

# Primjena hibridne metodologije upravljanja projektima u razvoju programskih sustava

---

Urlić, Svea

Professional thesis / Završni specijalistički

2023

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:168:546565>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-20**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Svea Urlić

**Primjena hibridne metodologije upravljanja  
projektima u razvoju programskih sustava**

SPECIJALISTIČKI RAD

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Svea Urlić

**Primjena hibridne metodologije upravljanja  
projektima u razvoju programskih sustava**

SPECIJALISTIČKI RAD

Zagreb, 2023.

UNIVERSITY OF ZAGREB  
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING  
SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**Svea Urlić**

**Application of hybrid project management  
methodology in development of software  
systems**  
**Primjena hibridne metodologije upravljanja  
projektima u razvoju programskih sustava**

SPECIALIST THESIS  
SPECIJALISTIČKI RAD

Zagreb, 2023.

Završni specijalistički rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva, na Zavodu za telekomunikacije.

Mentor: prof. dr sc. Željka Car

Završni rad ima: 90 stranica

Završni rad br.:

Povjerenstvo za ocjenu u sastavu:

1. prof. dr. sc. Krešimir Fertalj – predsjednik
2. prof. dr. sc. Željka Car – mentorica
3. nasl. izv. prof. dr. sc. Saša Dešić, KONČAR – Ericsson Nikola Tesla, Zagreb – član.

Povjerenstvo za obranu u sastavu:

1. prof. dr. sc. Krešimir Fertalj – predsjednik
2. prof. dr. sc. Željka Car – mentorica
3. nasl. izv. prof. dr. sc. Saša Dešić, KONČAR – Ericsson Nikola Tesla, Zagreb – član.

Datum obrane: 26. siječnja 2023.

## Sažetak

Odabir metodologije upravljanja projektima u razvoju programskih sustava vrlo je zahtjevan zadatak, posebice imajući u vidu da je svaki projekt jedinstven. U ovom specijalističkom radu analiziran je hibridni pristup kao kombinacija tradicionalnih i agilnih metodologija upravljanja projektima, s naglaskom na primjenu kombinacije Scruma i vodopadnog modela. Specijalistički rad daje pregled rezultata kvalitativnog istraživanja, provedenog kroz dubinske intervjue, i ankete vezano za metodologije upravljanja projektima korištene u projektima razvoja web-aplikacija. Rezultati kvalitativnog istraživanja pokazali su da većina projekata slijedi obrazac u kojem se vodopadni model kombinira s agilnim modelima, a agilne su metodologije korištene u projektnoj fazi Razvoja na gotovo svim projektima koji su bili predmet istraživanja. Kvantitativno istraživanje kroz anketni upitnik potvrdilo je kako su Scrum i vodopadni model najčešće korištene metodologije na projektima razvoja web-aplikacija te ukazalo na učestalost preklapanja projektnih faza Razvoja i Testiranja. Rad daje smjernice za kombiniranje vodopadnog modela s agilnim modelima te ukazuje na važnost planiranja, analize korisničkih zahtjeva i definiranja arhitekture sustava na početku projekta te isporučivanja korisnicima više manjih razvojnih verzija na testiranje.

**Ključne riječi:** Upravljanje projektima; razvoj softvera; vodopadni model; agilne metodologije upravljanja projektima; Scrum; hibridne metodologije upravljanja projektima; web-aplikacija

## **Abstract**

Selection of project management methodology in development of software systems is very challenging, especially taking into consideration that every project is unique. In this specialist thesis hybrid approach as a combination of traditional plan-driven and agile methodologies of project management, with emphasis on applying Scrum in waterfall project management, is analyzed. This specialist thesis reports on qualitative research through interviews and a survey on project methodologies used in web-application development projects. Results of qualitative research show that most projects follow a pattern in which waterfall methodology of project management is combined with agile practices - agile was used in almost all projects which were the subject of research in project phase of Development. Results of quantitative research conducted through a survey questionnaire confirmed that Scrum and waterfall are the most frequently used methodologies in web-application development projects and also indicated the frequency of overlapping of Development and Testing project phases. Specialist thesis provides guidelines for combining waterfall model with agile models and emphasizes the importance of planning, analyzing user requirements and defining system architecture at the beginning of the project, as well as delivering multiple smaller development versions to users for testing.

**Keywords:** Project management; software development; waterfall; agile project methodologies; Scrum, hybrid project methodologies; hybrid model; web-application



# Sadržaj

Sažetak.....	VI
Abstract.....	VII
Uvod.....	1
1. Pregled metodologija upravljanja projektima razvoja softvera .....	2
1.1. Tradicionalne metodologije upravljanja projektima .....	2
1.1.1. Vodopadni model.....	3
1.2. Agilne metodologije upravljanja projektima .....	4
1.2.1. Scrum .....	7
1.3. Usporedba vodopadnog modela i Scruma .....	11
1.4. Hibridne metodologije upravljanja projektima .....	13
1.4.1. Zastupljenost hibridnih metodologija upravljanja projektima s obzirom na veličinu tvrtke i industriju .....	14
1.4.2. Najzastupljenije metodologije upravljanja projektima u hibridnim modelima..	16
1.4.3. Hibridni modeli temeljeni na vodopadnom modelu i Scrumu.....	25
1.4.4. Utjecaj primjene hibridnih modela na uspješnost projekta.....	31
1.4.5. Komunikacija s korisnicima u hibridnim modelima sastavljenim od vodopadnog modela i agilnih projektnih metodologija .....	35
2. Kvalitativno istraživanje načina vođenja projekata razvoja web-aplikacija .....	40
2.1. Dubinski intervjui.....	44
2.1.1. Dubinski intervju_1 .....	44
2.1.2. Dubinski intervju_2 .....	48
2.1.3. Dubinski intervju_3 .....	52
2.1.4. Dubinski intervju_4 .....	55
2.1.5. Dubinski intervju_5 .....	58
2.1.6. Dubinski intervju_6 .....	62
2.1.7. Dubinski intervju_7 .....	65
2.2. Analiza rezultata kvalitativnog istraživanja .....	68
3. Kvantitativno istraživanje mišljenja članova projektnih timova o primijenjenim metodologijama upravljanja projektima na projektima razvoja web-aplikacija .....	72
3.1. Socioekonomski podaci.....	72
3.2. Radno iskustvo na projektima razvoja web-aplikacija.....	72
3.3. Iskustvo s posljednjeg projekta razvoja web-aplikacije .....	73
3.3.1. Trajanje projekta razvoja web-aplikacije.....	73

3.3.2.	Projektna uloga na projektu razvoja web-aplikacije.....	74
3.3.3.	Broj članova projektnog tima.....	74
3.3.4.	Najveći izazovi na projektu .....	75
3.3.5.	Korištena metodologija upravljanja projektom.....	76
3.3.6.	Korištene prakse razvoja softvera.....	77
3.3.7.	Preklapanje projektnih faza.....	79
3.3.8.	Uključivanje korisnika u testiranje.....	80
3.3.9.	Kašnjenje projekta.....	80
3.4.	Stav ispitanika o načinu upravljanja projektima razvoja web-aplikacija.....	81
4.	Prijedlozi i smjernice za kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima.....	84
	Zaključak .....	89
	Literatura .....	91
	Popis tablica .....	94
	Popis slika .....	94
	Životopis.....	96
	Biography.....	97

## Uvod

U ovom specijalističkom radu istražena je primjena hibridnih metodologija upravljanja projektima u projektima razvoja programskih rješenja zasnovanih na webu. Hibridne metodologije upravljanja projektima spoj su različitih metodologija. Takve metodologije nisu strogo definirane te ih razvojni timovi kroje na temelju svojeg iskustva i potreba projekata. Suvremeni razvoj softvera rijetko se provodi striktno po jednoj metodologiji upravljanja projektima pa većina projektnih timova koristi hibridne metodologije, a cilj ovog rada je identificirati koje se aktivnosti preuzimaju iz metodologije upravljanja projektima po tradicionalnom vodopadnom modelu, a koje se aktivnosti preuzimaju iz agilne metodologije Scrum.

U radu je naglasak stavljen na analizu hibridnih metodologija upravljanja projektima i identifikaciju projektnih aktivnosti koje su se pokazale učinkovitijima kada su rađene po tradicionalnim metodologijama, kao i identifikaciju projektnih aktivnosti koje je preporučljivo izvoditi po agilnom pristupu.

U sklopu rada napravljen je kratki pregled tradicionalnih metodologija s naglaskom na model vodopada i agilnih metodologija s naglaskom na Scrum te njihova usporedba, a zatim definicija i pregled hibridnih metodologija.

Analizirana je primjena hibridnih metodologija upravljanja projektima u razvoju softvera općenito s naglaskom na rješavanje izazova koji se pojavljuju u različitim fazama projekata i na različitim aktivnostima korištenjem vodopadnog modela i Scruma.

Kvalitativnim istraživanjem (dubinskim intervjuiranjem) u kojem su ispitanici voditelji projekata istražen je način vođenja projekata razvoja web-aplikacija s ciljem utvrđivanja najčešćih problema na projektima i načina na koji se rješavaju, a potom je provedeno i kvantitativno istraživanje s ostalim članovima projektnih timova kako bi se utvrdio njihov stav s obzirom na identificirane probleme i ukazalo na to koje su aktivnosti u primijenjenim hibridnim modelima preuzete iz tradicionalnih metodologija upravljanja projektima, a koje iz agilnih te koliko se njihova kombinacija pokazala uspješnom.

Naposljetku, definirane su smjernice i prijedlozi za učinkovito kombiniranje tradicionalnih i agilnih metodologija upravljanja projektima, kako bi se izbjegli nedostaci korištenja samo jedne metodologije i postigli optimalni rezultati.

# 1. Pregled metodologija upravljanja projektima razvoja softvera

## 1.1. Tradicionalne metodologije upravljanja projektima

Tradicionalni pristup vođenju projekata, sukladno smjernicama opisanim u katalogu znanja Project Management Body of Knowledge (PMBOK) [1], dijeli projekt na faze Inicijalizacije, Planiranja, Izvršenja, Kontrole i Zatvaranja. Skup svih faza čine životni ciklus projekta, a karakteristično je da su faze slijedne te da su izlazi, odnosno rezultati posla obavljenog u jednoj fazi ulazi za posao koji će se obaviti u sljedećoj fazi [1]. Kod tradicionalnog pristupa plan je temelj projekta te se stoga u literaturi tradicionalne metodologije upravljanja projektima često naziva „vođen planom“ (*plan-driven*).

Najistaknutiji tradicionalni modeli razvoja programskih proizvoda su [2]:

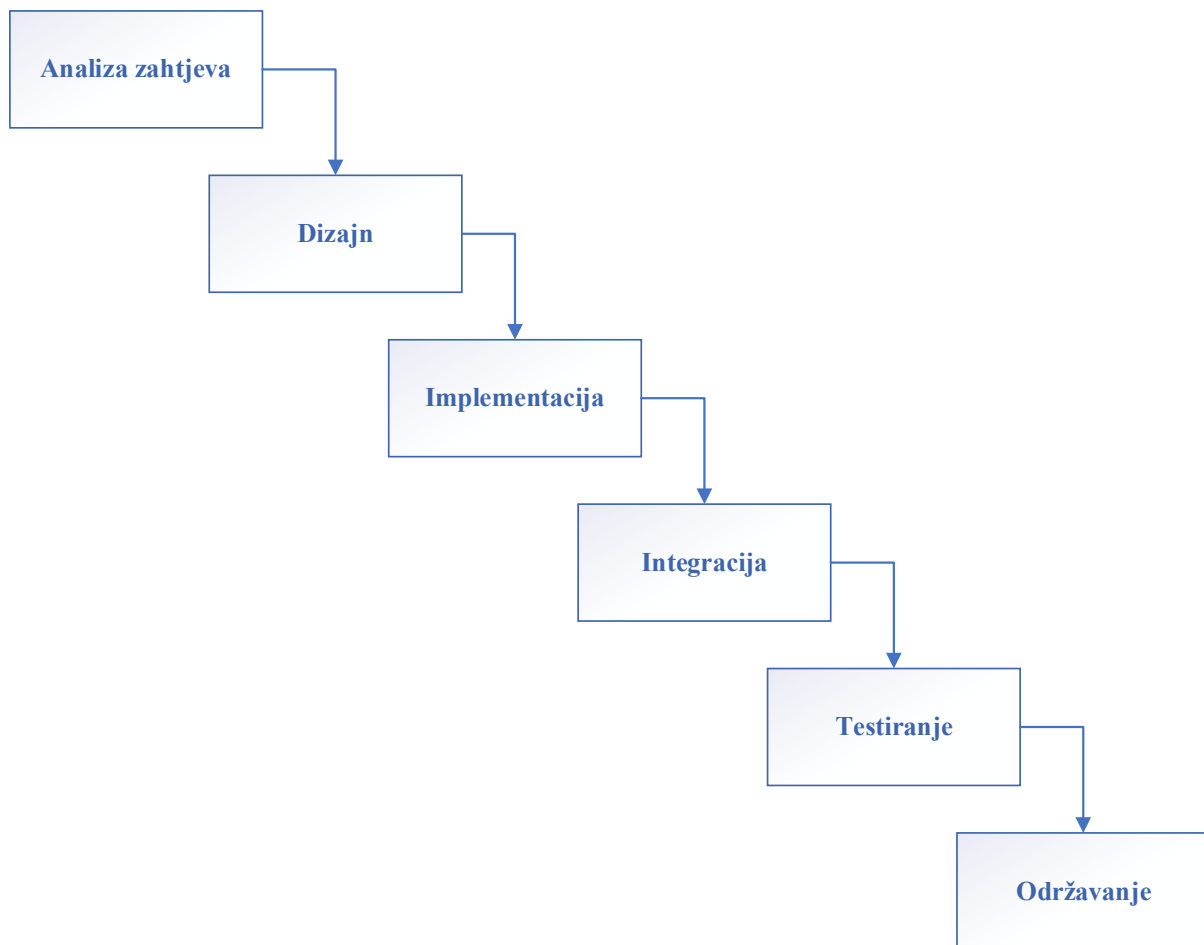
- Vodopadni model, koji je najstariji sustavni pristup razvoju programskih proizvoda. Za njega je karakteristično da se aktivnosti odvijaju kaskadno odnosno slijede jedna drugu.
- Evolucijski model, koji se zasniva na evoluciji početne verzije sustava. Početno se rješenje razvija te se u sklopu tog procesa proizvode međurješenja, sve dok se ne proizvede konačno rješenje.
- Iterativni model, koji kombinira vodopadni model s iterativnim pristupom evolucijskog. Prva je iteracija jezgri proizvod. Rješenje se razvija kroz niz iteracija. U svakoj se iteraciji dodaju nove funkcionalnosti unutar predefiniranog vremenskog rasporeda, a rješenje se smatra konačnim tek nakon posljednje iteracije [1].
- Spiralni model, koji je nadogradnja evolucijskog modela. Ovaj model ima oblik spirale te se u jednoj spirali razvija dio programskog proizvoda ili povećava njegova funkcionalnost. Razvoj projekta odvija se u fazama koje se definiraju s ciljem identifikacije i smanjenja rizika od neuspjeha te se stoga na početku svake faze radi procjena rizika, a u slučaju da je procijenjeni rizik prevelik projekt se prekida.
- Komponentni razvoj, kod kojeg je motivacija u pretpostavci ponovne iskoristivosti postojećih komponenata. Sustav se integrira višestrukom uporabom postojećih komponenata ili uporabom komercijalnih, gotovih komponenata (engl. *COTS – Commercial-of-the-shelf*) [3].
- Korisnički usmjeren razvoj programskog proizvoda (prototipiranje) zasniva se na izradi prototipa programskog proizvoda te na paradigmi „*User centered design*“, pri čemu

*design* ne označava arhitekturu već vanjski izgled i logiku funkcioniranja programskog proizvoda. Korisnički usmjeren razvoj programskog proizvoda počinje prikupljanjem zahtjeva korisnika, a zatim se provodi kontinuirana evaluacija predloženih prototipa. Dizajn se u tijeku evaluacije revidira te se analiziraju zahtjevi za promjenama, a tek se potom izrađuje plan implementacije.

U nastavku rada fokus će, vezano za tradicionalne metodologije upravljanja projektima, biti stavljen na vodopadni model budući da je sukladno iskustvu u softverskoj industriji riječ o najčešće korištenom tradicionalnom modelu razvoja programskih proizvoda.

### **1.1.1. Vodopadni model**

Vodopadni model (eng. *waterfall model*) je do kraja 20. stoljeća bio dominantna metoda razvoja softvera. Prvi je puta ovaj model upravljanja projektima u literaturi pojašnjen od strane Winstona Roycea 1970. godine, no sam naziv „vodopadni“ model prvi puta su spomenuli T. E. Bell te T.A. Thayer 1976. godine [4]. Riječ je o sekvencijalnoj metodologiji upravljanja projektima u kojoj se kroz projektne faze prolazi točno određenim redoslijedom (Slika 1). Sam naziv ukazuje na to da faze projekta „teku“ iz jedne u drugu te u ovoj projektnoj metodologiji nema vraćanja na prethodnu fazu.



Slika 1. Dijagram vodopadnog modela po fazama razvoja softvera [4]

## ***1.2. Agilne metodologije upravljanja projektima***

Tijekom 1990-ih godina započelo je formiranje agilnih metodologija upravljanja projektima kao odgovor na vodopadni model, ali i ostale modele razvoja jer je bilo očito da kod kompleksnih projekata postoji potreba za vraćanjem na prethodne faze projekta budući da se i sami korisnički zahtjevi mijenjaju protekom vremena. 2001. godine 17 je stručnjaka (Kent Beck, Mike Beedle, Arie van Bennekum, Alistar Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellot, Ken Schwaber, Jeff Sutherland i Dave Thomas) sastalo se u Snowbirdu u Utahu, gdje su kreirali 2 dokumenta - „Agilni manifest“ i „Načela agilnog razvoja“. Stoga je upravo 2001. godina bila prekretnica jer su agilne metodologije upravljanja projektima tada službeno definirane [4].

Agilne metodologije potiču iterativni i inkrementalni pristup razvoju, u kojem se kroz faze prolazi u svakoj iteraciji, a softver isporučuje evolucijski [4].

## **Načela na kojima se zasniva proglas o agilnom razvoju softvera:**

Rukovodimo se sljedećim načelima:

Najvažnije nam je zadovoljstvo naručitelja koje postizemo ranom i neprekinutom isporukom softvera koji nosi vrijednost.

Spremno prihvaćamo promjene zahtjeva, čak i u kasnoj fazi razvoja.

Agilni procesi uprežu promjene da naručitelju stvore kompetitivnu prednost.

Često isporučujemo upotrebljiv softver, u razmacima od nekoliko tjedana do nekoliko mjeseci, nastojeći da razmak bude čim kraći.

Poslovni ljudi i razvojni inženjeri moraju svakodnevno zajedno raditi, tijekom cjelokupnog trajanja projekta.

Projekte ostvarujemo oslanjajući se na motivirane pojedince. Pružamo im okruženje i podršku koja im je potrebna, i prepuštamo im posao s povjerenjem.

Razgovor uživo je najučinkovitiji način prijenosa informacija razvojnom timu i unutar tima.

Upotrebljiv softver je osnovno mjerilo napretka.

Agilni procesi potiču i podržavaju održivi razvoj. Pokrovitelji, razvojni inženjeri i korisnici trebali bi moći neograničeno dugo zadržati jednak tempo rada.

Neprekinuti naglasak na tehničkoj izvrsnosti i dobar dizajn pospješuju agilnost.

Jednostavnost, vještina povećanja količine posla kojeg ne treba raditi, je od suštinske važnosti.

Najbolje arhitekture, projektne zahtjeve i dizajn, stvaraju samoorganizirajući timovi.

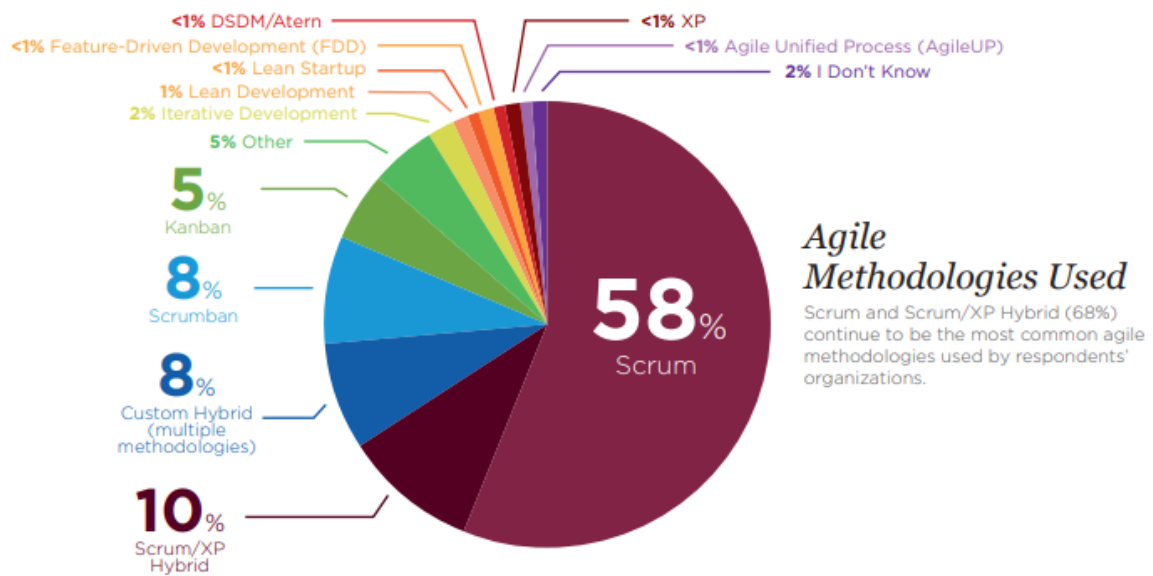
Tim u redovitim razmacima razmatra načine da postane učinkovitiji, a zatim usklađuje i prilagođava svoje ponašanje. [5]

Najpoznatije agilne metodologije razvoja softvera su Scrum, *eXtreme Programming*, *Lean project management* te Kanban [4].

Na tortnom dijagramu (Slika 2) prikazane su najkorištenije agilne metodologije upravljanja projektima prema istraživanju tvrtke VersionOne koja je isti objavila u 11. verziji publikacije „*State of Agile*“ [6]. Istraživanje je provedeno između srpnja i prosinca 2016. godine. Polovina ispitanika bila je iz Sjeverne Amerike, 28% iz Europe, 10% iz Azije, 5% iz Južne Amerike, 4% iz Oceanije te 2% iz Afrike. Od toga je 23% ispitanika iz industrije razvoja softvera te je 23% ispitanika zaposleno u ulozi voditelja projekata (neovisno o industriji).

Čak 58% ispitanika izjasnilo se da koristi Scrum metodologiju upravljanja projektima, a slijedile su hibridna projektna metodologija koja je kombinacija Scruma i *eXtreme Programminga* s 10%, hibridne metodologije koje su kombinacija većeg broja metodologija upravljanja projektima s 8%, kombinacija Scruma i Kanbana također s 8% te Kanban s 5% dok su ostale metodologije zajedno prikupile svega 11%.





Slika 2. Najkorištenije agilne metodologije 2017. [6]

### 1.2.1. Scrum

Scrum je najkorištenija agilna projektna metodologija [6].

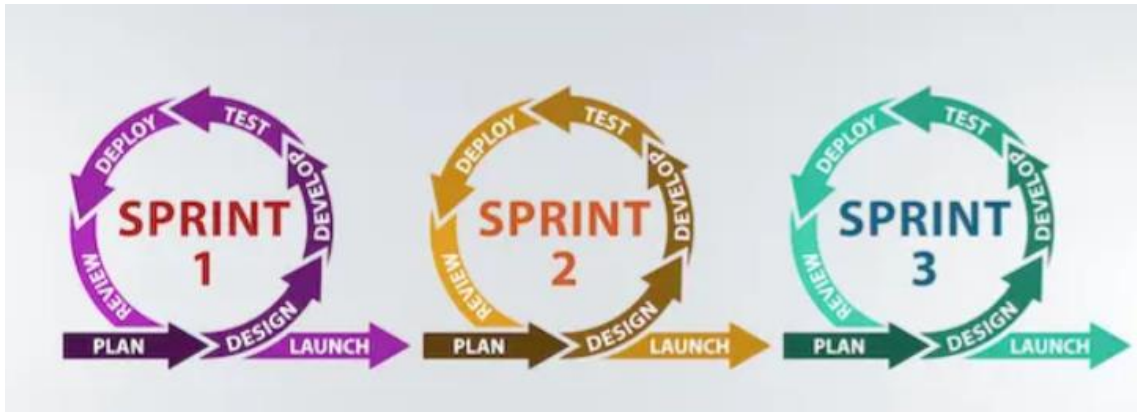
Sam je naziv preuzet iz sporta, odnosno iz ragbija, gdje označava snažnu formaciju igrača koji pokušavaju zajedno postići cilj.

Riječ je o brzom modelu razvoja softverskih proizvoda koji se temelji na iterativnom i inkrementalnom pristupu razvoju u kojem se kroz sve faze razvoja softvera prolazi u svakoj iteraciji, a softver se isporučuje nakon svake razvojne iteracije (Slika 3).

U Scrumu se često isporučuje uporabljiv softver, koji sadrži barem minimalni set funkcionalnosti da zadovolji poslovnu potrebu, kako bi se što prije dobila povratna informacija od korisnika i stalno korigirao proces.

Osnovna su obilježja Scruma da je lagan za primjenu (nema puno pravila i procedura), jednostavan (lako razumljiv) te donosi mjerljive rezultate.

Korisnik je ključni sudionik projektnog tima koji donosi odluke u stvarnom vremenu i kontinuirano komunicira s razvojnim timom.



Slika 3. Životni ciklus Scruma [7]

### 1.2.1.1. Osnovni pojmovi u Scrumu

Ciklusi razvoja softvera koji se opetovano ponavljaju kroz cijeli proces razvoja nazivaju se *Sprintovima*. *Sprintovi* su brze i relativno kratke iteracije u kojima nastaje mjerljivo i uporabljivo softversko rješenje. Uobičajeno *Sprintovi* traju 2 do 4 tjedna.

Scrum ima 3 artefakta. Artefakti, odnosno tvorevine Scruma su:

- *Backlog* proizvoda, koji predstavlja „rezervoar” svih zahtjeva koje je potrebno ispuniti da bi rješenje bilo konačno i potpuno. U *Product Backlogu* moraju se nalaziti svi zahtjevi, kako bi dovršenost bila mjerljiva i kako bi se mogao procijeniti točan napor.
- *Sprint Backlog* često nazivaju „mlađim bratom” *Backloga* proizvoda budući da sadrži podskup zahtjeva koje se razvojni tim obvezao isporučiti u tekućem *Sprintu*.
- Inkrement predstavlja potencijalno isporučiv proizvod koji je rezultat rada u jednom *Sprintu* – mora biti potpuno dovršen, testiran i dokumentiran.

### 1.2.1.2. Projektne uloge u Scrumu

*Product Owner* je uloga zadužena da iz vizije kreira korisničke zahtjeve i zapiše ih u *Backlog* proizvoda. Predstavlja sponu između razvojnog tima i korisnika. Naziva se i čuvarom vizije projekta i vrlo je važno da bude utjelovljen u jednoj osobi. Jedina je osoba koju razvojni tim pita za odluke oko zahtjeva. Odlučuje o sadržaju *Backloga* proizvoda, određuje koje se stavke razvijaju u nadolazećem *Sprintu*, na raspolaganju je razvojnom timu radi pojašnjenja detalja zahtjeva te odlučuje kada se proizvod isporučuje.

Scrum *Master* je čuvar tima i metodologije rada. On dirigira ritam rada Scrum tima i brine o poštivanju praksi i načela Scrum projektne metodologije.

Razvojni tim je multidisciplinarni tim osoba koji posjeduje sva potrebna znanja i vještine za isporuku inkrementa rješenja.

### **1.2.1.3. Komunikacija s korisnicima**

*Product Owner* utjelovljuje više interesnih strana, poput upravljačkih struktura i korisnika.

