

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Bojan Vončina

**Analiza sprejemljivosti metodologije
Scrum**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Viljan Mahnič
SOMENTOR: doc. dr. Tomaž Hovelja

Ljubljana, 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Analiza sprejemljivosti metodologije Scrum

Tematika naloge:

Uvedba metodologije Scrum predstavlja tehnološko inovacijo, ki lahko bistveno spremeni način dela razvijalcev programske opreme. Zato je smiselno analizirati faktorje, ki vplivajo na uspešnost uvedbe. V svoji nalogi proučite modele, ki obravnavajo omenjeno problematiko (npr. Cooper-Zmudov model in Rogersov model difuzije inovacij), ter na tej podlagi analizirajte faktorje, ki so vplivali na uspešnost uvedbe metodologije Scrum v eni izmed večjih slovenskih organizacij za razvoj programske opreme. Dobljene rezultate primerjajte z rezultati podobne študije, ki sta jo v zvezi z uvedbo metodologije Scrum v eni izmed nemških zavarovalnic opravila Overhage in Schlauderer.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Bojan Vončina, z vpisno številko **63100322**, sem avtor diplomskega dela z naslovom:

Analiza sprejemljivosti metodologije Scrum

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Viljana Mahniča in somentorstvom doc. dr. Tomaža Hovelje,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 25. januarja 2016

Podpis avtorja:

Rad bi se zahvalil izr. prof. dr. Viljanu Mahničju in doc. dr. Tomažu Hovelji za nasvete, napotke, pomoč in potrpežljivost pri izdelavi diplomskega dela. Prav tako bi se rad zahvalil zaposlenim v podjetju Comtrade, ki so si vzeli čas za izpolnjevanje vprašalnika, ter celotni družini in prijateljem za podporo, ki so mi jo nudili tekom celotnega študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Metodologija Scrum	3
2.1	Vloge	4
2.1.1	Produktni vodja	4
2.1.2	Skrbnik metodologije	5
2.1.3	Razvojna skupina	5
2.2	Razvojni proces metodologije Scrum	6
2.2.1	Iteracija	6
2.2.2	Sestanki	7
2.2.2.1	Sestanek za načrtovanje iteracije	7
2.2.2.2	Dnevni sestanek	7
2.2.2.3	Sestanek za pregled rezultatov iteracije	8
2.2.2.4	Retrospektivni sestanek	8
2.3	Izdelki metodologije Scrum	8
2.3.1	Seznam zahtev	9
2.3.2	Seznam nalog	9
2.3.3	Inkrement produkta	9
2.3.4	Diagram preostalega dela	9
2.4	Uporabniške zgodbe	11
2.4.1	Metoda Planning Poker	12
2.4.2	Metoda Team Estimation Game	12

KAZALO

2.5	Prednosti in slabosti	13
3	Modeli za analizo uvajanja tehnoloških inovacij	15
3.1	Cooper - Zmudov model	15
3.2	Rogersov model	16
3.3	Opis vprašalnika	17
4	Analiza uporabe metodologije Scrum v podjetju Comtrade	19
4.1	Predstavitev podjetja	19
4.2	Zbiranje podatkov	20
4.3	Normalnost porazdelitve pridobljenih podatkov	20
4.4	Zanesljivost pridobljenih podatkov	28
4.5	Sprejemljivost ključnih aktivnosti metodologije Scrum	29
4.6	Ključni faktorji uporabe metodologije Scrum	43
4.6.1	Vpliv faktorjev o inovaciji	43
4.6.2	Vpliv faktorjev o nalogah	46
4.6.3	Vpliv faktorjev o posamezniku	48
4.6.4	Vpliv faktorjev o okolju	50
4.6.5	Vpliv faktorjev o organizaciji	52
5	Primerjava z rezultati raziskave Overhageja in Schlaudererja	57
5.1	Predstavitev raziskave Overhageja in Schlaudererja	57
5.2	Preverjanje hipotez	60
6	Zaključek	63
	Literatura	64
A	Vprašalnik	67

Povzetek

Namen diplomske naloge je bil analizirati sprejetost metodologije Scrum, ki je ena izmed vodilnih agilnih metodologij, ter ugotoviti kateri so bili ključni faktorji, ki so vplivali na njeno sprejetje. Analiza je bila opravljena v podjetju Comtrade, ki je eno izmed največjih slovenskih podjetij na področju razvoja programske opreme.

Prvi del (teoretični del) vsebuje uvodno poglavje, podrobno predstavitev metodologije Scrum in predstavitev teoretičnih modelov, na katerih temelji praktični (raziskovalni) del diplomske naloge.

Drugi del (praktični del) tvorita dve poglavji. V prvem je predstavljena analiza uporabe metodologije Scrum v podjetju Comtrade, kjer glavni del tvorita analiza sprejetosti ključnih aktivnosti (elementov) metodologije Scrum in analiza ključnih faktorjev. Drugo poglavje pa je namenjeno predstavitvi primerjave naše raziskave z raziskavo dveh nemških strokovnjakov s področja razvoja programske opreme.

Ključne besede: metodologija Scrum, sprejetost ključnih aktivnosti, ključni faktorji.

Abstract

The purpose of the thesis was to analyse the acceptance of Scrum methodology, which has become one of the leading agile methodologies, and to find out which were the key factors that influenced the acceptance. The analysis was conducted in Comtrade, which is one of the largest Slovenian software development companies.

The First part (theoretical part) contains an introduction chapter, a detailed presentation of Scrum methodology and the presentation of theoretical models, on which practical (research) part of thesis is based.

The Second part (practical part) consists of two chapters. In the first one the analysis of Scrum methodology usage in Comtrade is presented, where the analysis of acceptance of key activities and the analysis of key factors form the main part. The second chapter is devoted to the presentation of comparison between our research and research done by two German software development experts.

Keywords: Scrum methodology, acceptance of key activities, key factors.

Poglavje 1

Uvod

Dandanes se pri razvoju programske opreme čedalje bolj uporablja agilne metodologije. Te so se pojavile sredi 90-ih let kot odgovor na vse večje kritike težkih (angl. *Heavyweight*) metodologij. Sprva so bile poimenovane kot lahke (angl. *Lightweight*) metodologije, leta 2001 pa je 17 izkušenih inženirjev programske opreme objavilo manifest za agilni razvoj programske opreme (angl. *Manifesto for Agile Software Development*), kjer se prvič pojavi poimenovanje agilne metodologije. V omenjenem manifestu so zapisali štiri vrednote in dvanajst principov, ki jim mora slediti vsaka agilna metodologija. Vrednote so sledeče [8]:

1. Posamezniki in komunikacija so bolj pomembni kot procesi in orodja
2. Delujoča programska oprema je bolj pomembna kot preveč obširna dokumentacija
3. Sodelovanje z naročnikom je bolj pomembno kot pogajanja o pogodbi
4. Odzivanje na spremembe je bolj pomembno kot sledenje načrtu

Ena izmed agilnih metodologij je tudi metodologija Scrum. Njena sprejemljivost in uporaba v praksi je tema te diplomske naloge. In sicer nas je zanimalo koliko slovenska podjetja sledijo svetovnim trendom razvoja programske opreme. Raziskavo smo opravili v podjetju Comtrade, ki je eno izmed največjih slovenskih podjetij na področju razvoja programske opreme. Pri tem nas je zanimalo kako dobro so posamezni elementi (aktivnosti) metodologije Scrum sprejeti in uporabljeni, ter

kateri so tisti (ključni) faktorji, ki vplivajo na sprejetje. Poleg tega smo opravili tudi primerjavo naše raziskave z raziskavo dveh nemških strokovnjakov (Sven Overhage in Sebastian Schlauderer), saj nas je zanimalo ali lahko najdemo kakšne skupne značilnosti, ki se pojavljajo pri uporabi metodologije Scrum. Podatke za raziskavo smo pridobili s pomočjo anketnega vprašalnika, ter jih analizirali z uporabo frekvenčnih tabel in izračuni korelacij. Diplomsko naloga se začne s podrobno predstavitevijo metodologije Scrum, ki je razdeljena na pet podpoglavij. V prvem sledi seznanitev z vlogami, ki jih pozna metodologija Scrum in podroben opis vsake izmed njih. Nato sledi predstavitev razvojnega procesa metodologije Scrum (predstavitev iteracije in sestankov). Tretje podpoglavje vsebuje predstavitev izdelkov metodologije Scrum, ki nastajajo tekom celotnega procesa. Predstavitev uporabniških zgodb in dveh metod za ocenjevanje uporabniških zgodb sledi v četrtem podpoglavju. Na koncu so predstavljene še prednosti in slabosti uporabe metodologije Scrum. Sledi poglavje, kjer sta predstavljena dva teoretična modela, ki sta osnova naše raziskave, in anketni vprašalnik. Četrto poglavje je osrednji del diplomske naloge, saj je v njemu predstavljena analiza uporabe metodologije Scrum v podjetju Comtrade. Začne se s predstavitevijo podjetja in podpoglavjem, kjer je opisan postopek zbiranja podatkov. Od tod naprej pa se začne analiza pridobljenih podatkov, ki obsega štiri podpoglavja, in sicer: normalnost porazdelitve pridobljenih podatkov, zanesljivost pridobljenih podatkov, sprejemljivost ključnih aktivnosti metodologije Scrum in ključni faktorji uporabe metodologije Scrum. Jedro diplomske naloge se zaključi s petim poglavjem, kjer je najprej predstavljena raziskava prej omenjenih nemških strokovnjakov, nato pa primerjava z našo raziskavo.

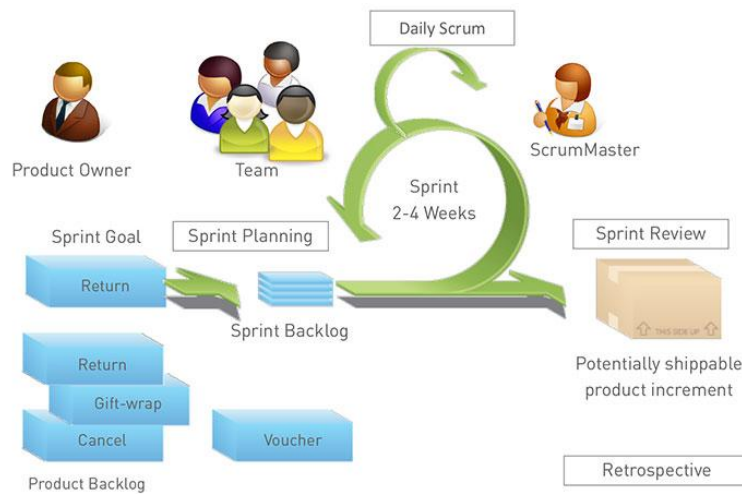
Poglavje 2

Metodologija Scrum

V začetku 90-ih Jeff Sutherland in Ken Schwaber naredita zasnovo celotnega procesa metodologije Scrum. Leta 1995 pa jo podrobneje definirata za potrebe konference OOPSLA (*Object Oriented Programming, Systems, Languages & Applications*) v Austinu (Teksas, ZDA), kjer je tudi prvič javno predstavljena. Temelji na treh enostavnih načelih [9]:

1. Vse je vidno vsem
2. Redno pregledovanje rezultatov
3. Prilagajanje procesa potrebam v danem trenutku

Danes jo označujejo kot eno izmed vodilnih agilnih metodologij, saj jo uporablja veliko število podjetij. Eden izmed razlogov je prav gotovo enostavnost, vendar pa je lahko ta enostavnost včasih tudi zavajajoča. Metodologija Scrum, tako kot ostale agilne metodologije, temelji na iterativnem, inkrementalnem procesnem ogrodju. Razvojni proces poteka v iteracijah, ki si sledijo ena za drugo. Običajno so dolge do 30 dni. Tekom posamezne iteracije razvojna skupina razvije nov inkrement produkta (prejšnjemu inkrementu se dodajo nove funkcionalnosti). Pomemben del metodologije Scrum pa so tudi sestanki, ki se dogajajo pred, med in po iteraciji. Na Sliki 2.1 je prikazan celotni proces, ki se ponavlja, dokler je projekt finančno podprt, oziroma je naročnik še zainteresiran za njegovo izvajanje.



Slika 2.1: Prikaz procesa Scrum [12]

2.1 Vloge

Pri metodologiji Scrum ločimo tri vloge: produktni vodja oziroma lastnik izdelka (angl. *Product Owner*), skrbnik metodologije (angl. *Scrum Master*) in razvojna skupina (angl. *Team*). Vse naloge in odgovornost na projektu so tako razdeljene med te tri vloge [1]. S projektom so lahko povezani tudi ostali zainteresirani (opazovalci), vendar pa se ne smejo vanj neposredno vmešavati in zanj niso odgovorni. Iz teh razlogov pri metodologiji Scrum poznamo dve skupini, in sicer prašiče in piščance. Poimenovanji izhajata iz šale, kjer piščanec in prašič želita odpreti restavracijo. Prašič se na začetku strinja, vendar ko piščanec predlaga ime za restavracijo ("Šunka in jajca") se prašič ne strinja več, saj bi bil s tem v celoti predan projektu, medtem ko bi bil piščanec le udeležen.

2.1.1 Produktni vodja

Produktni vodja oziroma lastnik izdelka predstavlja vse, ki so zainteresirani za projekt in njegove rezultate [2]. Glavna naloga, ki jo ima, je skrb za seznam zahtev (angl. *Product Backlog*). Naloga vključuje določanje funkcionalnosti v obliki uporabniških zgodb (angl. *User Story*), dodajanje teh v seznam zahtev in določanje prioritete funkcionalnostim oziroma uporabniškim zgodbam. Prav tako je odgovo-

ren, da se najpomembnejše funkcionalnosti (najpomembnejše z vidika naročnikov) realizirajo najprej, saj s tem maksimizira dodano vrednost (angl. *ROI - Return on Investment*) projekta. Naloga produktnega vodje je tudi odločanje, katere uporabniške zgodbe so uspešno realizirane in pripravljene za izdajo (angl. *Release*) in katere je potrebno še popraviti. Odločitve sprejema na podlagi rezultatov t.i. sprejemnih testov (angl. *Acceptance Test*).

2.1.2 Skrbnik metodologije

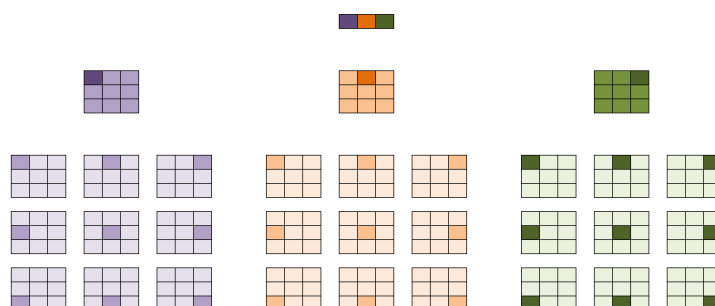
Skrbnik metodologije je odgovoren, da proces Scrum poteka pravilno. Vse sodelujoče na projektu uči in skrbi, da vsi upoštevajo pravila in določila, ki jih predpisuje metodologija. Razvojno skupino ščiti pred motečimi dejavniki in odstranjuje možne ovire, ki se lahko pojavijo med samim razvojnim procesom. S tem omogoča, da se lahko razvojna skupina v celoti posveti svojemu delu (kodiranju, testiranju itd.) in doseže zastavljene cilje.

2.1.3 Razvojna skupina

Razvojna skupina je tista, ki je v celoti zadolžena za implementacijo funkcionalnosti. Na kakšen način bodo člani razvojne skupine spremenili zahteve s seznama zahtev v končne funkcionalnosti znotraj vsake iteracije, je popolnoma prepuščeno njim. Zato so vsi člani razvojne skupine odgovorni za uspeh vsake iteracije in celotnega projekta. Scrum ne priporoča uporabe nazivov, kot so programer, tester, itd. pač pa so vsi člani razvojne skupine enakopravni in so "označeni" kot člani skupine (angl. *Team Members*). Tako naj bi vsak član razvojne skupine opravljal vsa dela (programiranje, testiranje, pisanje dokumentacije itd.).

