. 2003.
Ref Type: Generic

- Jung, H. W., Kim, S. G., & Chung, C. S. 2004, "Measuring software product quality: A survey of ISO/IEC 9126", *Ieee Software*, vol. 21, no. 5, p. 88-+.
- Keil, M., Beranek, P. M., & Konsynski, B. R. 1995, "Usefulness and Ease of Use - Field-Study Evidence Regarding Task Considerations", *Decision Support Systems*, vol. 13, no. 1, pp. 75-91.
- King, W. R. & He, J. 2006, "A meta-analysis of the technology acceptance model", *Information & Management*, vol. 43, no. 6, pp. 740-755.
- Kitchenham, B. & Pfleeger, S. L. 1996, "Software quality: The elusive target", *Ieee Software*, vol. 13, no. 1, p. 12-&.
- Klopping, I. M. & McKinney, E. 2004, "Extending the Technology Acceptance Model and the Task-Technology Fit Model to Consumer E-Commerce", *Information Technology, Learning and Performance Journal*, vol. 22, no. 1.
- Kontio, J., Chen, S.-F., Limperos, K., Tesoriero, R., Caldiera, G., & Deutsch, M. 1995, "A COTS Selection Method and Experiences of Its Use".
- Krajnc, A. 2006, *Dokumentiranje ogrodij informacijskih rešitev*, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.
- Krajnc, A. & Heričko, M. "Classification of Object-Oriented Frameworks", in *EUROCON 2003*.
- Krajnc, A. & Heričko, M. 2004, "Vloga ogrodij pri razvoju sodobnih informacijskih rešitev", *Uporabna informatika*, vol. 12, no. 2, pp. 68-79.
- Krajnc, A., Petr, C., Kos G., & Štok B. "Razvojna okolja kot ogrodja", in *Objektna tehnologija v Sloveniji - OTS'2005*, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, pp. 59-71.
- Kurt, J. S. 1986, "Object-oriented programming for the Macintosh", Hayden Books.
- Lajoie, R. & Keller, R. K. 1994, "Design and Reuse in Object-Oriented Frameworks: Patterns, Contracts, and Motifs in Concert", in *ACFAS Montreal Canada*.
- Landin, N. & Niklasson, A. 1995, *Development of Object-Oriented Frameworks*, Lund Institute of Technology, Lund University, Department of Communication Systems.
- Langelett, G. 2003, "The Qualitative Tradition: A Complimentary Paradigm for Research in Economic Education", *SSRN eLibrary*.
- Leach, R. J. 1996, *Software Reuse: Methods, Models and Costs* McGraw-Hill, Inc..
- Lederer, A. L., Maupin, D. J., Sena, M. P., & Zhuang, Y. 2000, "The technology acceptance model and the World Wide Web", *Decision Support Systems*, vol. 29, no. 3, pp. 269-282.
- Legris, P., Ingham, J., & Collerette, P. 2003, "Why do people use information technology? A critical review of the technology acceptance model", *Information & Management*, vol. 40, no. 3, pp. 191-204.