*Product Owner* često je zaposlenik naručitelja, makar ne mora biti. No *Product Owner* mora dobro razumjeti poslovno područje na koje se projekt odnosi, kao što mora imati i osjećaj za vrijednost proizvedenog jer je upravo *Product Owner* osoba koja odgovara za odnos uloženog rada i ostvarene vrijednosti.

Scrum predviđa da se, uz *Product Ownera*, korisnici uključe na:

- Planiranje *Sprinta*
- Pregled *Sprinta*.

Korisnici su također pozvani da pasivno sudjeluju na dnevnim sastancima tima.

Korisnici u Scrumu kontinuirano su upućeni u razvojne aktivnosti kako bi mogli reagirati na vrijeme, a ne tek na kraju projekta.

### **1.2.1.4. Ceremonije u Scrumu**

Scrum ima 4 ceremonije, odnosno vrste sastanaka, koje se održavaju sukladno jasno utvrđenim pravilima, a riječ je o:

1. Planiranju *Sprinta*
2. Dnevnom Scrum sastanku
3. Pregledu *Sprinta*
4. Retrospektivi *Sprinta*.

#### ***1.2.1.4.1. Planiranje Sprinta***

Prva je ceremonija u Scrumu Planiranje *Sprinta*.

Trajanje se određuje kao umnožak tjedana u *Sprintu* \* 2h, dakle za dvotjedni *Sprint* trajanje je 4h.

Na planiranju *Sprinta* sudjeluju svi članovi Scrum tima.

Izlazni rezultat ove ceremonije je *Sprint Backlog* odnosno jasno isplaniran inkrement sljedećeg *Sprinta*.

Provodi se u 2 dijela:

- 1) *Product Owner* predstavlja listu zahtjeva za koje smatra da bi trebali biti riješeni u nadolazećem *Sprintu*

U izravnoj komunikaciji *Product Owner* razjašnjava članovima tima detalje zahtjeva.

- 2) *Product Owner* više nije potreban, čak je dobra praksa da napusti sastanak.

Razvojni tim treba dogovoriti KAKO ostvariti cilj *Sprinta*. To je samoupravljujući tim, donosi odluke kako će ispuniti pojedini zahtjev te procjenjuje posao u radnim satima.

Kada procijeni sve zadatke, razvojni tim će odlučiti koliko od prvotnih želja *Product Ownera* može ispuniti (to može biti sve, ali i ne mora). Ako ima još vremena na raspolaganju, bitno je da im *Product Owner* ostavi i „rezervne” zahtjeve.

Na kraju razvojni tim *Product Owneru* i *Scrum Masteru* predstavlja *Sprint Backlog* i preuzima odgovornost da ga isporuči u idućem *Sprintu*.

#### **1.2.1.4.2. Dnevni Scrum sastanak**

Dnevni Scrum sastanak održava se svakodnevno i uobičajeno traje 10 do 15 minuta. Često ga se naziva i *daily standup*, odnosno stojeći sastanak jer je uobičajeno da članovi razvojnog tima stoje tijekom sastanka s ciljem da sastanak ne traje dulje od planiranog.

Na njemu sudjeluju svi članovi razvojnog tima.

Glavne teme su:

- Što sam radio/la jučer
- Što ću raditi danas
- Imam li prepreke za obaviti svoj današnji zadatak.

#### **1.2.1.4.3. Pregled Sprinta**

Pregled *Sprinta* je sastanak na kojem *Product Owner* potvrđuje jesu li dosegnuti ciljevi *Sprinta*, odnosno sastanak na kojem razvojni tim predstavlja posao odrađen u tijeku *Sprinta*.

Na ovom sastanku mogu sudjelovati i drugi zainteresirani dionici iz krugova Naručitelja.

Uobičajeno traje 2 do 4h.

Poželjno je da tim za ovaj sastanak pripremi demo verziju softvera proizvedenog u tijeku *Sprinta*, kako bi što vjerodostojnije prezentirao isporuku *Sprinta*.

Na kraju ovog sastanka *Product Owner* odlučuje hoće li inkrement isporučiti krajnjim korisnicima na korištenje.

#### **1.2.1.4.4. Retrospektiva**

Retrospektiva se često naziva i „Pogled unatrag”.

Na sastanku sudjeluju Razvojni tim i *Scrum Master*, ali bez *Product Ownera*.

Sastanak traje od 45 minuta do 3h, ovisno o duljini trajanja Srinta.

Primjerice:

- Ako je *Sprint* trajao 1 tjedan – retrospektiva će trajati 45 min
- Ako je *Sprint* trajao 2 tjedna - retrospektiva će trajati 1.5 h
- Ako je *Sprint* trajao 3 tjedna - retrospektiva će trajati 2.25 h
- Ako je *Sprint* trajao 4 tjedna - retrospektiva će trajati 3h.

Cilj retrospektive je analizirati što je u proteklom *Sprintu* bilo dobro, a što ne.

Ova je ceremonija najvažniji alat za kontinuirani napredak i povećanje učinkovitosti procesa razvoja softvera.

Dobra je praksa da se zaključci sa sastanka Retrospektive spremaju na dostupno mjesto kako bi i drugi *Scrum* timovi iz organizacije mogli vidjeti zaključke i naučiti na „tuđim greškama”.

### **1.3. Usporedba vodopadnog modela i Scruma**

Vodopadni model generalno se smatra modelom koji više pažnje pridaje vođenju samog projekta dok se *Scrum* smatra modelom koji se fokusira na proizvod.

U Tablici 1 navedene su osnovne razlike vodopadnog modela i *Scrum* modela po Schwaberu, K. i Beedleu, M. sukladno njihovoj knjizi [8].

Tablica 1. Usporedba vodopadnog modela i Scruma [8]

<b>Vodopadni model</b>	<b>Scrum</b>
Fokus je na projektu	Fokus je na proizvodu
Tradicionalni model sastavljen od jasno odvojenih faza	Agilni model sastavljen od <i>Sprintova</i>
Ne očekuje promjene u tijeku projekta	Očekuje i prihvaća promjene
Puno projektne dokumentacije	Malo projektne dokumentacije
Troškovi projekta utvrđuju se za vrijeme planiranja projekta	Projektne troškovi utvrđuju se u tijeku projekta
Niža vjerojatnost uspjeha	Viša vjerojatnost uspjeha
Fleksibilnost i kreativnost projektnog tima su ograničene	Fleksibilnost i kreativnost projektnog tima nisu ograničene
Sekvencijalan razvoj po fazama	Preklapanje faza

Mahalakshmi, M. i Sundararajan, M. u svome su radu [9] napravili komparativnu studiju vodopadnog modela i Scruma. U nastavku su navedeni zaključci njihove studije.

Prednosti vodopadnog modela su:

- Riječ je sekvencijalnom modelu, kojeg je lako implementirati
- Količina resursa potrebnih za implementaciju vodopadnog modela je minimalna
- Razvojne aktivnosti praćene su odgovarajućom dokumentacijom.

Nedostaci vodopadnog modela su:

- Problemi koji se pojave unutar jedne često imaju izvorište u nekoj od prethodnih faza projekta, a vraćanja na prethodne faze nema
- Ako klijent želi promijeniti zahtjev u tijeku projekta, promjena neće biti implementirana
- Opseg je „zamrznut“, a samim time i raspored isporuka.

Prednosti Scruma su:

- Zadovoljstvo klijenta uslijed optimizacije vremena isporuke rješenja i odziva na zahtjeve
- Povećanje kvalitete isporuka
- Prihvaćanje promjena zahtjeva

- Pružanje točnijih i bržih procjena potrebnog napora i trajanja razvojnih zadataka
- Brze isporuke iskoristivih dijelova programskog rješenja
- Zamrzavanje rasporeda razvojnih zadataka u malim dionicama (*Sprintovima*) – *Sprint* se ne mijenja kada započne.

Nedostaci Scruma su:

- Dokumentacije je jako malo
- Predanost članova tima da u sklopu *Sprinta* uistinu isporuče razvojne zadatke stavljene u *Sprint Backlog* je esencijalna
- Timski je rad presudan i ako članovi tima ne surađuju dobro projekt će se vjerojatno suočiti s neuspjehom.

Finalni je zaključak njihove studije da je vodopadni model pogodniji za projekte na kojima nema promjena zahtjeva, a Scrum za projekte na kojima su promjene zahtjeva česte.

#### ***1.4. Hibridne metodologije upravljanja projektima***

Hibridne metodologije upravljanja projektima prema definiciji predstavljaju kombinaciju metodologija upravljanja projektima, koju organizacijska jedinica usvoji i prilagodi svojim kontekstualnim potrebama [10].

U projektima razvoja programskih sustava uobičajeno je da se voditelji projekata bave uspostavljanjem ravnoteže između planiranja i kontrole nad projektom s jedne strane te agilnošću odnosno mogućnošću brzog reagiranja na promjene s druge strane. Dionici projekta često očekuju predvidljivost i jasan opseg projekta, ali i fleksibilnost u slučaju promjene zahtjeva projekta zbog promjene poslovnih prilika. Ni tradicionalne ni agilne metodologije upravljanja projektima nisu dostatne kako bi samostalno ispunile navedene zahtjeve te su stoga hibridne metodologije upravljanja projektima kao kombinacija tradicionalnih i agilnih metodologija sve zastupljenije [11].

Hibridne metodologije upravljanja projektima, koje kombiniraju tradicionalni i agilni pristup, postaju sve zastupljenije jer su odgovarajuće za provedbu većine projekata [11]. Naime, vodopadni model teško je implementirati u potpunosti jer u tijeku projekta gotovo uvijek dolazi do promjena zahtjeva i situacije na tržištu, a s druge strane agilni pristup predviđa kontinuirani

razvoj kroz veći broj isporuka koje nisu na početku projekta jasno definirane, što otežava definiranje roka završetka projekta.

Ljepota hibridnih metodologija upravljanja projektima upravo je u tome što omogućuju timu planiranje prije nego počnu raditi na projektu, ali također dijele razvojni ciklus na kratkoročne isporuke. Kod hibridnih modela preporučuje se početno planiranje odnosno analizu i definiranje korisničkih zahtjeva provesti po modelu vodopada, a razvoj i isporuke po agilnom modelu [11]. Ovakav hibridni pristup čini planiranje i procjene na projektu puno točnijima, a istovremeno tim može na vrijeme reagirati na tržišne promjene i isporučiti ono što situacija na tržištu zahtijeva, a ne nužno ono što je inicijalno isplanirano.

U istraživanje HELENA („*Hybrid dEveLopmENt Approaches in software systems development*“) [12] uloženi su višegodišnji napor kako bi se otkrilo kako projekti razvoja softvera u praksi kombiniraju tradicionalne i agilne metode razvoja softvera te koje se prakse najviše koriste.

Analiza rezultata istraživanja HELENA u radu [10] ukazala je da voditelji projekata razvijaju i nadograđuju hibridne modele postepeno na temelju iskustva i učenja iz prethodnih projekata - čak se 83,9% ispitanika izjasnilo da hibridne modele definiraju na temelju vlastitog iskustva s prethodnih projekata.

Više od četvrtine (27,5%) ispitanika reklo je da u svojoj organizaciji ima slobodu individualno odabrati pristup i prilagoditi ga svakom specifičnom projektu.

#### **1.4.1. Zastupljenost hibridnih metodologija upravljanja projektima s obzirom na veličinu tvrtke i industriju**

Autori istraživanja [10] utvrdili su da tvrtke koriste hibridne modele u razvoju softvera neovisno o svojoj veličini.

Istraživanje je pokazalo da se u prosjeku u tvrtkama koristi 5,5% tradicionalnih metodologija te 16% agilnih metodologija (Tablica 2). Agilne prakse pak koristi 40% ispitanika, a prakse i iz tradicionalnih i iz agilnih modela 25,7% ispitanika (Tablica 2). Zaključak je kako se u tvrtkama koriste kombinacije tradicionalnih i agilnih metodologija upravljanja projektima neovisno o njihovoj veličini.

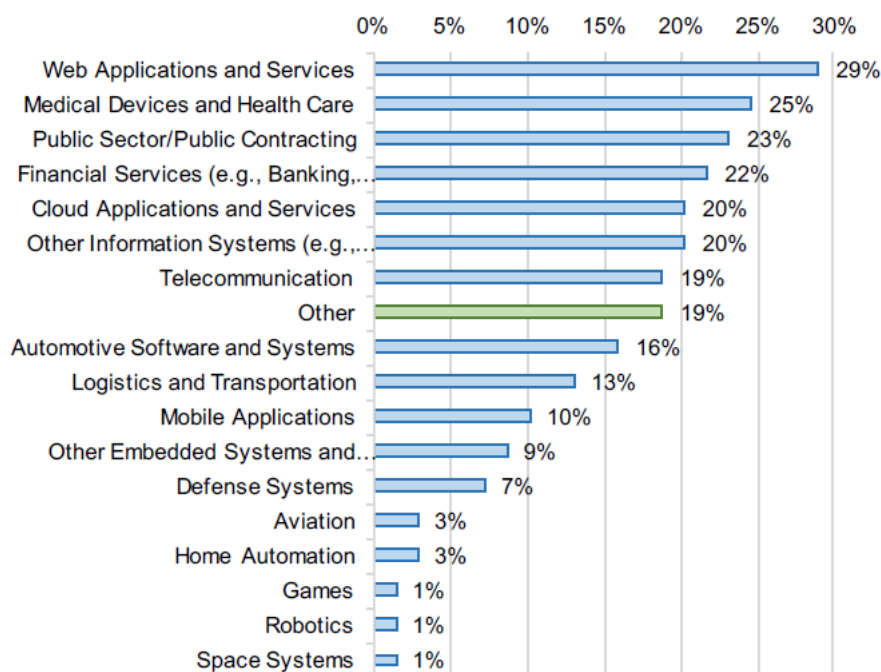


Tablica 2. Korištene metodologije upravljanja projektima i prakse s obzirom na veličinu tvrtke [10]

Veličina tvrtke	Metodologija			Praksa			Ostalo	n
	T	A	O	T	A	O		
<b>Mikro</b>	6,6%	15,4%	13,2%	5,5%	33,0%	25,3%	1,1%	91
<b>Mala</b>	3,2%	16,7%	7,7%	0,6%	47,4%	24,4%	-	156
<b>Srednja</b>	4,9%	16,2%	9,2%	1,4%	40,8%	26,8%	0,7%	142
<b>Velika</b>	4,3%	16,0%	7,4%	3,7%	40,5%	27,6%	0,6%	163
<b>Jako velika</b>	8,5%	15,8%	7,9%	5,1%	38,4%	24,3%	-	177
<b>Prosjek</b>	5,5%	16,0%	9,1%	3,3%	40,0%	25,7%	0,8%	
<b>Srednje apsolutno odstupanje</b>	1,6%	0,3%	1,7%	1,8%	3,5%	1,2%	0,2%	

Legenda: T= tradicionalno; A=agilno; O=oboje

Također, svojim su istraživanjem dokazali da tvrtke koriste hibridne metodologije upravljanja projektima neovisno o sektoru industrije za koju se razvija softver, pri čemu je čak 29% ispitanika radilo na razvoju web-aplikacija (Slika 4).



Slika 4. Sektori industrije za koje rade ispitanici [10]

Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S.G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M. [13] svojim su istraživanjem na uzorku od 1.467 ispitanika također dokazali da su hibridne metodologije upravljanja projektima postale uobičajene u suvremenom razvoju softvera te da je uporaba hibridnih metodologija upravljanja projektima jednako zastupljena u razvoju softvera u tvrtkama različite veličine te za različite sektore industrije odnosno dokazali

su da uporaba hibridnih metodologija upravljanja projektima ne ovisi o veličini tvrtke niti o sektoru industrije za koju se razvija softver.

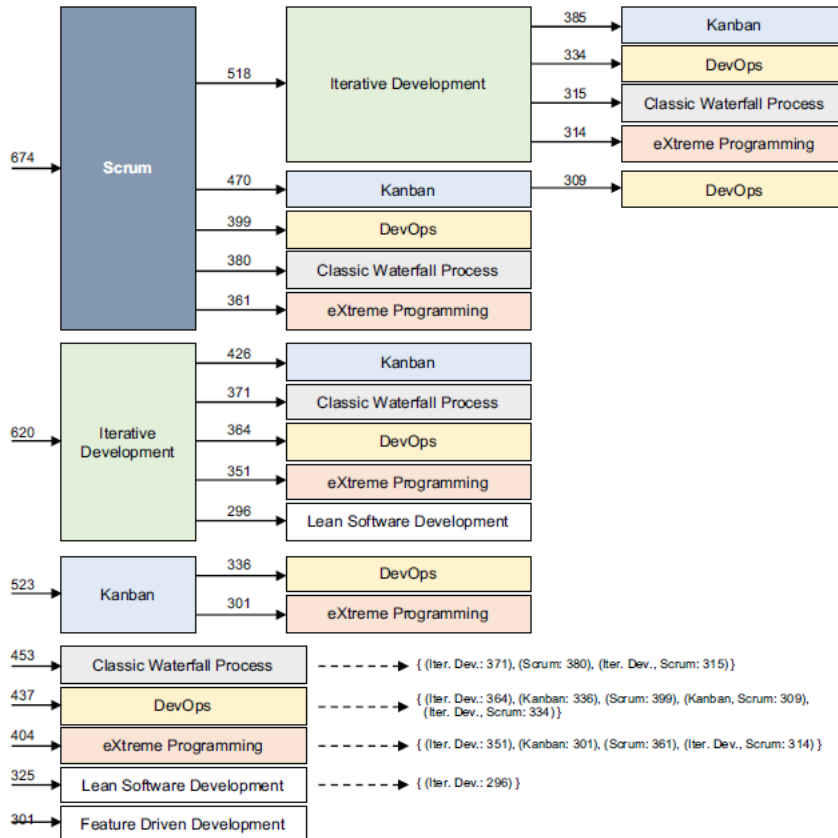
#### **1.4.2. Najzastupljenije metodologije upravljanja projektima u hibridnim modelima**

U svom istraživanju Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S.G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M. [13] istražili su na uzorku od 1.467 ispitanika kroz 38 pitanja koji se okviri, metode i prakse najčešće koriste u primjeni hibridnih metodologija upravljanja projektima. Anketno istraživanje distribuirano je među osobnim kontaktima istraživača, na konferencijama, putem *mailing* listi te društvenih medija, a na uzorku od 1.467 ispitanika prikupljeno je 690 potpunih odgovora.

Kroz istraživanje je identificirano 8 temeljnih metodologija upravljanja projektima od kojih razvojni timovi najčešće kroje svoje hibridne modele, a to su:

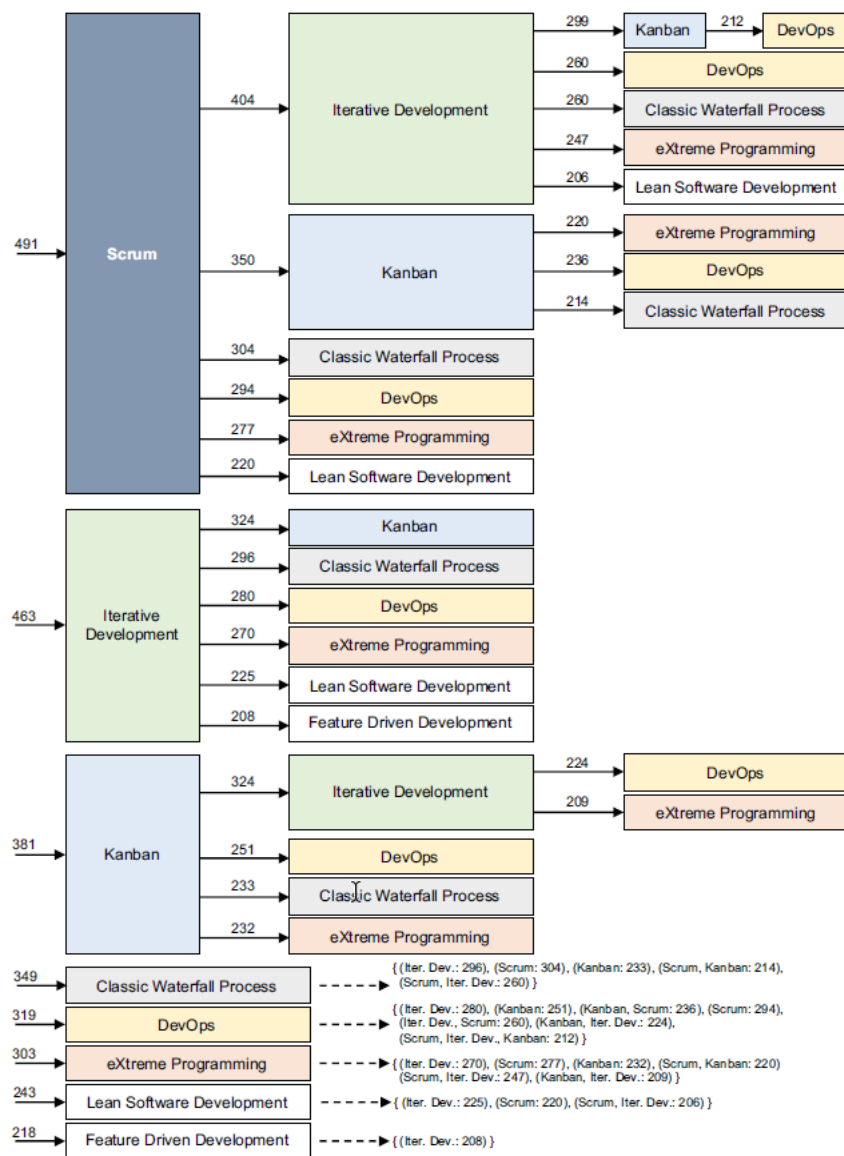
- Scrum
- Iterativni razvoj
- Kanban
- Vodopadni model
- *DevOps*
- *eXtreme Programming*
- *Lean*
- *Feature Driven Development*.

Kada se u obzir uzme cijeli uzorak od svih 1.467 ispitanika ispostavilo se da je Scrum najkorištenija metodologija upravljanja projektima koja se koristi od strane 674 ispitanika, a kombinaciju Scruma i vodopadnog modela (eng. "*Water-Scrum-Fall*") koristi 380 ispitanika. Vodopadni model pak koristi 453 ispitanika. Scrum se našao u velikom broju hibridnih metodologija upravljanja projektima u paru s nekom od ostalih metodologija upravljanja projektima te u svim kombinacijama koje sadrže ukupno 3 različite metodologije upravljanja projektima (Slika 5).



Slika 5. Temeljne metodologije upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na cijelom uzorku ispitanika [13]

Kada se u obzir uzme samo uzorak ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima u praksi, Scrum je i dalje najkorištenija projektna metodologija koju je naveo 491 ispitanik, a u kombinaciji s vodopadnim modelom koristi ga 304 ispitanika (Slika 6).



Slika 6. Temeljne metodologije upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na uzorku ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima [13]

U istraživanju je dodatno ispitanicima na odabir ponuđeno 36 praksi koje se primjenjuju u razvoju softvera (pitanje je bilo postavljeno po principu višestrukog izbora) kako bi se utvrdilo koje su prakse najčešće korištene u kombinaciji:

- Specifikacije arhitekture softvera (*eng. Architecture Specifications*)
- Automatizirano generiranje koda (*eng. Automated Code Generation*)
- Automatizirano dokazivanje teorema (*eng. Automated Theorem Proving*)
- Automatizirano jedinično testiranje (*eng. Automated Unit Testing*)
- Upravljanje *Backlogom* (*eng. Backlog Management*)
- *Burn-down* grafikom (*eng. Burn-down Charts*)

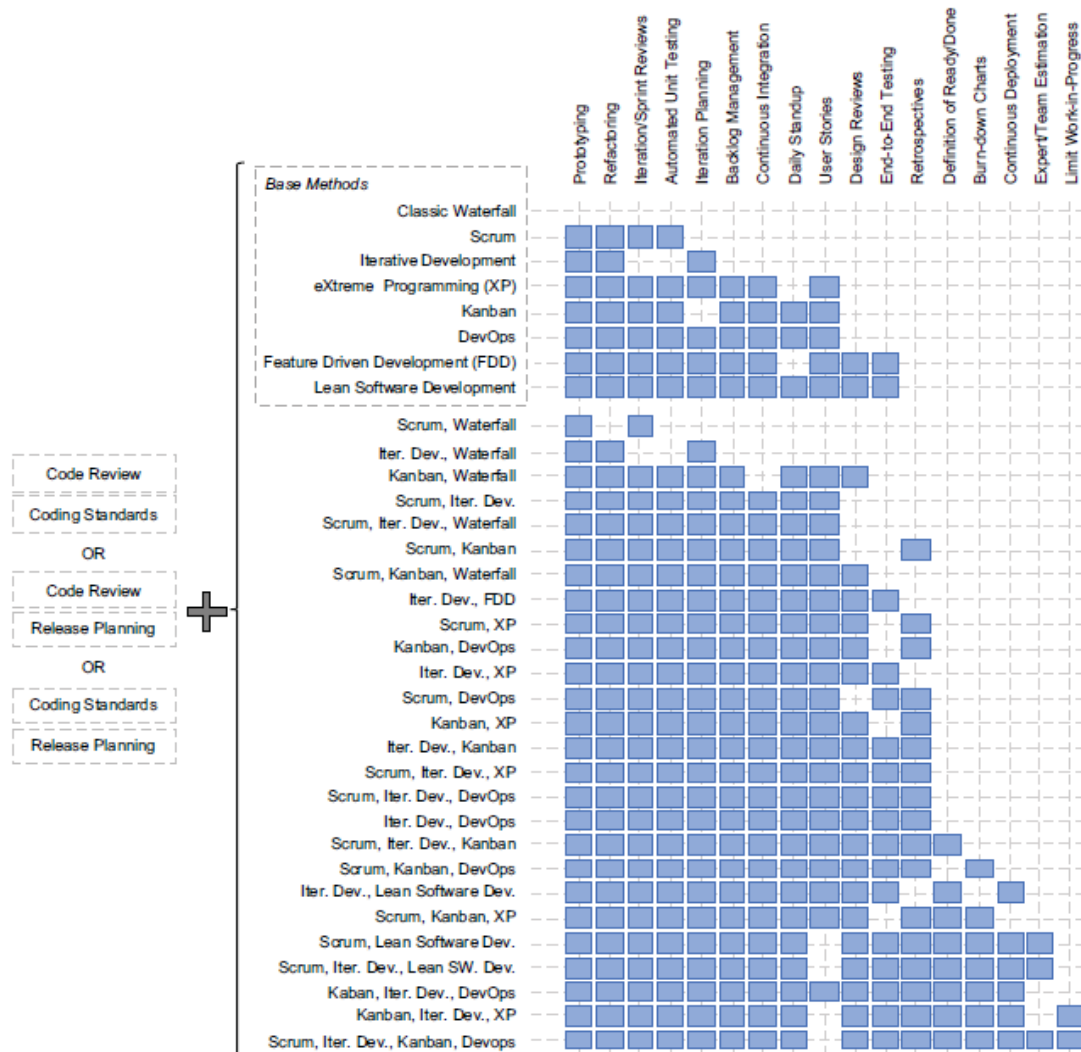
- Pregled koda (*eng. Code Review*)
- Postavljanje standarda u pisanju koda (*eng. Coding Standards*)
- Zajedničko vlasništvo nad kodom (*eng. Collective Code Ownership*)
- Kontinuirana isporuka (*eng. Continuous Deployment*)
- Kontinuirana integracija (*eng. Continuous Integration*)
- Dnevni sastanci (*eng. Daily Standup*)
- Definicija gotovog (*eng. Definition of Ready/Done*)
- Pregled dizajna (*eng. Design Reviews*)
- Destruktivno testiranje (*eng. Destructive Testing*)
- Specifikacije dizajna (*eng. Detailed Designs/Design Specifications*)
- Testiranje od kraja do kraja odnosno cijelog korisničkog scenarija (*eng. End-to-End Testing*)
- Procjena stručnjaka ili tima (*eng. Expert/Team Estimation*)
- Formalne procjene (*eng. Formal Estimation*)
- Formalne specifikacije (*eng. Formal Specifications*)
- Planiranje iteracija (*eng. Iteration Planning*)
- Pregled iteracije/*Sprinta* (*eng. Iteration/Sprint Reviews*)
- Ograničavanje rada u tijeku (*eng. Limit Work-in-Progress*)
- Provjera odgovara li finalni model sustava specifikaciji (*eng. Model Checking*)
- Klijent na licu mjesta (*eng. On-Site Customer*)
- Programiranje u paru (*eng. Pair Programming*)
- Prototipiranje (*eng. Prototyping*)
- Refaktoriranje koda (*eng. Refactoring*)
- Planiranje isporuke odnosno planiranje puštanja rješenja u produkciju (*eng. Release Planning*)
- Retrospektive (*eng. Retrospectives*)
- Podjela velikih timova na manje agilne timove od 5 do 10 osoba (*eng. Scrum of Scrums*)
- Sigurnosno testiranje (*eng. Security Testing*)
- Razvoj temeljen na testiranju (*eng. Test-driven development*)
- Modeliranje korisničkih scenarija (*eng. Use Case Modeling*)
- Korisničke priče (*eng. User Stories*)
- Planiranje temeljeno na brzini izvedbe (*eng. Velocity-based Planning*).