Glede optimalne velikosti razvojne skupine je pri metodologiji Scrum tudi precej govora. Optimalno velika skupina naj bi bila taka, ki je dovolj majhna, da ostane "okretna" in dovolj velika, da opravi vso pomembno delo posamezne iteracije [10]. Optimalno število članov razvojne skupine naj bi se tako gibalo med tremi in devetimi člani. V praksi se je namreč izkazalo, da v kolikor je razvojna skupina manjša oziroma večja od omenjenega intervala, prihaja do premajhne interakcije med člani oziroma je potrebno preveč koordinacije. V obeh primerih je posledica manjša produktivnost.

Vendar pa to ne pomeni, da v podjetjih z večjim številom razvijalcev, ni možno razvijati produkta po metodologiji Scrum. V takih primerih se uporabi pristop Scrum of Scrums. Razvijalci so ločeni v več razvojnih skupin, velikosti, ki jo priporoča metodologija Scrum. Vsaka izmed skupin izbere enega člana, ki se udeležuje Scruma na višjem nivoju. Primer na Sliki 2.2 prikazuje uporabo pristopa Scrum of Scrums, kjer vsaka celica predstavlja enega razvijalca.



Slika 2.2: Prikaz pristopa Scrum of Scrums z 243 razvijalci [2]

2.2 Razvojni proces metodologije Scrum

2.2.1 Iteracija

Iteracija (angl. *Sprint*) je glavni element v razvojnem procesu metodologije Scrum [11], ki običajno traja od 2 do 4 tedne. V tem času člani razvojne skupine razvijejo nov inkrement produkta. Poleg razvoja imajo tekom iteracije tudi administrativni nalogi, in sicer udeleževanje dnevnih sestankov (angl. *Daily Scrum Meeting*) in skrb za seznam nalog (angl. *Sprint Backlog*). Vsaka interacija se konča s sestankom za pregled rezultatov (angl. *Sprint Review Meeting*) in retrospektivnim sestankom (angl. *Retrospective Meeting*). Omenjeni sestanki so v nadaljevanju tudi podrobneje opisani.

V kolikor se izkaže, da cilji iteracije niso uresničljivi oziroma izvedljivi (npr. realizacija ni izvedljiva zaradi omejitev tehnologije oziroma tehnoloških omejitev), jo lahko produktni vodja prekine. Temu sledi takojšnje načrtovanje nove iteracije.

2.2.2 Sestanki

2.2.2.1 Sestanek za načrtovanje iteracije

Pred začetkom vsake iteracije se zgodi sestanek za načrtovanje iteracije, na katerem so prisotni produktni vodja, skrbnik metodologije in razvojna skupina. Sestanek običajno traja 8 ur in je razdeljen na dva dela (vsak del traja 4 ure).

V prvem delu produktni vodja predstavi uporabniške zgodbe članom razvojne skupine. Pri tem upošteva prioriteto uporabniških zgodb in se osredotoči na tiste z najvišjo prioriteto (pomembnejše uporabniške zgodbe). Ostale uporabniške zgodbe pa pusti za naslednje iteracije. Nato sledijo vprašanja članov razvojne skupine, v kolikor zahteve posameznih zgodb niso popolnoma jasne. Zahteve posameznih uporabniških zgodb morajo namreč biti popolnoma jasne vsem članom razvojne skupine še pred začetkom implementacije. Sledi ocenitev uporabniških zgodb, ki jo opravi razvojna skupina. Ob koncu prvega dela sestanka, razvojna skupina določi, katere uporabniške zgodbe bo v iteraciji realizirala.

V drugem delu člani razvojne skupine načrtujejo iteracijo. Uporabniške zgodbe, ki so jih izbrali v prvem delu sestanka, razbijejo na naloge (angl. *Tasks*). Prav tako si lahko že na tem sestanku okvirno razdelijo delo in ocenijo zahtevnost (čas trajanja) posameznih nalog. Rezultat drugega dela sestanka je tako seznam nalog.

2.2.2.2 Dnevni sestanek

Dnevni sestanek je sestanek, ki se zgodi vsak delovni dan (najbolje pred začetkom dela) in traja 15 minut. Na njem mora biti prisotna celotna razvojna skupina. V kolikor se kdo izmed članov ne more udeležiti sestanka, mora nekdo izmed prisotnih članov poročati namesto njega. Priporočljivo je, da so sestanki vsak dan na istem mestu ob isti uri. Na sestanku vsak član razvojne skupine odgovori na tri vprašanja:

1. Kaj si naredil od prejšnjega sestanka ?
2. Kaj boš naredil do naslednjega sestanka ?
3. Kakšne težave so se pojavile pri delu ?

Sestanek je namenjen pregledu sprotnega dela vseh članov razvojne skupine in usklajevanju le-tega. Člani razvojne skupine si lahko v primeru nejasnosti med seboj postavljajo kratka vprašanja in nanje odgovarjajo s kratkimi odgovori, vendar pa se ne spuščajo preveč v podrobnosti. V kolikor so stvari še nerazjasnjene, se lahko dogovorijo za dodaten sestanek.

2.2.2.3 Sestanek za pregled rezultatov iteracije

Po končani iteraciji sledi štiriurni sestanek za pregled rezultatov iteracije. Na začetku sestanka razvojna skupina predstavi produktnemu vodji in ostalim zainteresiranim, kakšni so bili cilji iteracije, ter kako uspešni so bili pri realizaciji le-teh. Osrednji del sestanka je predstavitev implementiranih funkcionalnosti in morebitna vprašanja produktnega vodje ter ostalih. Produktni vodja nato oceni, ali so predstavljene funkcionalnosti uspešno realizirane ali pa jih je potrebno še popraviti (postaviti v naslednjo iteracijo). Poleg omenjenega, je namen sestanka tudi okvirno določiti, kaj naj bi razvojna skupina naredila v naslednji iteraciji.

2.2.2.4 Retrospektivni sestanek

Zadnji sestanek pred začetkom načrtovanja nove iteracije je triurni retrospektivni sestanek za ocenitev celotnega razvojnega procesa zadnje iteracije. Na sestanku so prisotni člani razvojne skupine in skrbnik metodologije, ki ga tudi vodi. Vsak član razvojne skupine odgovori na naslednji vprašanji [1]:

1. Kaj je bilo dobro v iteraciji ?
2. Katere so možne izboljšave v naslednji iteraciji ?

Odgovore si zapiše skrbnik metodologije v obliki krajšega povzetka. Sestanek se zaključi z razpravo članov razvojne skupine o predlaganih izboljšavah, na podlagi katere izberejo tiste, za katere ocenijo, da so najpomembnejše. Realizacijo teh izboljšav nato preverjajo na naslednjem retrospektivnem sestanku.

2.3 Izdelki metodologije Scrum

Tekom celotnega procesa nastajajo in se uporabljajo, spreminjajo ter posodablajo t.i. izdelki metodologije Scrum (angl. *Scrum Artifacts*). V nadaljevanju so

podrobneje predstavljeni.

2.3.1 Seznam zahtev

Vse zahteve oziroma funkcionalnosti, ki jih mora vsebovati končni produkt, so podane v seznamu zahtev v obliki uporabniških zgodb. V celoti odgovoren za vsebino, prioritiziranje in zagotavljanje dostopnosti le-tega, je produktni vodja. Seznam zahtev je dinamičen. Običajno je na začetku projekta kratek in splošen, tekom projekta pa postaja daljši (se dopolnjuje) in bolj podroben. Obstaja toliko časa, kolikor traja projekt oziroma dokler obstaja produkt.

2.3.2 Seznam nalog

Seznam nalog vsebuje naloge, ki morajo biti opravljene v tekoči iteraciji. Določi jih razvojna skupina, in sicer z razbitjem izbranih uporabniških zgodb. Pri tem naj bi realizacija posamezne naloge obsegala od 4 do 16 ur dela. Naloge, ki so na začetku grobo definirane, je potrebno razbiti na manjše naloge. Tekom iteracije se lahko spreminja (posodablja) seznam nalog, vendar pa le s strani razvojne skupine. Seznam nalog je tudi pokazatelj (v realnem času), koliko dela je razvojna skupina že opravila in koliko dela še mora opraviti do zaključka tekoče iteracije.

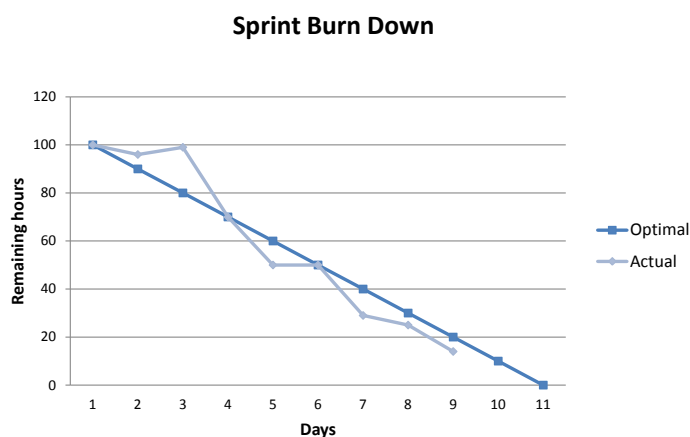
2.3.3 Inkrement produkta

Rezultat vsake iteracije je delujoča programska koda (angl. *Shippable functionality*) oziroma inkrement produkta (angl. *Product Increment*). Ta vsebuje v celoti stestirano, dobro strukturirano, dobro napisano programsko kodo ter uporabniško dokumentacijo.

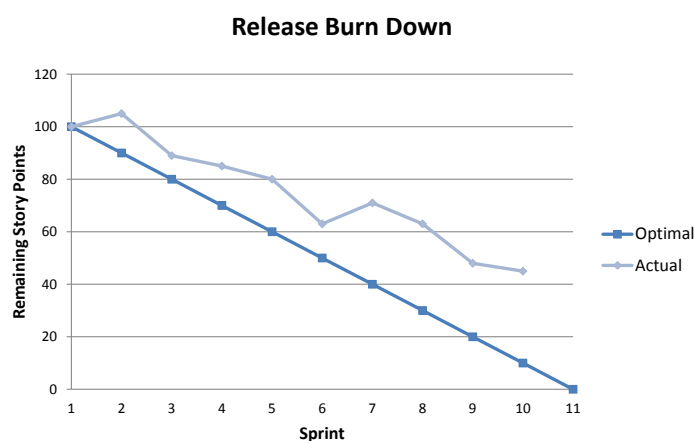
2.3.4 Diagram preostalega dela

Pomemben izdelek metodologije Scrum je tudi diagram preostalega dela (angl. *Burn Down Chart*), ki omogoča grafično spremljanje napredovanja projekta. Je dvodimenzionalen, kjer abscisna os predstavlja trajanje iteracije (v dnevih) oziroma projekta (v iteracijah), medtem ko ordinatna os predstavlja količino preostalega dela. Ta je lahko izražena v točkah, urah, dnevih ali v kakšni drugi enoti.

Diagram se običajno uporablja na nivoju iteracije in na nivoju projekta. V prvem primeru gre za t.i. *Sprint Burndown*, ki prikazuje količino preostalega dela znotraj ene iteracije. Primer diagrama *Sprint Burndown* je prikazan na Sliki 2.3. Na nivoju projekta pa se uporablja t.i. *Release Burndown*, ki služi za oceno, kdaj bo projekt zaključen. Slika 2.4 prikazuje primer diagrama *Release Burndown*.



Slika 2.3: Primer diagrama Sprint Burndown [2]



Slika 2.4: Primer diagrama Release Burndown [2]

2.4 Uporabniške zgodbe

Zahteve oziroma funkcionalnosti produkta se praviloma pri metodologiji Scrum, ter pri večini ostalih agilnih metodologijah, zapisuje v obliki uporabniških zgodb. Kot smo že omenili v podpoglavju 2.1.1, je produktni vodja tisti, ki je zadolžen, da zagotovi obstoj seznama zahtev. Vendar pa to ne pomeni, da je on vedno tudi tisti, ki je zadolžen za pisanje uporabniških zgodb. Običajno se v praksi raje oblikuje skupina naročnikov (angl. *Customer Team*), ki je zadolžena za pisanje, kot tudi določanje prioritete, uporabniških zgodb.

Poznamo dva načina pisanja uporabniških zgodb, in sicer: neformalen in formalen. Pri neformalnem načinu se uporabi vzorec ”<Vloga> lahko <opravi akcijo>”. Primer neformalnega zapisa uporabniške zgodbe bi tako lahko bil npr.: ”Uporabnik lahko ob prvi prijavi izbere možnost shranitve gesla”. Formalno zapisane uporabniške zgodbe pa imajo sledeč vzorec: ”Kot <vloga> želim imeti <zmožnost>, da lahko <korist>”. Prej omenjeni primer bi se tako v formalnem načinu lahko glasil: ”Kot uporabnik želim imeti ob prvi prijavi možnost shranitve gesla, da se lahko prijavim brez vnosa gesla ob ponovni prijavi”.

Za realizacijo vsake uporabniške zgodbe naj bi bilo potrebno od pol dneva do 2 tedna. V kolikor pa sta za realizacijo uporabniške zgodbe potrebna več kot 2 tedna, oziroma je le-ta preobsežna (obsega preveliko število funkcionalnosti), jo je potrebno razbiti na več manjših. Vsaki uporabniški zgodbi je potrebno določiti časovno zahtevnost oziroma le-to oceniti. Pri tem se uporablja t.i. točke (angl. *Story Points*). Po priporočilu eksperta za agilne metodologije, Mika Cohna, naj bi 1 točka predstavljala 1 idealen delovni dan [2].

Ocenjevanje v celoti opravi razvojna skupina. Naročnik je sicer lahko prisoten, vendar pa se ne sme vmešavati. Razvojna skupina opravi ocenjevanje z uporabo metode Planning Poker ali metode Team Estimation Game. Metodi sta v nadaljevanju podrobneje predstavljeni. Uporabniškim zgodbam se ponavadi pripiše tudi sprejemne teste. Ti odražajo pričakovanja uporabnikov oziroma naročnikov, kako naj bi končna funkcionalnost delovala [2]. Koristni pa so tudi razvojni skupini, kateri služijo kot indikator, kdaj je uporabniška zgodba končana.

2.4.1 Metoda Planning Poker

Planning Poker [3] je metoda oziroma tehnika, ki se uporablja za ocenjevanje uporabniških zgodb in temelji na soglasju članov razvojne skupine. Začne se s predstavitvijo uporabniške zgodbe, ki jo opravi produktni vodja ali pa predstavnik naročnika. Ta nato odgovarja na vprašanja članov razvojne skupine, dokler vsebina uporabniške zgodbe ni razjasnjena. Ko je vsebina uporabniške zgodbe jasna vsem članom razvojne skupine, sledi ocenjevanje. Vsak član razvojne skupine na svojo kartico napiše eno izmed že vnaprej določenih ocen. Kartice nato istočasno pokažejo. V kolikor se ocene med seboj razlikujejo, člana, ki sta dala najnižjo in najvišjo oceno, utemeljita svoji oceni. Sledi diskusija, ki lahko traja največ 5 minut. Nato se postopek ocenjevanja ponovi in se ponavlja toliko časa, dokler ocene niso poenotene.

2.4.2 Metoda Team Estimation Game

Druga metoda oziroma tehnika, ki se prav tako kot prej omenjena metoda Planning Poker uporablja za ocenjevanje uporabniških zgodb, je Team Estimation Game. Potek metode je sledeč [13]:

1. Prvi ocenjevalec oziroma prvi član razvojne skupine, ki bo podal oceno, vzame prvo kartico z uporabniško zgodbo in jo postavi na mizo ali tablo.
2. Vsak naslednji ocenjevalec oziroma član razvojne skupine ima 3 možnosti: vzame naslednjo kartico in jo postavi v isti stolpec ali levo (manj zahtevne zgodbe) ali desno (bolj zahtevne zgodbe), premakne eno izmed kartic, ki je že na mizi oziroma tabli ali pa preskoči potezo. Ko so vse kartice razporejene v stolpce in jih nihče izmed ocenjevalcev oziroma članov razvojne skupine ne premika več, je postopek končan.
3. Razvojna skupina vsakemu stolpcu določi ustrezno število točk, pri tem pa uporabi že vnaprej določene vrednosti.
4. Na koncu razvojna skupina preveri razmerja med uporabniškimi zgodbami v posameznih stolpcih (triangulacija).

Pomembno je vedeti, da se metoda osredotoča na razmerja in ne na ocene. To je tudi razlika med metodo Team Estimation Game in metodo Planning Poker.

2.5 Prednosti in slabosti

Tako kot vsaka stvar, ima tudi uporaba metodologije Scrum svoje prednosti [5]:

- Razvoj produkta poteka v zaporedju obvladljivih delov (iteracij).
- Delo poteka nemoteno, kljub spreminjajočim se zahtevam.
- Vsi na projektu so seznanjeni z vsem, kar se dogaja na projektu.
- Komunikacija in sodelovanje med člani razvojne skupine se izboljša.
- Zaslug za uspešno delo med in na koncu projekta je enakomerno deležen vsak član razvojne skupine.
- Naročnik lahko vidi sproten razvoj produkta in ob določenih rokih dobiva posamezne delujoče dele končnega produkta.
- Odnosi z naročnikom se izboljšajo.
- Ustvari se pozitivno vzdušje, kjer vsi verjamejo v končni uspeh projekta.

Kot tudi slabosti [14]:

- Če naloga ni dobro definirana, ocena časovne zahtevnosti ne bo točna. V tem primeru se lahko realizacija naloge razširi na več iteracij.
- V kolikor člani razvojne skupine niso v celoti predani projektu, bo le-ta ostal nedokončan oziroma bo propadel.
- Člani razvojne skupine morajo biti izkušeni, sicer je zelo verjetno, da projekt ne bo pravočasno končan.
- Scrum deluje dobro, kadar skrbnik metodologije zaupa razvojni skupini in njenem delu. Če tega zaupanja ni in razvojna skupina nima določene svobode pri razvoju (preveč strogo vodenje in nadzor razvojne skupine s strani skrbnika metodologije), lahko to pripelje do frustracij in padca morale razvojne skupine, ter posledično do propada projekta.
- Če kateri izmed članov razvojne skupine zapusti projekt sredi razvojnega procesa, povzroči precej negativnih posledic za celoten razvojni proces.

Seveda pa se je naštetim slabostim moč izogniti, s pravim pristopom k planiranju in nato s pametnimi odločitvami tekom nadaljevanja procesa.

Poglavje 3

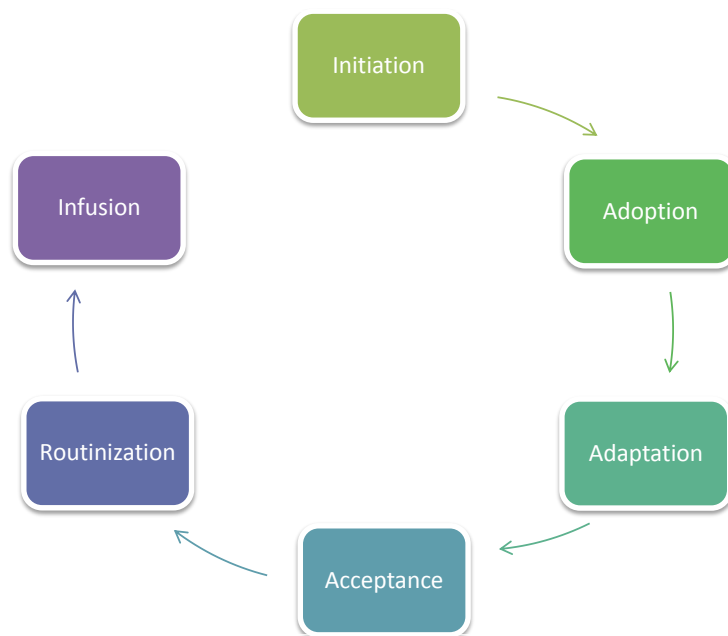
Modeli za analizo uvajanja tehnoloških inovacij

Poglavje je namenjeno predstavitvi dveh teoretičnih modelov, in sicer Cooper - Zmudovega in Rogersovega. Prvi opisuje proces, kako neka skupina sprejema in uveljavlja inovacijo, skozi neko časovno obdobje oziroma skozi stopnje. Drugi pa opisuje teorijo, ki raziskuje različne faktorje in njihov vpliv na sprejetje in uveljavitev neke inovacije. Omenjena modela sta teoretična osnova naše raziskave in sta v nadaljevanju tudi podrobneje predstavljena. Na koncu poglavja sledi še opis vpršalnika.

3.1 Cooper - Zmudov model

Leta 1990 Randolph B. Cooper in Robert W. Zmud predstavita model za sprejetje inovacij. Model je sestavljen iz šestih stopenj, kot prikazuje Slika 3.1. Pri tem vsaka izmed stopenj predstavlja nivo do katerega neka inovacija vpliva na skupino, ki sprejema inovacijo [6]. V prvi stopnji (angl. *Initiation*) skupina, ki sprejema inovacijo, primerja količino potrebnega truda za vpeljavo (sprejetje) inovacije in kolikšna je pridobitev, uporabe inovacije. Sledi druga stopnja (angl. *Adoption*), kjer skupina, ki sprejema inovacijo, na podlagi primerjave v predhodnji stopnji, poda odločitev o sprejetju inovacije. Tretja stopnja (angl. *Adaptation*) je stopnja, kjer se inovacija vpelje v obstoječ proces. V tej stopnji se sprejemajoča skupina

uči uporabljati inovacijo. V četrti stopnji (angl. *Acceptance*) skupina, ki sprejema inovacijo, prične z njeno uporabo. Sledi peta stopnja (angl. *Routinization*), kjer uporaba inovacije že postane vsakdanje opravilo sprejemajoče skupine. V zadnji (šesti) stopnji (angl. *Infusion*) skupina uporablja inovacijo na tak način, da ji ta prinaša korist in povečuje njeno učinkovitost.

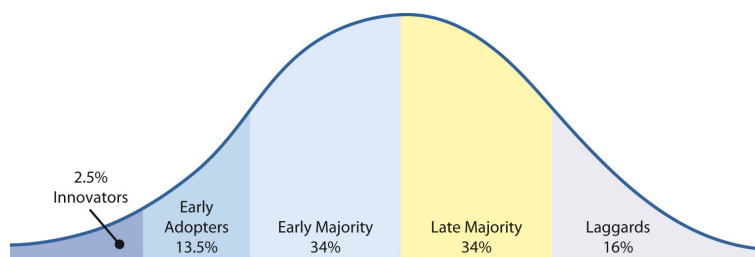


Slika 3.1: Cooper - Zmudov šeststopenjski model

3.2 Rogersov model

Everett M. Rogers leta 1962 predstavi teorijo *DOI - Diffusion Of Innovations* v kateri razlaga kako, zakaj in s kakšno hitrostjo, se nove ideje in tehnologija (inovacije) razširjajo oziroma uveljavljajo v družbenem sistemu [4]. V omenjeni teoriji razbije populacijo, glede na značilnosti posameznikov znotraj le-te (npr. glede na željo in pripravljenost sprejemanja inovacij, znanje ...), na pet segmentov oziroma kategorij. Te so: inovatorji (angl. *Innovators*), prvi uporabniki (angl. *Early Adoptors*), zgodnja večina (angl. *Early Majority*), pozna večina (angl. *Late Majority*) in zamudniki (angl. *Laggards*). Kategorije so tudi grafično predstavljene

na Sliki 3.2, iz katere lahko razberemo, da v kategoriji zgodnja in pozna večina "pade" večina populacije.



Slika 3.2: Razdelitev populacije v kategorije po teoriji DOI [15]

Poleg omenjenih kategorij, Rogers določi tudi pet sklopov dejavnikov oziroma faktorjev, ki vplivajo na sprejetje inovacije, in sicer dejavniki oziroma faktorji: o inovaciji, o posamezniku, o nalogah, o okolju in o organizaciji.

3.3 Opis vprašalnika

Podatke za analizo dolgoročne sprejemljivosti metodologije Scrum v podjetju Comtrade smo pridobili s pomočjo anketnega vprašalnika, ki je v celoti predstavljen v Dodatku A. Sestavljen je iz štirih delov. Prvi del tvorita vprašanji o dosedanjih vlogah na projektih (projekti, ki so potekali oziroma še potekajo po metodologiji Scrum) in izkušnjah z metodologijo Scrum. Drugi del sestavljajo vprašanja, ki se nanašajo na ključne aktivnosti oziroma elemente metodologije Scrum (izvajanje sestankov, uporaba uporabniških zgodb itd.) in njihovo stopnjo uporabe oziroma stopnjo sprejetja. Pri vsakem izmed vprašanj je podanih 6 odgovorov, kjer vsak odgovor predstavlja določeno stopnjo Cooper - Zmudovega modela sprejetja inovacij. In sicer prvi odgovor predstavlja prvo stopnjo Cooper - Zmudovega modela, oziroma šesti odgovor predstavlja šesto stopnjo Cooper - Zmudovega modela. Sledi tretji del, ki ga tvori pet sklopov faktorjev (3.2 Rogersov model). Pri tem je vsak sklop sestavljen iz trditve (vprašanj), kjer vsaka predstavlja posamezen faktor. Trditve so povzete po raziskavi *Why organizations adopt information system process innovations: a longitudinal study using Diffusion of Innovation theory* [7]. Zadnji (četrti) del vprašalnika pa tvorijo trditve, s katerimi smo preverjali hipoteze, ki sta

jih podala Sven Overhage in Sebastian Schlauderer v svoji raziskavi sprejemljivosti metodologije Scrum.

Poglavje 4

Analiza uporabe metodologije Scrum v podjetju Comtrade

4.1 Predstavitev podjetja

Začetki podjetja Comtrade v Sloveniji segajo v leto 1990, ko štirje partnerji ustanovijo podjetje HERMES SoftLab. Leta 1996 HERMES SoftLab odpre prvo podružnico v Sarajevu z željo prodora na tuje trge. Uspešnemu odprtju sledi odprtje nove podružnice v Kaliforniji (ZDA). Z začetkom novega tisočletja se podjetje še bolj utrdi na mednarodnem trgu, ko odpre podružnico v Stuttgartu. Leta 2003 se odprejo podružnice v Veliki Britaniji, Franciji in na Irskem, leta 2005 pa podružnica v Avstriji. Sledita še odprtji podružnic v Srbiji (2006) in v Črni gori (2007). Uspešnost in širitev podjetja sočasno rasteta tudi s prevzemi podjetij. Leta 2000 HERMES SoftLab uspešno prevzame podjetje Zaslon, ki je glavni pionir na področju varnega elektronskega bančništva in poslovanja v Sloveniji. Dve leti kasneje HERMES SoftLab in irsko podjetje Eurologic Systems ustanovita podjetje StorScape. Nato HERMES Softlab odkupi večinski delež srbskega podjetja Virtual Team. Leta 2006 pa postane edini lastnik le-tega. Leto kasneje sledi prevzem podjetja Coding v Črni gori. Leta 2008 se HERMES SoftLab pridruži skupini Comtrade, ki zaposluje 1.600 ljudi in deluje v 16 državah. Dve leti kasneje sledi prehod na skupno blagovno znamko skupine Comtrade. Takrat HERMES SoftLab spremeni ime v Comtrade d.o.o.

Strankam nudi razvoj visoko kvalitetnih aplikacij po meri in e-rešitev. Prav tako nudi vrhunsko strokovno znanje na področju poslovnih integracij in storitveno usmerjene arhitekture z računalništvom v oblaku, upravljanja IT storitev in nadzornih orodij, poslovne inteligence in upravljaljskega svetovanja.

Oddelek, kjer je bila anketa opravljena, zaposluje 49 ljudi, ki razvijajo programsko opremo za avtomatsko varnostno kopiranje (angl. *Backup*) in obnavljanje (angl. *Restore & Recovery*) podatkov. Z metodologijo Scrum so se prvič srečali na začetku leta 2011, vendar zgolj poskusno. Resneje jo začnejo uporabljati poleti 2012 in jo od takrat še vedno postopoma uvajajo v svoj razvojni proces.

4.2 Zbiranje podatkov

Kot smo že omenili v prejšnjem poglavju, smo podatke za analizo uporabe metodologije Scrum v podjetju Comtrade pridobili s pomočjo vprašalnika. Zaradi lažje izvedbe anketiranja in kasnejše obdelave podatkov smo vprašalnik iz Dodatka A predelali v elektronsko obliko, in sicer s pomočjo orodja *Google Forms*. Odgovore smo zbirali dober mesec in pol, ter v tem času uspeli pridobiti 40 odgovorov. Pridobljene podatke smo nato pretvorili v ustrezno obliko za analizo in jih analizirali z uporabo programa za statistično analizo (*SPPS - Statistical Package for the Social Sciences*).

4.3 Normalnost porazdelitve pridobljenih podatkov

Analizo smo začeli s preverjanjem normalnosti porazdelitve pridobljenih podatkov. Obstaja več načinov preverjanja. Eden izmed teh je tudi preverjanje s pomočjo koeficienta asimetrije (angl. *Skewness*) in koeficienta sploščenosti (angl. *Kurtosis*), katerega smo tudi mi izbrali pri naši analizi. Porazdelitev podatkov je približno normalna (realni podatki so namreč redkokdaj porazdeljeni popolnoma normalno), kadar je vrednost koeficienta asimetrije znotraj intervala $[-1,1]$ [16], vrednost koeficienta sploščenosti pa znotraj intervala $[-3,3]$ [16]. Če je vrednost koeficienta asimetrije negativna, je levi del porazdelitve daljši, v nasprotnem primeru

4.3. NORMALNOST PORAZDELITVE PRIDOBLJENIH PODATKOV²¹

(koeficient asimetrije je pozitiven) pa je desni del porazdelitve daljši. Koeficient asimetrije ima vrednost 0, ko je porazdelitev popolnoma simetrična. Porazdelitev je sploščena (platikurtična), kadar je koeficient sploščenosti negativen oziroma koničasta (leptokurtična), kadar je koeficient sploščenosti pozitiven. Popolnoma normalna porazdelitev ima ničelno sploščenost (porazdelitev je mezokurtična).

Če pogledamo Tabelo 4.1 vidimo, da pri šestih vprašanjih (1., 7., 19., 27., 31. in 34.) vrednost koeficienta asimetrije nekoliko presega prej omenjeni interval, v katerem je porazdelitev približno normalno porazdeljena. Vendar pa to ni imelo velikega pomena oziroma vpliva na nadaljno analizo. Medtem, ko so vrednosti koeficienta sploščenosti pri vseh vprašanjih, kot lahko vidimo iz Tabele 4.1, v mejah intervala, ki velja zanj. Na osnovi vrednosti koeficienta asimetrije, lahko razberemo, da je pri večini vprašanj (približno 72 % oziroma 38 od 53 vprašanj) porazdelitev zamaknjena v desno oziroma raztegnjena v levo. Pri približno 54 % vprašanj je porazdelitev nekoliko sploščena oziroma manj ostra od normalne porazdelitve. Gledano v celoti, lahko rečemo, da so podatki približno normalno porazdeljeni.