- Lewis, J. A. 1990, "An experiment to determine software reusability factors (abstract)", ACM, p. 405.
- Mamrak, S. A. & Sinha, S. 1999, "A case study: Productivity and quality gains using an object-oriented framework", *Software-Practice & Experience*, vol. 29, no. 6, pp. 501-518.
- Manolescu, D., Noble, J., & Voelter, M. 2006, "Patterns for Successful Object-oriented Framework Development," in *Pattern Languages of Program Design 5*, 1st edn, Addison Wesley Professional, pp. 401-431.
- Mason, R. O. 1978, "Measuring Information Output: A Communications System Approach", *Information and Management*, vol. 1, no. 5, pp. 219-234.
- Mathias, I., de Oliveira, T. C., & de Lucena, C. J. P. 2004, "A framework instantiation approach based on the Features Model", *Journal of Systems and Software*, vol. 73, no. 2, pp. 333-349.
- Mattsson, M. 1996, *Object-oriented Frameworks, A survey of methodological issues*, Lund University - Department of Computer Science.
- Mattsson, M. 2000, *Evolution And Composition Of Object-Oriented Frameworks*, Department of Software Engineering and Computer Science - University of Karlskrona/Ronneby.
- Mattsson, M. & Bosch, J. "Characterizing Stability in Evolving Frameworks", in *Proceedings of TOOLS Europe 99*, pp. 118-130.
- McBurney, D. H. & White, T. L. 2003, *Research Methods*, 6 edn, Wadsworth Publishing .
- McCall, J. A., Richards, P. K., & Walters, G. F. 1977, *Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality*. 1.
- McFarland, D. J. & Hamilton, D. 2006, "Adding contextual specificity to the technology acceptance model", *Computers in Human Behavior*, vol. 22, no. 3, pp. 427-447.
- Meier, P. 1998, "Critical Success Factors Of Object Oriented Frameworks", in *STJA 98*.
- Menzies, T. & Di Stefano, J. S. 2003, "More success and failure factors in software reuse", *Ieee Transactions on Software Engineering*, vol. 29, no. 5, pp. 474-477.
- Messerschmitt, D. G. & Szyperski, C. 2003, *Software Ecosystem: Understanding an Indispensable Technology and Industry* MIT Press.
- Michael, M. "Effort Distribution in a Six Year Industrial Application Framework Project", *IEEE Computer Society*, p. 326.
- Mili, H. 1995, "Reusing Software - Issues and Research Directions", *Ieee Transactions on Software Engineering*, vol. 21, no. 6, pp. 528-562.
- Morisio, M., Ezran, M., & Tully, C. 2002, "Success and failure factors in software reuse", *Ieee Transactions on Software Engineering*, vol. 28, no. 4, pp. 340-357.

- Morisio, M., Romano, D., & Stamelos, I. 2002, "Quality, productivity, and learning in framework-based development: An exploratory case study", *Ieee Transactions on Software Engineering*, vol. 28, no. 9, pp. 876-888.
- Moser, S. & Nierstrasz, O. 1996, "The effect of object-oriented frameworks on developer productivity", *Computer*, vol. 29, no. 9, p. 45-&.
- Nelson, D. L. 1990, "Individual Adjustment to Information-Driven Technologies - A Critical-Review", *MIS Quarterly*, vol. 14, no. 1, pp. 79-98.
- Neuman, W. L. 2005, *Social research methods : qualitative and quantitative approaches*, 5th ed. edn.
- Northrop, L. M. 1999, "A Framework for Software Product Line Practice", Springer-Verlag, pp. 365-366.
- Oliveira, T. C., Alencar, P. S. C., Filho, I. M., de Lucena, C. J. P., & Cowan, D. D. 2004, "Software process representation and analysis for framework instantiation", *Ieee Transactions on Software Engineering*, vol. 30, no. 3, pp. 145-159.
- Parsons, D., Rashid, A., Telea, A., & Speck, A. 2006, "An architectural pattern for designing component-based application frameworks", *Software-Practice & Experience*, vol. 36, no. 2, pp. 157-190.
- Philippow, I. & Riebisch, M. 2001, "Systematic Definition of Reusable Architectures", in *Eight Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems ECBS '01*.
- Pikkarainen, T., Pikkarainen, K., Karjaluoto, H., & Pahlila, S. 2004, "Consumer acceptance of online banking: an extension of the technology acceptance model", *Internet Research-Electronic Networking Applications and Policy*, vol. 14, no. 3, pp. 224-235.
- Poulin, J. S. 1994, "Measuring software reusability", in *Third International Conference on Software Reuse*, pp. 126-138.
- Rai, A., Lang, S. S., & Welker, R. B. 2002, "Assessing the validity of IS success models: An empirical test and theoretical analysis", *Information Systems research*, vol. 13, no. 1, pp. 50-69.
- Roberts, D. & Johnson, R. 1997, "Patterns for evolving frameworks," in *Pattern languages of program design 3*, Addison-Wesley Longman Publishing Co. Inc., pp. 471-486.
- Saga, V. L. & Zmud, R. W. 1994, "The Nature and Determinants of It Acceptance, Routinization, and Infusion", *Diffusion, Transfer and Implementation of Information Technology*, vol. 45, pp. 67-86.
- Sangdon, L., Hansuk, C., Youngjong, Y., & Sangduck, L. "Storage and management of object-oriented frameworks", in *International Conference on Systems, Man, and Cybernetics, 1999*, pp. 762-767.