Istraživanje je dokazalo da se kombinacija 2 od sljedeće 3 prakse razvoja softvera ponavlja u preko 85% odgovora svih ispitanika neovisno o korištenoj metodologiji upravljanja projektima, a to su:

- Pregled koda (eng. *Code review*)
- Postavljanje standarda u pisanju koda (eng. *Coding Standards*)
- Planiranje isporuke odnosno planiranje puštanja rješenja u produkciju (eng. *Release Planning*) (Slika 7).

Na uzorku samo onih ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima, preko 85% ispitanika složilo se oko 5 praksi, pri čemu su prve 3 jednake onima identificiranim na cijelom uzorku, a dodatne 2 prakse bile su:

- Prototipiranje (eng. *Prototyping*)
- Automatizirano jedinično testiranje (eng. *Automated Unit Testing*) (Slika 7).



Slika 7. Prikaz temeljnih metodologija upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na uzorku ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima [13]

Istraživanje HELENA [14] ukazuje na 6 najpopularnijih pristupa i praksi u razvoju softvera:

- 1) Pregled koda (eng. Code review)
- 2) Postavljanje standarda u pisanju koda (eng. *Coding standards*)
- 3) Kontinuirana integracija (eng. *Continuous integration*)
- 4) Refaktoriranje koda (eng. *Refactoring*)
- 5) Scrum
- 6) Jedinično testiranje (eng. *Unit testing*) (Slika 8).

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	Total
Agile Portfolio Management (APM)																0
Behavior-driven Development (BDD)	1				1											2
Code Review	1	1			1	1			1	1	1	1	1	1	1	11
Coding Standards	1				1	1	1	1		1	1		1	1	1	10
Collective Code Ownership					1											1
Continuous Deployment									1		1		1	1	1	5
Continuous Integration	1				1	1		1	1		1		1	1	1	9
Crystal Family																0
Daily Standups								1								1
Definition of Done	1				1			1		1						4
Definition of Ready	1									1						2
DevOps								1	1							2
Digital Taskboard											1					1
Disciplined Agile Delivery (DAD)																0
Expert-/Team-based estimation (e.g., Planning Poker)							1	1								2
eXtreme Programming (XP)	1											1				2
Feature-Driven Development (FDD)		1								1		1				3
Formal Estimation (e.g., COCOMO, FP)																0
Formal Specification						1										1
Iteration Planning	1	1			1											3
Iterative Development	1				1	1								1		4
Kanban											1				1	2
Large-Scale Scrum (LeSS)																0
Lean Development							1				1					2
On-Site Customer																0
Pair Programming					1									1		2
Prototyping										1		1		1	1	4
Rational Unified Process (custom variant)				1												1
Rational Unified Process (standard version)																0
Refactoring		1			1	1	1		1		1		1	1	1	9
Release Planning		1			1	1					1			1		5
Retrospectives	1				1			1								3
Scaled Agile Framework (SAFe)																0
Scrum	1	1						1	1			1		1	1	7
Spiral Model															1	1
Test-Driven Development (TDD)	1				1			1							1	4
Unit Testing	1	1			1	1		1	1	1		1	1		1	10
V-Model Derivate(s)													1	1		2
Waterfall/Phase Model	1							1				1			1	4
Other			1		1											2

Slika 8. Najpopularniji pristupi i prakse u razvoju projekata [14]

Noll, J. i Beecham, S. u svome radu [15] analizirali su dodatno rezultate istraživanja HELENA s ciljem utvrđivanja koliko projekata kombinira agilne i tradicionalne prakse u razvoju softvera. Istraživanje je provedeno na gotovo 700 projekata, a utvrđeno je da je na njih 66% korištena hibridna metodologija upravljanja projektima. Na 30% projekata nikada nisu korištene tradicionalne metode.

U Tablici 3 kolona TOTAL prikazuje koliko se često projekti izvode po agilnim metodologijama, a redak TOTAL koliko se često projekti izvode prema tradicionalnim metodologijama. Podaci iskazani **podebljano** klasificirani su kao korištenje hibridnih metodologija, a oni označeni **osjenčano** kao čisti agilni pristup.



Tablica 3. Učestalost kombiniranja tradicionalnih i agilnih metoda [15]

Agilne metode/Tradicionalne metode	Nikad	Rijetko	Ponekad	Često	Uvijek	TOTAL
<b>Nikad</b>	(21) 3%	(1) 0%	(3) 0%	(10) 1%	(8) 1%	(43) 6%
<b>Rijetko</b>	(2) 0%	<b>(1) 0%</b>	<b>(3) 0%</b>	<b>(13) 2%</b>	<b>(5) 1%</b>	(24) 3%
<b>Ponekad</b>	(16) 2%	<b>(9) 1%</b>	<b>(24) 3%</b>	<b>(24) 3%</b>	<b>(17) 2%</b>	(90) 13%
<b>Često</b>	<b>(55) 8%</b>	<b>(53) 8%</b>	<b>(56) 8%</b>	<b>(82) 12%</b>	<b>(20) 3%</b>	(266) 39%
<b>Uvijek</b>	<b>(116) 17%</b>	<b>(44) 6%</b>	<b>(42) 6%</b>	<b>(33) 5%</b>	<b>(32) 5%</b>	(267) 39%
<b>TOTAL</b>	(210) 30%	(108) 16%	(128) 19%	(162) 23%	(82) 12%	(690) 100%

Uočeno je da je na projektima na kojima je korištena hibridna projektna metodologija kao kombinacija tradicionalnog i agilnog pristupa znatno manja vjerojatnost korištenja agilnih praksi u odnosu na projekte na kojima su korištene isključivo agilne metode.

Tablica 4 prikazuje kako su pojedine prakse - Refaktoriranje koda, Kontinuirana integracija, Kontinuirana isporuka, Zajedničko vlasništvo nad kodom, Definicija gotovog te Dnevni sastanci, korištene na većini projekata vođenih po agilnim metodologijama upravljanja projektima dok su se u projektima vođenim po hibridnim metodologijama upravljanja projektima iste koristile znatno manje.

Prakse Programiranje u paru i Razvoj temeljen na testiranju koristile su se značajno češće na projektima vođenim po agilnim modelima, iako ne na većini.

Suprotno tome, Formalne procjene i Formalne specifikacije, prakse karakteristične za tradicionalni pristup, korištene su na manjini projekata vođenih po hibridnim metodologijama upravljanja projektima, a značajno manje na projektima vođenim po agilnim modelima.

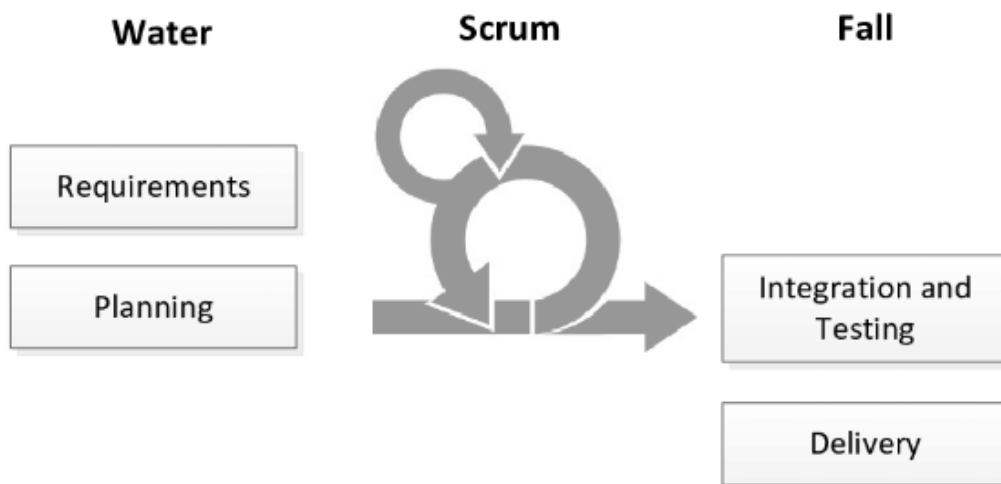
Tablica 4. Usporedba korištenih praksi na uzorku od 171 agilno i 458 hibridno vođenih projekata [15]

Prakse	Agilne metodologije (n=171)	Hibridne metodologije (n=458)	Razlika
1. Retrospektive	68%	48%	20%
2. Refaktoriranje koda	63%	43%	20%
3. Kontinuirana integracija	75%	56%	19%
4. Kontinuirana isporuka	56%	38%	18%
5. Automatizirano testiranje	67%	50%	17%
6. Programiranje u paru	30%	15%	15%
7. Procjena stručnjaka ili tima	56%	41%	15%
8. Upravljanje <i>Backlogom</i>	75%	60%	15%
9. Ograničavanje rada u tjeku	44%	30%	14%
10. Zajedničko vlasništvo nad kodom	51%	37%	14%
11. Pregled koda	73%	59%	14%
12. Planiranje temeljeno na brzini izvedbe	36%	23%	13%
13. Razvoj temeljen na testiranju	33%	20%	13%
14. Definicija gotovog	62%	49%	13%
15. Dnevni sastanci	69%	56%	13%
16. Pregled iteracije/ <i>Sprinta</i>	69%	58%	11%
17. Korisničke priče	63%	53%	10%
18. Planiranje isporuke odnosno planiranje puštanja rješenja u produkciju	69%	65%	4%
19. Klijent na licu mjesta	20%	19%	9%
20. Planiranje iteracija	58%	57%	1%
21. Podjela velikih timova na manje agilne timove od 5 do 10 osoba	19%	19%	0%
22. Postavljanje standarda u pisanju koda	70%	71%	-1%
23. Automatizirano dokazivanje teorema	2%	4%	-2%
24. Sigurnosno testiranje	25%	29%	-4%
25. <i>Burn-down</i> grafikon	37%	41%	-4%
26. Testiranje od kraja do kraja odnosno cijelog korisničkog scenarija	50%	55%	-5%
27. Automatizirano generiranje koda	16%	21%	-5%
28. Provjera odgovara li finalni model sustava specifikaciji	4%	10%	-6%
29. Destruktivno testiranje	6%	12%	-6%
30. Pregled dizajna	36%	43%	-7%
31. Prototipiranje	33%	41%	-8%
32. Formalne procjene	2%	13%	-11%
33. Modeliranje korisničkih scenarija	14%	30%	-16%
34. Specifikacije arhitekture softvera	32%	49%	-17%
35. Formalne specifikacije	6%	24%	-18%
36. Specifikacije dizajna	16%	34%	-18%

Pretpostavka koja proizlazi iz istraživanja Noll, J. i Beecham, S. [15] jest da način razmišljanja, točnije agilnost organizacije, a ne specifičnost i potrebe pojedinog projekta, određuje hoće li se na projektu usvojiti hibridni ili čisti agilni pristup. Rezultati istraživanja sugeriraju da hibridni modeli često zamjenjuju agilne prakse tradicionalnim praksama u početnim fazama projekta (što je posebice uočljivo na primjeni praksi modeliranja korisničkih scenarija, izradi specifikacija arhitekture softvera, davanju formalnih specifikacija i specifikacija dizajna), umjesto da proširuju agilne metodologije upravljanja projektima tradicionalnim praksama.

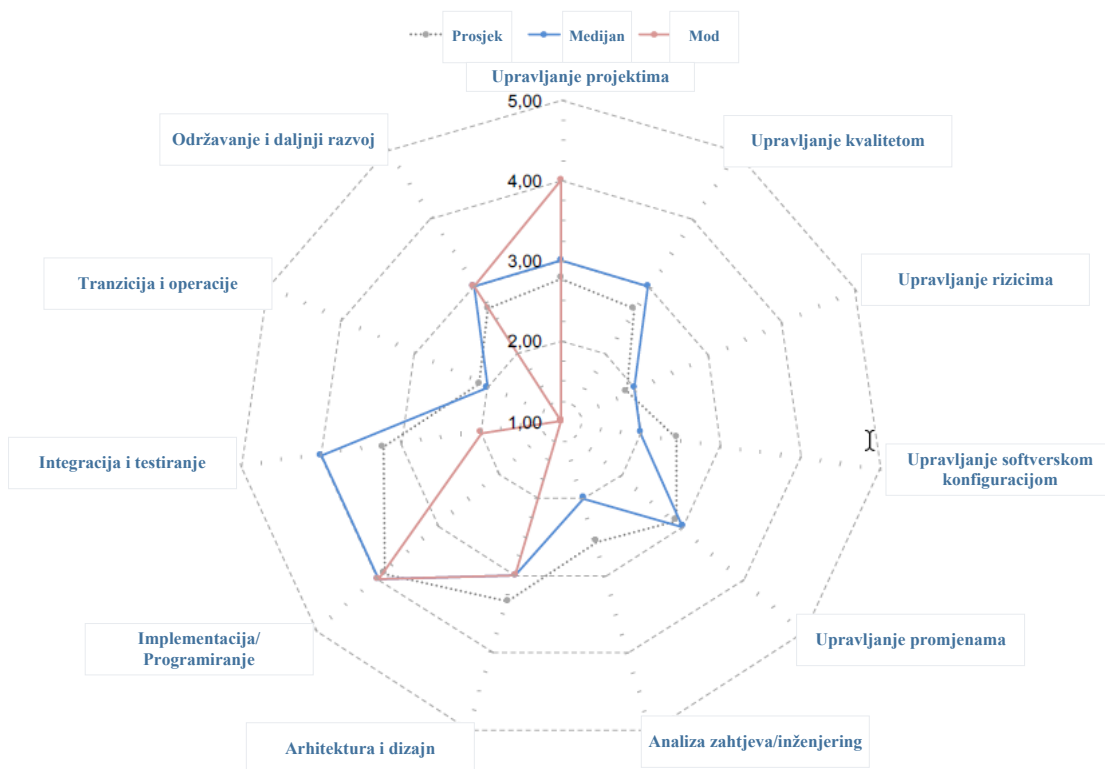
### 1.4.3. Hibridni modeli temeljeni na vodopadnom modelu i Scrumu

Kuhrmann, M., Münch, J., Diebold, P., Linssen, O., Prause, C.R. u svome radu [14] potvrđuju kako je „*Water-Scrum-Fall*“ (Slika 9) postao sve zastupljenija metodologija upravljanja projektima u brojnim poduzećima.



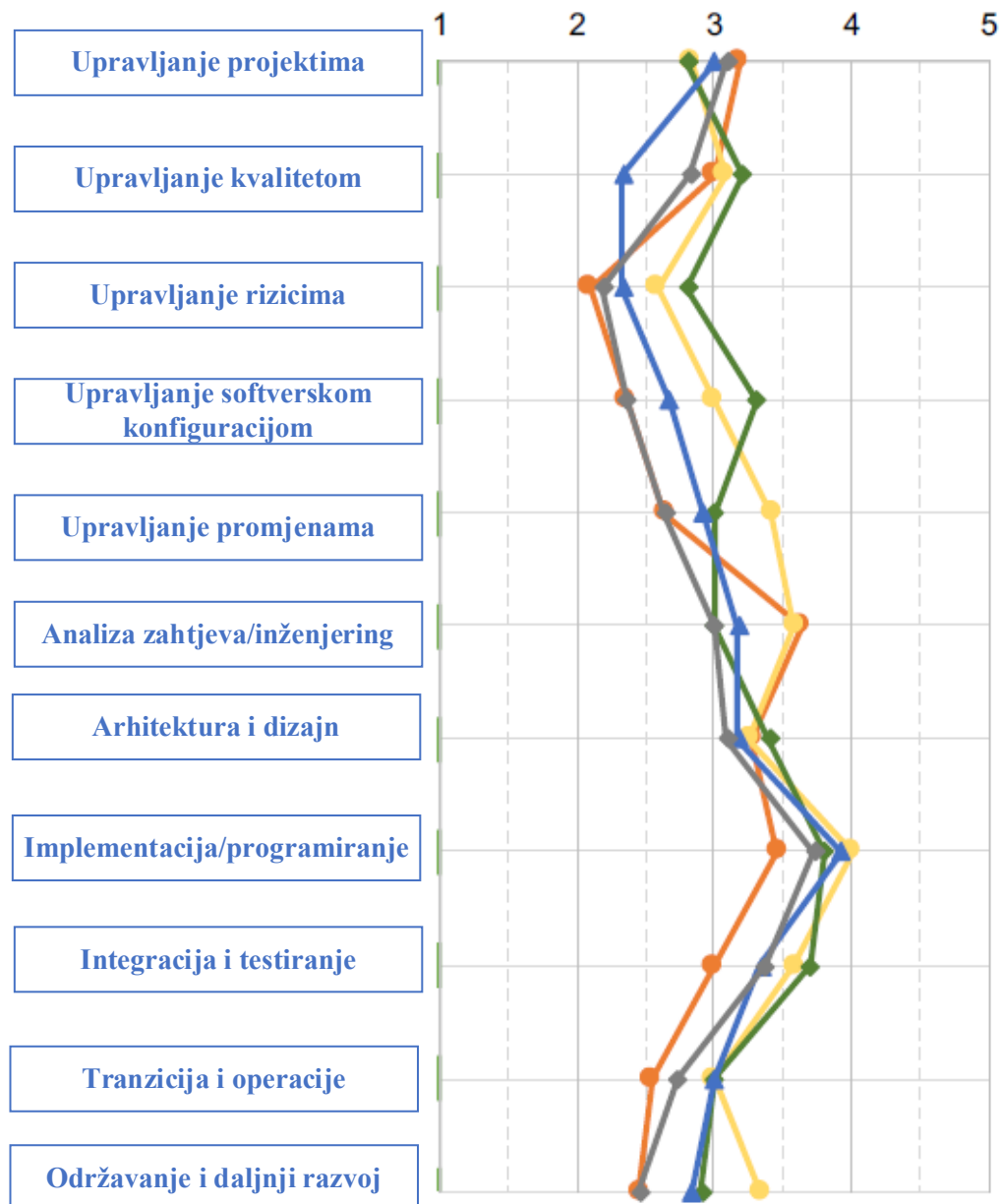
Slika 9. Water-Scrum-Fall [16]

Istraživanje je ukazalo da je agilni pristup zastupljen u aktivnostima vezanim za sami razvoj softvera dok su aktivnosti menadžmenta i dalje sklonije tradicionalnom pristupu. Upravljanje rizicima, konfiguracijom, tranzicijom i operacijama ukazuju na korištenje tradicionalnog pristupa. S druge strane, implementacija i programiranje, integracija i testiranje te izrada arhitekture i dizajna prednjače po pitanju agilnog pristupa. Sukladno rezultatima, upravljanje projektima uravnoteženo je između tradicionalnog i agilnog pristupa potvrđujući tezu sve češćeg korištenja hibridnih metodologija upravljanja projektima, s blagom tendencijom prema agilnom pristupu (Slika 10).



Slika 10. Zastupljenost tradicionalnog i agilnog pristupa projektnim aktivnostima po Likertovoj skali (1 – tradicionalni pristup; 3 – kombinacija tradicionalnog i agilnog pristupa; 5 – agilni pristup) [14]

Slični su rezultati dobiveni i u istraživanju [10] na uzorku od 56 ispitanika, gdje je vrlo očito da tvrtke kombiniraju „najbolje od oba svijeta“ kada su u pitanju tradicionalni i agilni pristup upravljanja projektima, a posebno je uočljiva tendencija prema agilnom pristupu za aktivnost Implementacije i programiranja (Slika 11).



Slika 11. Pregled tradicionalnog i agilnog pristupa u različitim aktivnostima razvoja softvera po Likertovoj skali ( 1 – tradicionalni pristup; 3 – kombinacija tradicionalnog i agilnog pristupa; 5 – agilni pristup) [10]

U svom radu Hayata, T. i Han, J. [17] bavili su se temom usvajanja agilnih metodologija upravljanja projektima u poslovnim organizacijama te predložili novi pristup razvoju i upravljanju IT (*Information technology*) projektima spajanjem Scruma i tradicionalnih metodologija upravljanja projektima, utemeljenih na planiranju svih aktivnosti unaprijed. U svom su radu predložili hibridni model koji aplicira Scrum u tradicionalnu metodologiju upravljanja projektima.

Mnogi istraživači ove tematike kreću od pretpostavke da su vodopadni model i agilne metodologije upravljanja projektima dva potpuno odvojena procesa, pri čemu se za velike projekte kao što je razvoj poslovnih aplikacija treba koristiti vodopadni model dok se agilne metodologije upravljanja projektima mogu primijeniti na projekte manjeg opsega. Hayata, T. i Han, J. [17] pak polaze od premise da se većina projekata razvoja softvera ipak svodi na unaprjeđenje postojećeg softvera, a ne na razvoj potpuno novog softvera te da ima smisla na razvoj softvera gledati kao na kontinuum razvoja i održavanja softvera. U tom smislu, njihova je pretpostavka da na velikom broju projekata krajnji korisnici već znaju što žele od funkcionalnosti softvera i fokus stavljaju na brzinu razvoja softvera kako bi što prije mogli početi koristiti isti i kapitalizirati njegovu poslovnu vrijednost. Članovi razvojnog tima u takvim slučajevima mogu odmah krenuti na fazu dizajna i implementacije rješenja, koristeći agilne metode.

Predloženi hibridni model apliciranja Scruma u tradicionalne metodologije upravljanja projektima naziva se V modelom, a zasnovan je na vodopadnom modelu na početku (eng. „*Waterfall-Up-Front*“) i vodopadnom modelu na kraju projekta (eng. „*Waterfall-At-End*“) u kojima se inicijalne aktivnosti (Analiza korisničkih zahtjeva i Izrada inicijalnog dizajna aplikacije) te završne aktivnosti (Završno testiranje i Integracijsko testiranje) provode po vodopadnom modelu uz detaljno planiranje aktivnosti unaprijed, dok se aktivnosti detaljnog dizajniranja aplikacije, implementacija i repetitivno testiranje odvijaju iterativno po načelima agilnih modela (Slika 12).

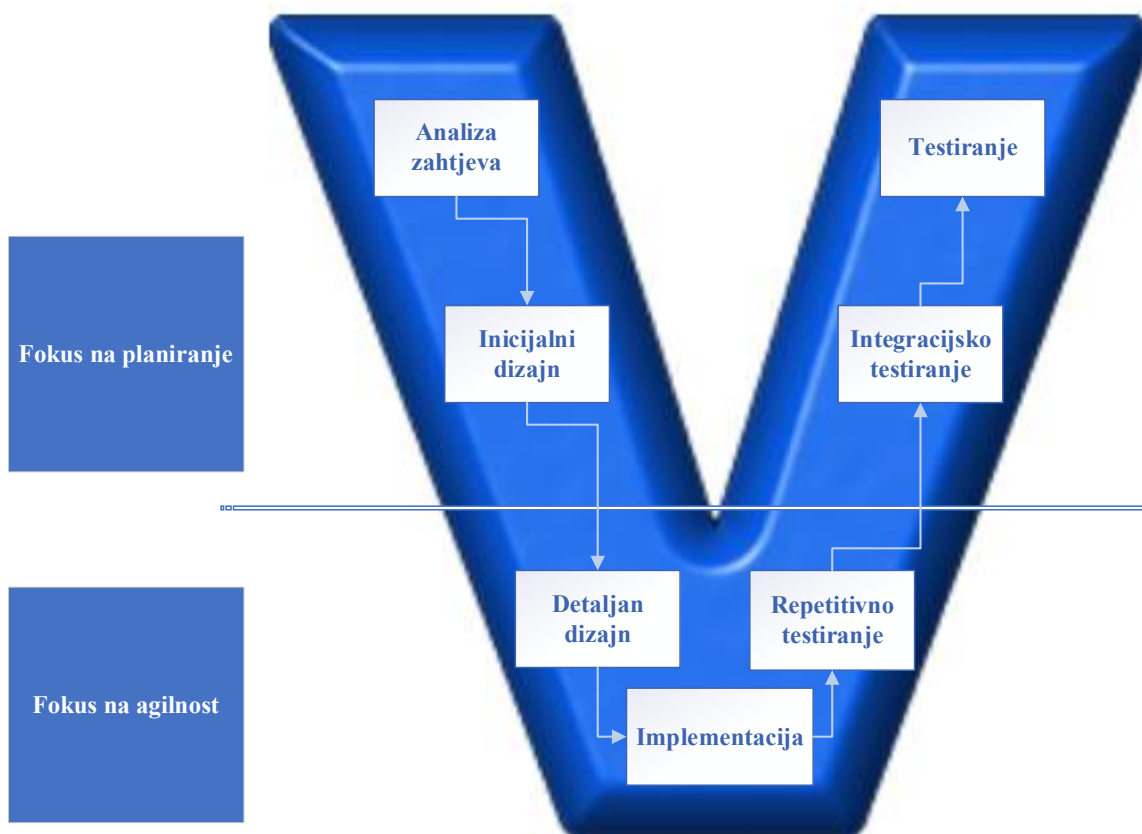
V model u literaturi prvi puta spominje Boehm, W.B. 1979. godine, a u kontekstu razvoja programskih sustava predstavili su ga Forsberg, K. i Mooz, H. na konferenciji NCOSE 1991. godine [18].

Koristi V modela su trostruke:

- 1) Projektni razvojni tim i korisnici specificiraju zahtjeve i kreiraju projektnu dokumentaciju po vodopadnom modelu te na temelju toga sklapaju ugovor kako bi se jasno utvrdile očekivane isporuke i ciljevi projekta te smanjio rizik za obje strane
- 2) U dizajniranju, implementaciji i testiranju primjenjuju se agilne metode, što ubrzava navedene aktivnosti i smanjuje rizik od kašnjenja i pomicanja rokova, koji se često

javljaju u sekvencijalnim tradicionalnim metodologijama upravljanja projektima (gdje jedna faza projekta mora završiti kako bi iduća započela)

- 3) Naposljetku, projektni razvojni tim i korisnici primjenjuju vodopadni model za završna testiranja i prihvata funkcionalnosti kako bi primopredajni proces bio formalan i programsko rješenje isporučeno korisniku.

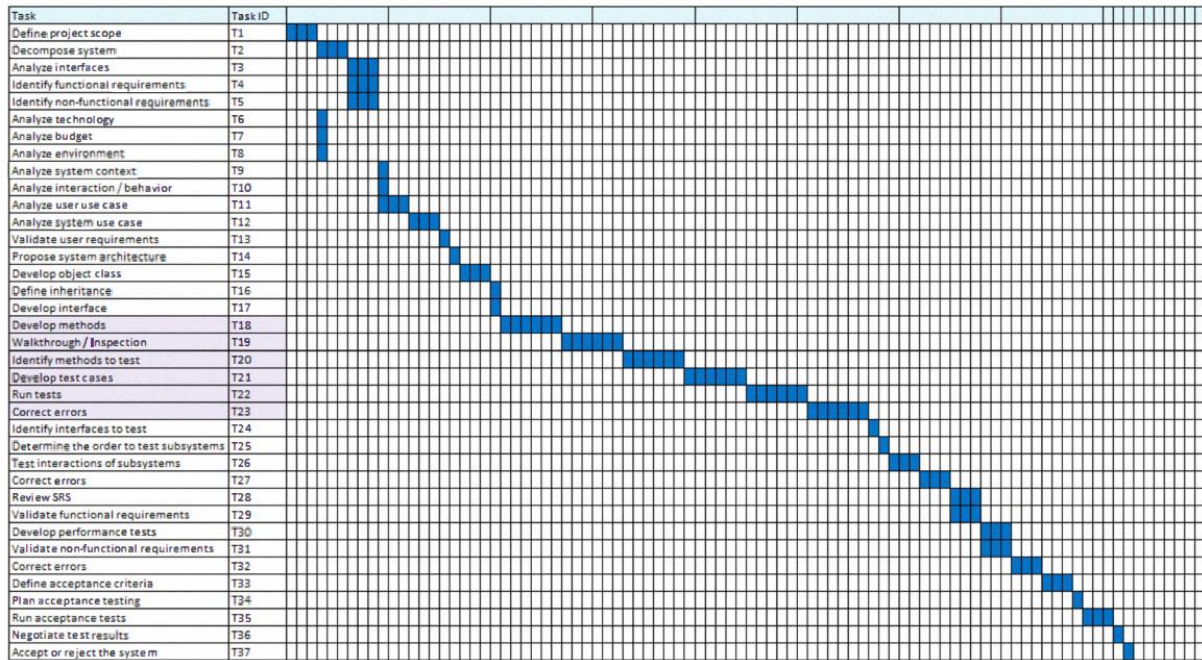


Slika 12. Hibridni model „V“ [17]

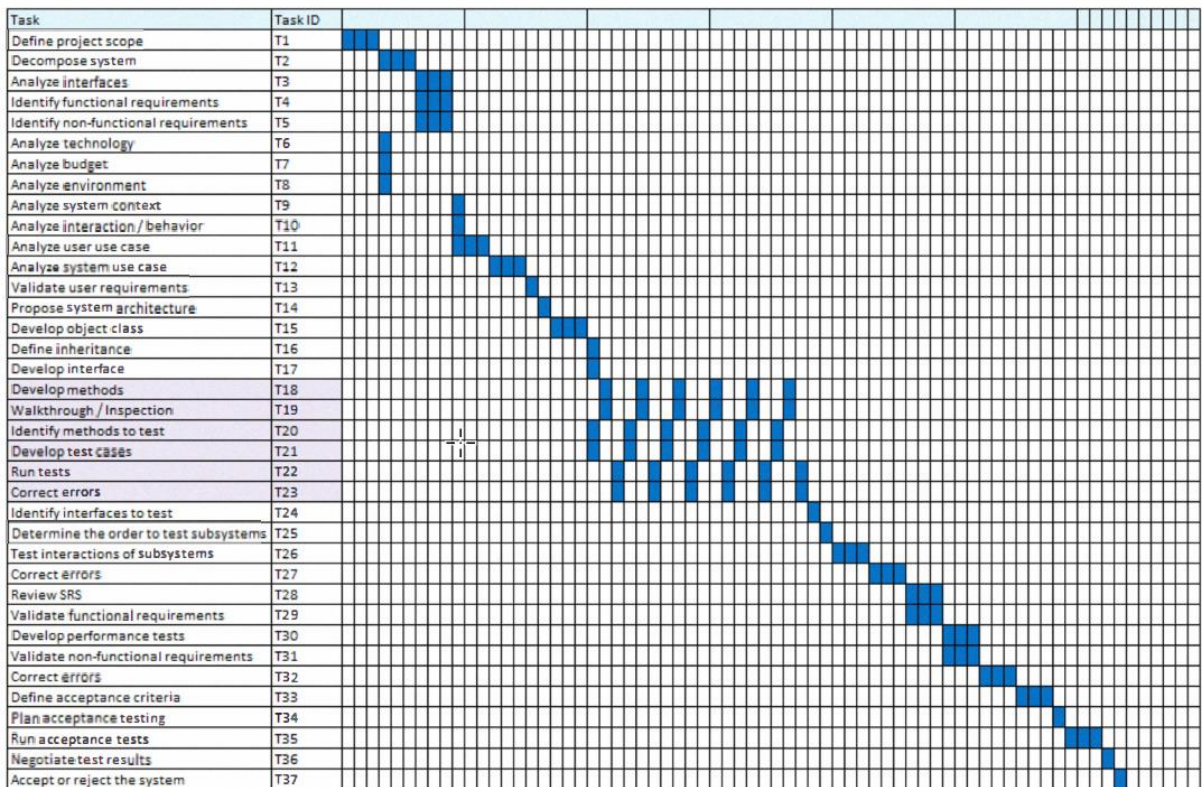
Nadalje, Hayata, T. i Han, J. [17] prikazali su kako je korištenje hibridnih modela temeljenih na uključivanju Scruma u vodopadni model korisno u smislu skraćivanja vremena trajanja projekata razvoja softvera.

Predstavili su isti tromjesečni projekt (90 dana) po vodopadnom modelu gdje se svaka faza mora završiti kako bi iduća započela (Slika 13) te po hibridnom V modelu u kojem se testiranje i implementacija rade paralelno i iterativno po načelima Razvoja temeljenog na testiranju (eng. *Test driven development*) (Slika 14).

Uz pretpostavku da se uzme upravljačka rezerva od 7 dana, njihov je zaključak da će projekt u slučaju primjene vodopadnog modela biti završen za 83 dana, a u slučaju primjene hibridnog V modela uz uštedu vremena od 23% odnosno za 64 dana, s uložnim istim naporom u čovjek-danima kao da je primijenjen vodopadni model.



Slika 13. Tromjesečni projekt vođen po vodopadnom modelu [17]



Slika 14. Tromjesečni projekt vođen po V modelu [17]



#### 1.4.4. Utjecaj primjene hibridnih modela na uspješnost projekta

Istraživanja u području metodologija upravljanja projektima većinom se svode na to trebaju li metodologije upravljanja projektima biti standardizirane ili je metodologiju potrebno prilagoditi svakom pojedinom projektu. Generalno, rezultati brojnih istraživanja ukazuju na to da su uspješniji oni projekti na kojima je korištena metodologija upravljanja projektom od onih na kojima nije uopće korištena metodologija upravljanja projektom, no isto tako ukazuju i na činjenicu da je prisutan značajan udio projekata vođenih po metodologijama upravljanja projektima koji nisu bili uspješni.

Joslin, R. i Müller, R. u svome istraživanju [19] fokusirali su se na vezu između metodologije upravljanja projektom i uspješnosti projekta, ali uzimajući u obzir specifičnosti projekata i projektno okruženje. Istraživanje je provedeno u formi polustrukturiranih dubinskih intervjua na uzorku od 19 ispitanika iz 11 različitih sektora industrije, uključujući razvoj softvera.

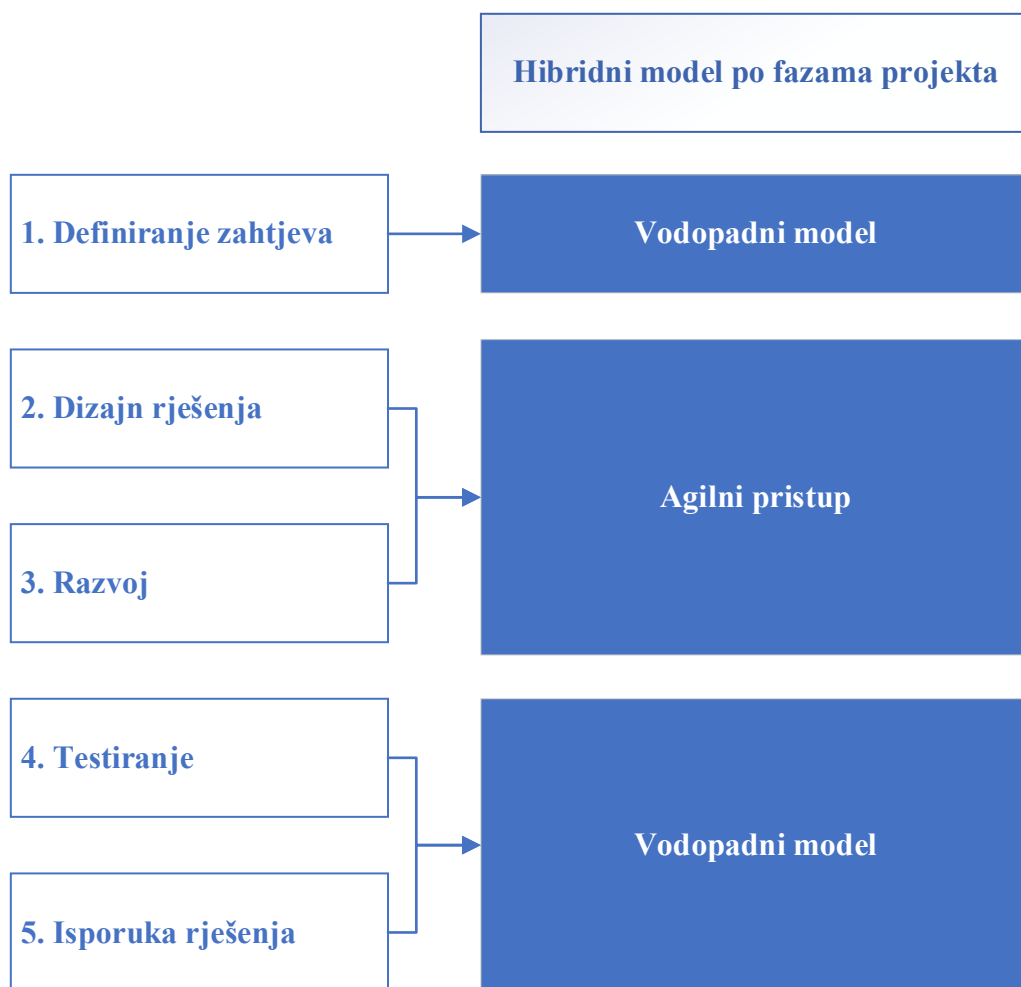
Rezultati istraživanja ukazali su da postoji pozitivna veza između metodologije upravljanja projektom i uspjeha projekta odnosno da odabrana metodologija upravljanja projektom utječe na uspješnost projekta.

Također, rezultati su pokazali da se ovisno o industriji i tipovima projekata koriste različite metodologije upravljanja projektima. Većina metodologija upravljanja projektima bila je prilagođena potrebama projekta i organizacije – 65% ispitanika reklo je da koristi metodologije utemeljene na međunarodnim normama, ali od toga se njih 75% njih izjasnilo kako su ih prilagodili svojim potrebama do određene razine. Ukupno 35% ispitanika kazalo je kako u njihovoj organizaciji postoji više od jedne metodologije upravljanja projektom po tipu projekta. Navedeno ukazuje na očitu potrebu organizacija da koriste specijalizirane metodologije ovisno o vrstama projekata i području poslovanja. Zaključak jest da vanjski čimbenici (projektno okruženje) i specifičnosti projekta imaju moderirajući učinak na vezu između metodologije upravljanja projektom i uspješnosti projekta. Pokazatelj toga je upravo činjenica da su organizacije ispitanika uložile u krojenje metodologija upravljanja projektima sukladno svojim potrebama.

Imani, T., Nakano, M. i Anantatmula, V. u svom su radu [20] otišli korak dalje i istražili utjecaj primjene hibridnih metodologija upravljanja projektima na uspješnost projekta. Proveli su kvantitativno istraživanje na uzorku od 117 ispitanika te 2 kvalitativna istraživanja u Japanu kako bi usporedili isplativost hibridnih, tradicionalnih i agilnih metodologija upravljanja projektima uzimajući u obzir veličinu projekata i učestalost promjena u tijeku projekta. U

kvantitativnom istraživanju preko 50% projekata koji su bili predmet istraživanja imali su više od 20 članova projektnog tima te je preko 50% projekata koji su bili predmet istraživanja trajalo dulje od godine dana, 70% ispitanika bili su voditelji projekata, a 70% od njih imali su preko 10 godina radnog iskustva u upravljanju projektima.

U svome su se radu fokusirali na hibridni model po fazama projekta u kojem se, slično kao i u već spomenutom V modelu, faza definiranja zahtjeva provodi po vodopadnom modelu, faza dizajna i razvoja po agilnom, a zatim testiranje i isporuka korisnicima ponovno po vodopadnom modelu (Slika 15).



Slika 15. Hibridni model po fazama projekta [20]

Hipoteze istraživanja bile su sljedeće:

- 1) Hibridni pristup po fazama može biti skalabilan (dakle, može se koristiti za projekte različitih veličina) za projekte s visokim stupnjem nesigurnosti po pitanju korisničkih zahtjeva.

2) Hibridni pristup po fazama povećava uspješnost projekta.

Rezultati istraživanja ukazali su na sljedeće:

- Hibridni pristup korišten je u projektima s višom stopom promjene zahtjeva u tijeku projekta u odnosu na projekte kojima je upravljano po vodopadnom modelu (Tablica 5)
- Hibridni modeli korišteni su na projektima većeg opsega od onih na kojima su se koristile isključivo agilne metodologije upravljanja projektima (Tablica 5)
- Hibridni modeli pokazali su se uspješnijima u odnosu na tradicionalni vodopadni model i agilne metodologije upravljanja projektima kada su u pitanju troškovi projekta (Tablica 6)
- Agilne su metodologije zauzele 1. mjesto po pitanju kvalitete izvedbe projekta i zadovoljstva sponzora projekta (Tablica 6).

Tablica 5. Prosječne vrijednosti veličine i trajanja projekta te stope promjena zahtjeva u tijeku projekta po Likertovoj skali ( 1 – malo; 5 – puno) [20]

Metodologija upravljanja projektima	Veličina projekta	Trajanje projekta	Promjene zahtjeva u tijeku projekta
Hibridna	1,64	3,45	3,47
Agilna	1,56	3,67	<b>3,89</b>
Vodopadni model	<b>2,21</b>	<b>3,73</b>	3,10

Tablica 6. Prosječne vrijednosti uspješnosti po pitanju kvalitete, troškova i zadovoljstva sponzora projekta po Likertovoj skali (1 – malo; 5 – puno) [20]

Metodologija upravljanja projektima	Kvaliteta	Trošak	Zadovoljstvo sponzora projekta
Hibridna	4,04	<b>3,81</b>	4,19
Agilna	<b>4,33</b>	3,78	<b>4,39</b>
Vodopadni model	3,98	3,52	4,15

Dodatno su provedena 2 kvalitativna istraživanja.

Prvo je provedeno na velikom projektu implementacije sustava za nabavu u IT tvrtki. Na projektu je radilo više timova, a korisnici su se nalazili u 6 različitih država. Sponzoru projekta bilo je najvažnije da projekt bude dovršen za 9 mjeseci, iako oko 50% zahtjeva na početku

projekta još nije bilo jasno definirano. Projektni tim radio je po hibridnom modelu – na početku su proveli 2,5 mjeseca u definiranju zahtjeva po vodopadnom modelu, a zatim su u fazi razvoja koristili Scrum metodologiju. Voditelj projekta izjavio je kako procjenjuje da je primjena Scruma rezultirala smanjenjem troška rada za oko 15%, ali da isto tako smatra ključnim planiranje arhitekture sustava i definiranje zahtjeva po vodopadnom modelu na početku projekta.

Drugo je kvalitativno istraživanje provedeno za mali projekt na kojem je radio jedan razvojni tim, a riječ je bila o implementaciji web-aplikacije za upravljanje ljudskim potencijalima u tvrtki. Programski sustav zahtijevao je visoki nivo sigurnosti i rigidnu kontrolu pristupa. Postojeći sustav nije bio dobro dokumentiran pa zahtjevi na početku projekta nisu bili jasno definirani. Projektni tim radio je po hibridnom modelu na način da su u početnoj fazi dizajniranja programskog sustava koristili vodopadni model, a u fazi detaljnog dizajniranja sustava i samoj implementaciji Scrum. Testiranje je provedeno prema vodopadnom modelu. Iako se u tijeku projekta dogodilo čak 20 promjena po pitanju zahtjeva, projekt je ostvario 8% manje troškova nego što bi ostvario da je vođen po vodopadnom modelu. Navedeno je utvrđeno u suradnji s voditeljem projekta, prema čijoj su kalkulaciji dodatni troškovi uslijed promjena zahtjeva u tijeku projekta trebali iznositi do 20% ukupne vrijednosti projekta da se koristio isključivo vodopadni model jer bi se tada isporučilo rješenje sukladno inicijalno definiranim korisničkim zahtjevima, a zatim bi se inicijalno isporučeni softver morao prekrajati i u drugoj fazi isporučiti s traženim promjenama. Trošak isporuke rješenja prema početnim zahtjevima i trošak naknadnog prekrajanja procijenjeni su na 8% vrijednosti projekta, no do navedenog troška nije došlo jer je korištena hibridna metodologija upravljanja projektom pa su promjene odmah ugrađene u inicijalno isporučenu verziju softvera te je sukladno tome dodatni trošak izazvan promjenama korisničkih zahtjeva u tijeku projekta bio za 40% manji od predviđenog.

Kvantitativno i oba kvalitativna istraživanja potvrdila su postavljene hipoteze – da hibridni modeli mogu biti uspješno primijenjeni na projekte različitih veličina te da pozitivno utječu na uspješnost projekata, posebice kada su u pitanju troškovi odnosno njihova redukcija. Istraživanje je ukazalo na to da hibridni modeli omogućuju uštede u odnosu na vodopadni model na velikim projektima razvoja programskih sustava za 8 do 15%.

### **1.4.5. Komunikacija s korisnicima u hibridnim modelima sastavljenim od vodopadnog modela i agilnih projektnih metodologija**

S obzirom na trend formiranja hibridnih metodologija upravljanja projektima uključivanjem agilnih projektnih metodologija u vodopadni model, Schmitz, K., Mahaparta, R. i Nerur, S. u svom su radu [21] istražili uključenost korisnika na projektima vođenim po hibridnim modelima.

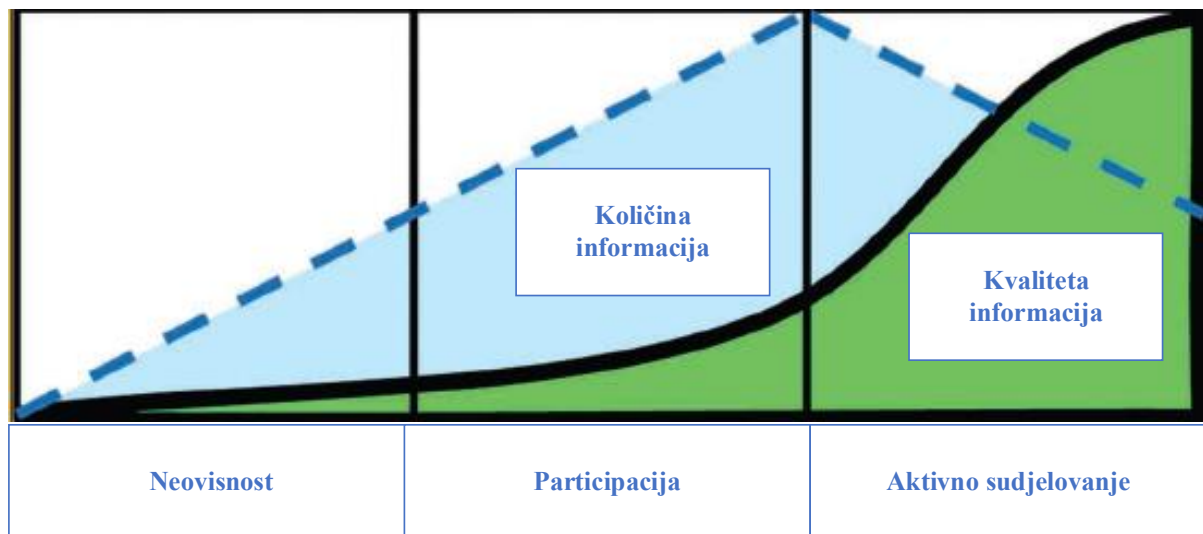
Komunikacija s korisnicima već je dugo prepoznata kao jedan od glavnih faktora koji utječe na uspješnost projekta, a oni su u svom istraživanju utvrdili kako je krojenje hibridne projektne metodologije iz tradicionalne i agilne utjecalo na komunikaciju na projektu, koji su se problemi u komunikaciji pojavili u hibridnim modelima te dali smjernice za bolje upravljanje komunikacijom na projektima vođenim po hibridnim modelima upravljanja.

Razlikuju 3 vrste komunikacije s obzirom na angažman korisnika u komunikaciji i kvalitetu komunikacije:

- Participaciju odnosno sudjelovanje (eng. *Participation*) – naglasak se stavlja na frekvenciju odnosno učestalost komunikacije između razvojnog tima i korisnika; kvaliteta komunikacije je upitna
- Uključenost (eng. *Involvement*) – naglasak je na motivaciji korisnika da komuniciraju s razvojnim timom, no iako su korisnici motivirani za rad na projektu u sklopu kojeg se razvija novi sustav zbog svakodnevnog rada na starom sustavu i posljedice ograničenosti da vide širu sliku često ne daju relevantne povratne informacije na novi sustav
- Aktivno sudjelovanje (eng. *Engagement*) – naglasak je na stvarnoj smislenoj razmjeni relevantnih i pravovremenih informacija, što čini komunikaciju kvalitetnom; aktivno sudjelovanje predstavlja željenu manifestaciju uključenosti korisnika u komunikaciju s razvojnim timom.

Komunikacijska sprega (eng. *Communication coupling*) je kontinuum koji počinje s neovisnošću, gdje je razmjena informacija s korisnicima izrazito niska, te prolazi kroz faze Participacije i u idealnom slučaju kulminira u fazi Aktivnog sudjelovanja [21]. Značajna participacija korisnika može dovesti do velike količine informacija, od kojih je velik dio

irelevantan za projekt, a u fazi Aktivnog sudjelovanja dolazi do smanjenja količine informacija i porasta njihove kvalitete (Slika 16).



Slika 16. Komunikacijska sprega [21]

Mahaparta, R. i Nerur, S. u svom su radu [21] proveli istraživanje na 2 projekta razvoja softvera vođenih po hibridnim modelima, od kojih je jedan vođen primarno po vodopadnom modelu obogaćenom agilnim praksama, a drugi primarno po Scrumu obogaćenom praksama iz vodopadnog modela. Prikupljeni su sekundarni podaci za oba projekta iz razgovora s po 4 dionika projekta (Tablica 7).

Tablica 7. Karakteristike projekata koji su bili predmet istraživanja [21]

<b>Projekt</b>	<b>1. <i>Green Giant</i></b>	<b>2. <i>Blue Velvet</i></b>
Industrija	Farmaceutika	Farmaceutika
Broj zaposlenika	350-500	350-500
Metodologija upravljanja projektom	Hibridni model s dominantnim vodopadnim modelom	Hibridni model s dominantnim agilnim pristupom
Poslovno područje za koje se razvija programski sustav	Prilagodna gotovog softvera	Izrada programskog sustava po mjeri
Lokacija projektnog tima	Svi na istom mjestu, korisnici su koristili isti stari softver	Na 4 lokacije, na svakoj su korisnici koristili različiti softver do tad
Tip projekta	Dorada i implementacija	Razvoj i implementacija
Veličina razvojnog tima	15	5
Trajanje projekta	10-12 mjeseci	8-9 mjeseci
Dionici projekta s kojima je rađeno kvalitativno istraživanje	Voditelj ključnih korisnika, Voditelj razvojnog tima, Voditelj projekta, Razvojni inženjer	Voditelj ključnih korisnika, Voditelj razvojnog tima, Scrum <i>Master</i> , Razvojni inženjer

U prvom projektu pod nazivom *Green Giant* riječ je bila o zamjeni postojećeg sustava za skladištenje i logistiku. Stari je sustav bio na platformi za koju je proizvođač odlučio ukinuti programsku podršku. Tvrtka je imala internacionalne distribucijske centre i zaključila da je svakako došlo vrijeme za novi softver. Odlučili su se projekt provesti fazno i tretirati svaki distribucijski centar kao novi projekt u skladu sa specifičnostima s obzirom na lokaciju, zakonsku regulativu i procedure.

Na projektu *Green Giant* korisnici su bili uključeni i participirali su često. No, projekt *Green Giant* je uz promjenu softvera uključivao i promjenu poslovnih procesa pa su stoga korisnici, mahom djelatnici na skladištu, istovremeno morali svladati novi softver i nove poslovne procese.

Projekt je doživio veliki neuspjeh po pitanju edukacije korisnika, koji nakon produkcije nisu znali obavljati svoje svakodnevne zadatke u novom softveru.

Glavni razlozi za to bili su sljedeći:

- 1) korisnici nisu bili posvećeni projektu zbog nedostatka vremena - čitavo vrijeme trajanja testiranja i edukacija morali su izvršavati i svoje redovne radne zadatke
- 2) testna okolina nije odgovarala stvarnoj situaciji – bila je suviše pojednostavljena i sadržavala je značajno manji skup podataka od produkcijske.

Korisnici su za vrijeme testiranja pokušavali „prevariti“ novi softver kopirajući procese iz starog sustava, koji nisu bili kompatibilni s novim sustavom niti s novim poslovnim procesima. Unatoč uložnim naporima, korisnici nisu na temelju testiranja i dobivenih edukacija razumjeli novi sustav, a još je gore što je to otkriveno tek nakon puštanja novog sustava u produkciju. Ovaj primjer pokazuje kako je teško postići razumijevanje novih procesa i sustava kada korisnici nastoje „živjeti“ stare procese i stari sustav. Iako su korisnici odradili testiranje zapravo su mehanički prošli kroz testne scenarije pod utjecajem starog sustava i poslovnih procesa. Pseudorealna testna okolina dodatno je otežala korisnicima razumijevanje novog sustava.

Drugi projekt pod nazivom *Blue Velvet* za cilj je imao zamijeniti softver za vođenje rasporeda medicinskih tretmana liječenja u sklopu istraživačke medicine u jednom laboratoriju. Projekt je načelno vođen po metodologiji upravljanja projektima Scrum, no napravljeni su kompromisi po pitanju komunikacije s korisnicima budući da su korisnici bili disperzirani u većem broju laboratorija.

Razvojni tim je pokazivao korisnicima razvijeni dio softvera i napredak dva puta tjedno, no ispostavilo se da kritično bitne informacije nisu razmijenjene između razvojnog tima i korisnika.

Glavni problemi bili su sljedeći:

- 1) Demo radionice vodio je razvojni tim i stvorio „umjetno“ iskustvo za korisnike; korisnici nisu razumjeli ograničenja sustava, a razvojni inženjeri korisničke scenarije
- 2) Razvojni se tim fokusirao na detalje, ne gledajući širu sliku.

Demo radionice bile su previše posvećene slavljenju onoga što je napravljeno, a razvojni inženjeri fokusirani na idući *Sprint* umjesto na povratnu informaciju vezano za prethodni.

Ovaj primjer ukazuje na to kako nije postignuto međusobno razumijevanje između razvojnog tima i korisnika, ali i upozorava da se kod hibridnih modela koji su dominantno agilni javlja dominacija vizije razvojnih inženjera u odnosu na zahtjeve korisnika.



Situacija se popravila kada su korisnici počeli testirati sustav bez da ih razvojni tim vodi te su tada korisnici počeli davati relevantne povratne informacije.

Zaključak istraživanja je da se na prvom projektu koji je vođen po hibridnom modelu dominantno utemeljenom na vodopadnom modelu problemi u komunikaciji rezultirali time da korisnici ne razumiju novi sustav i ne znaju se njime služiti u produkciji. Naime, posljedica je to neuključenosti korisnika u pojedine faze projekta u vodopadnom modelu. Kod drugog projekta koji je bio predmet istraživanja, a koji je vođen po hibridnom modelu dominantno temeljenom na Scrumu, problemi u komunikaciji sveli su se na nerazumijevanje korisničkih zahtjeva od strane razvojnog tima uslijed snažne vizije programera vezano za korisničke priče (eng. *User story*) koja je stvorila barijeru u komunikaciji s korisnicima. Posljedica je to fiktivne uključenosti korisnika na ceremonijama djelomično preuzetih iz Scruma, a na kojima je razvojnim inženjerima fokus bio pokazati što su dotad napravili umjesto dobiti povratnu informaciju od korisnika.

Hibridni modeli moraju posebnu pažnju pridati aktivnom sudjelovanju korisnika, a učinkovit angažman korisnika bit će bolji ako se demo radionice i testiranja u kontroliranim uvjetima zamijene korisničkim testiranjem na stvarnim slučajevima.

S obzirom na to da su hibridni modeli sastavljeni od različitih metodologija upravljanja projektima, nemaju svoje ceremonije i standarde komunikacije već ih projektni timovi kroje po svojim potrebama. Stoga je posebno bitno osigurati uravnoteženo sudjelovanje i uključenost svih dionika projekta, a posebno korisnika, u komunikaciji te na vrijeme prepoznati probleme u komunikaciji.

## **2. Kvalitativno istraživanje načina vođenja projekata razvoja web-aplikacija**

U sklopu izrade ovog rada provedeno je kvalitativno istraživanje s ciljem utvrđivanja koje su metodologije upravljanja projektima korištene, koliko je učestalo kombiniranje različitih metodologija upravljanja projektima te koliko se njihova kombinacija pokazala uspješnom za projekte razvoja web-aplikacija.

Fokus istraživanja bio je na primijenjenoj metodologiji upravljanja projektom razvoja web-aplikacije, izazovima koji su se na projektu pojavili te načinu na koji su riješeni.

Istraživanjem je utvrđeno koje su se projektne faze, u slučaju korištenja hibridnih modela, provodile po kojoj metodologiji upravljanja projektom.

U obzir su prilikom obrade rezultata istraživanja uzete karakteristike predmetnog projekta (trajanje projekta, karakteristike projektnog tima i procijenjena tehnička zahtjevnost web-aplikacije) te uspješnost projekta s obzirom na rokove, budžet i kvalitetu.

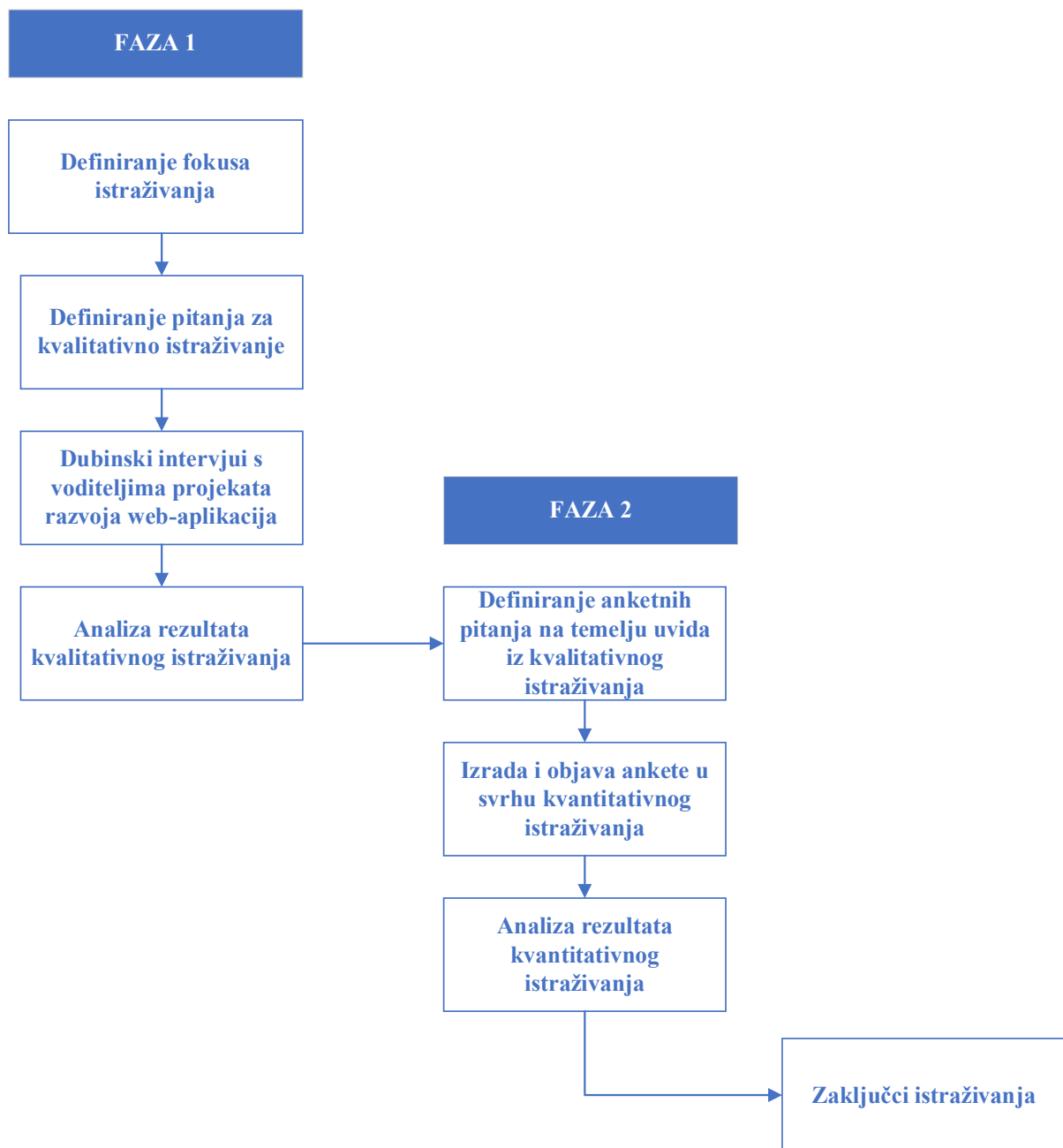
Kvalitativno istraživanje provedeno je u formi dubinskih intervjua na uzorku od 7 voditelja projekata razvoja web-aplikacija.

Kvalitativno istraživanje predstavljalo je osnovu za definiranje anketnih pitanja za kvantitativno istraživanje koje je uključilo ostalim ulogama projektnih timova kako bi se utvrdio njihov stav prema:

- metodologijama upravljanja projektima
- identificiranim problemima koji se pojavljuju na projektima
- korištenju hibridnih modela u upravljanju projektima, s naglaskom na to koje projektne aktivnosti su pogodnije za izvođenje po tradicionalnim metodologijama, a koje po agilnim.

Zaključno, cilj cjelokupnog istraživanja bio je utvrditi učestalost i način primjene hibridnih modela na projektima razvoja web-aplikacija s obzirom da je područje razvoja softvera vrlo široko te je opseg istraživanja uključivao područje razvoja web-aplikacija. Na temelju rezultata istraživanja definirane su smjernice za rješavanje najčešćih identificiranih problema koji se pojavljuju na projektima, uz pomoć primjene hibridnih metodologija upravljanja projektima.

Tijek istraživanja prikazan je dijagramom na slici 17.



Slika 17. Tijek provedenog istraživanja

Pitanja za kvalitativno istraživanje grupirana su u 8 logičkih cjelina na sljedeći način:

**1) Demografska pitanja vezana za kompetencije Voditelja projekata (P1-P3)**

P1 – godine u struci; P2 – dob; P3 – formalna edukacija iz Upravljanja projektima

**2) Svojstva projekta (P4-P6)**

P4 – trajanje projekta; P5 – broj članova projektnog tima; P6 - Projektni tim s obzirom na komunikaciju

### **3) Tehničke karakteristike projekta (P7-P8)**

P7 – nova web-aplikacija/dorada postojeće; P8 – kompleksnost integracija

### **4) Učestalost izmjena u tijeku projekta (P9-P11)**

P9 – definiranost korisničkih zahtjeva; P10 – učestalost izmjena korisničkih zahtjeva; P11 – izmjene arhitekture sustava

### **5) Izazovi koji su se pojavili na projektu (P12-P15)**

P12 – greške u kodu prije produkcije; P13 – greške u kodu nakon produkcije; P14 – najveći izazovi generalno; P15 – načini rješavanja izazova

### **6) Korištene metodologije upravljanja projektima (P16-P20)**

P16 – korištena metodologija upravljanja projektom; P17 – korištenje hibridnih modela; P18 – inicijalni plan projekta; P19 – zatvaranje faza projekta/preklapanje faza; P20 – uključenost korisnika u tijeku projekta

### **7) Uspješnost projekta (P21-P23)**

P21 - budžet; P22 - rok; P23 – kvaliteta

### **8) Naučene lekcije (P24)**

P24 – uspješnost korištene metodologije upravljanja projektom i prijedlog preinaka iste na temelju iskustva

Pitanja za dubinske intervju:

PITANJE 1 (P1): Koliko ste dugo u ulozi voditelja projekata?

PITANJE 2 (P2): Koliko imate godina?

PITANJE 3 (P3): Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

PITANJE 4 (P4): Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

PITANJE 5 (P5): Koliko je bilo članova projektnog tima?

PITANJE 6 (P6): Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

PITANJE 7 (P7): Je li se u sklopu tog projekta razvijala nova web-aplikacija ili doradivala postojeća?

PITANJE 8 (P8): Je li projekt uključivao i integraciju drugih programskih sustava?

PITANJE 9 (P9): Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

PITANJE 10 (P10): Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

PITANJE 11 (P11): Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

PITANJE 12 (P12): Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

PITANJE 13 (P13): Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

PITANJE 14 (P14): Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

PITANJE 15 (P15): Kako ste riješili izazove iz Pitanja 14?

PITANJE 16 (P16): Koju ste metodologiju upravljanja projektima koristili?

PITANJE 17 (P17): Jeste li kombinirali različite metodologije upravljanja projektima?

Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

PITANJE 18 (P18): Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

PITANJE 19 (P19): Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

PITANJE 20 (P20): Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

PITANJE 21 (P21): Je li projekt završen u okviru zadanog budžeta?

PITANJE 22 (P22): Je li projekt završen u roku?

PITANJE 23 (P23): Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

PITANJE 24 (P24): Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

## ***2.1. Dubinski intervjui***

Dubinski intervjui provedeni su na uzorku od 7 voditelja projekata razvoja web-aplikacija koji rade u softverskoj industriji u Hrvatskoj. U nastavku rada nalaze se rezultati kvalitativnog istraživanja te analiza navedenih rezultata.

### **2.1.1. Dubinski intervju\_1**

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 3 godine.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 29.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Posjedujem Udemy certifikat PMP Preparation Course koji sam položio u svrhu mogućnosti prijave za PMP certifikat, za koji se trenutno pripremam.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 18 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Projektni tim imao je ukupno 11 članova (1 poslovni analitičar, 1 junior konzultant za potrebe edukacije krajnjih korisnika, 1 *Product Owner*, 5 programera, 1 specijalist za testiranje softvera, 1 inženjer za API dio odnosno za integraciju, voditelj projekta). Scrum tim činili su *Product Owner*, 5 programera i specijalist za testiranje softvera – dakle, njih 7 od 11.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Projektni tim je prešao s prethodnog projekta na kojem su radili u istom sastavu tako da su se dobro poznavali i uspješno komunicirali pa nije postojao izazov po pitanju međusobnog upoznavanja članova projektnog tima i uspostavljanja komunikacije.

Scrum tim je imao svoje Scrum ceremonije, a projektni menadžer sudjelovao na *Sprint review* sastanku jednom u 2 tjedna budući da se radilo u dvotjednim *Sprintovima*. Jednom tjedno održavali su se tjedni sastanci na kojem su bili svi članovi projektnog tima osim dijela programera, koji su priključivali po potrebi.

U slučaju kašnjenja Scrum tim je bio transparentan prema voditelju projekta. Kašnjenja nije bilo puno sa strane Scrum tima, većinom su se događala kao posljedica nepoznanica po pitanju poslovnih procesa koje je tad Poslovni analitičar morao dodatno istražiti. Voditelj projekta je također posjećivao klijenta radi razjašnjavanja pojedinih zahtjeva.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili dorađivala postojeća?

**ODGOVOR 7 (O7):** Dorađivala se postojeća web-aplikacija.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Da, uključivao je integraciju web-aplikacije s 2 vanjska sustava.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Smatrali smo da jesu, međutim u tijeku projekta se ispostavilo da nisu bili dobro definirani jer je klijent pokušavao digitalizirati jedan potpuno novi dio poslovanja (koji do tada nije imao pa je i za njega samog bilo puno nepoznanica).

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj

- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Da, bilo je značajnih izmjena zahtjeva.

Zahtjevi za izmjenama pojavili su se u fazi 3) Dizajniranje rješenja i tu je otkriveno otprilike 90% zahtjeva za izmjenom, a preostalih 10% u fazi 4) Razvoj i većinom su bili vezani za tehničku nemogućnost izvedbe ili integracije na način koji je bio inicijalno zamišljen.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):** Nije došlo do izmjene arhitekture budući da se radilo o nadogradnji postojeće web-aplikacije.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Je, bio je veći od 10 - prvenstveno iz razloga što se pojavio veliki broj rubnih slučajeva koji nisu bili predviđeni. Bilo je otprilike 100 grešaka u kodu prijavljenih u tijeku testiranja, od čega 10 ocjenjujem kao kritične.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije, jer su se svi treninzi i edukacije održavali uživo kod Klijenta pa su kritične greške uočene i ispravljene prije produkcije.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):**

- 1) Voditi Klijenta kroz Scrum proces i iterativni razvoj jer poslovni ljudi nisu bili upoznati s konceptom Scruma i bilo ih je teško uvjeriti u takav koncept rada.
- 2) Drugi izazov bio je digitalizirati poslovni proces koji do tada uopće nije postojao i osmišljavao se u hodu od strane Klijenta.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?



**ODGOVOR 15 (O15):**

- 1) Čestim i transparentnim informiranjem Klijenta te čestim posjećivanjem Klijenta na njegovoj lokaciji kako bih mu pokazao rezultate dosadašnjeg rada i napredak projekta.
- 2) Direktnim povezivanjem programera i krajnjih korisnika radi boljeg pojašnjenja zahtjeva (bez šumova u kanalu). Time je zapravo preskočen menadžment Klijenta i približili smo se operativnom timu koji je ipak bolje znao razjasniti poslovne potrebe budući da su radili operativne zadatke u domeni projekta.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Scrum. Glavni je razlog bio što smo „išli u nepoznato“ odnosno nismo znali ukupan opseg projekta pa nam je bilo jednostavnije raditi po agilnoj metodologiji upravljanja projektima kako bi mogli dorađivati proizvod kako budemo otkrivali koje su sve dorade potrebne.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Jesmo.

Prvenstveno je Planiranje napravljeno po vodopadnom modelu, kao i većina ostalih faza projekta. Planirani su okvirni budžet, opseg i rokovi projekta.

Samo smo 4) Razvoj radili po Scrumu.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Napravljen je kompletan plan projekta na početku projekta, ali budući da se broj korisničkih zahtjeva drastično povećao i da je došlo do izmjena korisničkih zahtjeva u fazi 3) Dizajniranje rješenja inicijalni plan se nije mogao provesti i više nije bio važeći.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Došlo je do preklapanja faza – razvoj i testiranje su se sukladno Scrum metodologiji konstantno preklapali. Dizajniranje rješenja preklapalo se, uslijed izmjena zahtjeva i dorada, također s razvojem i testiranjem. Uz navedeno, migracija podataka te edukacija Krajnjih korisnika odvijali su se paralelno s Razvojem i Testiranjem budući da su Krajnji korisnici bili uključeni u testiranje.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Korisnici su aktivno testirali i kao nositelji domenskog znanja bili su najkompetentniji da daju povratnu informaciju o ispravnosti rada web-aplikacije.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Je, ali samo zato što je veliki dio inicijalnih zahtjeva postao nevažeći i nije se uopće razvijao. Tijekom faza 2) Definiranja zahtjeva i 3) Dizajniranja rješenja ukazali smo Klijentu da dio inicijalno postavljenih zahtjeva neće donijeti dodatnu vrijednost poslovanju.

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Nije, kasnio je 6 mjeseci.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** Je, pokriveni su svi poslovni scenariji koje je Klijent definirao u svojim finalnim zahtjevima.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Vodio bih sličan projekt po istoj metodologiji upravljanja projektima. Mislim da nije bilo boljeg pristupa za projekt ovakvog tipa s puno nepoznanica. Nije bilo zahtjeva Klijenta za izmjenama nakon produkcije i Klijent nas je kasnije ponovno angažirao. Eventualno bih produljio fazu Planiranja i analize korisničkih zahtjeva i proveo je s Klijentom u njihovom radnom okruženju (na lokaciji Klijenta).

### **2.1.2. Dubinski intervju\_2**

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 18 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 50.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** CROZ tečaj vezan za Upravljanje projektima – softvera te CROZ priprema za PMI certifikaciju.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 6 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Razvojni tim imao je 9 članova - 7 programera, *Product Owner* i voditelj projekta je bio ista osoba, 1 specijalist za testiranje softvera. Sa strane Klijenta je postojao testni tim od 4-5 članova, a dodatno je Klijent osigurao ključne korisnike – njih 5. Dio ključnih korisnika je u tijeku projekta prešao u testni tim Klijenta.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Trudili smo se biti agilni Scrum tim te smo imali Scrum *Mastera*. Suradivali smo s Klijentom i sudjelovali s njima na definiranju korisničkih zahtjeva i prioritizirali ih svaka 2 tjedna. Radili smo u dvotjednim *Sprintovima*. Unutar razvojnog tima članovi su bili transparentni prema meni kao voditelju projekta u slučaju kašnjenja.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili doradivala postojeća?

**ODGOVOR (O7):** Oboje. Tehnički je bila riječ o 3 aplikacije – jedna se nadograđivala; druga je bila nova, ali se prikazivala kao dodatna funkcionalnost postojeće web-aplikacije (korisnicima); treća je bila potpuno nova i bila je namijenjena za prodaju odnosno za krajnje kupce korisnika.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Je, bilo je puno integracija. Te 3 web-aplikacije čine jedan sustav koji je još dodatno integriran s 5 poslovnih sustava, a tehnički gledano riječ je o 20 servisa.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** U smislu vodopadnog modela ne. Korisnici su na početku dali popis željenih funkcionalnosti na temelju kojih je *Product Owner* raspisao bazične *User Stories*, a koji su se kasnije razrađivali na radionicama s Korisnicima.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Teško je odgovoriti na ovo pitanje jer smo radili po agilnoj metodologiji upravljanja projektima pa su faze bile preklapajuće, no novi su se zahtjevi počeli pojavljivati u zadnjoj trećini trajanja projekta.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):** Nije došlo do promjene arhitekture sustava u tijeku projekta.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Da, ukupni broj *bugova* bio preko 100.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Oko 5. (više od 3 manje od 10)

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):** Prebacivanje rješenja na produkcijsku okolinu. Imali smo 4 okoline (razvojnu, testnu, predprodukcijsku i produkcijsku). Kako smo selili rješenje na više instance tako su se na različitim okolinama pojavljivale nove pogreške. Kada smo došli do produkcijske okoline razvojnom je timu bilo jako teško otkriti uzrok *bugova* jer kao razvojni tim nemamo pristup produkcijskoj bazi i vidimo samo grešku, ali ne i *logove* i teško nam je otkriti njen uzrok. Često je bila riječ o neispravnoj ili različitoj konfiguraciji na okolinama.

Ključni korisnici i IT sa strane Klijenta bili su zauzeti drugim projektima i vlastitim poslom pa je izazova bilo i po pitanju njihove dostupnosti.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** Nismo ih u potpunosti riješili, ali trudili smo se rješavati jedan po jedan problem. Zbog povjerljivosti podataka nažalost nismo mogli raditi kopije baze podataka po okolinama tako da nam je razlika među okolinama bila glavni neriješeni problem.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Scrum.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Kombinirali smo Scrum i Kanban. Pred kraj projekta prešli smo na Kanban jer nam je dvotjedni Sprint postao trom. To je bila faza u kojoj smo već riješili veće funkcionalnosti i većinom rješavali prijave iz testiranja.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Nije, bile su napravljene *User Stories*, ali u kontekstu vodopadnog modela ne.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Sve su se faze preklapale.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Aktivno su sudjelovali u testiranju tijekom razvoja, s time da su ipak pred kraj projekta intenzivirali testiranje i doveli nove dodatne testere. Bilo bi bolje da su testirali i više u tijeku razvoja.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Da.

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Kasnio je 1 mjesec. Da nije bilo međuovisnosti i integracije s drugim projektima koji su se paralelno razvijali ne bi došlo do kašnjenja.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** *User Stories* su sadržavale *Acceptance criteria*, a tester i sa strane Klijenta dodatno su imali svoje testne scenarije. Generalno je projekt zadovoljio postavljene standarde kvalitete.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Vodio bih ga po istoj metodologiji upravljanja projektima, čak razmišljam i o agilnijim modelima (poput XP). Glavni je razlog tome što na početku projekta naprosto nismo znali sve zahtjeve.

### 2.1.3. Dubinski intervju\_3

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 20 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 49.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Položila sam PMI. Ujedno sam pohađala tečajeve za agilno upravljanje projektima.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 7 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Maksimalno u jednom trenutku projekta 11. Krenuli smo s 2 programera i jednim poslovnim analitičarem, no kako je posao na projektu bio opsežniji tako smo širili tim na 5-8 programera i 1-2 poslovna analitičara. Poslovni analitičari su ujedno radili i poslove testiranja. Uz njih, cijelo je vrijeme na projektu bio i voditelj projekta.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Tim se počeo oformljavati u svrhu ovog projekta pa je na početku bilo problema razvijanja tima, no u vrlo su kratkom vremenu profunkcionirali. Odnos prema voditelju projekta je bio transparentan. Komunikacija je bila dobra i tim je na vrijeme informirao Voditelja projekta o kašnjenjima. Komunikacija s Klijentom bila je otežana, čak pomalo kaotična, no trudili smo se da se to ne prenese na naš interni razvojni tim.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili dorađivala postojeća?

**ODGOVOR 7 (O7):** Dorađivala se postojeća – radile su se nove funkcionalnosti te se razvijala nova usluga, koja je bila potpuno novi poslovni segment.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Da, bilo je više integracija – od toga 2 velike, 3-4 manje.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Ne u potpunosti. Postojala je specifikacija koju je Klijent napisao, no nije bila dovoljno detaljna pa smo održali nekoliko radionica u fazi Analize korisničkih zahtjeva te

smo u sklopu njih razbijali korisničke zahtjeve do razine *Taskova*. Tako da su *Taskovi* bili definirani na početku projekta.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Da, u fazama 4) Razvoj i 2) Testiranje. Klijent je tada shvatio da procesi koje su na početku definirali dijelom neće zadovoljiti njihovu poslovnu potrebu pa je došlo do izmjena zahtjeva.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):** Ne, arhitektura nije mijenjana.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Nije.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):**

- 1) Komunikaciju s Klijentom – dio informacija je do nas dolazio preko posrednika, dobivali smo neispravne informacije poput toga da je sustav s kojim se trebamo integrirati završen, a nije bio.
- 2) Pritisak menadžmenta Klijenta da izađu na tržište prvi s predmetnom web-aplikacijom (u odnosu na konkurenciju) pa se često ubrzavalo testiranje i umanjivala važnost provjere kvalitete.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** Štitila sam svoj tim od nepotrebnih informacija kako ih ne bi demotivirale, ali bila transparentna prema njima.

Razgovarala sam redovno s voditeljem projekta sa strane Klijenta, a ponekad i eskalirala prema visokom menadžmentu kada je bilo potrebe.

Također, redovno sam slala zapisnike sa statusom projekta nadređenima i sa svoje strane i sa strane Klijenta.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Hibridni model – kombinaciju vodopadnog modela i agilnog pristupa (kombinacija Scruma i Kanbana).

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Da,

- 1) Planiranje i analiza korisničkih zahtjeva – počelo je po vodopadnom modelu, no završilo po hibridnom
- 2) Definiranje zahtjeva - počelo je po vodopadnom modelu, no završilo po hibridnom
- 3) Dizajniranje rješenja – agilni pristup
- 4) Razvoj – agilni pristup
- 5) Testiranje – agilni pristup

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Da, napravljen je plan projekta u fazi 1) Planiranje i analiza korisničkih zahtjeva. Opseg je bio jasno definiran, ali ne i isporuke (u smislu rasporeda).

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Došlo je do preklapanja navedenih faza.



**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Korisnici su željeli raditi testiranje tek na završetku razvoja, no inzistirali smo na testiranju u tijeku razvoja tako da su aktivno sudjelovali.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Je, završen je u okviru zadanog budžeta.

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Je, završen je u roku. I, što je Klijentu bilo vrlo bitno, prije konkurencije.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** Klijent je imao dobro raspisane testne scenarije i projekt je zadovoljio standarde kvalitete.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Vjerojatno bih, iako je projekt bio dosta specifičan pa sam tražila najbolju kombinaciju metodologija za konkretan slučaj. Glavni uspjeh na ovom projektu je bio uključivanje Klijenta u agilni razvoj i njihovo testiranje u tijeku razvoja. Iako ugovaranje nije bilo agilno, u konačnici smo ih uspjeli uključiti u razvoj i dobiti od njih povratnu informaciju na vrijeme, a ne tek pred kraj projekta.

#### **2.1.4. Dubinski intervju\_4**

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 18 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 47.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Da, PMP i SAP *Activate*.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 18 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Projektni tim sastojao se od 26 članova – voditelj projekta, arhitekt rješenja, *Product Owner*, 9 programera, ostatak tima činili su poslovni analitičari.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** U početku su prikrivali kašnjenja i prošli smo kroz tzv. *storming* fazu dok se tim nije formirao. Dio tima sam mijenjao (od 9 programera 3 smo smijenili i uveli nove u posao). Na početku je bilo dosta izazovno formirati tim, ali uspjeli smo.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili dorađivala postojeća?

**ODGOVOR 7 (O7):** U sklopu projekta razvijala se nova web-aplikacija.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Uključivao je integraciju s 5 drugih programskih sustava.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Jesu.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Nije došlo do značajnih promjena zahtjeva u tijeku projekta. Promjene koje su se pojavile uočene su u fazi 3) Dizajniranje rješenja, a većinom su bilo tehničke prirode ili vezane za arhitekturu sustava.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj

5) Testiranje

6) Produkcija

**ODGOVOR 11 (O11):** Je, u fazi 3) Dizajniranje rješenja. Mijenjali smo testne uređaje jer oni koje smo inicijalno odabrali nisu zadovoljavali kompleksnost podataka koje smo imali. Mijenjali smo i *Payment Gateway*.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Je, bilo ih je više od 10. Na otprilike 30% aplikacije smo imali *bugove*.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije, imali smo 2 *buga*.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):** Integracija s *Payment Gatewayem* i logistički procesi (misli se na implementaciju aplikacije na opremu na kojoj će se izvršavati web-aplikacija – bilo je riječ o otprilike 250 komada opreme).

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** Prvi izazov smo riješili tako što smo isprogramirali potpuno novu funkcionalnost za plaćanje po vlastitim potrebama, a drugu dobrom organizacijom (izdvajali smo 20 po 20 komada opreme i na njima vršili instalaciju budući da je oprema već bila u uporabi).

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Koristio sam *Lean* metodologiju, a Razvoj je išao po Scrumu. Smatrao sam da je to najbolji pristup, vodeći se po PMI-u i PMP-u, kako bi projekt pratio plan, a razvojni tim imao dovoljno slobode.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Kombinirao sam *Lean* projektnu metodologiju i Scrum. Po Scrumu se radio Razvoj.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Je, na početku projekta je napravljen kompletan Plan projekta i to je ujedno bila prva isporuka na projektu prema Klijentu.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Bilo je preklapanja – preklapale su se faze definiranja zahtjeva i testiranja. Prvo smo razvijali minimalni set funkcionalnosti, a zatim smo krenuli raditi dodatne funkcionalnosti pa je tim bio podijeljen na dio koji je već provodio testiranje osnovnih funkcionalnosti i na dio koji je raspisivao dodatne funkcionalnosti. Testiranje je bilo provedeno na 6 demo radionica na kojima smo detektirali velike *bugove*, nakon toga smo ih ispravili i ponovili testiranje zajedno s korisnicima, a još smo na kraju imali *end-to-end* testiranje. Testiranje i razvoj su se, dakle, također preklapali.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Korisnici su sudjelovali u testiranju.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (P21):** Je, projekt je završen u okviru budžeta .

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (P22):** 2 mjeseca je kasnio.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (P23):** Nismo imali unaprijed definirane testne scenarije, no Klijent je bio zadovoljan isporučenim programskim rješenjem.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (P24):** Vodio bih ga na isti način, mislim da je bio najučinkovitiji.

### **2.1.5. Dubinski intervju\_5**

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 8 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 33.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Tečajevi – *PMP preparation for exam*, tečajevi: Upravljanje teškim korisnicima, *Project management versus Product management*, Vještine komuniciranja i upravljanja te Metodologije vođenja projekata.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 30 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Bilo je 17 članova razvojnog tima – 1 arhitekt rješenja, 9 konzultanata, voditelj projekta, a ostali su bili programeri. Projektni se tim mijenjao u tijeku projekta.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Tim se međusobno slagao, ali nisu se osjećali odgovornima za rokove, nisu transparentno komunicirali i nisu bili dorasli poslu. Sve je to u kombinaciji s koronom i radom od kuće rezultiralo velikim izazovima na projektu.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili dorađivala postojeća?

**ODGOVOR 7 (O7):** U sklopu projekta razvijala se nova web-aplikacija.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Je, bila je potrebna integracija sa 6 programskih sustava.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Nisu.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):**

Je, u fazi 5) Testiranje.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):**

Je, također u fazi 5) Testiranje.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Da, 15-20% web-aplikacije je imalo *bugove*.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije, bila su 3.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):** Loše napravljena analiza korisničkih zahtjeva. Shodno tome su funkcionalne specifikacije bile manjkave. Nije napravljena ni zadovoljavajuća *fit-gap* analiza. Konzultanti nisu bili dorasli zadatku. Rad od kuće nije doprinio. Ja sam došao u ulogu voditelja projekta nakon što je projekt već trajao 8 mjeseci, a 4 mjeseca kasnije (dakle, 12 mjeseci od početka projekta) promijenio se voditelj projekta i sa strane Klijenta. U fazi analize arhitekt rješenja i prvi voditelj projekta sa strane Isporučitelja nisu razgovarali uslijed konflikta, stoga arhitekt nije sudjelovao u fazi Analize korisničkih zahtjeva. Ključni korisnik nije jasno specificirao zahtjeve i mijenjao ih više puta u tijeku projekta.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** Sastavili smo listu svih otvorenih pitanja odnosno kreirali tzv. *issue list*. Nakon što smo riješili otvorena pitanja odobren je novi *scope* od strane Nadzornog odbora projekta.

15 mjeseci od početka projekta smijenili smo sve konzultante na projektu i zamijenili ih kompetentnijima.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Koristio sam vodopadni model i *Lean* metodologiju, a Razvoj je išao po Scrumu.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Jesmo, Razvoj je išao po Scrumu, Planiranje i analiza su išli po vodopadnom modelu. Završetak je bio po vodopadnom model.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Je, ali nije uključivao sve funkcionalnosti koje su zapravo trebale biti napravljene odnosno bio je manjkav.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Došlo je do preklapanja svih faza. Krenuli smo po vodopadnom modelu, ali nakon prvog testiranja sve se počelo raditi paralelno.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Sudjelovali su tek kad bi konačno rješenje bilo razvijeno, ali za svaku funkcionalnost zasebno (ne za cijeli projekt).

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Ne, utrošak resursa bio je 4x veći od planiranog.

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Ne, kašnjenje je bilo 12 mjeseci.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** Nije. Testni scenariji su bili raspisani nakon Analize u vrijeme dok se radio Razvoj. Problem je bio što su scenariji pisani kad je već počeo razvoj. Produkcija je puštena s minimalnim opsegom (*minimum viable product*), ali još uvijek se dorađuje.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Ne, napravio bih na početku projekta *fit-gap* analizu na način da prvo pokažem funkcionalnosti našeg produkta i na temelju toga popisao potrebne dorade. Za dorade bih odmah definirao testne scenarije koji moraju biti zadovoljeni da bi se isporuka smatrala valjanom. Ne bih se upustio u projekt s nekompetentnim timom. Sve faze osim Razvoja i Testiranja bih radio po vodopadnom modelu, a Razvoj i Testiranje po Scrumu.

## 2.1.6. Dubinski intervju\_6

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 15 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 52.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Nemam, prošao sam seminar na temu Vođenja projekata po vodopadnom modelu.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 22 mjeseca.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Ukupno 11. Interni razvojni tim je imao 6 članova – voditelja projekta, 2 poslovna analitičara koji su ujedno radili testiranje i 3 programera. Vanjski razvojni tim imao je 5 članova – 1 poslovni analitičar i 4 programera.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Bilo je problema po pitanju komunikacije između internog i vanjskog razvojnog tima. Rokovi su bilo vrlo teško dostižni pa je postojao problem čak i unutar internog tima jer su članovi tima pokušavali prebaciti odgovornost na druge (u smislu da su oni napravili „svoj dio“, ali netko drugi nije). Članovi tima nisu skrivali kašnjenja, ali su ih komunicirali u posljednji čas.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili doradivala postojeća?

**ODGOVOR 7 (O7):** Razvijala se nova web-aplikacija, ali je rađena na gotovom *frame worku* koji se doradivao.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Je, uključivao je 3 integracije.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Nisu bili jasno definirani, korisnici su očekivali i definirali da žele da sve funkcionira kao na starom sustavu koji se trebao zamijeniti ovom novom web-aplikacijom.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?



Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Je, s time da ni na početku zapravo nisu bili definirani, promjene su bile najintenzivnije u fazi 5) Testiranje.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):** Minimalno, promjene koje su se dogodile nisu bile velike i odnose se na dodavanje nekih novih sučelja i odustajanje od dijela sučelja, ali koncept se nije mijenjao. Promjene koje su se dogodile bile u fazi 4) Razvoj.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Je, bilo ih je otprilike 20. Uz *bugove*, imali smo problema s podizanjem verzija softvera jer se događalo da nam vanjski razvojni tim „pregazi“ ispravke koje je interni razvojni tim napravio u svojoj posljednjoj verziji.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije, bio je manji od 5.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):** Na početku projekta komunikacija se odvijala putem e-maila, a radne smo zadatke vodili u Excelu pa su se informacije gubile i tim nije imao pregled cijele komunikacije.

Procjene su bile pogrešne jer opseg projekta nije bio jasno definiran na početku projekta pa je došlo do nerazumijevanja ukupnog posla kojeg je bilo potrebno odraditi.

Bilo je problema i po pitanju komunikacije između internog i vanjskog razvojnog tima.

Rokovi su bili nerealni i teško dostižni pa su timovi pokušavali prebaciti posao jedni na druge, a čak i unutar tima na druge članove tima.

Nismo imali ključnog korisnika odnosno korisnici nisu aktivno sudjelovali na projektu.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** U tijeku projekta uveli smo JIRAu pa je to riješilo problem komunikacije u smislu transparentnosti i praćenja napretka po radnim zadacima.

Uvedeni su operativni sastanci s razvojnim timovima čak 2 puta dnevno kako bi se pratio napredak i definirali prioriteti.

Općenito, jako sam puno vremena i truda uložio u komunikaciju i sastanke kako bih otklonio način razmišljanja „to nije moja briga“ i oformio tim.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Kanban.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Ne, koristili smo Kanban.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Ne, imali smo okvirni *break down structure*, ali nije bilo napravljeno planiranje.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Faze se nisu službeno zatvarale i došlo je do preklapanja svih navedenih faza. Definiranje zahtjeva protezalo se praktički kroz sve faze jer na početku nisu bili jasno definirani.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Testiranje su korisnici vršili tek po završetku razvoja. Imali smo 3 modula koja smo puštali u produkciju. Tek kad bi bio dovršen cijeli modul korisnici su pristupili testiranju.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Ne, budžet je premašen za 1,5 puta (odnosno potrošeno je 150% više od planiranog)

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Ne, kašnjenje je bilo 16 mjeseci.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** Ne, nije bio zadovoljen osnovni zahtjev da web-aplikacija ima sve funkcionalnosti koje je imao stari sustav. Web-aplikacija se stoga još uvijek doraduje. Testni scenariji nisu bili unaprijed definirani.

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Ne bih. Rascjepkao bih projekt na isporuke odnosno cjeline koje treba u sklopu njega napraviti, a koje mogu samostalno biti puštene u produkciju. Po navedenim cjelinama osigurao bih da od korisnika dobijemo jasno definirane zahtjeve za sljedeću cjelinu. Nakon toga bismo napravili razvoj po definiranim zahtjevima i prošli s korisnikom napravljeni. Na temelju povratne informacije od korisnika zatvorili bismo cjelinu i pustili je u produkciju ili definirali dodatne zahtjeve ako ih ima. Dakle, odlučio bih se za agilni iterativni pristup, ali na puno uređeniji način uz jasno definiranje korisničkih zahtjeva koje je potrebno zadovoljiti u idućoj razvojnoj iteraciji. Da je projekt za vanjskog Klijenta svakako bih planiranje odradio po vodopadnom modelu radi preciznog definiranja budžeta i rokova, a sam bih Razvoj radio agilno.

### **2.1.7. Dubinski intervju\_7**

**PITANJE 1 (P1):** Koliko ste dugo u ulozi voditelja projekata?

**ODGOVOR 1 (O1):** 5 godina.

**PITANJE 2 (P2):** Koliko imate godina?

**ODGOVOR 2 (O2):** 33.

**PITANJE 3 (P3):** Imate li formalnu edukaciju iz Upravljanja projektima? Ako je odgovor da, o kojoj je edukaciji riječ?

**ODGOVOR 3 (O3):** Nemam.

**PITANJE 4 (P4):** Koliko je trajao posljednji projekt razvoja web-aplikacije kojeg ste vodili (izraziti se u mjesecima)?

**ODGOVOR 4 (O4):** 7 mjeseci.

**PITANJE 5 (P5):** Koliko je bilo članova projektnog tima?

**ODGOVOR 5 (O5):** Razvojni tim imao je 5 članova - voditelja projekta, 2 programera, specijalista za testiranje softvera i *Product Ownera*. Sa strane Klijenta bila su 2 člana tima.

**PITANJE 6 (P6):** Kako biste opisali projektni tim u smislu komunikacije, interakcija s Vama, odnosa prema radu, zadovoljavanja rokova i pregovaranja?

**ODGOVOR 6 (O6):** Do samog roka razvojni me je tim uvjeravao da će stići, međutim pogriješili su u procjenama i u posljednji me čas informirali da će doći do kašnjenja pa bih rekla da komunikacija nije bila transparentna.

**PITANJE 7 (P7):** Je li se u sklopu tog projekta razvijala nova web-aplikacija ili doradivala postojeća?

**ODGOVOR 7 (O7):** Razvijala se nova web-aplikacija.

**PITANJE 8 (P8):** Je li projekt uključivao i integraciju drugih programskih sustava?

**ODGOVOR 8 (O8):** Da, uključivao je integraciju s 2 programska sustava.

**PITANJE 9 (P9):** Jesu li korisnički zahtjevi bili jasno definirani na početku projekta?

**ODGOVOR 9 (O9):** Nisu, u tijeku projekta je otkriveno da Klijent ima dodatnih očekivanja od web-aplikacije, a koja nam na početku nisu bila poznata.

**PITANJE 10 (P10):** Je li u tijeku projekta došlo do značajnih izmjena zahtjeva?

Ako da, u kojoj fazi razvoja su promjene bile najintenzivnije?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva
- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 10 (O10):** Da, promjene zahtjeva dogodile su se u fazi 5) Testiranje.

**PITANJE 11 (P11):** Je li u tijeku projekta došlo do promjene arhitekture sustava jer se inicijalno zamišljena arhitektura pokazala nezadovoljavajućom?

Ako da, u kojoj fazi razvoja se to dogodilo?

Faze u životnom ciklusu projekta raspoložive za odabir su:

- 1) Planiranje i analiza korisničkih zahtjeva

- 2) Definiranje zahtjeva
- 3) Dizajniranje rješenja
- 4) Razvoj
- 5) Testiranje
- 6) Produkcija

**ODGOVOR 11 (O11):** Da, nažalost nakon puštanja produkcije uočili smo da je sustav prespor pa smo nakon produkcije (dakle, u fazi 6) Produkcija) radili refaktoriranje koda i izmjene arhitekture.

**PITANJE 12 (P12):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) pronađenih u tijeku testiranja bio veći od 10?

**ODGOVOR 12 (O12):** Da, bilo je oko 15.

**PITANJE 13 (P13):** Je li broj kritičnih grešaka u kodu (eng. *bugs*) uočenih nakon produkcije bio veći od 5?

**ODGOVOR 13 (O13):** Nije, bila su 3 kritična *buga* u produkciji.

**PITANJE 14 (P14):** Što biste izdvojili kao najveće izazove koji su se pojavili u projektu?

**ODGOVOR 14 (O14):** Najveći izazov bili su novi zahtjevi i zahtjevi za izmjenama koji su se pojavili u tijeku projekta.

**PITANJE 15 (P15):** Kako ste riješili izazove iz Pitanja 14?

**ODGOVOR 15 (O15):** Podijelila sam projekt na 2 faze - u prvoj fazi isporučene su osnovne funkcionalnosti iz inicijalnog opsega, a u drugoj fazi napravljeni su dodatni zahtjevi.

**PITANJE 16 (P16):** Koju ste metodologiju upravljanja projektima koristili?

**ODGOVOR 16 (O16):** Vodopadni model.

**PITANJE 17 (P17):** Jeste li kombinirali različite metodologije upravljanja projektima? Ako je odgovor da, koje ste metodologije kombinirali i koje ste projektne faze/aktivnosti provodili po kojoj metodologiji?

**ODGOVOR 17 (O17):** Nismo kombinirali metodologije upravljanja projektima, koristili smo vodopadni model.

**PITANJE 18 (P18):** Je li na početku projekta napravljen kompletan plan projekta (uključujući jasno definirani opseg projekta i korisničke zahtjeve) po fazama projekta?

**ODGOVOR 18 (O18):** Je, na početku projekta napravljen je kompletan plan projekta.

**PITANJE 19 (P19):** Jesu li se faze projekta službeno „zatvarale“ prije prelaska na sljedeću fazu ili je došlo do preklapanja u fazama definiranja zahtjeva, dizajniranja rješenja, razvoja i testiranja?

**ODGOVOR 19 (O19):** Faze se nisu službeno zatvarale s Klijentom, ali se nisu preklapale.

**PITANJE 20 (P20):** Jesu li korisnici aktivno sudjelovali u fazi testiranja koja su se provodila za vrijeme razvoja ili su testiranje vršili tek po završetku razvoja cijelog programskog sustava?

**ODGOVOR 20 (O20):** Korisnici su testiranje vršili tek po završetku razvoja cijelog programskog sustava.

**PITANJE 21 (P21):** Je li projekt završen u okviru zadanog budžeta?

**ODGOVOR 21 (O21):** Inicijalni opseg (Faza 1) završen je u okviru budžeta, no s obzirom na dodatne zahtjeve koji su se pojavili u tijeku projekta – ne.

**PITANJE 22 (P22):** Je li projekt završen u roku?

**ODGOVOR 22 (O22):** Inicijalni opseg isporučen je u roku, međutim zbog dodatnih zahtjeva projekt se oduljio za dodatnih mjesec dana.

**PITANJE 23 (P23):** Je li projekt zadovoljio standarde kvalitete koji su bili postavljeni?

**ODGOVOR 23 (O23):** Je, projekt je zadovoljio standarde kvalitete i prošao sve testne scenarije (koji su bili unaprijed definirani).

**PITANJE 24 (P24):** Ako u budućnosti trebate voditi sličan projekt, biste li ga vodili po istoj metodologiji upravljanja projektima? Ako je odgovor ne, što biste promijenili?

**ODGOVOR 24 (O24):** Ne bih, vodila bih 4) Razvoj po Scrum metodologiji jer smatram da smo korisnike odnosno Klijenta trebali uključiti u testiranje puno ranije dok je još trajao razvoj kako bismo na vrijeme dobili od njih povratnu informaciju.

## ***2.2. Analiza rezultata kvalitativnog istraživanja***

Na većini projekata koji su bili predmet kvalitativnog istraživanja glavni su izazov predstavljale promjene korisničkih zahtjeva u tijeku projekta i nepoznat opseg projekta na početku projekta odnosno promjena opsega u tijeku projekta. Čak i projekti na kojima je na početku napravljeno planiranje naišli su na brojne izmjene u tijeku projekta pa inicijalni plan projekta više nije bio važeći.

Izmjene korisničkih zahtjeva događale su se najčešće pred sam kraj projekta u fazama Razvoja i/ili čak Testiranja.

Dodatno, kao čest problem na projektima istaknuta je komunikacija među dionicima - prvenstveno s Klijentom, ali i između razvojnih timova ili čak između članova istog razvojnog tima.

Svi su projekti koji su bili predmet kvalitativnog istraživanja iziskivali integraciju s minimalno 2 vanjska programska sustava.

Izmjene u arhitekturi sustava u tijeku projekta u odnosu na inicijalno zamišljenu arhitekturu dogodile su se na polovici projekata koji su bili predmet istraživanja, no u različitim fazama projekta.

Što je potreba za izmjenom korisničkih zahtjeva ili arhitekture detektirana kasnije na projektu, to su problemi na projektu bili veći.

Gotovo svi projekti (njih 6 od 7) u sklopu ovog istraživanja imali su prijavljeno više od 10 kritičnih neispravnosti u fazi Testiranja. Većina projekata (također njih 6 od 7) je pak u produkciju izašla s manje od 5 kritičnih neispravnosti.

Na većini projekata korištena je kombinacija tradicionalnih i agilnih metodologija upravljanja projektima odnosno hibridna metodologija upravljanja projektima.

Od tradicionalnih metodologija najčešće je korišten vodopadni model, a od agilnih modela korišteni su Scrum i Kanban.

Agilni su modeli većinom korišteni u fazi Razvoja i Testiranja, a početne faze projekta Planiranje i analiza korisničkih zahtjeva, Definiranje zahtjeva i Dizajniranje rješenja vođeni su po vodopadnom modelu (ili je barem tako bilo planirano).

Na većini projekata na kojima nisu korištene hibridne metodologije upravljanja projektima korištene su samo agilne metodologije upravljanja projektima – Scrum ili Kanban, a samo je jedan projekt u potpunosti vođen po vodopadnom modelu.

Na 6 od 7 projekata koji su bili predmet istraživanja došlo je do preklapanja u fazama projekta.

Više od polovice projekata završeno je u okviru budžeta, no na gotovo svim projektima (na njih 6 od 7) došlo je do kašnjenja produkcije odnosno prekoračenja postavljenih rokova.

Na većini projekata (na njih 4 od 7) korisnici su sudjelovali u testiranju za vrijeme razvoja, a ne tek po završetku razvoja.

Uočena je jasna korelacija između kašnjenja projekata i nesudjelovanja korisnika u testiranju za vrijeme razvoja odnosno projekti na kojima su korisnici testirali tek po završetku razvoja najviše su kasnili jer su potrebe za izmjenom zahtjeva (ili dodatnim zahtjevima) i neispravnosti uočeni prilično kasno.

Ipak, većina je projekata na poslijetku zadovoljila postavljene standarde kvalitete.

Rezultati kvalitativnog istraživanja sistematizirani su u Tablici 8.

Tablica 8. Presjek rezultata kvalitativnog istraživanja

Mjera/Rbr intervjua	1	2	3	4	5	6	7
Broj godina staža u vođenju projekata	3	18	20	18	8	15	5
Dob voditelja projekta	29	50	49	47	33	52	33
Formalno obrazovanje iz upravljanja projektima	NE	NE	DA	DA	NE	NE	NE
Trajanje projekta (u mjesecima)	18	6	7	18	30	22	7
Broj članova projektnog tima	11	9	11	26	17	11	5
Nova web-aplikacija ili dorada postojeće	Dorada	Oboje	Dorada	Nova	Nova	Nova	Nova
Broj integracija	2	5	6	5	6	3	2
Izmjene korisničkih zahtjeva u tijeku projekta	DA	DA	DA	NE	DA	DA	DA
Faza(e) projekta u kojoj je došlo do izmjene korisničkih zahtjeva	Dizajniranje rješenja; Razvoj	Razvoj; Testiranje	Razvoj; Testiranje	-	Testiranje	Testiranje	Testiranje
Izmjene u arhitekturi sustava u tijeku projekta u odnosu na inicijalni nacrt	NE	NE	NE	DA	DA	DA	DA
Faza(e) projekta u kojoj je došlo do izmjene arhitekture	-	-	-	Dizajniranje rješenja	Testiranje	Razvoj	Produkcija
Broj kritičnih bugova uočenih u testiranju	>10	>10	<10	>10	>10	>10	>10
Broj kritičnih bugova u produkciji	<5	<10	<5	<5	<5	<5	<5
Korištena metodologija(e) upravljanja projektima	Vodopadni model; Scrum	Scrum	Vodopadni model; Scrum; Kanban	Lean; Scrum	Vodopadni model; Lean; Scrum	Kanban	Vodopadni model
Na početku napravljen plan projekta	DA	NE	DA	DA	DA	NE	DA
Preklapanje faza projekta	DA	DA	DA	DA	DA	DA	NE
Korisnici sudjelovali u	DA	DA	DA	DA	NE	NE	NE



<b>testiranju za vrijeme razvoja</b>							
<b>Projekt završen u okviru budžeta</b>	DA	DA	DA	DA	NE	NE	NE
<b>Kašnjenje projekta (u mjesecima)</b>	6	1	-	2	12	16	1
<b>Zadovoljeni postavljeni standardi kvalitete</b>	DA	DA	DA	DA	NE	NE	DA

### 3. Kvantitativno istraživanje mišljenja članova projektnih timova o primijenjenim metodologijama upravljanja projektima na projektima razvoja web-aplikacija

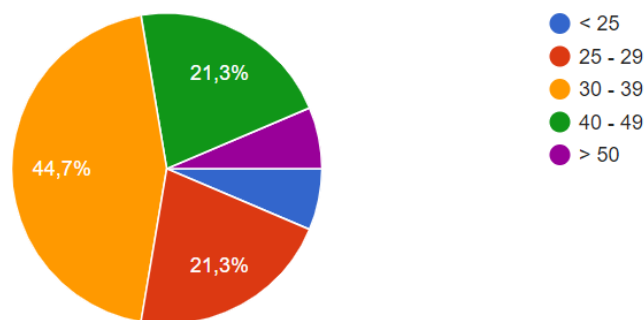
Za kvantitativno istraživanje mišljenja članova projektnih timova o projektima razvoja web-aplikacija te njihovih stavova vezano za primjenu metodologija upravljanja projektima u različitim projektnim fazama korišten je anketni upitnik generiran u aplikaciji *Google Forms* (Dodatak). Odgovori su prikupljeni u periodu od 29. kolovoza do 5. rujna 2022. godine. Anketni je upitnik prosljeđen ispitanicima iz 9 različitih tvrtki putem elektroničke pošte i osobnih kontakata.

#### 3.1. Socioekonomski podaci

U sklopu ankete ukupno je prikupljeno 47 odgovora.

Najveći udio ispitanika pripada dobnoj skupini 30-39 godina (44,7%), nakon čega slijede dobne skupine 25-29 godina i 40-49 godina (obje s udjelom od 21,3%). U dobnim skupinama do 25 godina i iznad 50 godina nalazi se po 6,4% ispitanika (Slika 18).

47 odgovora

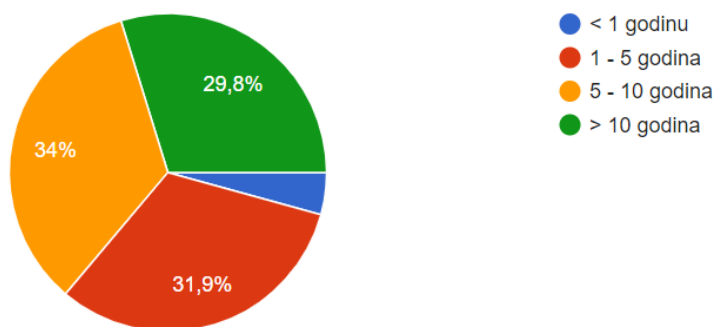


Slika 18. Dob ispitanika (u godinama)

#### 3.2. Radno iskustvo na projektima razvoja web-aplikacija

Ukupno 34% (16) ispitanika ima između 5 i 10 godina radnog iskustva na projektima razvoja web-aplikacija. Njih 31,9% (15) ima između 1 i 5 godina iskustva, a 29,8% (14) preko 10 godina. Tek 4,3% (2) ispitanika ima manje od 1 godine radnog iskustva na projektima razvoja web-aplikacija (Slika 19).

47 odgovora



Slika 19. Radno iskustvo na projektima razvoja web-aplikacija (u godinama)

### ***3.3. Iskustvo s posljednjeg projekta razvoja web-aplikacije***

Pitanja 3-14 odnosila su se na posljednji projekt razvoja web-aplikacije na kojem je ispitanik sudjelovao.

#### **3.3.1. Trajanje projekta razvoja web-aplikacije**

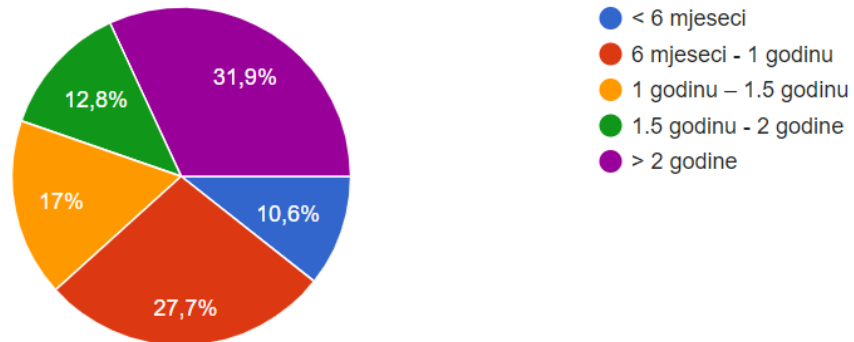
Većina ispitanika (31,9% odnosno njih 15) navelo je da je posljednji projekt razvoja web-aplikacije na kojem su radili trajao više od 2 godine.

Ukupno 27,7% odnosno 13 ispitanika navelo je da je njihov posljednji projekt razvoja web-aplikacije trajao između 6 mjeseci i 1 godine.

Ukupno 17% odnosno 8 ispitanika navelo je trajanje između 1 i 1,5 godine, a 12,8% odnosno 6 ispitanika između 1,5 i 2 godine.

Najmanje ispitanika (10,6% odnosno njih 5) navelo je da je njihov posljednji projekt razvoja web-aplikacije trajao kraće od pola godine (Slika 20).

47 odgovora



Slika 20. Trajanje posljednjeg projekta razvoja web-aplikacije na kojem je ispitanik sudjelovao

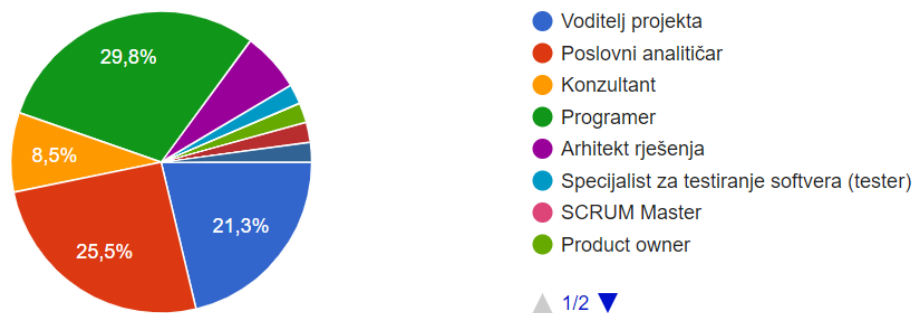
### 3.3.2. Projektna uloga na projektu razvoja web-aplikacije

Najviše ispitanika na svom se posljednjem projektu razvoja web-aplikacije našlo u ulozi programera (njih 29,8% odnosno 14).

Značajan udio imale su i uloge poslovnog analitičara (25,5% ispitanika odnosno njih 12) te voditelja projekta (21,3% ispitanika odnosno njih 10).

U ulozi konzultanta bilo je 8,5% ispitanika odnosno njih 4, a u ulozi arhitekta rješenja 6,4% ispitanika odnosno njih 3 (Slika 21).

47 odgovora



Slika 21. Projektna uloga na posljednjem projektu razvoja web-aplikacije

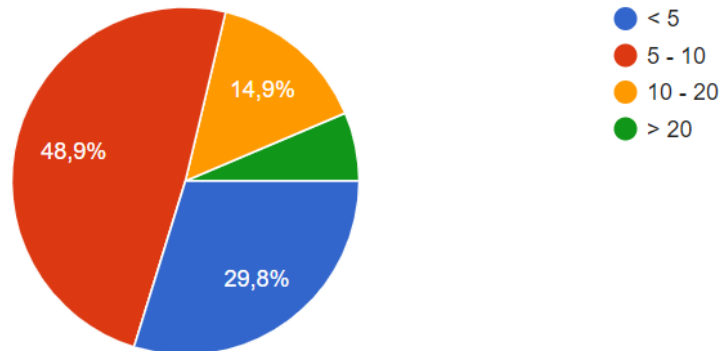
### 3.3.3. Broj članova projektnog tima

Većina (48,9% odnosno 23) ispitanika navelo je da je projektni tim na posljednjem projektu razvoja web-aplikacije imao između 5 i 10 članova.

Usljedili su projektni timovi koji su imali manje od 5 članova, što je bio slučaj kod 29,8% odnosno 14 ispitanika. 14,9% odnosno 7 ispitanika bili su članovi projektnih timova koji su

brojali između 10 i 20 članova. Najmanje ispitanika (6,4% odnosno 3) izjasnilo se da je projektni tim imao više od 20 članova (Slika 22).

47 odgovora



Slika 22. Broj članova projektnog tima

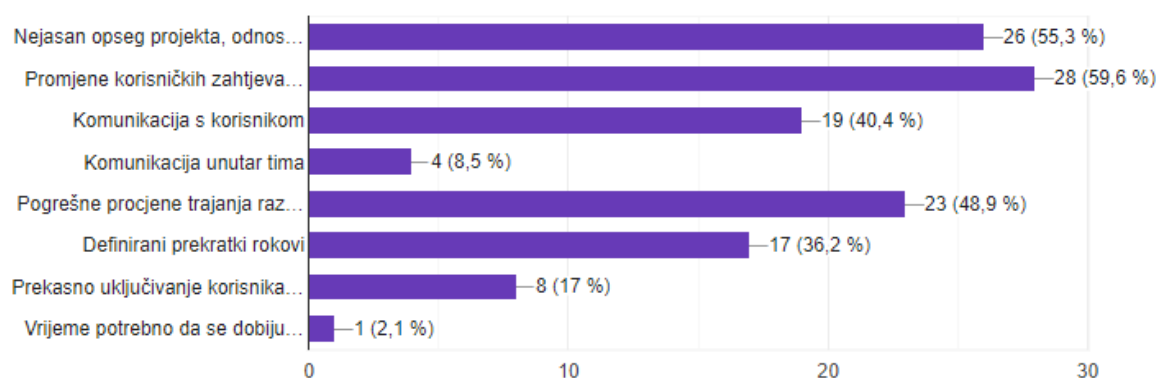
### 3.3.4. Najveći izazovi na projektu

Na pitanju 6 vezanom za najveće izazove koji su se pojavili na posljednjem projektu razvoja web-aplikacije na kojem je ispitanik sudjelovao bio je dozvoljen višestruk odabir.

Izazovi su poredani sljedećim redoslijedom od najvećih prema najmanjima (Slika 23):

- Promjene korisničkih zahtjeva u tijeku projekta (59,6% odnosno 28 odgovora)
- Nejasan opseg projekta, odnosno nejasni korisnički zahtjevi na početku projekta (55,3% odnosno 26 odgovora)
- Pogrešne procjene trajanja razvojnih aktivnosti (48,9% odnosno 23 odgovora)
- Komunikacija s korisnikom (40,4% odnosno 19 odgovora)
- Definirani prekratki rokovi (36,2% odnosno 17 odgovora)
- Prekasno uključivanje korisnika u testiranje (17% odnosno 8 odgovora)
- Komunikacija unutar tima (8,5% odnosno 4 odgovora)
- Vrijeme potrebno da se dobiju svi pristupi potrebni za početak razvoja (2,1% odnosno 1 odgovor).

47 odgovora

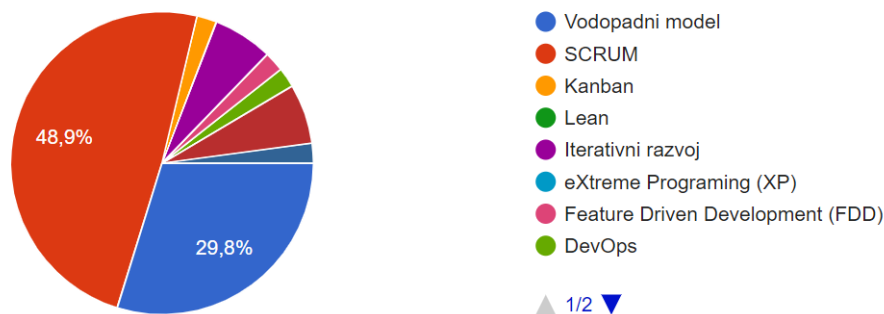


Slika 23. Najveći izazovi na projektu razvoja web-aplikacije

### 3.3.5. Korištena metodologija upravljanja projektom

Čak 48,9% odnosno 23 ispitanika izjasnilo se da je na posljednjem projektu razvoja web-aplikacije primarna metodologija upravljanja projektom bila Scrum. Slijedio je vodopadni model s 29,8% odnosno 14 odgovora (Slika 24).

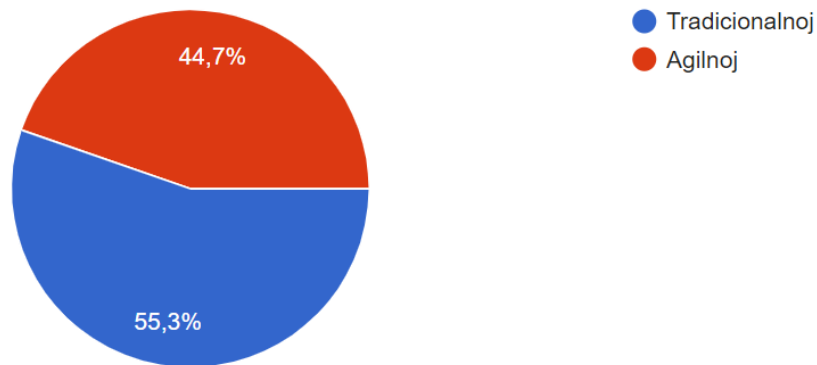
47 odgovora



Slika 24. Metodologija upravljanja projektom po kojem je općenito vođen projekt razvoja web-aplikacije

Ukupno 55,3% odnosno 26 ispitanika izjasnilo se da projekt u projektnoj fazi Planiranja i analize korisničkih zahtjeva vođen po tradicionalnoj metodologiji upravljanja projektima, a njih 44,7% odnosno 21 da je u navedenoj fazi vođen po agilnoj metodologiji upravljanja projektima (Slika 25).

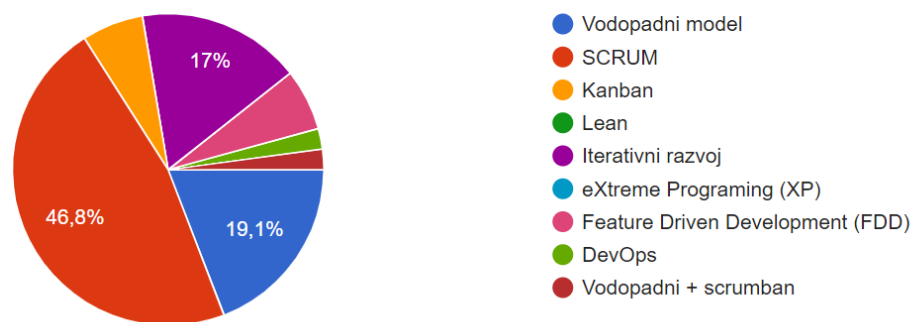
47 odgovora



Slika 25. Metodologija upravljanja projektom u projektnoj fazi Planiranja i analize korisničkih zahtjeva

Ukupno 46,8% odnosno 22 ispitanika izjasnilo se da projekt u projektnoj Razvoja (odnosi se na programiranje) vođen po Scrumu, a njih 19,1% odnosno 9 da je u navedenoj fazi vođen po vodopadnom modelu. Na trećem mjestu našao se iterativni razvoj sa 17% odnosno 8 odgovora (Slika 26).

47 odgovora



Slika 26. Metodologija upravljanja projektom u projektnoj fazi Razvoja

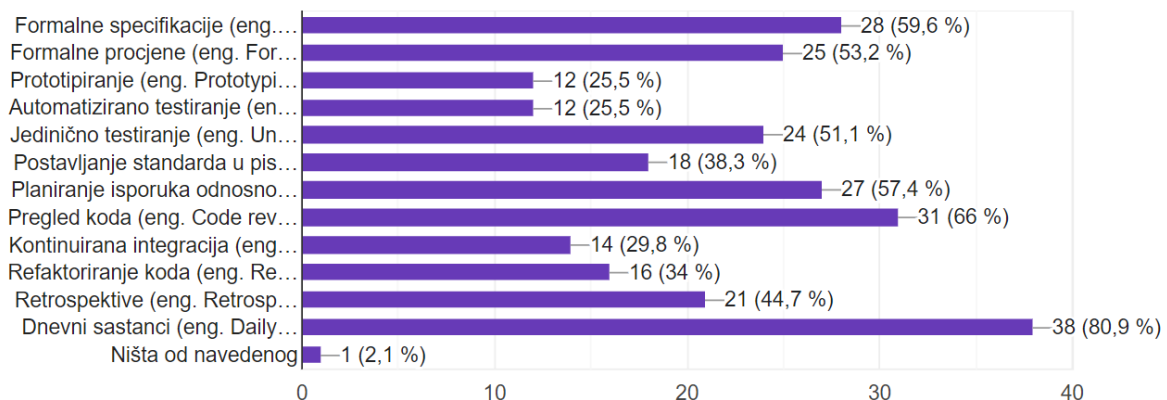
### 3.3.6. Korištene prakse razvoja softvera

Na pitanju 10 za odabir korištenih praksi razvoja softvera bio je dozvoljen višestruki odabir. Poredak praksi razvoja softvera od onih najčešće korištenih prema onim rjeđe korištenima bio je sljedeći (Slika 27):

- Dnevni sastanci (eng. *Daily Standup*) – 80,9% odnosno 38 odgovora
- Pregled koda (eng. *Code review*) – 66% odnosno 31 odgovor

- Formalne specifikacije (eng. *Formal Specifications*) – 59,6% odnosno 28 odgovora
- Planiranje isporuka odnosno planiranje puštanja rješenja u produkciju (eng. *Release Planning*) – 57,4% odnosno 27 odgovora
- Formalne procjene (eng. *Formal Estimation*) – 53,2% odnosno 25 odgovora
- Jedinično testiranje (eng. *Unit Testing*) – 51,1% odnosno 24 odgovora
- Retrospektive (eng. *Retrospectives*) – 44,7% odnosno 21 odgovor
- Postavljanje standarda u pisanju koda (eng. *Coding Standards*) – 38,3% odnosno 18 odgovora
- Refaktoriranje koda (eng. *Refactoring*) – 34% odnosno 16 odgovora
- Kontinuirana integracija (eng. *Continous integration*) – 29,8% odnosno 14 odgovora
- Prototipiranje (eng. *Prototyping*) – 25,5% odnosno 12 odgovora
- Automatizirano jedinično testiranje (eng. *Automated Unit Testing*) - 25,5% odnosno 12 odgovora.

47 odgovora



Slika 27. Korištene prakse razvoja softvera

Navedeno ukazuje na to da su se u velikoj mjeri koristile i prakse karakteristične za agilne metodologije upravljanja projektima (kao što su primjerice Dnevni sastanci) te prakse karakteristične za tradicionalne metodologije upravljanja projektima (kao što su primjerice Formalne specifikacije i Formalne procjene).

Svih 5 praksi (Pregled koda (eng. *Code review*), Postavljanje standarda u pisanju koda (eng. *Coding Standards*), Planiranje isporuke odnosno planiranje puštanja rješenja u produkciju (eng. *Release Planning*), Prototipiranje (eng. *Prototyping*), Automatizirano jedinično testiranje (eng. *Automated Unit Testing*)) koje su detektirane kao često korištene u hibridnim metodologijama upravljanja projektima u radu [13] korištene su na posljednjem projektu razvoja web-aplikacija



više od četvrtine ispitanika, što ukazuje na kombiniranje metodologija upravljanja projektima odnosno korištenje hibridnih modela.

### 3.3.7. Preklapanje projektnih faza

Pitanje 11 odnosilo se na preklapanje projektnih faza i pokazalo je da se najčešće s nekom od ostalih projektnih faza preklapala faza Razvoja/programiranja, i to na posljednjem projektu razvoja web-aplikacije čak 74,5% (35 odgovora) ispitanika.

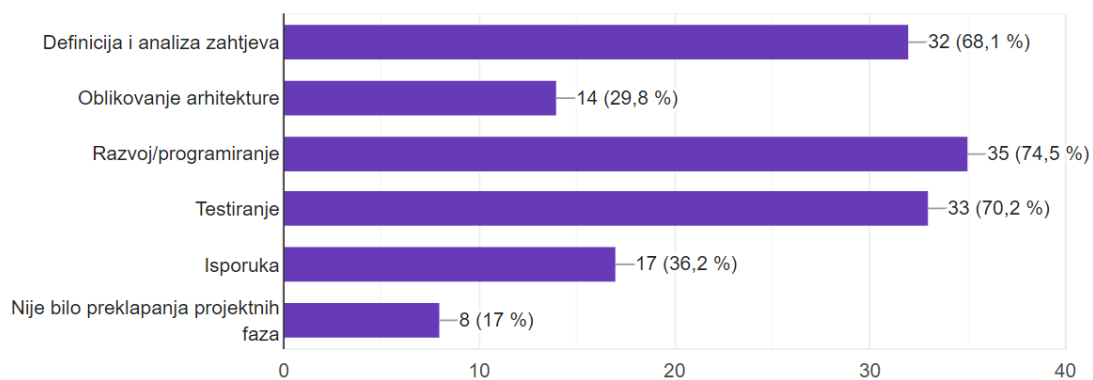
Slijedile su faze Testiranja s 70,2% (33 odgovora) i Definicije i analize zahtjeva sa 68,1% (32 odgovora).

Isporuca se s nekom od ostalih projektnih faza preklapala na projektima 36,2% ispitanika (17 odgovora), a projektna faza Oblikovanje arhitekture kod 29,8% (14 odgovora).

17% ispitanika (8 odgovora) navelo je kako na njihovom posljednjem projektu razvoja web-aplikacije uopće nije došlo do preklapanja projektnih faza (Slika 28).

Zaključak je da je kod velike većine projekata razvoja web-aplikacija (preko 80%) koji su bili predmet istraživanja došlo do preklapanja projektnih faza.

47 odgovora



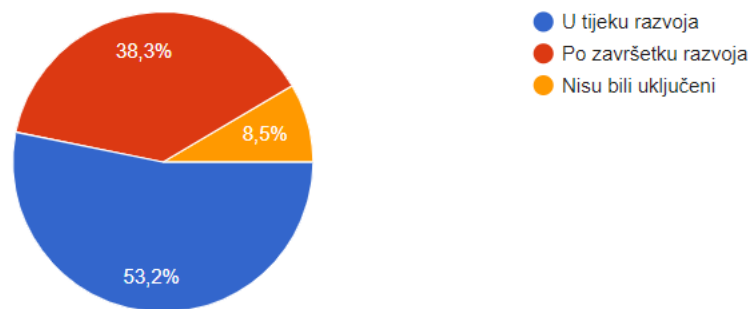
Slika 28. Projektne faze koje su se preklapale s drugim projektnim fazama

Sljedeće pitanje za cilj je imalo utvrditi koje su se faze međusobno najčešće preklapale. Zaključak je da su se najčešće međusobno preklapale faze Razvoja/programiranja i Testiranja, a na drugom mjestu našla se kombinacija Definicije i analize zahtjeva te Razvoja/programiranja.

### 3.3.8. Uključivanje korisnika u testiranje

Više od polovine (53,2%) ispitanika navelo je kako su korisnici bili uključeni u testiranje u tijeku razvoja. Njih 38,3% navelo je kako su korisnike u testiranje uključili tek po završetku Razvoja, a samo 8,5% ispitanika navelo je da korisnici na njihovom posljednjem projektu razvoja web-aplikacije uopće nisu bili uključeni u testiranje (Slika 29).

47 odgovora



Slika 29. Uključivanje korisnika u testiranje na projektima razvoja web-aplikacija

### 3.3.9. Kašnjenje projekta

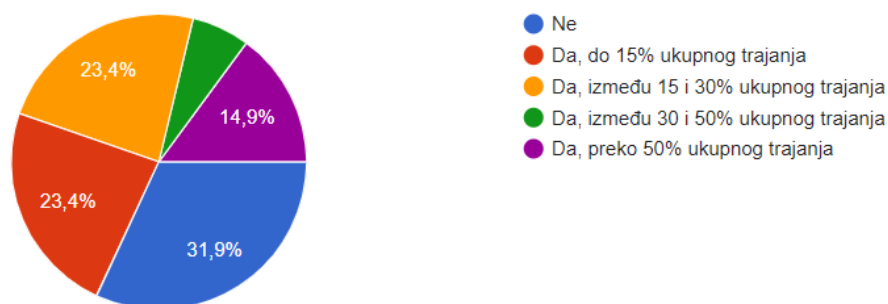
31,9% (15) ispitanika radilo je na projektu razvoja web-aplikacije na kojem nije došlo do kašnjenja.

Navedeno ukazuje na činjenicu da je na većini projekata razvoja web-aplikacija došlo do kašnjenja odnosno rokovi nisu bili dostignuti.

Kašnjenja su bila zastupljena u sljedećim postocima (Slika 30):

- Kod 23,4 % (11) ispitanika – do 15% ukupnog trajanja projekta
- Kod 23,4 % (11) ispitanika – između 15 i 30% ukupnog trajanja projekta
- Kod 14,9% (7) ispitanika – preko 50% ukupnog trajanja projekta
- Kod 6,4% (3) ispitanika – između 30 i 50% ukupnog trajanja projekta.

47 odgovora

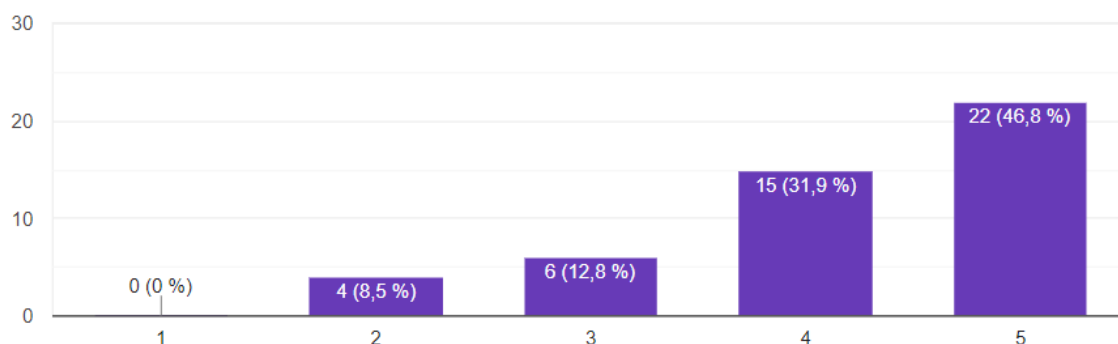


Slika 30. Kašnjenje projekata razvoja web-aplikacija

### 3.4. Stav ispitanika o načinu upravljanja projektima razvoja web-aplikacija

Značajna većina ispitanika smatra da je Iznimno važno (46,8%) i Važno (31,9%) da se planiranje, analiza korisničkih zahtjeva i definiranje arhitekture sustava provedu maksimalno moguće detaljno na početku projekta, a tek onda krene s ostalim razvojnim fazama. Tek 4 ispitanika (8,5%) smatraju da navedeno nije važno (Slika 31).

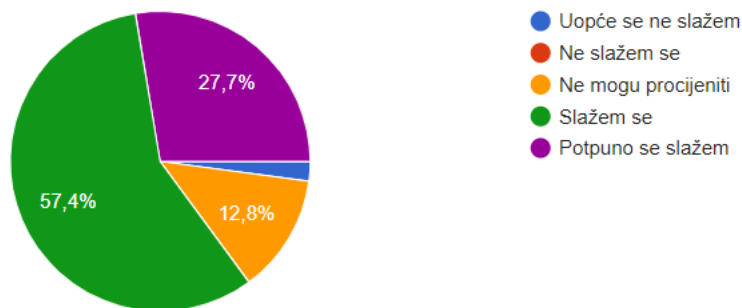
47 odgovora



Slika 31. Važnost provođenja planiranja, analize korisničkih zahtjeva i definiranja arhitekture sustava maksimalno moguće detaljno na početku projekta po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno)

Značajna većina ispitanika smatra da je u fazi Razvoja (odnosi se na programiranje) bolje koristiti Agilne metodologije upravljanja projektima. Njih 57,4% (27) s navedenim se slaže, a njih 27,7% (13) potpuno se slaže (Slika 32).

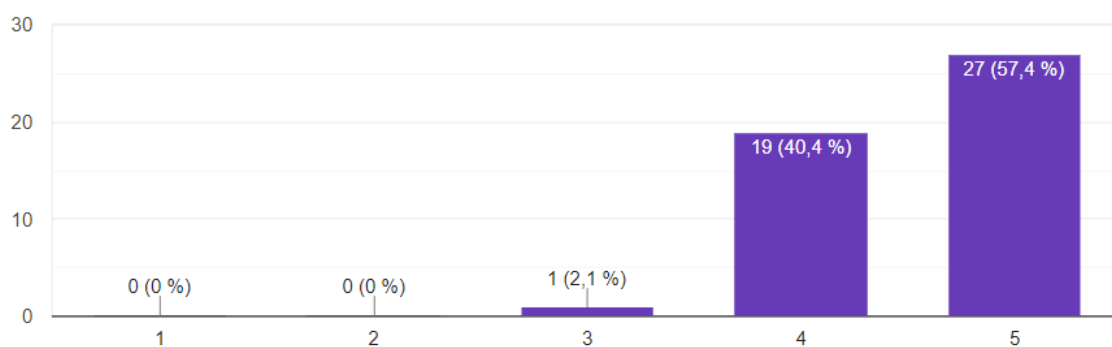
47 odgovora



Slika 32. Stav prema tvrdnji da je u fazi Razvoja/programiranja bolje koristiti Agilne metodologije upravljanja projektima

Gotovo svi ispitanici smatraju da je Iznimno važno (57,4% odnosno njih 27) ili Važno (40,4% odnosno njih 19) isporučivati korisnicima više manjih razvojnih verzija na testiranje (Slika 33).

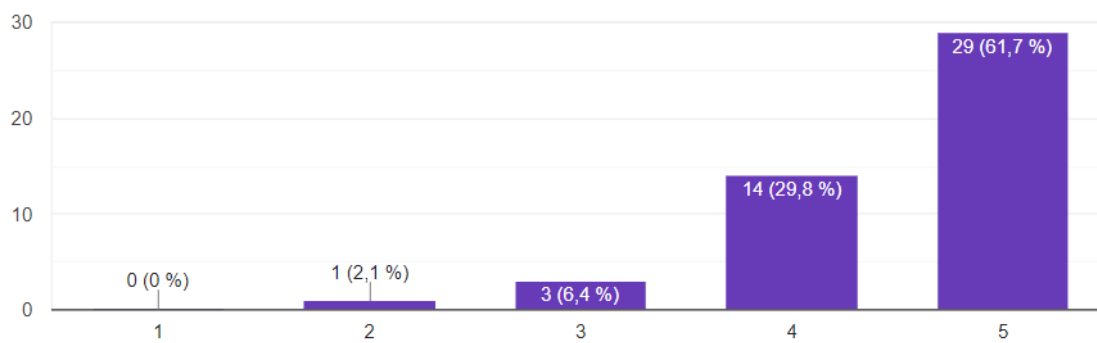
47 odgovora



Slika 33. Važnost isporučivanja više manjih razvojnih verzija korisniku na testiranje po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno)

Značajna većina ispitanika smatra da je Iznimno važno (61,7% odnosno njih 29) ili Važno (29,8% odnosno njih 14) dobro dokumentirati zahtjeve i arhitekturu na početku projekta u smislu da olakšavaju razvoj i održavanje softvera kasnije (Slika 34).

47 odgovora



Slika 34. Važnost dobro dokumentiranih zahtjeva i arhitekture sustava na početku projekta po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno)

## **4. Prijedlozi i smjernice za kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima**

Slijedom provedenih istraživanja, vidljivo je da projektni timovi kombiniraju različite metodologije upravljanja projektima odnosno koriste hibridne modele kako bi najbolje odgovorili na izazove projekta.

U sklopu kvalitativnog istraživanja načina vođenja projekata razvoja web-aplikacija te kvantitativnog istraživanja mišljenja članova projektnih timova o primijenjenim metodologijama upravljanja projektima razvoja web-aplikacija identificirano je da su najveći izazovi na projektima razvoja web-aplikacija promjene korisničkih zahtjeva u tijeku projekta, nejasan opseg projekta, odnosno nejasno definirani korisnički zahtjevi na početku projekta, pogrešne procjene trajanja razvojnih aktivnosti, komunikacija s korisnikom te definirani prekratki rokovi. Uzroci problema kriju se, dakle, u nejasnom definiranju korisničkih zahtjeva te posljedično pogrešnim procjenama trajanja razvojnih aktivnosti i roka završetka projekta.

Promjene korisničkih zahtjeva u tijeku projekta zauzele su prvo mjesto kada su u pitanju izazovi na projektima razvoja web-aplikacija, no njih nikako ne možemo izjednačiti s problemom nejasno definiranih korisničkih zahtjeva. Naime, promjene korisničkih zahtjeva su realnost i događat će se na gotovo svim projektima jer u tijeku projekta naprosto dolazi do promjena na tržištu i korisnik mora ići u korak s njima.

Također, u spomenutim je istraživanjima identificirano da velika većina ispitanika smatra da se planiranje, analiza korisničkih zahtjeva i definiranje arhitekture sustava trebaju provesti maksimalno moguće detaljno na početku projekta, a tek onda krenuti s ostalim razvojnim fazama te da je od velikog značaja dobro dokumentirati korisničke zahtjeve i arhitekturu sustava na početku projekta (u smislu da navedeno olakšava razvoj i održavanje softvera kasnije).

Uz to, većina je ispitanika navela kako smatraju da je važno uključiti korisnike u testiranje još u tijeku razvoja.

Slijedom navedenog, smjernice za kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima razvoja web-aplikacija po projektnim fazama pojašnjene su u Tablici 9.

Tablica 9. Smjernice za kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima razvoja web-aplikacija

Projektna faza	Preporučeni pristup	Smjernice
Definicija i analiza zahtjeva	Vodopadni model	<ul style="list-style-type: none"> <li>• Preporuka je prije početka narednih projektnih faza identificirati i jasno definirati korisničke zahtjeve</li> <li>• Korisničke je zahtjeve potrebno dokumentirati</li> <li>• Na temelju jasno definiranih korisničkih zahtjeva potrebno je definirati plan projekta te identificirati troškove i dostižne rokove projekta</li> <li>• Jasno definirani i dokumentirani korisnički zahtjevi značajno će olakšati razvoj i održavanje web-aplikacije u budućnosti</li> </ul>
Oblikovanje arhitekture	Vodopadni model	<ul style="list-style-type: none"> <li>• Nakon Definicije i analize korisničkih zahtjeva, preporuka je definirati arhitekturu sustava prije prelaska na sljedeće projektne faze</li> <li>• Arhitekturu sustava je potrebno dokumentirati</li> <li>• Dokumentirana arhitektura sustava olakšat će razvoj i održavanje web-aplikacije u budućnosti</li> </ul>

Razvoj/programiranje	Agilni pristup	<ul style="list-style-type: none"> <li>• Preporučuje se Razvoj/programiranje raditi u iteracijama i isporučivati inkremente rješenja</li> <li>• Faza Razvoja/programiranja trebala bi se preklapati s projektnom fazom Testiranja</li> </ul>
Testiranje	Agilni pristup	<ul style="list-style-type: none"> <li>• Preporučuje se Testiranje raditi u iteracijama i isporučivati korisnicima veći broj manjih razvojnih verzija na testiranje</li> <li>• Projektna faza Testiranja preklapat će se stoga s projektnom fazom Razvoja/programiranja</li> <li>• Preporučuje se pripremiti testne scenarije koji moraju biti zadovoljeni kako bi se isporučena funkcionalnost smatrala ispravnom i potpunom (eng. <i>Acceptance criteria</i>)</li> <li>• U fazi Testiranja preporučuje se intenzivna komunikacija s korisnicima i prihvaćanje povratnih informacija od njih kako bi se na vrijeme otklonile nepravilnosti i kako bi konačno rješenje</li> </ul>



		zadovoljilo postavljene standarde kvalitete
Isporuka rješenja	Vodopadni model	<ul style="list-style-type: none"> <li>• Isporuku rješenja, koja se odnosi na puštanje određene verzije web-aplikacije u produkciju, preporučuje se napraviti po vodopadnom modelu kao odvojenu projektnu fazu koja se ne preklapa s ostalima</li> <li>• Preporučljivo je formalno zatvoriti projektnu fazu Isporuke rješenja potpisivanjem primopredajnog zapisnika</li> </ul>

Što se praksi u razvoju softvera tiče, u sklopu hibridnih modela preporučuje se, sukladno rezultatima istraživanja, korištenje kombinacije praksi razvoja softvera od kojih su neke karakteristične za tradicionalni pristup poput primjerice Formalnih specifikacija i Formalnih procjena, a neke pak za agilni pristup razvoju softvera poput primjerice Dnevni statusnih sastanaka i Retrospektiva (Tablica 10):

Tablica 10. Preporučene prakse razvoja softvera u hibridnim modelima

<b>Preporučena praksa u razvoju softvera u hibridnim modelima</b>
Formalne specifikacije ( <i>eng. Formal Specifications</i> )
Formalne procjene ( <i>eng. Formal Estimation</i> )
Specifikacije dizajna ( <i>eng. Detailed Designs/Design Specifications</i> )
Modeliranje korisničkih scenarija ( <i>eng. Use Case Modeling</i> )
Specifikacije arhitekture softvera ( <i>eng. Architecture Specifications</i> )
Prototipiranje ( <i>eng. Prototyping</i> )
Postavljanje standarda u pisanju koda ( <i>eng. Coding Standards</i> )
Pregled koda ( <i>eng. Code review</i> )
Refaktoriranje koda ( <i>eng. Refactoring</i> )

Automatizirano jedinično testiranje (eng. <i>Automated Unit Testing</i> )
Jedinično testiranje (eng. <i>Unit Testing</i> )
Planiranje isporuka odnosno planiranje puštanja rješenja u produkciju (eng. <i>Release Planning</i> )
Kontinuirana integracija (eng. <i>Continuous integration</i> )
Retrospektive (eng. <i>Retrospectives</i> )
Dnevni statusni sastanci (eng. <i>Daily Standup</i> )

Zaključno, izuzetno je važno na projektu razvoja web-aplikacije ostvariti pravovremenu i otvorenu komunikaciju s korisnicima te se u tom smislu preporučuje:

- Definiranje i analiza korisničkih zahtjeva zajedno s korisnicima na početku projekta
- Uključivanje korisnika u testiranje još u tijeku razvoja odnosno isporučivanje manjih razvojnih verzija korisnicima na testiranje
- Transparentno evidentiranje promjena ili novih korisničkih zahtjeva koji su se pojavili u tijeku projekta.

## Zaključak

Ovim specijalističkim radom istražena je primjena hibridnih metodologija upravljanja projektima u razvoju programskih sustava, s naglaskom na projekte razvoja web-aplikacija.

Analizom istraživanja primjene hibridnih metodologija upravljanja projektima, kao kombinacije tradicionalnih i agilnih metodologija upravljanja projektima, identificirano je da se u tvrtkama koriste hibridne metodologije upravljanja projektima neovisno o njihovoj veličini te neovisno o sektoru industrije za koju se razvija softver. Također, identificirano je 8 temeljnih metodologija upravljanja projektima od kojih razvojni timovi najčešće kroje svoje hibridne modele, a riječ je o: Scrumu, Iterativnom razvoju, Kanbanu, Vodopadnom modelu, *DevOpsu*, Vodopadnom modelu, *eXtreme Programmingu*, *Leanu* te *Feature Driven Developmentu*.

Analiza istraživanja ukazala je da je agilni pristup zastupljeniji u aktivnostima vezanim za sami razvoj softvera dok su aktivnosti menadžmenta i dalje sklonije tradicionalnom pristupu, potvrđujući time tezu sve češćeg korištenja hibridnih metodologija upravljanja projektima, s blagom tendencijom prema agilnom pristupu.

Na većini projekata razvoja web-aplikacija koji su bili predmet dubinskih intervjua korištena je hibridna metodologija upravljanja projektima, a pritom je od tradicionalnih modela najčešće korišten vodopadni model, a od agilnih Scrum i Kanban.

Početne faze projekata Planiranje i analiza korisničkih zahtjeva, Definiranje zahtjeva i Dizajniranje rješenja vođene su po vodopadnom modelu, a agilni su modeli većinom korišteni u fazama Razvoja i Testiranja.

Dubinski intervjui dali su sliku i o najčešćim izazovima na projektima razvoja web-aplikacija, a riječ je o promjenama korisničkih zahtjeva u tijeku projekta, nepoznatom opsegu projekta na početku projekta i lošoj komunikaciji s korisnikom. Također, uočena je korelacija između kašnjenja projekata i nesudjelovanja korisnika u testiranju za vrijeme razvoja.

Kvantitativno istraživanje kroz anketni upitnik potvrdilo je kako su Scrum i vodopadni model najčešće korištene metodologije na projektima razvoja web-aplikacija, ali i dodatno ukazalo na učestalost preklapanja projektnih faza pri čemu su se najčešće međusobno preklapale faze Razvoja/programiranja i Testiranja. Ispitanici su istaknuli važnost planiranja, analize korisničkih zahtjeva i definiranja arhitekture sustava na početku projekta te isporučivanja korisnicima više manjih razvojnih verzija na testiranje.

Slijedom navedenog, u posljednjem su poglavlju dane smjernice za učinkovito kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima. Analizu i definiciju zahtjeva i Oblikovanje arhitekture preporučuje se provesti po vodopadnom modelu, a Razvoj i Testiranje po agilnom, a posebna je preporuka dana za komunikaciju s korisnicima u smislu definiranja i analize korisničkih zahtjeva zajedno s korisnicima na početku projekta te uključivanje korisnika u testiranje još u tijeku razvoja.



## Literatura

- [1] „A guide to the project management book of knowledge – Seventh edition and The standard for project management“, Project management Institute, Pennsylvania, 2021.
- [2] Car, Ž., „Razvoj komunikacijske programske podrške – nastavni materijali“, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Zagreb, 2021.
- [3] Jović, A., Frid, N., Ivošević, D., „Procesi programskog inženjerstva“, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Zagreb, 2019.
- [4] Roić, R., Ferlež, L., „Agilni razvoj softvera“, Vlastita naklada, Zagreb, 2016.
- [5] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, J., „Agile Manifesto“, dostupno na: <https://agilemanifesto.org/> (27. lipnja 2022.)
- [6] VersionOne, „11th annual State of Agile“, dostupno na: <https://www.agile247.pl/wp-content/uploads/2017/04/versionone-11th-annual-state-of-agile-report.pdf> (27. lipnja 2022.)
- [7] Blake, S., „Your Guide To Agile Software Development Life Cycles“ dostupno na: <https://www.easyagile.com/blog/agile-software-development-life-cycle/> (29. siječnja 2023.)
- [8] Schwaber, K., Beedle, M., „Agile Software development with Scrum“, Prentice Hall PTR, Upper Saddle River, New Jersey, SAD, 2001.
- [9] Mahalakshmi, M., Sundararajan, M., „Traditional SDLC Vs Scrum Methodology – A Comparative Study“, International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 6, lipanj 2013., str. 192-196.
- [10] Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trekter, K., McCaffery, F., Linssen, O., Hanser, E., Prause, C.R., „Hybrid software

and system development in practice: waterfall, SCRUM and beyond“, ICSSP International Conference on Software and System processes, Pariz, Francuska, 2017., str. 30-39.

- [11] Robinson, R.D., „Hybrid: A new management approach“, dostupno na: <https://www.cio.com/article/230639/hybrid-a-new-project-management-approach.html> (07. kolovoza 2022.)
- [12] Kuhrmann, M., Tell, P., Klünder, J., Hebig, R., Licorish, S., MacDonell, S., „HELENA Stage 2 Results“, dostupno na [https://www.researchgate.net/publication/329246439\\_HELENA\\_Stage\\_2\\_Results](https://www.researchgate.net/publication/329246439_HELENA_Stage_2_Results) (06. kolovoza 2022.)
- [13] Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S.G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M., „What are Hybrid Development Methods Made Of? An Evidence-based Characterization“, IEEE/ACM International Conference on Software and System Processes, Montreal, Kanada, 2019., str. 105-114.
- [14] Kuhrmann, M., Münch, J., Diebold, P., Linssen, O., Prause, C.R., „On the Use of Hybrid Development Approaches in Software and Systems Development: Construction and Test of the HELENA Survey“, Projektmanagement und Vorgehensmodelle, Paderborn, Njemačka, 2016., str. 59-68
- [15] Noll, J., Beecam, S., „How Agile Is Hybrid Agile? An Analysis of the HELENA Data“, International Conference on Product Focused Software Development and Process Improvement (PROFES), Barcelona, Španjolska, 2019., str. 341-349.
- [16] Wysocki, W., Orłowski, C., „A multi-agent model for planning hybrid software processes“, Procedia Computer Science, Vol. 159, siječanj 2019., str. 1688-1697.
- [17] Hayata, T., Han, J., „A hybrid model for IT project with Scrum“, IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI, Peking, Kina, 2011., str. 285-290.

- [18] Weilkiens, T., Lamm, J.G., Roth, S., Walker, M., „Model-Based System Architecture“, John Wiley & Sons, Inc., Hoboken, New Jersey, 2016.
- [19] Joslin, R., Müller, R., „The impact of project methodologies on project success in different project environments“, Journal of Managing Projects in Business, Vol. 9, No. 2, veljača 2016., str. 364-388.
- [20] Imani, T., Nakano, M., Anantatmula, V., „Does a Hybrid Approach of Agile and Plan-Driven Methods Work Better for IT System Development Projects?“, Int. Journal of Engineering Research and Application, Vol. 7, No. 3, ožujak 2017., str. 39-46.
- [21] Schmitz, K., Mahaparta, R., Nerur, S., „User engagement in the Era of Hybrid Agile Methodology“, IEEE Software, Vol. 36, No. 4, lipanj 2019., str. 32-40.
- [22] Binfire, „Hybrid project management manifesto“, dostupno na: <https://www.binfire.com/hybrid-project-management-manifesto/> (20. srpnja 2022.)
- [23] Klünder, J., Karajic, D., Tell, P., Karras, O., Münkel, J., Münch, J., MacDonnel, S.G., Hebig, R., Kuhrmann, M., „Determining Context Factors for Hybrid Development Methods with Trained Models“, International Conference on Software and Systems Process (ICSSP), Seoul, Južna Koreja, 2020., str. 61-70.
- [24] Kuhrmann, M., Nakatumba-Nabende, J., Pfeiffer, R.H., Tell, P., Klünder, J., Conte, T., MacDonnel, S.G., Hebig, R., „HELENA-Edu Results“, dostupno na [https://www.researchgate.net/publication/330349801\\_Helena-Edu\\_Results](https://www.researchgate.net/publication/330349801_Helena-Edu_Results) (06. kolovoza 2022.)
- [25] Schwaber, K., Sutherland, J., „The Scrum Guide“, dostupno na: <https://scrumguides.org> (27. lipnja 2022.)
- [26] Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S.G., Münch, J., Pfahl, D., Linssen, O., Kuhrmann, M., „Towards the statistical construction of hybrid development methods“, Journal of Software: Evolution and Process, Vol. 33, No. 1, siječanj 2021., str. 1-16.

## Popis tablica

Tablica 1. Usporedba vodopadnog modela i Scruma [8] .....	12
Tablica 2. Korištene metodologije upravljanja projektima i prakse s obzirom na veličinu tvrtke [10].....	15
Tablica 3. Učestalost kombiniranja tradicionalnih i agilnih metoda [15] .....	23
Tablica 4. Usporedba korištenih praksi na uzorku od 171 agilno i 458 hibridno vođenih projekata [15].....	24
Tablica 5. Prosječne vrijednosti veličine i trajanja projekta te stope promjena zahtjeva u tijeku projekta po Likertovoj skali ( 1 – malo; 5 – puno) [20] .....	33
Tablica 6. Prosječne vrijednosti uspješnosti po pitanju kvalitete, troškova i zadovoljstva sponzora projekta po Likertovoj skali (1 – malo; 5 – puno) [20] .....	33
Tablica 7. Karakteristike projekata koji su bili predmet istraživanja [21] .....	37
Tablica 8. Presjek rezultata kvalitativnog istraživanja .....	70
Tablica 9. Smjernice za kombiniranje vodopadnog modela i agilnih metodologija upravljanja projektima razvoja web-aplikacija .....	85
Tablica 10. Preporučene prakse razvoja softvera u hibridnim modelima .....	87

## Popis slika

Slika 1. Dijagram vodopadnog modela po fazama razvoja softvera [4] .....	4
Slika 2. Najkorištenije agilne metodologije 2017. [6].....	7
Slika 3. Životni ciklus Scruma [7].....	8
Slika 4. Sektori industrije za koje rade ispitanici [10].....	15
Slika 5. Temeljne metodologije upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na cijelom uzorku ispitanika [13].....	17
Slika 6. Temeljne metodologije upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na uzorku ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima [13] .....	18
Slika 7. Prikaz temeljnih metodologija upravljanja projektima iz kojih se formiraju hibridne metodologije upravljanja projektima – temeljeno na uzorku ispitanika koji su se izjasnili da koriste hibridne metodologije upravljanja projektima [13] .....	21
Slika 8. Najpopularniji pristupi i prakse u razvoju projekata [14] .....	22
Slika 9. Water-Scrum-Fall [16].....	25
Slika 10. Zastupljenost tradicionalnog i agilnog pristupa projektnim aktivnostima po Likertovoj skali (1 – tradicionalni pristup; 3 – kombinacija tradicionalnog i agilnog pristupa; 5 – agilni pristup) [14] .....	26
Slika 11. Pregled tradicionalnog i agilnog pristupa u različitim aktivnostima razvoja softvera po Likertovoj skali ( 1 – tradicionalni pristup; 3 – kombinacija tradicionalnog i agilnog pristupa; 5 – agilni pristup) [10].....	27
Slika 12. Hibridni model „V“ [17] .....	29
Slika 13. Tromjesečni projekt vođen po vodopadnom modelu [17] .....	30
Slika 14. Tromjesečni projekt vođen po V modelu [17] .....	30
Slika 15. Hibridni model po fazama projekta [20].....	32
Slika 16. Komunikacijska sprega [21].....	36
Slika 17. Tijek provedenog istraživanja .....	41
Slika 18. Dob ispitanika (u godinama) .....	72
Slika 19. Radno iskustvo na projektima razvoja web-aplikacija (u godinama).....	73



Slika 20. Trajanje posljednjeg projekta razvoja web-aplikacije na kojem je ispitanik sudjelovao .....	74
Slika 21. Projektna uloga na posljednjem projektu razvoja web-aplikacije .....	74
Slika 22. Broj članova projektnog tima .....	75
Slika 23. Najveći izazovi na projektu razvoja web-aplikacije .....	76
Slika 24. Metodologija upravljanja projektom po kojem je općenito vođen projekt razvoja web-aplikacije .....	76
Slika 25. Metodologija upravljanja projektom u projektnoj fazi Planiranja i analize korisničkih zahtjeva .....	77
Slika 26. Metodologija upravljanja projektom u projektnoj fazi Razvoja .....	77
Slika 27. Korištene prakse razvoja softvera .....	78
Slika 28. Projektna faze koje su se preklapale s drugim projektnim fazama .....	79
Slika 29. Uključivanje korisnika u testiranje na projektima razvoja web-aplikacija .....	80
Slika 30. Kašnjenje projekata razvoja web-aplikacija .....	81
Slika 31. Važnost provođenja planiranja, analize korisničkih zahtjeva i definiranja arhitekture sustava maksimalno moguće detaljno na početku projekta po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno) .....	81
Slika 32. Stav prema tvrdnji da je u fazi Razvoja/programiranja bolje koristiti Agilne metodologije upravljanja projektima .....	82
Slika 33. Važnost isporučivanja više manjih razvojnih verzija korisniku na testiranje po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno) .....	82
Slika 34. Važnost dobro dokumentiranih zahtjeva i arhitekture sustava na početku projekta po Likertovoj skali (1 – Uopće nije važno; 5 – Iznimno važno) .....	83

## Životopis

Svea Urlić (rođ. Vodanović) rođena je 23. veljače 1990. u Zagrebu. Završava opću VII. Gimnaziju u Zagrebu te školovanje nastavlja na Ekonomskom fakultetu Sveučilišta u Zagrebu, gdje 2013. godine završava diplomski studij, smjer Financije. 2014. godine počinje raditi kao Aplikacijski konzultant za Microsoft Dynamics ERP softver AX (Axapta) u poduzeću Adacta d.o.o. (koje danas posluje pod imenom BE-terna d.o.o.), a svoj poslovni put nastavlja u tvrtki eKupi d.o.o. u području IT razvoja, gdje je trenutno na poziciji Voditelja sektora za razvoj poslovnih sustava.

U trenutnom portfelju ističe se rad na sljedećim projektima:

- SAP Commerce (Hybris) implementacija *web* trgovine za tvrtke M SAN d.o.o., Kimtec, Pakom i AskTec za tržišta Hrvatske, Bosne i Hercegovine, Republike Srbije, Crne Gore, Sjeverne Makedonije i Kosova)
- SAP Commerce (Hybris) implementacija *web* trgovine za ekupi.hr, ekupi.ba, ekupi.rs, ekupi.me i ekupi.mk, a što obuhvaća i:
  - integraciju s *Internet Payment Gatewayem*, ERP-om i sustavom za skladišno poslovanje
  - *Custom* rješenje za prodaju školskih udžbenika
  - *Custom* rješenje za maržiranje maloprodajnih cijena na SAP HANA sustavu
- Implementacija softverskog rješenja za fiskalizaciju u Crnoj Gori
- Implementacija softverskog rješenja za *pick up* lokacije za preuzimanje paketa naručenih preko *web* trgovine i rješenja za digitalno potpisivanje na tabletu prilikom preuzimanja robe na *pick up* lokacijama (Android aplikacija)
- Implementacija SAP Qualtrics platforme za napredno anketiranje i integracija navedene platforme sa CMS sustavom
- Dorada *open-source* platforme za *online* naručivanje hrane
- Integracija sa sustavom globalnog distributera hotelskog smještaja HotelBeds
- Integracija sa sustavom globalnog distributera avionskih karata Amadeus
- Implementacija Microsoft AX 2012 R2 ERP-a za tvrtku koja posluje u području gospodarenja otpadom.

Posjeduje certifikat Business Value Oriented Project Management, koji je položila 2021. godine. Također, završila je SQL tečaj te tečaj brzog čitanja.

Materinji jezik joj je hrvatski, a uz njega govori engleski, njemački i ruski jezik.

## Biography

Svea Urlić (maiden name: Vodanović) was born on February, 23th 1990 in Zagreb. Completes secondary education at VIIIth Gymnasium in Zagreb and continues her education at University of Zagreb Faculty of Economics, where she completes graduate program Finance in 2013. In 2014. starts working at Adacta d.o.o. (company currently operating under the name BE-terna d.o.o.) as Application consultant for Microsoft Dynamics ERP Software named Axapta (AX), and continues her business journey at eKupi d.o.o. in the field of IT development, where she currently holds the position of Head of Business Systems Development Sector.

In current portfolio most significant is participation on the following projects:

- Implementation of SAP Commerce (Hybris) web shop solution for M SAN d.o.o., Kimtec, Pakom and AskTec for the following markets: Croatia, Bosnia & Herzegovina, Republic of Serbia, Monte Negro, North Macedonia and Kosovo)
- Implementation of SAP Commerce (Hybris) web shop solution for ekupi.hr, ekupi.ba, ekupi.rs, ekupi.me and ekupi.mk, which includes:
  - Internet Payment Gateway, ERP and Warehouse Management System integrations
  - Custom solution for school books sales
  - Custom solution for defining margins and retail prices on SAP HANA system
- Implementation of software for fiscalization in Monte Negro
- Implementation of software solution for collection of packages ordered via web shop on pick up locations and software solution for digital signing on a tablet when picking up packages at pick up locations (Android application)
- Implementation of SAP Qualtrics platform for advanced surveying and integration of the mentioned platform with Content Management System
- Custom development on open-source platform for online food ordering
- Integration with the system of global distributor of hotel accommodation - HotelBeds
- Integration with the system of global air ticket distributor – Amadeus
- Implementation of Microsoft AX 2012 R2 ERP for a company operating in the field of waste management.

In 2021. Svea Urlić got a certificate in Business Value Oriented Project Management. She completed SQL and Speed-reading courses.

Her native language is Croatian and she also speaks English, German and Russian.

# Anketni upitnik o primjeni hibridnih metodologija upravljanja projektima u razvoju web-aplikacija

Ovaj anketni upitnik provodi se za potrebe specijalističkog rada "Primjena hibridne metodologije upravljanja projektima u razvoju programskih sustava" na poslijediplomskom specijalističkom studiju „Upravljanje projektima“ na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, pod mentorstvom prof. dr. sc. Željke Car.

Cilj anketnog upitnika je istražiti stav članova projektnih razvojnih timova vezano za najčešće probleme s kojima se susreću na projektima razvoja web-aplikacija te otkriti koriste li se na projektima hibridne metodologije upravljanja projektima koje kombiniraju tradicionalne i agilne metodologije upravljanja projektima s naglaskom na vodopadni model i SCRUM. Također, fokus je na utvrđivanju koje su aktivnosti u primijenjenim hibridnim modelima preuzete iz tradicionalnih metodologija upravljanja projektima, a koje iz agilnih te koliko se njihova kombinacija pokazala uspješnom.

Anketni je upitnik anonimn, a predviđeno vrijeme ispunjavanja je 5 minuta. Anketni upitnik možete ispuniti jednom. Rezultati upitnika bit će dostupni isključivo istraživaču te će biti obrađeni zbirno i koristit će se isključivo u svrhu istraživanja.

---

**\*Obavezno**

1. 1) Dob (u godinama): \*

*Označite samo jedan oval.*

< 25

25 - 29

30 - 39

40 - 49

> 50

## 2. 2) Koliko dugo radite na projektima razvoja softvera? \*

Označite samo jedan oval.

- < 1 godinu
- 1 - 5 godina
- 5 - 10 godina
- > 10 godina

## 3. 3) Koliko je trajao posljednji projekt razvoja web-aplikacije na kojem ste sudjelovali? \*

Označite samo jedan oval.

- < 6 mjeseci
- 6 mjeseci - 1 godinu
- 1 godinu – 1.5 godinu
- 1.5 godinu - 2 godine
- > 2 godine

4. 4) Koja je bila Vaša uloga na posljednjem projektu razvoja web-aplikacije (u daljnjim pitanjima ovaj projekt će se nazivati „predmetni projekt“)? Moguć je jedan odabir: Ako ste na projektu imali više uloga, označite onu koja je bila vodeća. \*

Označite samo jedan oval.

- Voditelj projekta
- Poslovni analitičar
- Konzultant
- Programer
- Arhitekt rješenja
- Specijalist za testiranje softvera (tester)
- SCRUM Master
- Product owner
- Ostalo: \_\_\_\_\_

## 5. 5) Koliko je članova imao projektni tim? \*

Označite samo jedan oval.

- < 5
- 5 - 10
- 10 - 20
- > 20

6. 6) Što je predstavljalo najveći izazov na tom projektu? Moguć je višestruki odabir. \*

Odaberite sve točne odgovore.

- Nejasan opseg projekta, odnosno nejasni korisnički zahtjevi na početku projekta
- Promjene korisničkih zahtjeva u toku projekta
- Komunikacija s korisnikom
- Komunikacija unutar tima
- Pogrešne procjene trajanja razvojnih aktivnosti
- Definirani prekratki rokovi
- Prekasno uključivanje korisnika u testiranje
- Ostalo: \_\_\_\_\_

## 7. 7) Po kojoj je metodologiji upravljanja projektom općenito vođen predmetni projekt? \*

Označite samo jedan oval.

- Vodopadni model
- SCRUM
- Kanban
- Lean
- Iterativni razvoj
- eXtreme Programming (XP)
- Feature Driven Development (FDD)
- DevOps
- Ostalo: \_\_\_\_\_

8. 8) Po kojoj je metodologiji upravljanja projektom vođen predmetni projekt u **projektnoj fazi Planiranja i analize korisničkih zahtjeva?** \*

*Označite samo jedan oval.*

- Tradicionalnoj
- Agilnoj
- Ostalo: \_\_\_\_\_

9. 9) Po kojoj je metodologiji upravljanja projektom vođen predmetni projekt u **projektnoj fazi Razvoja (odnosi se na programiranje)?** \*

*Označite samo jedan oval.*

- Vodopadni model
- SCRUM
- Kanban
- Lean
- Iterativni razvoj
- eXtreme Programming (XP)
- Feature Driven Development (FDD)
- DevOps
- Ostalo: \_\_\_\_\_

10. 10) Odaberite prakse razvoja softvera korištene na predmetnom projektu: \*  
Moguć je višestruki odabir.

*Odaberite sve točne odgovore.*

- Formalne specifikacije (eng. Formal Specifications)
- Formalne procjene (eng. Formal Estimation)
- Prototipiranje (eng. Prototyping)
- Automatizirano testiranje (eng. Automated Unit Testing)
- Jedinično testiranje (eng. Unit Testing)
- Postavljanje standarda u pisanju koda (eng. Coding Standards)
- Planiranje isporuka odnosno planiranje puštanja rješenja u produkciju (eng. Release Planning)
- Pregled koda (eng. Code review)
- Kontinuirana integracija (eng. Continuous integration)
- Refaktoriranje koda (eng. Refactoring)
- Retrospektive (eng. Retrospectives)
- Dnevni sastanci (eng. Daily Standup)
- Ništa od navedenog

11. 11) Odaberite projektne faze koje su se preklapale: Moguć je višestruki odabir. \*

*Odaberite sve točne odgovore.*

- Definicija i analiza zahtjeva
- Oblikovanje arhitekture
- Razvoj/programiranje
- Testiranje
- Isporuka
- Nije bilo preklapanja projektnih faza

12. 12) Koje su se projektne faze iz prethodnog pitanja najznačajnije preklapale? Ukoliko je odgovor bio "Nije bilo preklapanja projektnih faza" možete preskočiti ovo pitanje.

---

---

---

---

---



13. 13) U kojoj su fazi projekta korisnici uključeni u testiranje? \*

*Označite samo jedan oval.*

- U tijeku razvoja
- Po završetku razvoja
- Nisu bili uključeni

14. 14) Je li projekt kasnio? \*

*Označite samo jedan oval.*

- Ne
- Da, do 15% ukupnog trajanja
- Da, između 15 i 30% ukupnog trajanja
- Da, između 30 i 50% ukupnog trajanja
- Da, preko 50% ukupnog trajanja

15. 15) Procijenite koliko je važno da se planiranje, analiza korisničkih zahtjeva i definiranje arhitekture sustava provedu maksimalno moguće detaljno na početku projekta, a tek onda krene s ostalim razvojnim fazama. \*

*Označite samo jedan oval.*

1      2      3      4      5

---

Uopće nije važno      Iznimno je važno

---

16. 16) U fazi Razvoja (odnosi se na programiranje) bolje je koristiti Agilne metodologije upravljanja projektima. \*

*Označite samo jedan oval.*

- Uopće se ne slažem  
 Ne slažem se  
 Ne mogu procijeniti  
 Slažem se  
 Potpuno se slažem

17. 17) Procijenite važnost isporučivanja više manjih razvojnih verzija korisniku na testiranje. \*

*Označite samo jedan oval.*

	1	2	3	4	5	
Uopće nije važno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Iznimno je važno

18. 18) Procijenite važnost dobro dokumentiranih zahtjeva i arhitekture sustava na početku projekta (u smislu smatrate li da olakšavaju razvoj i održavanje softvera kasnije). \*

*Označite samo jedan oval.*

	1	2	3	4	5	
Uopće nije važno	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Iznimno je važno

---

Google nije izradio niti podržava ovaj sadržaj.

Google Obrasci