Tabela 4.1: Vrednosti koeficientov asimetrije in sploščenosti za vsako vprašanje

Vprašanje	N	Povprečje	Mediana	Standardni odklon	Asimetrija	Sploščenost
1. Uporabniška vloga	40	1,50	1,00	0,97	1,92	2,50
2. Izkušnje z uporabo metodologije Scrum	40	3,15	3,00	0,66	0,39	0,67
3. Vzdrževanje seznama zahtev	40	3,26	3,00	1,28	0,84	-0,16
4. Uporaba uporabniških zgodb	40	4,05	4,00	1,26	0,15	-0,98
5. Sodelovanje s produktnim vodjo	40	4,33	4,00	1,27	-0,18	-1,08
6. Ocenjevanje uporabniških zgodb po metodi Planning Poker	40	3,00	3,00	1,54	0,18	-1,20

7. Ocenjevanje uporabniških zgodb po metodi Team Estimation Game	40	2,23	2,00	1,33	1,15	0,62
8. Načrtovanje izdaje	40	3,13	3,00	1,34	0,63	-0,82
9. Načrtovanje iteracije	40	3,40	3,00	1,35	0,32	-0,82
10. Vzdrževanje seznama nalog	40	3,42	3,00	1,38	0,29	-0,64
11. Izvajanje dnevnih sestankov	40	4,68	5,00	1,37	-0,71	-0,74
12. Uporaba diagrama preostalega dela	40	3,10	3,00	1,18	0,60	-0,08
13. Dosledno upoštevanje koncepta "done"	40	3,95	4,00	1,40	0,21	-1,29
14. Izvajanje sestankov za pregled rezultatov iteracije	40	4,03	4,00	1,42	0,12	-1,19
15. Izvajanje retrospektivnih sestankov.	40	3,78	4,00	1,48	0,26	-1,25
16. Poznavanje aktivnosti, ki jih izvajajo posamezne uporabniške vloge.	40	4,15	4,00	1,49	-0,03	-1,55
17. Metodologija Scrum je veliko boljša od tradicionalnih.	40	5,15	5,00	1,03	-0,76	1,14
18. Metodologijo Scrum je veliko lažje razumeti in uporabljati od tradicionalnih.	40	4,93	5,00	1,21	-0,68	0,22
19. Metodologija Scrum se veliko bolj ujema z mojim načinom dela, vrednotami in izkušnjami od tradicionalnih.	40	5,25	5,00	1,21	-1,23	2,70

4.3. NORMALNOST PORAZDELITVE PRIDOBLJENIH PODATKOV²³

20. Metodologija Scrum omogoča, da brez težav opazujem kako jo uporabljajo druge skupine Scrum.	40	4,88	5,00	1,40	-0,77	0,46
21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.	40	4,80	5,00	1,34	-0,48	-0,31
22. Uporaba metodologije Scrum mi ne povzroča veliko stroškov in napora.	40	4,88	5,00	1,52	-0,93	0,15
23. Metodologija Scrum razrešuje veliko problemov pri razvoju prog. opreme.	40	4,63	5,00	1,39	-0,54	-0,40
24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.	40	5,40	6,00	0,96	-0,16	-0,98
25. Metodologija Scrum je tehnično bolj napredna od tradicionalnih.	40	4,55	5,00	1,18	-0,62	0,50
26. Metodologija Scrum pomembno prispeva k večjemu zadovoljstvu naročnikov s končno prog. rešitvijo.	40	4,88	5,00	1,45	-0,77	-0,22
27. Metodologija Scrum je skladna z mojimi potrebami pri razvoju prog. rešitev.	40	5,10	5,00	1,34	-1,28	1,86
28. Uporaba metodologije Scrum olajša izvedbo zahtevnejših nalog pri razvoju prog. rešitev.	40	4,65	5,00	1,66	-0,47	-0,80

29. Z metodologijo Scrum lahko samostojno eksperimentiram.	40	4,28	5,00	1,24	-0,90	0,21
30. Kolegi in prijatelji mi večinoma priporočajo uporabo metodologije Scrum.	40	4,38	5,00	1,23	-0,69	0,50
31. Enostavna prilagoditev metodologije Scrum različnim načinom dela razvojne skupine.	40	4,88	5,00	1,26	-1,12	1,85
32. Metodologija Scrum mi olajša pridobivanje novih znanj pri razvoju prog. rešitev.	40	4,50	5,00	1,59	-0,49	-0,37
33. Razvijalci brez težav sprejemamo(-jo) spremembe v delu, ki jih zahteva metod. Scrum.	40	4,10	5,00	1,53	-0,40	-0,51
34. Imamo vso potrebno tehn. infrastrukturo za uporabo metod. Scrum.	40	5,55	6,00	1,36	-1,50	1,76
35. Razvijalci uporabljamo(-jo) metod. Scrum točno tako kot je predpisano v literaturi.	40	3,18	3,00	1,39	0,15	-1,12
36. Na razpolago imamo vse potrebne vire za uspešno in učinkovito uporabo metod. Scrum.	40	5,15	5,50	1,29	-0,74	0,07

4.3. NORMALNOST PORAZDELITVE PRIDOBLJENIH PODATKOV²⁵

37. Sodelavci, ki že uporabljajo metod. Scrum, večinoma močno priporočajo njeno uporabo.	40	4,50	4,00	1,28	-0,81	0,95
38. V skupini Scrum razvijam prog, rešitev skupaj s sodelavci, s katerimi se veliko družim.	40	4,82	5,00	1,39	-0,87	0,38
39. Metodologija Scrum omogoča veliko spontane in neformalne komunikacije med člani skupine Scrum.	40	4,95	5,00	1,57	-0,88	0,13
40. Z uporabo metodologij razvoja prog. opreme imam veliko izkušenj.	40	5,15	5,00	1,19	-0,77	0,98
41. Metodologija Scrum omogoča, da sami načrtujemo delo in razrešimo večino problemov in težav, do katerih je med razvojem prog. rešitve prišlo.	40	4,65	5,00	1,49	-0,96	0,31
42. Osebe, katerih mnenje spoštujem, večinoma močno priporočajo uporabo metod. Scrum.	40	4,33	4,00	1,40	-0,68	0,01
43. Vsak dodaten uporabnik metod. Scrum močno poveča koristnost uporabe metod. Scrum.	40	4,10	4,00	1,32	0,16	-0,63

44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.	40	4,40	4,50	1,01	-0,27	-0,49
45. Zahteva podjetja, da mora razvoj potekati po metod. Scrum, je glavni razlog da jo uporabljam.	40	4,75	5,00	1,51	-0,25	-0,85
46. Metodologija Scrum omogoča hitrejši razvoj programske opreme od tradicionalnih.	40	4,65	5,00	1,61	-0,64	-0,19
47. Metodologija Scrum omogoča razvoj prog. opreme, ki bolj ustreza zahtevam naročnika, od tradicionalnih.	40	4,98	5,50	1,69	-0,84	-0,32
48. Metodologija Scrum omogoča da se pri razvoju prog. rešitve naučimo več kot pri tradicionalnih.	40	4,03	4,50	1,56	-0,47	-0,68
49. Metodologija Scrum omogoča večje zadovoljstvo programerjev z razvito prog rešitvijo od tradicionalnih.	40	4,38	5,00	1,56	-0,75	-0,19
50. Metodologija Scrum omogoča boljši pregled nad potekom razvoja prog. rešitve od tradicionalnih.	40	4,95	5,00	1,52	-0,79	0,18

4.3. NORMALNOST PORAZDELITVE PRIDOBLJENIH PODATKOV

51. Metodologija Scrum omogoča boljše sodelovanje programerjev pri razvoju prog. rešitve od tradicionalnih.	40	4,95	5,50	1,63	-0,81	-0,29
52. Metodologija Scrum omogoča nižjo kompleksnost procesa razvoja prog. rešitve od tradicionalnih.	40	4,50	5,00	1,47	-1,00	0,90
53. Metodologija Scrum zahteva več discipline pri razvoju prog. rešitve od tradicionalnih.	40	4,83	5,00	1,63	-0,75	0,01

4.4 Zanesljivost pridobljenih podatkov

Nato je sledilo preverjanje zanesljivosti pridobljenih podatkov. Le-to smo opravili z izračunom korelacijskega koeficienta ICC (angl. *Intraclass Correlation Coefficient*) v že omenjenem programu za statistično analizo (*SPSS*). Dobili smo dva rezultata, in sicer t.i. *Single Measure* in *Average Measure*. Prvi je korelacijski koeficient ICC za odgovore posameznega anketiranca, drugi pa za povprečje odgovorov vseh anketirancev. Za našo analizo je bil slednji pomembnejši, saj nas je zanimalo v kolikšni meri so si podobni odgovori anketirancev, kar pa posledično pove ali je anketni vprašalnik zanesljiv (ali se z njegovo uporabo dobi zanesljive podatke).

Vrednost korelacijskega koeficienta ICC (*Average Measure*) je tako bila 0,93. Pri tem velja, da v kolikor je vrednost korelacijskega koeficienta ICC enaka ali večja od 0,7 je zanesljivost podatkov sprejemljiva. Če je vrednost enaka ali večja od 0,8 je zanesljivost podatkov dobra, če pa je enaka ali večja od 0,9 je zanesljivost podatkov zelo dobra. Oba omenjena rezultata sta tudi predstavljena v Tabeli 4.2.

Tabela 4.2: Rezultata izračuna korelacijskega koeficienta ICC

	Korelacijski koeficient ICC	95 % interval zaupanja	
		Spodnja meja	Zgornja meja
a	0,25	0,18	0,35
b	0,93	0,90	0,96

Legenda:

a - Vrednost korelacijskega koeficienta ICC za odgovore posameznega anketiranca (*Single Measure*)

b - Vrednost korelacijskega koeficienta ICC za povprečje odgovorov vseh anketirancev (*Average Measure*)

4.5 Sprejemljivost ključnih aktivnosti metodologije Scrum

Analizo smo nadaljevali s sprejemljivostjo ključnih aktivnosti oziroma elementov metodologije Scrum v podjetju Comtrade. Pri tem nas je zanimalo kako dobro poznajo posamezne aktivnosti metodologije Scrum, kakšne izkušnje imajo z uporabo (izvajanjem) le-teh, ter do katere stopnje Cooper - Zmudovega modela (predstavljen v podpoglavju 3.1) so jih sprejeli. Analizo smo opravili s pomočjo frekvenčnih tabel.

1. Vzdrževanje seznama zahtev

Iz Tabele 4.3 lahko vidimo, da je vseh 40 anketirancev že slišalo za seznam zahtev. Od teh je približno tretjina (35 %) takšnih, ki še nikoli niso vzdrževali seznama zahtev. 11 anketirancev je odgovorilo, da ima dovolj potrebnega znanja, da lahko vzdržuje seznam zahtev. Nekaj več kot tretjina (15 od 40 oziroma 37,5 %) anketirancev ima že pozitivne izkušnje z vzdrževanjem seznama zahtev. Skoraj tretjina od teh (4 od 15) pa izvaja vzdrževanje seznama zahtev rutinsko, učinkovito in celovito.

2. Uporaba uporabniških zgodb za predstavitev zahtevane funkcionalnosti

Uporabniške zgodbe in uporaba le-teh za predstavitev zahtevane funkcionalnosti je dobro znana zaposlenim v podjetju Comtrade, kot lahko razberemo iz Tabele 4.4. Namreč le 10 % anketirancev se še ni srečalo z uporabniškimi zgodbami. Medtem ko imata skoraj dve tretjini (25 od 40 oziroma 62,5 %) anketirancev že pozitivne izkušnje z uporabo uporabniških zgodb. Enako število anketirancev (11), kot pri prejšnjem elementu (aktivnosti) metodologije Scrum, je odgovorilo, da ima dovolj potrebnega znanja za uporabo uporabniških zgodb.

Tabela 4.3: Vzdrževanje seznama zahtev

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	14	35	35	35
3	11	27,5	27,5	62,5
4	9	22,5	22,5	85
5	2	5	5	90
6	4	10	10	100
Σ	40	100	100	100

Tabela 4.4: Uporaba uporabniških zgodb za predstavitev zahtevane funkcionalnosti

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	4	10	10	10
3	11	27,5	27,5	37,5
4	11	27,5	27,5	65
5	7	17,5	17,5	82,5
6	7	17,5	17,5	100
Σ	40	100	100	100

3. Sodelovanje s produktnim vodjo pri razčiščevanju podrobnosti uporabniških zgodb

Zagotovo ena izmed pomembnejših aktivnosti metodologije Scrum, saj se v primeru, da je sodelovanje članov razvojne skupine s produktnim vodjo slabo oziroma sodelovanja ni, lahko zgodi da člani razvojne skupine zaradi drugačnega razumevanja uporabniške zgodbe kot produktni vodja, le-te ne realizirajo uspešno (po željah produktnega vodje). Kar pa lahko pomeni, da je potrebno uporabniško zgodbo "prenesti" v naslednjo iteracijo. Poleg tega pa lahko to povzroči nezadovoljstvo stranke (npr. zaradi premika uporabniške zgodbe v naslednjo iteracijo, stranka ne dobi delujoče funkcionalnosti ob prvotno dogovorjenem datumu). Člani razvojne skupine v podjetju Comtrade dobro sodelujejo s produktnim vodjo pri razčiščevanju podrobnosti uporabniških zgodb, kot lahko vidimo iz Tabele 4.5. Namreč kar 70 % (28 od 40) anketirancev ima pozitivne izkušnje s sodelovanjem. Skoraj 70 % oziroma natančneje 68 % (19 od 28) od teh pa je s produktnim vodjo sodelovalo že tolikokrat, da jim je to postalo rutinsko opravilo. Medtem ko zgolj trije anketiranci (7,5 %) še niso sodelovali s produktnim vodjo.

Tabela 4.5: Sodelovanje s produktnim vodjo pri razčiščevanju podrobnosti uporabniških zgodb

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	3	7,5	7,5	7,5
3	9	22,5	22,5	30
4	9	22,5	22,5	52,5
5	10	25	25	77,5
6	9	22,5	22,5	100
Σ	40	100	100	100

4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker

Kot smo že pri predstavitvi metodologije Scrum omenili, je Planning Poker ena izmed metod za ocenjevanje uporabniških zgodb. V literaturi je velikokrat omenjena kot najbolj znana in najbolj pogosto uporabljena metoda za ocenjevanje uporabniških zgodb. Spodnja Tabela 4.6 prikazuje, kako dobro jo poznajo in koliko jo uporabljajo v podjetju Comtrade. 40 % anketirancev še ni ocenjevalo uporabniških zgodb z uporabo metode Planning Poker oziroma dobra polovica (22,5 % vseh anketirancev) teh še nikoli ni slišala za ocenjevanje uporabniških zgodb po metodi Planning Poker. Okoli tretjina (14 od 40 oziroma 35 %) anketirancev ima z ocenjevanjem uporabniških zgodb po metodi Planning Poker že pozitivne izkušnje. Kar 71 % (10 od 14) od teh pa opravlja ocenjevanje rutinsko. Četrtnina vseh anketirancev je odgovorila, da ima dovolj potrebnega znanja za ocenjevanje uporabniških zgodb po metodi Planning Poker.

Tabela 4.6: Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	9	22,5	22,5	22,5
2	7	17,5	17,5	40
3	10	25	25	65
4	4	10	10	75
5	9	22,5	22,5	97,5
6	1	2,5	2,5	100
Σ	40	100	100	100

5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game

Nekoliko manj znana in manj uporabljena metoda za ocenjevanje uporabniških zgodb je metoda Team Estimation Game (prav tako predstavljena v prejšnjem poglavju), kar lahko tudi potrdimo s pomočjo Tabele 4.7. Več kot tretjina (15 od 40 oziroma 37,5 %) anketirancev še nikoli ni ocenjevala uporabniških zgodb po metodi Team Estimation Game. Prav tako jih je nekaj več kot tretjina (14 od 40 oziroma 35 %), ki še nikoli niso slišali za ocenjevanje uporabniških zgodb po metodi Team Estimation Game. Oziroma, če obe številki združimo, vidimo da več kot dve tretjini (29 od 40 oziroma 72,5 %) anketirancev še nista ocenjevali uporabniških zgodb po omenjeni metodi. Le petina anketirancev (8 od 40 oziroma 20 %) je odgovorila, da ima že izkušnje z uporabo metode Team Estimation Game. Poleg tega lahko vidimo, da so v primerjavi s prejšnjimi elementi (aktivnostmi) metodologije Scrum, tokrat le trije anketiranci odgovorili, da imajo dovolj potrebnega znanja za ocenjevanje uporabniških zgodb po metodi Team Estimation Game.

Tabela 4.7: Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	14	35	35	35
2	15	37,5	37,5	72,5
3	3	7,5	7,5	80
4	5	12,5	12,5	92,5
5	2	5	5	97,5
6	1	2,5	2,5	100
Σ	40	100	100	100

6. Načrtovanje izdaje na osnovi ocenjene hitrosti razvojne skupine

Naslednja ključna aktivnost metodologije Scrum, ki smo jo analizirali v podjetju Comtrade, je načrtovanje izdaje. Le-to precej bolje poznajo in uporabljajo, kot recimo prej analizirano uporabo metode Team Estimation Game za ocenjevanje uporabniških zgodb. Iz Tabele 4.8 lahko vidimo, da zgolj en anketiranec še ni slišal za načrtovanje izdaje na osnovi ocenjene hitrosti razvojne skupine. Kljub temu, da je 18 anketirancev (45 % vseh anketirancev) odgovorilo, da izdaje še ni načrtovalo pa lahko vidimo, da ima več kot tretjina (15 od 40 oziroma 37,5 %) anketirancev že pozitivne izkušnje z načrtovanjem izdaje. Od teh pa kar dobra polovica (8 od 15) izvaja načrtovanje izdaje rutinsko. Poleg tega je 6 anketirancev odgovorilo, da ima dovolj potrebnega znanja za načrtovanje izdaje na osnovi ocenjene hitrosti razvojne skupine.

Tabela 4.8: Načrtovanje izdaje na osnovi ocenjene hitrosti razvojne skupine

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	1	2,5	2,5	2,5
2	18	45	45	47,5
3	6	15	15	62,5
4	7	17,5	17,5	80
5	6	15	15	95
6	2	5	5	100
Σ	40	100	100	100

7. Načrtovanje vsebine naslednje iteracije na sestanku za načrtovanje iteracije

Če pogledamo Tabelo 4.9 vidimo, da je delež anketirancev, ki že imajo izkušnje z načrtovanjem iteracije oziroma načrtovanjem vsebine le-te, nekoliko večji kot delež anketirancev, ki še nimajo nobene izkušnje. In sicer prvo omenjenih je 47,5 % (19 od 40), medtem ko je slednjih 40 %. 9 anketirancev od teh, ki že imajo izkušnje, je odgovorilo, da jim je načrtovanje že postalo rutinsko opravilo. Dva anketiranca (5% od vseh) pa izvajata načrtovanje iteracije rutinsko, učinkovito in celovito, ter s tem olajšata izvedbo ostalih ključnih aktivnosti metodologije Scrum. 12,5 % anketirancev meni, da ima dovolj potrebnega znanja za načrtovanje iteracije.

Tabela 4.9: Načrtovanje vsebine naslednje iteracije na sestanku za načrtovanje iteracije

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	16	40	40	40
3	5	12,5	12,5	52,5
4	8	20	20	72,5
5	9	22,5	22,5	95
6	2	5	5	100
Σ	40	100	100	100

8. Vzdrževanje seznama nalog

Ena izmed nalog razvojne skupine, ki smo jo predstavili v prejšnjem Poglavju 2, je tudi vzdrževanje seznama nalog. Skoraj tretjina (12 od 40 oziroma 30 %) anketirancev še nikoli ni izvajala omenjene aktivnosti metodologije Scrum, kot lahko razberemo iz Tabele 4.10. Dva anketiranca od teh še nikoli nista slišala za vzdrževanje seznama nalog. Nekaj manj kot četrtnina (22,5%) anketirancev pa je odgovorila, da ima dovolj potrebnega znanja, da lahko vzdržuje seznam nalog. Pozitivne izkušnje z vzdrževanjem seznama nalog ima slaba polovica anketirancev (19 od 40 oziroma 47,5 %). Približno 21 % od teh je vzdrževanje opravljalo že tolikokrat, da jim je to postalo rutinsko opravilo. Enak odstotek anketirancev opravlja vzdrževanje seznama nalog rutinsko, učinkovito in celovito. S tem pa izboljšujejo hitrost in kakovost razvoja programske opreme.

Tabela 4.10: Vzdrževanje seznama nalog

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	2	5	5	5
2	10	25	25	30
3	9	22,5	22,5	52,5
4	11	27,5	27,5	80
5	4	10	10	90
6	4	10	10	100
Σ	40	100	100	100

9. Izvajanje dnevnih sestankov

Zelo pomemben element metodologije Scrum so dnevni sestanki, saj se z njimi preverja sproti opravljeno delo. Prav tako pa se lahko z izvajanjem le-teh pravočasno odkrije in odpravi težave, v kolikor se te pojavijo. V spodnji Tabeli 4.11 vidimo, da je vseh 40 anketirancev že slišalo za izvajanje dnevnih sestankov. Pri tem pa je zgolj 10 % (4 od 40) anketirancev, ki jih še niso izvajali. Dobre tri četrtine (31 od 40 oziroma 77,5 %) anketirancev ima že vsaj pozitivne izkušnje z izvajanjem omenjenega sestanka. Največji delež teh pa predstavljajo anketiranci, ki izvajajo dnevne sestanke rutinsko, učinkovito in celovito, in sicer jih je slaba polovica (15 od 31 oziroma 48 %). Oziroma za skoraj dve tretjini (25 od 40 oziroma 62,5 %) vseh anketirancev, je izvajanje dnevnih sestankov postalo rutina. 5 anketirancev pa je odgovorilo, da ima dovolj potrebnega znanja, da lahko izvaja dnevne sestanke. Iz predstavljenih podatkov lahko sklepamo, da v podjetju Comtrade zelo dobro poznajo dnevne sestanke in jih redno izvajajo.

Tabela 4.11: Izvajanje dnevnih sestankov

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	4	10	10	10
3	5	12,5	12,5	22,5
4	6	15	15	37,5
5	10	25	25	62,5
6	15	37,5	37,5	100
Σ	40	100	100	100

10. Uporaba diagrama preostalega dela za nadzor poteka dela

Eden izmed načinov za nadzor poteka dela je uporaba diagrama preostalega dela, ki pa ga v podjetju Comtrade, kot vidimo v Tabeli 4.12, ne uporabljajo pogosto oziroma nimajo izkušenj z njegovo uporabo. In sicer dobra tretjina (14 od 40 oziroma 35 %) anketirancev še nikoli ni uporabljala diagrama preostalega dela. Dva od teh, še nikoli nista slišala za diagram preostalega dela. Anketirancev, ki imajo že vsaj pozitivne izkušnje z uporabo diagrama preostalega dela oziroma je za njih to rutinsko opravilo, je nekaj več kot četrtnina (11 od 40 oziroma 27,5 %). Največji delež (15 od 40 oziroma 37,5 %) anketirancev je odgovoril, da ima dovolj potrebnega znanja za nadzorovanje poteka dela z uporabo diagrama preostalega dela.

Tabela 4.12: Uporaba diagrama preostalega dela za nadzor poteka dela

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	2	5	5	5
2	12	30	30	35
3	15	37,5	37,5	72,5
4	5	12,5	12,5	85
5	5	12,5	12,5	97,5
6	1	2,5	2,5	100
Σ	40	100	100	100

11. Dosledno upoštevanje koncepta "done"

Prav tako eden izmed pomembnejši elementov metodologije Scrum, in sicer predstavlja zahteve, ki jih mora izpolnjevati uporabniška zgodba ob koncu iteracije. Namreč, kot smo že omenili pri predstavitvi metodologije Scrum, mora biti uporabniška zgodba v celoti realizirana in delujoča, da jo ob koncu iteracije lahko damo v uporabo. Tabela 4.13 prikazuje, koliko dosledno upoštevajo koncept "done", zaposleni v podjetju Comtrade. Dobra polovica (21 od 40 oziroma 52,5 %) anketirancev ima že pozitivne izkušnje z doslednim upoštevanjem koncepta "done".

Za 62 % (13 od 21) od teh, pa je upoštevanje le-tega, že postalo rutinsko opravilo. Le 15 % oziroma 6 anketirancev, koncepta "done" še ni dosledno upoštevalo. Slaba tretjina (13 od 40 oziroma 32,5 %) anketirancev je odgovorila, da ima dovolj potrebnega znanja za dosledno upoštevanje omenjenega koncepta.

Tabela 4.13: Dosledno upoštevanje koncepta "done"

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	6	15	15	15
3	13	32,5	32,5	47,5
4	6	15	15	62,5
5	7	17,5	17,5	80
6	8	20	20	100
Σ	40	100	100	100

12. Izvajanje sestankov za pregled rezultatov posameznih iteracij

Kot že omenjeno pri predstavitvi metodologije Scrum, se iteracija zaključi s sestankom na katerem se pregleda in oceni rezultate le-te. Tabela 4.14 prikazuje kako dobro poznajo oziroma kakšne izkušnje imajo anketiranci z izvajanjem sestankov za pregled rezultatov posameznih iteracij. Kot vidimo je vseh 40 anketirancev že slišalo za izvajanje omenjenega sestanka, kar pa ni presenetljivo, saj gre za zelo pomembno aktivnost metodologije Scrum. Čeprav je 7 anketirancev odgovorilo, da še nikoli ni izvajalo sestankov za pregled rezultatov iteracij, pa podatek, da imata skoraj dve tretjini (25 od 40 oziroma 62,5 %) anketirancev že vsaj pozitivne izkušnje, odtehta le-to. Poleg tega lahko vidimo, da 40 % od teh (oziroma četrtnina vseh anketirancev) izvaja sestanke za pregled rezultatov posameznih iteracij rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti metodologije Scrum in izboljša hitrost in kakovost razvoja programske opreme. Petina vseh anketirancev pa je odgovorila, da ima dovolj potrebnega znanja, da lahko izvaja sestanke za pregled rezultatov iteracij.

Tabela 4.14: Izvajanje sestankov za pregled rezultatov posameznih iteracij

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	7	17,5	17,5	17,5
3	8	20	20	37,5
4	12	30	30	67,5
5	3	7,5	7,5	75
6	10	25	25	100
Σ	40	100	100	100

13. Izvajanje retrospektivnih sestankov

Pred začetkom načrtovanja nove iteracije sledi še zadnji sestanek, in sicer retrospektivni sestanek (prav tako predstavljen v Poglavju 2), na katerem se analizira dobre in slabe stvari pretekle iteracije. Če pogledamo Tabelo 4.15 vidimo, da ima nekaj več kot polovica (22 od 40 oziroma 55 %) anketirancev že vsaj pozitivne izkušnje z izvajanjem retrospektivnih sestankov. Dobra tretjina od teh pa izvaja retrospektivni sestanek rutinsko, celovito in učinkovito. Na drugi strani pa je 11 anketirancev oziroma nekaj več kot četrtnina, ki omenjenega sestanka še nikoli niso izvajali. 17,5 % anketirancev meni, da ima dovolj potrebnega znanja za izvajanje retrospektivnih sestankov. Če primerjamo omenjene podatke s podatki prej analiziranega izvajanja sestankov za pregled rezultatov posameznih iteracij, vidimo da je izvajanje retrospektivnih sestankov nekoliko manj sprejeto med zaposlenimi v podjetju Comtrade.

Tabela 4.15: Izvajanje retrospektivnih sestankov

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	11	27,5	27,5	27,5
3	7	17,5	17,5	45
4	10	25	25	70
5	4	10	10	80
6	8	20	20	100
Σ	40	100	100	100

14. Poznavanje aktivnosti, ki jih izvajajo posamezne uporabniške vloge: produktni vodja, vodja metodologije in član razvojne skupine

Zadnji element oziroma ključna aktivnost metodologije Scrum, ki smo jo analizirali, je poznavanje aktivnosti posamezne uporabniške vloge. Več kot polovica (22 od 40 oziroma 55 %) anketirancev ima že pozitivne izkušnje z aktivnostmi posameznih uporabniških vlog, kot lahko razberemo iz Tabele 4.16. Dobri tretjini od teh (8 od 22 oziroma 36 %), so aktivnosti posameznih uporabniških vlog postale rutinsko opravilo, polovica (11 od 22 oziroma 50 %) pa jih opravlja rutinsko, celovito in učinkovito, kar olajša izvedbo ostalih ključnih aktivnosti metodologije Scrum in pohitri ter izboljša kakovost razvoja programske rešitve. Le 6 anketirancev je odgovorilo, da se še ni soočilo z aktivnostmi posameznih uporabniških vlog. Anketirancev, ki so odgovorili, da imajo dovolj potrebnega znanja, da poznajo aktivnosti posameznih uporabniških vlog, pa je dvakrat več (12 od 40 oziroma 30 %).

Tabela 4.16: Poznavanje aktivnosti, ki jih izvajajo posamezne uporabniške vloge: produktni vodja, vodja metodologije in član razvojne skupine

Stopnja uporabe	Frekvenca	Odstotek [%]	Veljavni odstotek [%]	Kumulativni odstotek [%]
1	0	0	0	0
2	6	15	15	15
3	12	30	30	45
4	3	7,5	7,5	52,5
5	8	20	20	72,5
6	11	27,5	27,5	100
Σ	40	100	100	100

Če povzamemo analizo sprejemljivosti ključnih aktivnosti (elementov) metodologije Scrum, vidimo da so v podjetju Comtrade najbolj sprejeti sestanki. Med njimi pa je najbolj sprejet oziroma uveljavljen (uporabljen) dnevni sestanek. Prav tako zelo dobro poznajo uporabniške zgodbe za predstavitev funkcionalnosti, ter jih rutinsko uporabljajo. Nekoliko manj kot uporabniške zgodbe, poznajo in uporabljajo metode za ocenjevanje le-teh. Pri tem metodo Planning Poker bolje poznajo in bolj uporabljajo kot metodo Team Estimation Game, kar pa ne preseneča, saj je metoda Planning Poker, kot smo že omenili, splošno bolj znana (v literaturi pogosto omenjena kot najbolj znana in uporabljena metoda za ocenjevanje uporabniških zgodb). Sodelovanje članov razvojnih skupin s produktnimi vodji pri razčiščevanju podrobnosti uporabniških zgodb, v podjetju Comtrade, je prav tako zelo dobro. Iz tega dejstva lahko sklepamo, da nejasnosti uporabniških zgodb rešujejo pravočasno, kar verjetno posledično pomeni, da uporabniške zgodbe realizirajo uspešno in pravočasno. Dodatno lahko našo domnevo potrdimo z dejstvom, da dobro poznajo in sledijo konceptu "done" (uprabniške zgodbe morajo biti v celoti realizirane). Eden izmed elementov metodologije Scrum, ki so najmanj sprejeti, je diagram preostalega dela za nadzor poteka dela. Kot pa smo že omenili, zelo dobro poznajo in uporabljajo dnevne sestanke. Možen sklep bi tako lahko bil,

da nadzor poteka dela opravljajo kar pri dnevni sestanekih. Prav tako v podjetju Comtrade dobro poznajo aktivnosti posameznih uporabniških vlog.

Za zaključek analize sprejemljivosti ključnih aktivnosti metodologije Scrum lahko čez palec ocenimo, da je sprejemanje metodologije Scrum kot inovacije, v podjetju Comtrade, nekje med četrto (*Acceptance*) in peto (*Routinization*) stopnjo Cooper - Zmudovega modela (predstavljen v pod poglavju 3.1).

4.6 Ključni faktorji uporabe metodologije Scrum

Analizo uporabe metodologije Scrum v podjetju Comtrade smo zaključili z analizo ključnih faktorjev oziroma t.i. faktorjev DOI (Rogersov model). In sicer nas je zanimalo kateri so tisti, ki najbolj vplivajo na sprejetje oziroma uporabo metodologije Scrum. Le-to smo ugotavljali z izračuni korelacij med posameznimi elementi (aktivnostmi) metodologije Scrum in omenjenimi faktorji. V nadaljevanju so za vsak sklop faktorjev posebej predstavljeni in analizirani rezultati. Pri tem pa smo se osredotočili na korelacije, ki so statistično značilnejše.

4.6.1 Vpliv faktorjev o inovaciji

Iz Tabele 4.17 lahko razberemo, da faktorji o inovaciji najbolj vplivajo na sprejetje prvih treh elementov (aktivnosti) metodologije Scrum, in sicer na vzdrževanje seznama zahtev, uporabo uporabniških zgodb in sodelovanje s produktnim vodjo. Kot vidimo, na sprejetje oziroma uveljavljanje vsakega izmed omenjenih elementov vpliva pet od skupno devetih faktorjev o inovaciji. Pri tem pa najbolj vplivata oziroma sta najbolj pomembna (pojavljata se pri vseh treh omenjenih elementih) enostavnost metodologije Scrum in pa koliko problemov razrešuje uporaba le-te pri razvoju programske opreme. Elementi (aktivnosti) metodologije Scrum na sprejetje katerih, v podjetju Comtrade, faktorji o inovaciji niso imeli vpliva, so: načrtovanje izdaje, izvajanje dnevni sestanekov, uporaba Burn Down diagrama in izvajanje retrospektivni sestanekov. Faktor oziroma trditev o inovaciji, ki najbolj vpliva na sprejetje in uveljavljanje metodologije Scrum v podjetju Comtrade, je trditev, da metodologija Scrum postaja standardna metodologija v podjetjih za

razvoj programske opreme. Iz tega bi lahko sklepali, da v podjetju Comtrade upoštevajo in sledijo standardom oziroma trendom razvoja programske opreme.

Tabela 4.17: Korelacije med elementi metodologije Scrum in faktorji oziroma trditvami o inovaciji

Element (aktivnost) metodologije Scrum	Faktorji DOI	Število faktorjev DOI
1. Vzdrževanje seznama zahtev	<p>17. Metodologija Scrum je veliko boljše od tradicionalnih metodologij.**</p> <p>18. Metodologijo Scrum je veliko lažje razumeti in uporabljati od tradicionalnih.*</p> <p>21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.**</p> <p>23. Metodologija Scrum razrešuje veliko problemov pri razvoju prog. opreme.*</p> <p>24. Metodologija Scrum postaja standardna metoda v podjetjih za razvoj prog. opreme.*</p>	5
2. Uporaba uporabniških zgodb	<p>17. Metodologija Scrum je veliko boljše od tradicionalnih metodologij.*</p> <p>19. Metodologija Scrum se veliko bolj ujema z mojim načinom dela, vrednotami in izkušnjami od tradicionalnih.**</p> <p>21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.**</p> <p>22. Uporaba metodologije Scrum mi ne povzroča veliko stroškov in napora.**</p> <p>23. Metodologija Scrum razrešuje veliko problemov pri razvoju prog. opreme.**</p>	5

3. Sodelovanje s produktnim vodjo	<p>19. Metodologija Scrum se veliko bolj ujema z mojim načinom dela, vrednotami in izkušnjami od tradicionalnih.*</p> <p>21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.*</p> <p>22. Uporaba metodologije Scrum mi ne povzroča veliko stroškov in napora.*</p> <p>23. Metodologija Scrum razrešuje veliko problemov pri razvoju prog. opreme.*</p> <p>24. Metodologija Scrum postaja standardna metoda v podjetjih za razvoj prog. opreme.*</p>	5
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.**	1
5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	<p>20. Metodologija Scrum omogoča, da brez težav opazujem kako jo uporabljajo druge skupine Scrum.*</p> <p>21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.*</p> <p>24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.*</p>	3
6. Načrtovanje izdaje	/	0
7. Načrtovanje iteracije	24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.*	1
8. Vzdrževanje seznama nalog	24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.*	1
9. Izvajanje dnevnih sestankov	/	0
10. Uporaba diagrama preostalega dela	/	0

11. Dosledno upoštevanje koncepta "done"	17. Metodologija Scrum je veliko boljša od tradicionalnih metodologij.* 21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.** 22. Uporaba metodologije Scrum mi ne povzroča veliko stroškov in napora.*	3
12. Izvajanje sestankov za pregled rezultatov iteracije	24. Metodologija Scrum postaja standardna metodologija v podjetjih za razvoj prog. opreme.*	1
13. Izvajanje retrospektivnih sestankov	/	0
14. Poznavanje aktivnosti posameznih uporabniških zgodb	19. Metodologija Scrum se veliko bolj ujema z mojim načinom dela, vrednotami in izkušnjami od tradicionalnih.* 21. Metodologijo Scrum je moč zelo enostavno preizkusiti v delovnem okolju.*	2

Legenda:

(*) - korelacija je značilna pri stopnji značilnosti 0,05

(**) - korelacija je značilna pri stopnji značilnosti 0,01

4.6.2 Vpliv faktorjev o nalogah

Precej manj kot faktorji o inovaciji, na sprejetje metodologije Scrum v podjetju Comtrade, vplivajo faktorji o nalogah. Kot vidimo iz Tabele 4.18, tudi ti vplivajo na sprejetje oziroma uveljavitev prvih treh elementov (aktivnosti) metodologije Scrum. Pri tem imajo največji vpliv na sprejetje prvega elementa (vzdrževanje seznama zahtev), in sicer dva od skupno treh faktorjev. Poleg prvih treh elementov metodologije Scrum je zgolj še en element, na sprejetje kateraga vpliva kateri izmed faktorjev o nalogah, in sicer dosledno upoštevanje koncepta "done". Rezultati kažejo, da je skladnost metodologije Scrum in potreb posameznika pri razvoju programskih rešitev, faktor, ki najbolj (izmed faktorjev o nalogah) vpliva na sprejetje oziroma uveljavitev metodologije, v podjetju Comtrade. Na osnovi rezultatov lahko tudi sklepamo, da uporaba metodologije Scrum, bolj malo prispeva k večjemu zadovoljstvu naročnikov s končno programsko rešitvijo. Prav tako upo-

raba metodologije Scrum, ne olajša znatno izvedbe zahtevnejših nalog pri razvoj programske opreme, v primerjavi s tradicionalnimi metodologijami.

Tabela 4.18: Korelacije med elementi metodologije Scrum in faktorji oziroma trditvami o nalogah

Element (aktivnost) metodologije Scrum	Faktorji DOI	Število faktorjev DOI
1. Vzdrževanje seznama zahtev	26. Metodologija Scrum pomembno prispeva k večjemu zadovoljstvu naročnikov s končno prog. rešitvijo.* 27. Metodologija Scrum je skladna z mojimi potrebami pri razvoju prog. rešitev.**	2
2. Uporaba uporabniških zgodb	27. Metodologija Scrum je skladna z mojimi potrebami pri razvoju prog. rešitev.**	1
3. Sodelovanje s produktnim vodjo	27. Metodologija Scrum je skladna z mojimi potrebami pri razvoju prog. rešitev.**	1
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	/	0
5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	/	0
6. Načrtovanje izdaje	/	0
7. Načrtovanje iteracije	/	0
8. Vzdrževanje seznama nalog	/	0
9. Izvajanje dnevnih sestankov	/	0
10. Uporaba diagrama preostalega dela	/	0

11. Dosledno upoštevanje koncepta "done"	27. Metodologija Scrum je skladna z mojimi potrebami pri razvoju prog. rešitev.**	1
12. Izvajanje sestankov za pregled rezultatov iteracije	/	0
13. Izvajanje retrospektivnih sestankov	/	0
14. Poznavanje aktivnosti posameznih uporabniških zgodb	/	0

Legenda:

(*) - korelacija je značilna pri stopnji značilnosti 0,05

(**) - korelacija je značilna pri stopnji značilnosti 0,01

4.6.3 Vpliv faktorjev o posamezniku

Najmanj vpliva na sprejetje metodologije Scrum v podjetju Comtrade, imajo od vseh faktorjev DOI faktorji o posamezniku. Vendar pa to ni presenečenje, saj je sprejemanje oziroma uveljavljanje neke inovacije v veliki večini odvisno od odločitve celotnega podjetja (organizacije) in ne posameznikov. Iz Tabele 4.19 vidimo, da le eden izmed faktorjev o posamezniku, in sicer samostojno eksperimentiranje z metodologijo Scrum, vpliva na sprejetje dveh elementov metodologije Scrum. To sta: dosledno upoštevanje koncepta "done" in poznavanje aktivnosti posameznih uporabniških zgodb.

Tabela 4.19: Korelacije med elementi metodologije Scrum in faktorji oziroma trditvami o posamezniku

Element (aktivnost) metodologije Scrum	Faktorji DOI	Število faktorjev DOI
1. Vzdrževanje seznama zahtev	/	0

2. Uporaba uporabniških zgodb	/	0
3. Sodelovanje s produktnim vodjo	/	0
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	/	0
5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	/	0
6. Načrtovanje izdaje	/	0
7. Načrtovanje iteracije	/	0
8. Vzdrževanje seznama nalog	/	0
9. Izvajanje dnevnih sestankov	/	0
10. Uporaba diagrama preostalega dela	/	0
11. Dosledno upoštevanje koncepta "done"	29. Z metodologijo Scrum lahko samostojno eksperimentiram.*	1
12. Izvajanje sestankov za pregled rezultatov iteracije	/	0
13. Izvajanje retrospektivnih sestankov	/	0
14. Poznavanje aktivnosti posameznih uporabniških zgodb	29. Z metodologijo Scrum lahko samostojno eksperimentiram.*	1

Legenda:

(*) - korelacija je značilna pri stopnji značilnosti 0,05

(**) - korelacija je značilna pri stopnji značilnosti 0,01

4.6.4 Vpliv faktorjev o okolju

Nekoliko bolj kot faktorji o nalogah in faktorji o posamezniku, imajo vpliv na sprejetje oziroma uveljavitev metodologije Scrum, faktorji o okolju. Kar pa je pričakovano, saj se le-ti med drugim nanašajo na različne vire (tehnološka infrastruktura, denarna sredstva, itd.), ki so nujno potrebni pri sprejemanju neke inovacije. Kot lahko vidimo iz Tabele 4.20 faktorji o okolju vplivajo, v podjetju Comtrade, na sprejetost petih elementov metodologije Scrum. Pri tem na nobena ne vpliva več kot en faktor. V podpoglavju 4.5 smo že omenili, da so v podjetju Comtrade dobro sprejeti (dobro znani in uporabljeni) sledeči elementi: vzdrževanje seznama zahtev, uporaba uporabniških zgodb in izvajanje dnevnih sestankov. Iz spodnjih rezultatov pa lahko sklepamo, da je na to precej vplivala razpoložljivost vseh potrebnih virov, ki jim je omogočila uspešno in učinkovito uporabo metodologije Scrum. V podjetju Comtrade skupinskega ocenjevanja uporabniških zgodb po metodi Team Estimation Game, ne poznajo dobro in posledično tudi nimajo veliko izkušenj z uporabo le-tega, kot smo to ugotovili v podpoglavju 4.5. Prav tako smo omenili, da je metoda Planning Poker veliko bolj razširjena in predpisana metoda za ocenjevanje uporabniških zgodb, kot pa metoda Team Estimation Game. Če ob tem upoštevamo še spodnje rezultate, bi lahko dejali, da v podjetju Comtrade metodologijo Scrum, razvijalci uporabljajo dokaj skladno s predpisi v literaturi (metodo Planning Poker, ki je bolj razširjena in predpisana v literaturi, bolje poznajo in uporabljajo kot metodo Team Estimation Game). Prvič lahko tudi vidimo negativno korelacijo, in sicer med elementom uporaba diagrama preostalega dela in faktorjem o tehnološki infrastrukturi. V podpoglavju 4.5 so rezultati pokazali, da v podjetju Comtrade dokaj dobro poznajo diagram preostalega dela, vendar pa ga ne uporabljajo oziroma nimajo veliko izkušenj z njegovo uporabo. Na podlagi spodnjih rezultatov lahko ugotovimo, da razlog ni pomanjkanje ustrezne tehnološke infrastrukture.

Tabela 4.20: Korelacije med elementi metodologije Scrum in faktorji oziroma trditvami o okolju

Element (aktivnost) metodologije Scrum	Faktorji DOI	Število faktorjev DOI
1. Vzdrževanje seznama zahtev	36. Na razpolago imamo vse potrebne vire za uspešno in učinkovito uporabo metod. Scrum.*	1
2. Uporaba uporabniških zgodb	36. Na razpolago imamo vse potrebne vire za uspešno in učinkovito uporabo metod. Scrum.*	1
3. Sodelovanje s produktnim vodjo	/	0
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	/	0
5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	35. Razvijalci uporabljamo (-jo) metodologijo Scrum točno tako kot je predpisano v literaturi.*	1
6. Načrtovanje izdaje	/	0
7. Načrtovanje iteracije	/	0
8. Vzdrževanje seznama nalog	/	0
9. Izvajanje dnevnih sestankov	36. Na razpolago imamo vse potrebne vire za uspešno in učinkovito uporabo metod. Scrum.*	1
10. Uporaba diagrama preostalega dela	34. Imamo vso potrebno tehn. infrastrukturo za uporabo metod. Scrum.* [△]	1
11. Dosledno upoštevanje koncepta "done"	/	0
12. Izvajanje sestankov za pregled rezultatov iteracije	/	0
13. Izvajanje retrospektivnih sestankov	/	0

14. Poznavanje aktivnosti posameznih uporabniških zgodb	/	0
---	---	---

Legenda:

- (*) - korelacija je značilna pri stopnji značilnosti 0,05
- (**) - korelacija je značilna pri stopnji značilnosti 0,01
- (Δ) - negativna korelacija

4.6.5 Vpliv faktorjev o organizaciji

Iz Tabele 4.21 lahko vidimo, da faktorji o organizaciji najbolj izmed vseh faktorjev DOI, vplivajo na sprejete in uveljavitev metodologije Scrum. Kot pri analizi prejšnjih faktorjev DOI, lahko tudi tu vidimo, da noben izmed elementov metodologije Scrum ne izstopa močno po številu faktorjev, ki vplivajo na njegovo sprejetje. Na sprejetje in uveljavitev petih elementov metodologije Scrum (3., 4., 8., 13. in 14.) vplivajo po trije faktorji, kar je tudi največje število faktorjev o organizaciji, ki vplivajo na posamezen element metodologije Scrum. Pri tem se v štirih primerih (sodelovanje s produktnim vodjo, skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker, izvajanje retrospektivnih sestankov in poznavanje aktivnosti posameznih uporabniških zgodb) pojavijo isti faktorji, in sicer faktor o izkušnjah (40.), faktor o samostojni uporabi novih tehnologij (44.) in faktor o zahtevi podjetja oziroma managementa (45.). Še posebej zanimiv je slednji faktor (45.), saj lahko vidimo, da je z vsakim elementom metodologije Scrum, kjer je korelacija statistično značilna (pri 8-ih od skupno 14-ih elementov), negativno koreliran oziroma z drugimi besedami povedano, glavni razlog za uporabo več kot polovice elementov metodologije Scrum, ni bila zahteva podjetja. Pomemben faktor o organizaciji, ki vpliva na sprejetje in uveljavitev metodologije Scrum v podjetju Comtrade, so izkušnje z uporabo metodologij razvoja programske opreme, kot lahko razberemo iz Tabele 4.21. In sicer, če se navežemo na podpoglavje 4.5, kjer smo ugotovili, da je izvajanje dnevni sestankov najbolj sprejet element metodologije Scrum, ob tem pa še upoštevamo spodnje rezultate, lahko rečemo, da so na to precej vplivale izkušnje. Verjetno so že pred metodologijo Scrum imeli podobne sestanke. Poleg tega vidimo, da na dosledno upoštevanje koncepta "done" prav

tako vplivajo izkušnje, ki jih imajo z razvojem programske opreme. Oziroma lahko sklepamo, da so tudi pred uporabo metodologije Scrum zahteve naročnikov realizirali v celoti uspešno (v celoti delujoča programska oprema). Faktor, ki izmed vseh faktorjev o organizaciji, najbolj vpliva na sprejetje in uveljavitev metodologije Scrum, je posameznikova samostojna uporaba novih tehnologij (posameznik začne z uporabo nove tehnologije še preden jo začnejo uporabljati ostali), in sicer vpliva na sprejetje desetih elementov.

Tabela 4.21: Korelacije med elementi metodologije Scrum in faktorji oziroma trditvami o organizaciji

Element (aktivnost) metodologije Scrum	Faktorji DOI	Število faktorjev DOI
1. Vzdrževanje seznama zahtev	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj.* 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.*	2
2. Uporaba uporabniških zgodb	44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.**	1
3. Sodelovanje s produktnim vodjo	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj.** 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.* 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam.* [△]	3
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj.* 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.** 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam.** [△]	3

5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	/	0
6. Načrtovanje izdaje	39. Metodologija Scrum omogoča veliko spontane in neformalne komunikacije med člani skupine Scrum. * Δ	1
7. Načrtovanje iteracije	45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam. ** Δ	1
8. Vzdrževanje seznama nalog	43. Vsak dodaten uporabnik metod. Scrum močno poveča koristnost uporabe metod. Scrum. * Δ 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji. ** 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam. ** Δ	3
9. Izvajanje dnevnih sestankov	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj. **	1
10. Uporaba diagrama preostalega dela	44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji. * 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam. ** Δ	2
11. Dosledno upoštevanje koncepta "done"	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj. ** 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji. *	2

12. Izvajanje sestankov za pregled rezultatov iteracije	44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.* 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam.* Δ	2
13. Izvajanje retrospektivnih sestankov	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj.* 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.** 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam.** Δ	3
14. Poznavanje aktivnosti posameznih uporabniških zgodb	40. Z uporabo metodologij razvoja programske opreme imam veliko izkušenj.** 44. Pogosto začnem z uporabo novih tehnologij prej kot moji sodelavci in prijatelji.** 45. Zahteva podjetja, da mora razvoj potekati po metodologiji Scrum, je glavni razlog da jo uporabljam.* Δ	3

Legenda:

(*) - korelacija je značilna pri stopnji značilnosti 0,05

(**) - korelacija je značilna pri stopnji značilnosti 0,01

(Δ) - negativna korelacija

Poglavje 5

Primerjava z rezultati raziskave Overhageja in Schlaudererja

Jedro diplomske naloge zaključujemo s preverjanjem hipotez, ki sta jih v svoji raziskavi dolgoročne sprejemljivosti metodologije Scrum predstavila Sven Overhage in Sebastian Schlauderer [6]. In sicer nas je zanimalo, v kolikšni meri so naši podatki primerljivi z njunimi, oziroma koliko njunih hipotez lahko naši podatki potrdijo. V ta namen smo na koncu vprašalnika dodali 8 dodatnih trditev izmed katerih vsaka predstavlja eno hipotezo Svena Overhagea in Sebastiana Schlaudererja. V nadaljevanju najprej sledi predstavitev njune raziskave in rezultatov, ki sta jih dobila, nato pa še predstavitev naših rezultatov in primerjava.

5.1 Predstavitev raziskave Overhageja in Schlaudererja

Leta 2012 Sven Overhage in Sebastian Schlauderer predstavita raziskavo *Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects*, v kateri sta predvsem raziskovala, ali sprejetje metodologije Scrum v primerjavi z uporabo tradicionalnih metodologij prinaša korist ali pa se njene pomankljivosti izkažejo kot potencialne ovire.

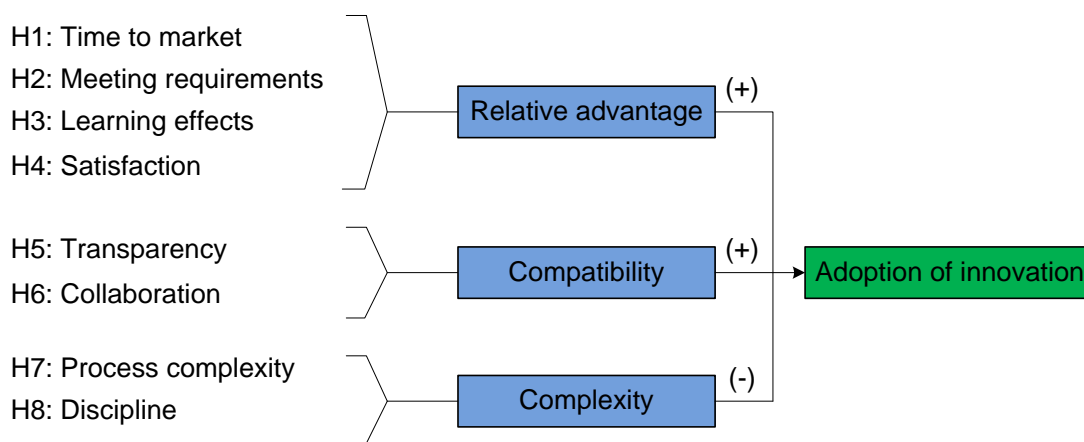
V raziskavi sta preverila osem hipotez, in sicer [6]:

- H1: Razvijalci opažajo hitrejši razvoj programske opreme kot pri tradicionalnih načinih razvoja programske opreme.*
- H2: Razvijalci opažajo, da so zahteve naročnika izpolnjene bolje kot pri tradicionalnih načinih razvoja programske opreme.*
- H3: Razvijalci opažajo, da se pri razvoju rešitve naučijo več kot pri tradicionalnih načinih razvoja programske opreme.*
- H4: Razvijalci so bolj zadovoljni z razvito programsko opremo kot pri tradicionalnih načinih razvoja programske opreme.*
- H5: Razvijalci opažajo boljši pregled nad potekom razvoja programske opreme kot pri tradicionalnih načinih razvoja programske opreme.*
- H6: Razvijalci opažajo boljše medsebojno sodelovanje kot pri tradicionalnih načinih razvoja programske opreme.*
- H7: Razvijalci opažajo nižjo kompleksnost razvojnega procesa kot pri tradicionalnih načinih razvoja programske opreme.*
- H8: Razvijalci opažajo, da je potrebno več discipline kot pri tradicionalnih načinih razvoja programske opreme.*

Omenjene hipoteze služijo za oceno faktorjev sprejemljivosti, ki so po mnenju Overhageja in Schlaudererja odločilni za sprejetje neke inovacije (Slika 5.1) [6]:

- Relativna prednost (angl. *Relative advantage*): stopnja, do katere je inovacija zaznana kot boljša od prejšnjega načina dela.
- Skladnost (angl. *Compatibility*): stopnja, do katere je inovacija zaznana kot skladna z dejanskimi potrebami in preteklimi izkušnjami.
- Kompleksnost (angl. *Complexity*): stopnja, do katere je inovacija zaznana kot zahtevna za uporabo.

Torej prve štiri hipoteze (*H1-H4*) predpostavljajo, da metodologija Scrum prinaša razvijalcem prednost v primerjavi s tradicionalnimi metodologijami. Hipotezi *H5* in *H6* predpostavljata, da je metodologija Scrum bolj združljiva z delovnimi navadami razvijalcev kot pa tradicionalne metodologije. Zadnji dve hipotezi (*H7* in *H8*) pa predstavljata kompromis med nižjo kompleksnostjo razvojnega procesa in potrebo po višji stopnji discipline.



Slika 5.1: Model faktorjev sprejetja [6]

Predstavljene hipoteze sta preverjala v vodilni nemški zavarovalnici, ki se z metodologijo Scrum sreča leta 2007. Podatke za raziskavo sta pridobila leta 2009, ko je v zavarovalnici že velik delež projektov potekal po metodologiji Scrum (okoli 200 razvijalcev naj bi delalo na 23 projektih, ki potekajo po agilnih metodologijah) [6]. In sicer sta jih pridobila z izvedbo t.i. delno-strukturiranega (oziroma pol-strukturiranega) intervjuja s 5 tipičnimi uporabniki.

Podatki, ki sta jih pridobila, so potrdili kar sedem od osmih hipotez, in sicer hipoteze od *H1* do *H6* in hipotezo *H8*. Pri sedmi hipotezi pa so imeli razvijalci, kot sta ugotovila, mešana stališča. In sicer po eni strani so pri metodologiji Scrum opazali manjšo kompleksnost posameznih razvojnih nalog, kot pri tradicionalnih metodologijah. Po drugi strani pa so imeli predvsem težave pri zagonu projektov. Prav tako sta ugotovila, da so se razvijalci vedno težje osredotočali na razvoj nove programske opreme, saj so morali sočasno izboljševati že izdano programsko opremo (zaradi obstoječih napak v implementaciji in spreminjajočih se zahtev

naročnikov). Kot že omenjeno sta potrdila tudi osmo hipotezo, kjer se izkaže, da razvijalci zaznavajo potrebno disciplino, kot slabost, ki negativno vpliva na sprejetje metodologije Scrum.

Na koncu raziskave sta še dodala, da se kljub številnim faktorjem, ki spodbujajo razvijalce k sprejetju metodologije Scrum, lahko pojavijo potencialne ovire, ki bi lahko ogrozile njeno sprejetje.

5.2 Preverjanje hipotez

Tako kot pri analizi ključnih faktorjev smo tudi tokrat preverjanje hipotez opravili z izračuni korelacij med njimi in stopnjo uporabe elementov metodologije Scrum. Pri tem smo se zopet osredotočili na statistično značilnejše, ki pa jih ni veliko, kot lahko vidimo iz spodnje Tabele 5.1. Kljub temu lahko rečemo, da anketiranci opažajo, hitrejši razvoj programske opreme, in sicer na to pomembno vpliva poznavanje in vzdrževanje seznama zahtev. Prav tako opažajo, da metodologija Scrum omogoča razvoj programske opreme, ki bolj ustreza zahtevam naročnika, od tradicionalnih metodologij (tudi tu v korelaciji z vzdrževanjem seznama zahtev). V podjetju Comtrade opažajo tudi boljše sodelovanje programerjev pri razvoju programskih rešitev, na kar vpliva uporaba uporabniških zgodb, kot lahko vidimo. Poleg omenjenih korelacij, sta v tabeli tudi dve negativni korelaciji. In sicer tisti anketiranci, ki so že slišali za načrtovanje izdaje oziroma iteracije in imajo izkušnje z omenjenima aktivnostima menijo, da metodologija Scrum ne zahteva več discipline pri razvoju programske rešitve od tradicionalnih metodologij. Medtem, ko tisti z manj izkušnjami oziroma brez izkušenj, menijo ravno obratno. To se je izkazalo tudi pri korelacijah s preostalimi elementi metodologije Scrum (vse korelacije so bile negativne). Kljub temu, da niso bile statistično značilne (vendar pa dokaj blizu), lahko rečemo, da v podjetju Comtrade ne opažajo potrebe po večji disciplini pri uporabi metodologije Scrum.

Tabela 5.1: Korelacije med elementi metodologije Scrum in trditvami (hipotezami)

Element (aktivnost) metodologije Scrum	Trditve	Število trditvev
1. Vzdrževanje seznama zahtev	46. Metodologija Scrum omogoča hitrejši razvoj programske opreme od tradicionalnih.** 47. Metodologija Scrum omogoča razvoj prog. opreme, ki bolj ustreza zahtevam naročnika, od tradicionalnih.*	2
2. Uporaba uporabniških zgodb	51. Metodologija Scrum omogoča boljše sodelovanje programerjev pri razvoju prog. rešitve od tradicionalnih.*	1
3. Sodelovanje s produktnim vodjo	/	0
4. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker	/	0
5. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game	/	0
6. Načrtovanje izdaje	53. Metodologija Scrum zahteva več discipline pri razvoju prog. rešitve od tradicionalnih.* Δ	1
7. Načrtovanje iteracije	53. Metodologija Scrum zahteva več discipline pri razvoju prog. rešitve od tradicionalnih.* Δ	1
8. Vzdrževanje seznama nalog	/	0
9. Izvajanje dnevnih sestankov	/	0
10. Uporaba diagrama preostalega dela	/	0

11. Dosledno upoštevanje koncepta "done"	/	0
12. Izvajanje sestankov za pregled rezultatov iteracije	/	0
13. Izvajanje retrospektivnih sestankov	/	0
14. Poznavanje aktivnosti posameznih uporabniških zgodb	/	0

Legenda:

- (*) - korelacija je značilna pri stopnji značilnosti 0,05
- (**) - korelacija je značilna pri stopnji značilnosti 0,01
- (Δ) - negativna korelacija

Če primerjamo naše rezultate z rezultati raziskave Svena Overhage in Sebastiana Schlaudererja vidimo, da naši rezultati potrjujejo zgolj tri (*H1*, *H2* in *H6*) od osmih hipotez (ostalih ni mogoče potrditi zaradi mešanih mnenj), medtem ko njihovi rezultati potrjujejo kar sedem od osmih hipotez. Eden izmed razlogov je prav gotovo razširjenost metodologije Scrum v posameznem podjetju. In sicer kot smo že omenili, v nemškem podjetju, ki sta ga analizirala Overhage in Schlauderer, velik delež projektov poteka po metodologiji Scrum. V podjetju Comtrade pa še ni v celoti sprejeta in uporabljena, kot smo to lahko videli v podpoglavju 4.5. Na rezultate je verjetno nekoliko vplival tudi način pridobivanja podatkov. Pomembno razliko lahko vidimo pri zadnji hipotezi (*H8*), kjer sta Overhage in Schlauderer ugotovila, da metodologija Scrum zahteva več discipline od razvijalcev, kot pa tradicionalne metodologije. Naši rezultati pa kažejo ravno obratno (oziroma so odvisni od posameznikovega poznavanja posameznih elementov metodologije Scrum).

Poglavje 6

Zaključek

V diplomski nalogi smo analizirali sprejemljivost metodologije Scrum, ter poskušali prepoznati ključne faktorje, ki vplivajo na njeno sprejetje. Analize smo se lotili s preučevanjem modelov, ki obravnavajo omenjeno problematiko. Pri tem smo se osredotočili na dva modela, in sicer Cooper-Zmudov in Rogersov. Na osnovi omenjenih modelov smo sestavili vprašalnik s katerim smo pridobili podatke za analizo.

Osrednji del analize sta tvorili analiza sprejetosti ključnih aktivnosti (elementov) metodologije Scrum v podjetju Comtrade (do katere stopnje so posamezni elementi sprejeti) in analiza ključnih faktorjev. Prvo smo opravili s pomočjo frekvenčnih tabel, drugo pa z izračuni korelacij. Pri tem smo ugotovili, da so v podjetju Comtrade najbolj sprejeti sestanki, ki se dogajajo pred, med in po posameznih iteracijah. Med njimi pa je najbolj sprejet dnevni sestanek. Prav tako smo ugotovili, da imajo precej izkušenj z uporabo uporabniških zgodb, ter da v veliki meri dosledno upoštevajo koncept "done". Na drugi strani pa smo opazili slabše poznavanje in nižjo stopnjo uporabe metode Team Estimation Game in diagrama preostalega dela (omenjena elementa sta najbolj odstopala izmed elementov, katerih stopnja sprejetosti je nizka). Gledano skozi prizmo celotnih rezultatov smo ocenili, da je sprejetost metodologije Scrum v podjetju Comtrade med četrto in peto stopnjo Cooper-Zmudovega modela.

Pri analizi ključnih faktorjev smo ugotovili, da imajo na sprejetje metodologije Scrum največji vpliv faktorji o organizaciji in faktorji o inovaciji, najmanjši pa faktorji o posamezniku. Tak rezultat smo tudi pričakovali, saj je običajno spreje-

tje neke inovacije v veliki meri odvisno od odločitve podjetja in tega kaj prinaša inovacija. Vendar pa nas je vseeno pozitivno presenetilo dejstvo, da na sprejetost dobre polovice elementov metodologije Scrum v podjetju Comtrade, ni odločilno vplivala zahteva le-tega. Iz tega bi lahko tudi sklepali, da je bila uporaba metodologije Scrum morda deloma zahteva naročnika (stranke).

Za zaključek smo opravili primerjavo dobljenih rezultatov z rezultati podobne študije (raziskave), ki sta jo opravila Overhage in Schlauderer. Iz rezultatov primerjave se lahko deloma potrdi, katere naj bi bile prednosti metodologije Scrum v primerjavi s tradicionalnimi metodologijami razvoja programske opreme (kakšne prednosti naj bi prinesla uporaba metodologije Scrum v primerjavi s tradicionalnimi metodologijami).