Seddon, P. B. 1997, "A respecification and extension of the DeLone and McLean model of IS success", *Information Systems research*, vol. 8, no. 3, pp. 240-253.

Seddon, P. B., Staples, S., Patnayakuni, R., & Bowtell, M. 1999, "Dimensions of Information Systems Success", *Communications of AIS*, vol. 2, no. 3.

SEI. Capability Maturity Model(R) Integration, CMMISM for Software Engineering, Version 1.1., CMU/SEI-2002-TR-029.

<http://www.sei.cmu.edu/publications/documents/02.reports/02tr029.html> . 2002. SEI. 4-6-2005.

Ref Type: Electronic Citation

Shai, B.-Y. 1997, "Pattern Language for Framework Construction", in *The 4th Pattern Language of Programming Conference*.

Shannon, C. E. & Weaver, W. 1949, "The Mathematical Theory of Communication", University of Illinois Press.

Sharp, J. H. 2006, "Development, Extension, and Application: A Review of the Technology Acceptance Model".

Shaw, M. 1990, "Prospects for An Engineering Discipline of Software", *Ieee Software*, vol. 7, no. 6, pp. 15-24.

Sheikh I.Ahmed, Alex Pezewski, & Al Pezewski "Towards Framework Selection Criteria and Suitability for an Application Framework", in *International Conference on Information Technology: Coding and Computing ITCC'04*.

Shull, F., Lanubile, F., & Basili, V. R. 2000, "Investigating reading techniques for object-oriented framework learning", *Ieee Transactions on Software Engineering*, vol. 26, no. 11, pp. 1101-1118.

Sindre, G., Conradi, R., & Karlsson, E. A. 1995, "The Reboot Approach to Software Reuse", *Journal of Systems and Software*, vol. 30, no. 3, pp. 201-212.

Sparks, S., Benner, K., & Faris, C. 1996, "Managing object-oriented framework reuse", *Computer*, vol. 29, no. 9, p. 52-&.

Speier, C., Urbaczewski, A., & Morris, M. G. Technology Acceptance: Has "Technology" Been Too Broadly categorized. 2005.

Ref Type: Unpublished Work

Srinivasan, S. 1999, "Design patterns in object-oriented frameworks", *Computer*, vol. 32, no. 2, p. 24-+.

Stevens, J. 1986, *Applied multivariate statistics for the social sciences*, 4 edn, Lawrence Erlbaum Associates Inc..

Succi, M. J., Succi, M. J., & Walter, Z. D. "Theory of user acceptance of information technologies: an examination of health care professionals", in *System Sciences*, 1999.

- HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on, Z. D. Walter, ed., pp. 1-7.
- Taligent Inc. 1995, *The Power of Frameworks: For Windows and Os/2*, 1 edn, Taligent Press.
- Todd, H. 1997, "Development of successful object-oriented frameworks", *ACM*, pp. 115-119.
- Tomer, A., Goldin, L., Kuflik, T., Kimchi, E., & Schach, S. R. 2004, "Evaluating software reuse alternatives: A model and its application to an industrial case study", *Ieee Transactions on Software Engineering*, vol. 30, no. 9, pp. 601-612.
- Trochim, W. M. K. 1989, "Concept Mapping - Soft Science Or Hard Art", *Evaluation and Program Planning*, vol. 12, no. 1, pp. 87-110.
- Uebersax, J. S. *Likert Scales: Dispelling the Confusion*.
<http://ourworld.compuserve.com/homepages/jsuebersax/likert.htm> . 2006. 8-7-2008.
Ref Type: Electronic Citation
- van Gurp, J. & Bosch, J. 2001, "Design, implementation and evolution of object oriented frameworks: concepts and guidelines", *Software-Practice & Experience*, vol. 31, no. 3, pp. 277-300.
- Venkatesh, V. & Davis, F. D. 1996, "A model of the antecedents of perceived ease of use: Development and test", *Decision Sciences*, vol. 27, no. 3, pp. 451-481.
- Venkatesh, V. & Davis, F. D. 2000, "A theoretical extension of the Technology Acceptance Model: Four longitudinal field studies", *Management Science*, vol. 46, no. 2, pp. 186-204.
- Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. 2003, "User acceptance of information technology: Toward a unified view", *MIS Quarterly*, vol. 27, no. 3, pp. 425-478.
- Westland, J. C. & Clark, T. H. K. 2000, *Global Electronic Commerce: Theory and Case Studies* MIT Press, Cambridge.
- Yu, J., Ha, I., Choi, M., & Rho, J. 2005, "Extending the TAM for a t-commerce", *Information & Management*, vol. 42, no. 7, pp. 965-976.
- Zigurs, I. & Buckland, B. K. 1998, "A theory of task/technology fit and group support systems effectiveness", *MIS Q.*, vol. 22, no. 3, pp. 313-334.

STVARNO KAZALO

AGFI ..125, 130, 134, 139, 140, 144, 145, 214	Neto prednosti uporabe ogrodja 109
AMOS125, 130, 131, 132, 134, 136, 150, 217	NFI.....125, 130, 134, 139, 140, 144, 145, 215
Anketa.....5, 58, 119, 154	NNFI 130, 134, 139, 140, 144, 145, 215
CFI.....125, 130, 134, 139, 140, 144, 145, 214, 215	Notranja kompleksnost razreda..... 174
COTS 183, 188, 191	Notranja veljavnost..... 154
D&M model 30, 31, 32	Ogrodje..... 15
Dinamično povezovanje 168	Opisna statistika..... 126, 127, 205
Diskriminacijska veljavnost..... 124	OTSO..... 183, 184, 185, 186, 188, 191
Dojeta enostavnost uporabe... 36, 78, 83, 95, 106, 115	Pearsonov koeficient korelacije..... 206
Dojeta uporabnost.....36, 106, 116, 163	Ponazorilna metoda..... 71
Domensko specifični jeziki.....28	Ponovna uporaba..... 1, 13, 14
Eksperimentalna veljavnost..... 154	Ponovna uporabnost 18, 78, 89
Empirična raziskava 4, 9	Postopno približevanje..... 70
Ethnograph 71	Povprečje slučajnega vzorca 205
Faktorska analiza 210, 211	Prenosljivost ogrodja 100, 101
FCM 41, 46, 47, 50, 79, 83, 84, 91, 94, 97	Proces izbire ogrodja..... 185, 186, 188
Fokusna skupina.....56	Proces izboljševanja sprejetosti ogrodij . 180
Generatorji aplikacij.....28	Proces razvoja programske opreme 20
GFI125, 130, 134, 139, 140, 144, 145, 214	Produktne linije 27
Graf..... 171	Programska komponenta 24
Hi-kvadrat 213	Programsko inženirstvo..... 1, 12
Informacijski sistem..... 12	Programsko ogrodje2, 13, 15, 17, 50, 87, 190
Informatika 12	Računalniška znanost 12
ISO/IEC 9126 ... 40, 41, 42, 44, 45, 46, 47, 51, 77, 84, 100, 173	Razširljivost.....19, 89, 179
Jedro ogrodja..... 167	Razumljivost ogrodja 100
Kakovost programske opreme40	REBOOT ..6, 14, 20, 46, 47, 51, 77, 78, 83, 84, 88, 99, 100, 173
Knjižnice razredov..... 25, 26	Regresijska analiza..... 208
Kompleksnost dostopa razredov..... 175	Relevantnost koncepta 81
Kompozitna zanesljivost..... 124	RMSEA.....125, 130, 134, 139, 140, 144, 145, 216
Konceptualni model ogrodij 171, 188	RMSR 130, 134, 136, 139, 140, 144, 145, 215
Korelacija..... 206	Seddonov model uspešnosti..... 32
Kovarianca 207	Semantični diferencial 114
Kumulativni raziskovalni pristop..... 6	Smernice razvoja ogrodij..... 169
Likertova lestvica..... 114, 164	Sourceforge.net.....3
Linearna regresija..... 208	Sprejetost ... 7, 31, 34, 50, 107, 109, 116, 166
McCabejevo ciklomatično število 174	SPSS 81, 91, 125, 132, 217
Merljivi model..... 123, 124	Standardna deviacija 206
Metoda anketiranja 58	Statistična veljavnost 154
Metoda za izbiro ogrodja 187	Strukturni model..... 123
Metrike prileganja..... 125, 132, 152	TAM ...8, 35, 36, 37, 38, 39, 40, 48, 49, 50, 77, 78, 83, 98, 99, 103, 104, 105, 107, 108, 110, 111, 113, 140, 144, 145, 148, 158, 159, 160, 162, 164, 190
Mnogoternost81	Tehnike kodiranja..... 70
Model za ocenitev sprejetosti ogrodja..... 163	Tehnološki razkorak..... 103, 158, 161
Modeliranje strukturnih enačb 123, 211	Teoretični model..... 98, 110, 113, 160, 190
Modularnost..... 18, 78, 179	TLI 125, 130, 134, 140, 144, 145, 215
Mreža determinant sprejetosti ogrodij... 164	
Multipla regresija 121	
Naključno vzorčenje..... 118	
Namera trajne uporabe 107, 109, 116	