Omenili smo, da je metodologija Scrum ena izmed vodilnih agilnih metodologij razvoja programske opreme, ter da na to precej vpliva njena enostavnost. Vendar pa uvedba le-te ni vedno preprosta, saj lahko bistveno spremeni način dela razvijalcev programske opreme. Zatorej lahko podjetjem, ki se odločajo o uvedbi metodologije Scrum, uporaba raziskav kot je naša in pa ostalih podobnih služi kot primer, ki prikazuje kaj vpliva na uspešno uvedbo metodologije Scrum in kje se lahko pojavijo težave.

Analiza v diplomski nalogi bi se lahko nadgradila (izboljšala) s segmentacijo anketirancev glede na njihovo uporabniško vlogo in leta izkušenj z uporabo metodologije Scrum. Le-te žal nismo uspeli narediti na naših podatkih, saj je pri obeh vprašanjih (uporabniška vloga in izkušnje z uporabo metodologije Scrum) precej izstopal en odgovor. Ena izmed možnih izboljšav bi lahko bila tudi večkratna izvedba anketnega vprašalnika, s čimer bi se lahko analizirala uporaba metodologije Scrum skozi neko časovno obdobje. Oziroma bi se pri tem lahko uporabil kakšen drug način pridobivanja podatkov (npr. intervju) ali pa kombinacija večih načinov.

Literatura

- [1] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, 2004
- [2] V. Mahnič, *Tehnologija programske opreme in agilne metode (Prosojnice)*, Fakulteta za računalništvo in informatiko, 2014
- [3] M. Cohn, *Agile Estimating and Planning*, Prentice Hall, 2006
- [4] E. M. Rogers, *Diffusion of Innovations, Fourth Edition*, The Free Press, 1995
- [5] L. Rising, N. S. Janoff, *The Scrum Software Development Process for Small Teams*, revija IEEE Software, letnik 17 št. 4, str. 26-32, julij 2000
- [6] S. Overhage, S. Schlauderer, *Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects*, v zborniku 45th Hawaii International Conference on System Sciences, str. 5452-5461, 2012
- [7] E. M. Ollila, K. Lyytinen, *Why organizations adopt information system process innovations: a longitudinal study using Diffusion of Innovation theory*, Information Systems Journal, letnik 13 št. 3, str. 275-297, julij 2003
- [8] The Agile Manifesto, dostopno na:
<http://www.agilealliance.org/the-alliance/the-agile-manifesto/>
- [9] Scrum Project Management as a Simple Rules Framework, dostopno na: <http://www.leadingagile.com/2008/03/scrum-project-management-as-a-simple-rules-framework/>
- [10] The Scrum Guide, dostopno na: <https://www.scrumguides.org>

- [11] Scrum (software development), dostopno na:
[https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))
- [12] Whitepaper: Kanban vs Scrum, dostopno na:
<http://www.belatrixsf.com/index.php/kanban-vs-scrum>
- [13] Team Estimation, dostopno na:
<http://www.netobjectives.com/files/team-estimation-game.pdf>
- [14] Benefits & Pitfalls of using Scrum Software Development Methodology,
dostopno na: <http://blog.belatrixsf.com/benefits-pitfalls-of-using-scrum-software-development-methodology/>
- [15] Where are You on the Recruitment Technology Curve ?, dostopno na:
<http://beforeitsnews.com/science-and-technology/2014/12/oldie-but-goodie-where-are-you-on-the-recruitment-technology-curve-2736640.html>
- [16] Normality Testing - Skewness and Kurtosis, dostopno na:
<https://help.gooddata.com/display/doc/Normality+Testing+-+Skewness+and+Kurtosis>

Dodatek A

Vprašalnik

Vloga na projektih in dosedanje izkušnje z metodologijo Scrum

1. V kakšni vlogi ste nastopali v projektih, ki so potekali oziroma še potekajo po metodologiji Scrum ? V kolikor ste nastopali v več vlogah, izberite tisto, ki vam je najbolj ustrezala.

- Produktni vodja (angl. *Product Owner*)
- Skrbnik metodologije (angl. *Scrum Master*)
- Član razvojne skupine (angl. *Team Member*)
- Drugo

2. Koliko izkušenj že imate z uporabo metodologije Scrum ?

- nimam še izkušenj z uporabo metodologije Scrum
- manj kot 1 leto
- 1 do 3 leta
- 3 do 5 let
- več kot 5 let

Vprašanja o stopnji uporabe ključnih aktivnosti Scruma

3. Vzdrževanje seznama zahtev (angl. *Product Backlog*)

Seznam zahtev je gonilo razvoja. Seznam zahtev ni nikoli dokončen in se ves čas projekta stalno dopolnjuje.

- 1 - za vzdrževanje seznama zahtev slišim prvič
- 2 - seznama zahtev še nisem vzdrževal
- 3 - imam dovolj potrebnih znanj, da lahko vzdržujem seznam zahtev
- 4 - z vzdrževanjem seznama zahtev imam že pozitivne izkušnje
- 5 - seznam zahtev sem vzdrževal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - vzdrževanje seznama zahtev izvajam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

4. Uporaba uporabniških zgodb (angl. *User Stories*) za predstavitev zahtevane funkcionalnosti

Namesto podrobne specifikacije zahtev, ki lahko »ubije« projekt, zadostuje kratek zapis zahtevane funkcionalnosti, ki služi kot opomnik, da vemo, o čem se je treba pogovarjati. Oblikovati je treba primerno velike uporabniške zgodbe, ki predstavljajo osnovo za ocenjevanje zahtevnosti in planiranje projekta.

- 1 - za uporabo uporabniških zgodb za predstavitev zahtevane funkcionalnosti slišim prvič
- 2 - uporabniških zgodb za predstavitev zahtevane funkcionalnosti nisem še uporabljal
- 3 - imam dovolj potrebnih znanj, da lahko uporabljam uporabniške zgodbe za predstavitev zahtevane funkcionalnosti

- 4 - z uporabo uporabniških zgodb za predstavitev zahtevane funkcionalnosti imam že pozitivne izkušnje
- 5 - uporabniške zgodbe za predstavitev zahtevane funkcionalnosti sem uporabljal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - uporabniške zgodbe za predstavitev zahtevane funkcionalnosti uporabljam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

5. Sodelovanje s produktnim vodjo pri razčiščevanju podrobnosti uporabniških zgodb

Podrobnosti, povezanih z realizacijo posameznih uporabniških zgodb, ne dokumentiramo, ampak jih razčiščujemo v pogovorih s produktnim vodjo.

- 1 - za sodelovanje s produktnim vodjo slišim prvič
- 2 - s produktnim vodjo še nisem sodeloval
- 3 - imam dovolj potrebnih znanj, da sodelujem s produktnim vodjo
- 4 - z sodelovanjem s produktnim vodjo imam že pozitivne izkušnje
- 5 - s produktnim vodjo sem sodeloval že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - s produktnim vodjo sodelujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

6. Skupinsko ocenjevanje uporabniških zgodb po metodi Planning Poker

Metoda zagotavlja enakopravno sodelovanje vseh razvijalcev in onemogoča prevlado posameznikov.

- 1 - za ocenjevanje uporabniških zgodb po metodi Planning Poker slišim prvič

- 2 - uporabniških zgodb po metodi Planning Poker še nisem ocenjeval
- 3 - imam dovolj potrebnih znanj, da lahko ocenjujem uporabniške zgodbe po metodi Planning Poker
- 4 - z ocenjevanjem uporabniških zgodb po metodi Planning Poker imam že pozitivne izkušnje
- 5 - uporabniške zgodbe po metodi Planning Poker sem ocenjeval že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - uporabniške zgodbe po metodi Planning Poker ocenjujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

7. Skupinsko ocenjevanje uporabniških zgodb po metodi Team Estimation Game

Tako kot metoda Planning Poker, tudi metoda Team Estimation Game zagotavlja enakopravno sodelovanje vseh razvijalcev in onemogoča prevlado posameznikov.

- 1 - za ocenjevanje uporabniških zgodb po metodi Team Estimation Game slišim prvič
- 2 - uporabniških zgodb po metodi Team Estimation Game še nisem ocenjeval
- 3 - imam dovolj potrebnih znanj, da lahko ocenjujem uporabniške zgodbe po metodi Team Estimation Game
- 4 - z ocenjevanjem uporabniških zgodb po metodi Team Estimation Game imam že pozitivne izkušnje
- 5 - uporabniške zgodbe po Team Estimation Game sem ocenjeval že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - uporabniške zgodbe po metodi Team Estimation Game ocenjujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

8. Načrtovanje izdaje (angl. *Release Planning*) na osnovi ocenjene hitrosti razvojne skupine

Uporabniške zgodbe razporedimo po iteracijah glede na njihovo prioriteto. Seštevek točk vseh uporabniških zgodb v vsaki iteraciji mora biti enak ocenjeni hitrosti razvojne skupine. Pomembno je, da razvojna skupina čim bolj točno oceni svojo hitrost.

- 1 - za načrtovanje izdaje slišim prvič
- 2 - izdaje še nisem načrtoval
- 3 - imam dovolj potrebnih znanj, da lahko načrtujem izdaje
- 4 - z načrtovanjem izdaje imam že pozitivne izkušnje
- 5 - izdajo sem načrtoval že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - izdajo načrtujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

9. Načrtovanje vsebine naslednje iteracije na sestanku za načrtovanje iteracije (angl. *Sprint Planning Meeting*)

Načrtovanje naslednje iteracije zahteva prilagoditev ocene za hitrost (angl. *Velocity*), določitev uporabniških zgodb (ki jih bomo realizirali), razdelitev zgodb na naloge, določitev izvajalcev za posamezne naloge, ocenjevanje zahtevnosti nalog.

- 1 - za načrtovanje vsebine naslednje iteracije slišim prvič
- 2 - vsebino naslednje iteracije še nisem načrtoval
- 3 - imam dovolj potrebnih znanj, da lahko načrtujem vsebine naslednje iteracije
- 4 - z načrtovanjem vsebine naslednje iteracije imam že pozitivne izkušnje
- 5 - vsebino naslednje iteracije sem načrtoval že tolikokrat, da je to zame postalo rutinsko opravilo

- 6 - ivsebino naslednje iteracije načrtujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

10. Vzdrževanje seznama nalog (angl. *Sprint Backlog*)

Med izvajanjem iteracije je potrebno dodajanje in (po potrebi) podrobnejše razčlenjevanje posameznih nalog ter beleženje količine vložene in preostalega dela.

- 1 - za vzdrževanje seznama nalog slišim prvič
- 2 - seznama nalog še nisem vzdrževal
- 3 - imam dovolj potrebnih znanj, da lahko vzdržujem seznam nalog
- 4 - z vzdrževanjem seznama nalog imam že pozitivne izkušnje
- 5 - seznam nalog sem vzdrževal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - seznam nalog vzdržujem rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

11. Izvajanje dnevnih sestankov (angl. *Daily Scrum*)

Ti sestanki omogočajo sproten vpogled v potek projekta in takojšnje ukrepanje v primeru težav.

- 1 - za izvajanje dnevnih sestankov slišim prvič
- 2 - dnevnih sestankov še nisem izvajal
- 3 - imam dovolj potrebnih znanj, da lahko izvajam dnevne sestanke
- 4 - z izvajanjem dnevnih sestankov imam že pozitivne izkušnje
- 5 - dnevne sestanke sem izvajal že tolikokrat, da je to zame postalo rutinsko opravilo

- 6 - dnevne sestanke izvajam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

12. Uporaba diagrama preostalega dela za nadzor poteka dela

Diagram prikazuje, kako se zmanjšuje količina neopravljenega dela. Uporablja se na ravni projekta kot celote (angl. *Release Burndown*) in na ravni iteracije (angl. *Sprint Burndown*). Razberemo lahko, kdaj bo projekt končan, oziroma ali bo delo opravljeno do konca iteracije.

- 1 - za uporabo diagrama preostalega dela slišim prvič
- 2 - diagrama preostalega dela še nisem uporabljal
- 3 - imam dovolj potrebnih znanj, da lahko uporabljam diagram preostalega dela
- 4 - z uporabo diagrama preostalega dela imam že pozitivne izkušnje
- 5 - diagram preostalega dela sem uporabljal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - diagram preostalega dela uporabljam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

13. Dosledno upoštevanje koncepta "done"

Vsaka uporabniška zgodba mora biti v celoti končana, da jo ob koncu iteracije lahko damo v uporabo (Rezultat vsake iteracije mora biti "shippable functionality").

- 1 - za dosledno upoštevanje koncepta "done" slišim prvič
- 2 - koncepta "done" še nisem dosledno upošteval
- 3 - imam dovolj potrebnih znanj, da lahko dosledno upoštevam koncept "done"
- 4 - z doslednim upoštevanjem koncepta "done" imam že pozitivne izkušnje

- 5 - koncept "done" sem dosledno upošteval že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - koncept "done" dosledno upoštevam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

14. Izvajanje sestankov za pregled rezultatov iteracije (angl. *Sprint Review Meeting*)

Predstavitev nove funkcionalnosti (rezultatov zadnje iteracije) naročniku.

- 1 - za izvajanje sestankov za pregled rezultatov sestankov slišim prvič
- 2 - sestankov za pregled rezultatov še nisem izvajal
- 3 - imam dovolj potrebnih znanj, da lahko izvajam sestanke za pregled rezultatov
- 4 - z izvajanjem sestankov za pregled rezultatov imam že pozitivne izkušnje
- 5 - sestanke za pregled rezultatov sem izvajal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - sestanke za pregled rezultatov izvajam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

15. Izvajanje retrospektivnih sestankov (angl. *Sprint Retrospective Meeting*)

Analiza dobrih in slabih strani dela v pretekli iteraciji. Dogovor o izboljšavah za naprej.

- 1 - za izvajanje retrospektivnih sestankov slišim prvič
- 2 - retrospektivnih sestankov še nisem izvajal
- 3 - imam dovolj potrebnih znanj, da lahko izvajam retrospektivne sestanke

- 4 - z izvajanjem retrospektivnih sestankov imam že pozitivne izkušnje
- 5 - retrospektivne sestanke sem izvajal že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - retrospektivne sestanke izvajam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

16. Poznavanje aktivnosti, ki jih izvajajo posamezne uporabniške vloge (produktni vodja, skrbnik metodologije, razvojna skupina)

Produktni vodja kot nosilec vizije, skrbnik metodologije kot skrbnik, ki skrbi da delo poteka nemoteno in razvojna skupina, ki deluje po načelu samoorganizacije in kolektivno odgovarja za realizacijo.

- 1 - za aktivnosti posameznih uporabniških vlog slišim prvič
- 2 - z aktivnostmi posameznih uporabniških vlog se še nisem soočil
- 3 - imam dovolj potrebnih znanj, da poznam aktivnosti posameznih uporabniških vlog
- 4 - z aktivnostmi posameznih uporabniških vlog imam že pozitivne izkušnje
- 5 - z aktivnostmi posameznih uporabniških vlog sem se soočil že tolikokrat, da je to zame postalo rutinsko opravilo
- 6 - z aktivnostmi uporabniških vlog se soočam rutinsko, učinkovito in celovito, kar olajša izvedbo ostalih ključnih aktivnosti Scruma in pohitri ter izboljša kakovost razvoja programske rešitve

50. Scrum metodologija omogoča boljši pregled nad potekom razvoja programske rešitve od tradicionalnih načinov razvoja programske opreme z vnaprejšnjim načrtovanjem (RUP, UML...).	○	○	○	○	○	○	○
51. Scrum metodologija omogoča boljše sodelovanje programerjev pri razvoju programske rešitve od tradicionalnih načinov razvoja programske opreme z vnaprejšnjim načrtovanjem (RUP, UML...).	○	○	○	○	○	○	○
52. Scrum metodologija omogoča nižjo kompleksnost procesa razvoja programske rešitve od tradicionalnih načinov razvoja programske opreme z vnaprejšnjim načrtovanjem (RUP, UML...).	○	○	○	○	○	○	○
53. Scrum metodologija zahteva več discipline pri razvoju programske rešitve od tradicionalnih načinov razvoja programske opreme z vnaprejšnjim načrtovanjem (RUP, UML...).	○	○	○	○	○	○	○