TTF.....	39, 50, 102, 103, 105, 115, 132, 135, 136, 140, 141, 144, 145, 148, 155, 158, 159, 185	Zamrznjene točke	167
Učinkovitost ogrodja.....	102	Zanesljivost ponovnega testa.....	85
Ujemanje zahtev in ogrodja	102, 103, 115	Zanesljivost posameznih in skupnih rezultatov	86
Uspešnost IS	29, 33, 34	Zanesljivost razcepa podatkov.....	86
Varianca	206	Zaupanje v ogrodje.....	101, 163, 172
Vprašalnik.....	59, 114, 119, 120	Zunanja kompleksnost skupine razredov	176
Vroče točke	167	Zunanja veljavnost	154
Vzorci načrtovanja.....	26, 168, 182	Z-vrednost.....	206
Zadovoljstvo uporabe ogrodja.....	109		

ŽIVLJENJEPIS KANDIDATA

12. 5. 1976 Rojen v Mariboru.
- 1983-1991 Obiskovanje osnovne šole v Jarenini.
- 1991-1995 Obiskovanje Srednje šole elektrotehniške in računalniške usmeritve v Mariboru.
- 1995 Opravljena matura.
- 1995-2001 Dodiplomski študij, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, smer Informatika.
- 2001 Zaposlitev na UM - FERI, Inštitut za informatiko.
- 2001 Vpis na podiplomski študij UM - FERI, Računalništvo in informatika.
- 2001 Izvolitev v pedagoški naziv asistent.
- 2003 Odobren prehod na direktni doktorski študij.
- 2001-2008 Izvajanje vaj pri predmetih Informacijska družba, Metode komuniciranja, Organizacija in vodenje informacijskih obdelav, Programiranje 2, Informatika v medijih, Vodenje projektov informacijskih sistemov, Komuniciranje in delo v skupinah in Metode in tehnologije komuniciranja.

UNIVERZA V MARIBORU

Fakulteta za elektrotehniko, računalništvo in informatiko

IZJAVA DOKTORSKEGA KANDIDATA

Podpisani-a Gregor Polančič, vpisna številka 95023888

izjavljam,

da je doktorska disertacija z naslovom **Integriran model ocenjevanja uspešnosti programskih ogrodij**

- rezultat lastnega raziskovalnega dela,
- da so rezultati korektno navedeni in
- da nisem kršil-a avtorskih pravic in intelektualne lastnine drugih.

Podpis doktorskega-e kandidata-ke:

