**UNAPREĐENJE PREDUZETNIČKE ORGANIZACIJE PRIMENOM SCRUM METODOLOGIJE**

Sadržaj

Sadržaj 2

1 UVOD 5

2 AGILE METODOLOGIJA 6

2.1 Opšti pojmovi 6

2.2 Preduslovi i preporuke za uspešnu primenu 7

2.3 Vrednosti metode 8

2.4 Vrste Agile pristupa 8

2.5 Agile versus Waterfall metoda 9

3 SCRUM METODOLOGIJA 12

3.1 Poreklo i definicija metodologije 12

3.2 Vrednosti 13

4 SCRUM model 17

4.1 Uloge 17

4.1.1 Vlasnik proizvoda 17

4.1.2 Scrum Master 18

4.1.3 Tim za razvoj 20

4.1.4 Menadžment 20

4.1.5 Menadžer promena 21

4.1.6 Scrum konsultant 21

4.1.7 Pile i prase 22

4.2 Artefakti 23

4.2.1 Korak u razvoju proizvoda (Product Increment) 23

4.2.2 Skladište proizvoda (Product Backlog) 23

4.2.3 Skladište Sprintova (Sprint Backlog) 26

4.3 Vremenska osnova 30

4.3.1 Sprint 30

4.3.2 Planiranje Sprint-a - Sprint Planning 33

4.3.3 Dnevni sastanak – Daily scrum meeting 33

4.3.4 Sastanak za pregled – Sprint review meeting 33

4.3.5 Retrospektiva – Sprint Retrospective 33

4.3.6 Planiranje lansiranja proizvoda – Release Planning 34

5 IMPLEMENTACIJA SCRUM METODE U PREDUZEĆU 35

5.1 Pristupi 35

5.1.1 Top-Down pristup 35

5.1.2 Bottom-up pristup 36

5.1.3 Podmornica 36

5.2 Oblici 37

5.2.1 Scrum PRN 37

5.2.2 Façade Scrum 37

5.2.3 Scrum Softver Studio 38

5.2.4 Virtualni Scrum Softver Studio 39

5.2.5 Temeljni Scrum 39

5.2.6 Održivi temeljni Scrum 40

5.2.7 Povezanost ciljeva i različitih oblika Scrum metode 40

5.3 Metode 41

5.3.1 Sastanak za procene 41

5.3.2 Poker – igra planiranja 41

5.3.3 Poker – igra planiranja kroz apsolutne vrednosti 42

5.3.4 Vremenska kutija – Timebox 42

5.3.5 Brzina 42

5.4 Finansijski aspekti 44

5.4.1 Troškovi projekta 44

5.4.2 Investicije 44

5.4.3 Nadgledanje performansi projekta 47

5.5 Dokumentacija 50

5.6 Testiranje 51

5.7 Izazovi i rizici primene Scrum metode 55

5.7.1 Promene i izazovi 55

5.7.2 Rizici 56

6 PRAKTIČNA PRIMENA SCRUM METODE U START - UP PROJEKTNO ORIJENTISANIM ORGANIZACIJAMA 58

6.1 Preduzetnička organizacija 58

6.2 Osvrt na metode projektnog organizovanja na našim prostorima 58

6.3 Primer primene Scrum metode u Start-Up organizaciji 60

6.4 Primena Scrum metode kroz Jira Agile software 62

7 ZAKLJUČAK 64

8 LITERATURA 65

**Slike**

Slika 1 Uporedni prikaz Waterfall i Agile metodologije 11

Slika 2 Scrum: Podela zaposlenih u timove 12

Slika 3 Scrum: Podela posla 13

Slika 4 Scrum: Podela vremena 13

Slika 5 Product Backlog 24

Slika 6 Release burndown chart 26

Slika 7 Sprint Backlog 27

Slika 8 Sprint Burndown chart 27

Slika 9 Sumaran prikaz artefakata 28

Slika 10 Vremenski okvir Scrum metode 30

Slika 11 MVP koncept 31

Slika 12 Odnos vrednosti i vremena, troškova i karakteristika proizvoda 32

Slika 13 Funkcija očikivane brzine tokom Sprint-ova 43

Slika 14 Prelomna tačka 45

Slika 15 Indeks troškovnih performansi 48

Slika 16 Indeks peformansi rasporeda 49

Slika 17 Dokumentacija kroz Sprint-ove 50

Slika 18 Scenario 1 52

Slika 19 Scenario 2 53

Slika 20 Scenario 3 54

**Tabele**

Tabela 1 Vrednosti i antiteze SCRUM metode 14

Tabela 2 Meke veštine Scrum mastera 19

Tabela 3 dodatna Scrum dokumentacija 51

Tabela 4 Hijerarhijska struktura 59

# UVOD

Kontinuirana potreba za poboljšanjem procesa i aktivnosti u svim organizacijama ne zaobilazi ni preduzetničke organizacije. Način na koje one funkcionišu je obično poznat samo zaposlenima u istim, I gotovo nikada se ne može unificirati, i predstaviti na jedinstven način. Poenta je u fluidnosti i fleksibilnosti. No, ne treba zanemariti činjenicu da i ovde postoji potreba za ustaljenim metodama organizovanja i realizacije aktivnosti. Stoga, pristupa se istraživanju u Scrum metode, u cilju unapređenja procesa razvoja proizvoda u Start-up okruženju.

Problem koji se razmatra se bazira na upotrebi tradicionalnih metoda upravljanja projektima i proizodima, i naglašavaju se određeni nedostaci, i nedoslednosti. Navedeno usporava sam proces funkcionisanja organizacije, i zadovoljavanje potreba klijenata ,što bitno utiče na krajnji uspeh. Scrum metoda se ovde javlja u ulozi oponenta, naglašavajući svoje prednosti i generalno, kreativan I sadržajan pristup pomenutom problemu. Stoga, cilj rada je da čitaocima, profesorima I studentima, približi primenu napomenute metode, u oblasti upravljanja razvojem proizvoda, I to u oblasti informaciono-komunikacionih tehnologija. Preciznije, reč je o razvoju softvera, uz deo razmatranja vezan za ostale oblasti poslovanja.

Prvi deo rada ima za cilj da čitaocu predstavi metodologiju koja je okrilje Scrum metode: Agile metodologiju. Pored osnovnih pojmova I principa, biće predstavljeni preduslovi, preporuke za primenu, pristupi, kao i tradicionalna Waterfall metodologija upravljanja projektima.

U drugom delu rada sledi pregled osnova i načela Scrum metode, uz vrednosti iste. Čak I ako nikada do sada nisu čuli za Scrum, čitaoci će vrlo brzo shvatiti principe iste.

Sledeći deo rada se nadovezuje na prethodni, uz prikaz Scrum modela koji se sastoji od uloga, artefakata i vremenske osnove same metode. Ovde nailazimo na onu kreativnu komponentu koja odlikuje samo Scrum, uz korisne smernice nephodne samom pristupu razvoja proizvoda,

Impelentacija same metode je tema sledećeg dela rada, te se razmatraju praktični aspekti primene iste. Razmatraju se oblici metode, pristupi, finansijski aspketi, rizici I slične prateće komponente.

Završni deo rada je težište razmtranja problema, gde se kroz uvid u konkretno preduzetničko okruženje i konkretne primere, kao i primenu softverskog alata, stiče pojam o vrednosti same metode.

Lična motivacija za proučavanje ove teme dolazi iz praktičnog iskustva u radu sa ovom metodom I pratećim softverskim alatom, te postoji velika želja da se podeli pozitivno lično iskustvo sa čitaocima. Naizgled komplikovana, ova metoda sa lakoćom može doprineti mnogim segmentima organizacije, a zaposlenima učiniti posao zabavnim I kreativnim. Tako posmatrano nije sve u samom cilju, poenta je živeti put do cilja

# AGILE METODOLOGIJA

## Opšti pojmovi

Uspeh u poslovanju preduzetničke organizacije podrazumeva kontinuelni rad na unapređenju svakog segmenta funkcionisanja iste, i motivaciju za neprestanim uvođenjem inovacija i poboljšanja. Veliki doprinos pomenutim potrebama daje Agile metodologija, koja sa svojim konceptom pruža mogućnost efikasnijeg organizovanja određenih procesa i proizvoda koji predstavljaju suštinu poslovanja. Ovde dolazi do sukobljavanja tradicionalne definicje uspeha nekog projekta, i inovativne, koju zastupa Agile metodologija. Naime, uspeh nije definisan samo vremenskim kriterijumima, neophodnim budžetom i specifikacijom prozvoda, i poštovanje istih ne znači nužno kvalitet, što će se videti dalje kroz rad.

Pojam "agilan"se prvi put pominje u proklamaciji *Agile Manifesto*, i označava felksibilnost i mobilnost. (Agile Alliance, 2015)

Neke od definicija Agile metodologije su:

„Agile podrazumeva skup metoda i metodologija koje pomažu da vaš tim misli efektivnije, radi efikasnije, i donosi bolje odluke.“(Stellman & Greene, 2014)

„Agile metodologija je alternativa tradicionalnom upravljanju projektima, koja se obično koristi u razvoju softvera. Pomaže timu da reaguje na nepredvidivosti kroz postepeni, iterativni rad u kadencama, koje su poznate kao sprint.“ (Anonymos, 2008)

Oblast ove metodologije podrazumeva tradicionalni softverski inženjering, uključujući upravljanje projektima, kao i poboljšanje organizacionih procesa, o čemu se i govori u ovom radu. Koncept je podržan praktičnim segmentom koji je pojednostavljen i optimizovan tako da postoji mogućnost lakog usvajanja. Ova metodologija je i specifičan *način razmišljanja*, jer može napraviti veliki pomak u funkcionisanju timova sastavljenih od zaposlenih. Ovaj način razmišljanja pomaže ljudima da razmenjuju informacije jedni sa drugima, tako da oni mogu zajedno donositi važne odluke vezane za procese i projekte, umesto menadžera koji bi iste donosio sam. Razlika u odnosu na tradicionalan način organizovanja je što u ovom pristupu svi imaju uvid u sve informacije, i svako ima pravo na mišljenje i iskazivanje stava o načinu funkcionisanja određenog procesa ili razvoja proizvoda. (Stellman & Greene, 2014)

Uspeh koji se može postići primenom ove metodologije se grana na organizacioni, tehnički i lični uspeh.*Organizacioni uspeh* se postiže fokusiranjem na isporuku željene vrednosti i smanjenje troškova, što direktno dovodi do povećanog povraćaja na investicije. Ova metodologija postavlja određena očekivanja na početku projekta, tako da ako projekat pretenduje da dostigne organizacioni uspeh, saznaće se dovoljno rano kako bi se prekinuo, pre nego što organizacija potroši mnogo novca. Najvredniji zahtevi i karakteristike projekta će biti razvijeni na početku, što će povećava vrednost istog, a ako dođe do promena zahteva timovi će biti u mogućnosti da ih ispoštuju, pa čak i predvide njihovo nastajanje. Takođe, zbog tehničke izvrsnosti, doćiće do smanjenja troškova jer se greške svode na minimum, i sprečavaju se prekidi i zamena projekta jednostavnijim.

*Tehnički uspeh* je omogućen zajedničkim radom svih članova tima, što pomaže praćenju odvijanja procesa i kontrole kvaliteta istog. Fokus je na završetku svake iteracije u potpunosti, pre nastupanja naredne.

*Lični uspeh* je individualna stvar, i ne postoji metodologija koja bi ispunila sve zahteve jedne osobe na tom planu, ali svaka pozicija u organizacionoj hijerarhiji nalazi nešto za sebe. Sledi prikaz određenih benefiti, na primeru organizacije koja se bavi razvojem softvera: (Shore, 2007)

* Rukovodioci i viši menadžment - oni će ceniti fokus tima koji radi na optimalnom povraćaju uloženih sredstava i dugovečnosti razvijenog softvera.
* Korisnici, stejkholderi, stručnjaci, i menadžeri proizvoda - oni će ceniti njihovu sposobnost da utiču na pravac razvoja softvera, posvećenost tima razvoju korisnog i vrednog softvera, i povećanoj učestalosti isporuke.
* Menadžeri proizvoda i projekta - oni će visoko vrednovati njihovu sposobnost da vršepromenu pravca poslovanja u pravcu dominatnih zahteva, kao i sposobnost uticaja na ispunjenje obaveza tima i zadovoljstvo stejkholdera.
* Programeri- ceniće poboljšan kvalitet života usled povećanog tehničkog kvaliteta , veći uticaj na procenei rasporeda, i autonomiju tima.
* Zaposleni zaduženi za testiranje - biće zadovoljni statusom prvoklasnog člana tima, sposobnošću da utiču nakvalitet u svim fazama projekta, i više izazovan, manje repetitivan rad.

## Preduslovi i preporuke za uspešnu primenu

Kako bi se omogućila kvalitetna i nesmetana primena metodologije, neophodni su određeni preduslovi, koji imaju sledeći izgled:(Shore, 2007)

1. Podrška upravljačke strukture-upravljačka struktura i njena posvećenost procesu primene su ključni, i to u domenu odobrenja resursne i logističke podrške, autoriteta, delegiranja i koordinacije svih učesnika. Tu je i domen kompenzacije, međuljdskih odnosa, kao i motivacije.
2. Saglasnost zaposlenih- ako grupa zaposlenih koja čini tim za uvođenje i primenu metodologije nije saglasna sa istom, sam proces je osuđen na na neuspeh, jer prisilan rad ne vodi dobrim rezultatima. Dakle, neophodno je stvoriti odnos razumevanja i podrške, kroz upoznavanje sa konceptom.
3. Tim koji "sedi u istoj sobi"- kako bi se postigla dobra komunikacija i sagledali svi zahtevi određenog projekta ili proizvoda, neophodno je da tim bude konzistentan, i to počev od fizičke blizine i efektivne interakcije, jer neprisustvo samo jednog člana otežava proces primene.
4. Klijenti "na licu mesta"- kada u pitanju npr. razvoj softvera, mišljenje krajnjih korisnika je od velikog značaja, jer oni iznose svoje zahteve, i time određuju vrednost softvera. Dakle, česta komunikacija i klijentove revizije su od esencijalnog značaja.
5. Prava veličina tima-ako se u obzir uzme prethodni primer, maksimalna veličina podrazumeva 20 ljudi.
6. Korišćenje celokupnog znanja i prakse koje metoda nudi- ne propuštati i ne zanemarivati nijednu komponentu koncepta, ključ je u holističkom pristupu.

Autori Agilnog Manifesta se zalažu za 12 sledećih principa, uzimajući u obzir razvoj softvera: (Agile Alliance, 2015)

1. Zadovoljan klijent je naš vrhunski prioritet, koji ostvarujemo blagovremenom i kontinuiranom isporukom vrhunskog softvera.
2. Spremno prihvatamo promene zahteva, čak i u kasnoj fazi razvoja. Agilni procesi omogućavaju

uspešno prilagođavanje izmenjenim zahtevima što za rezultat ima prednost naših klijenata u odnosu na konkurenciju.

1. Redovno isporučujemo primenljiv softver, u periodu od nekoliko nedelja do nekoliko meseci, dajući prednost kraćim intervalima.
2. Poslovni ljudi i developeri svakodnevno da sarađuju u toku celokupnog trajanja projekta.
3. Projekte ostvarujemo uz pomoć motivisanih pojedinaca. Obezbeđujemo im ambijent i podršku koja im je potrebna i prepuštamo im posao s poverenjem.
4. Za najproduktivniji i najefikasniji metod prenosa informacije do i unutar razvojnog tima smatramo kontakt licem u lice.
5. Primenljiv softver je osnovno merilo napretka.
6. Procesi promovišu održivi razvoj. Pokrovitelji, developeri i korisnici moraju biti u stanju da kontinuirano rade usklađenim tempom, nezavisno od perioda trajanja projekta.
7. Stalna posvećenost vrhunskom tehničkom kvalitetu i dobar dizajn pospešuju agilnost.
8. Jednostavnost – veština dovođenja do najvišeg stepena količine rada koji nije potrebno uraditi – je od suštinske važnosti.
9. Najbolje arhitekture, zahtevi i dizajn, rezultat su rada samo–organizovanih timova.
10. Timovi u redovnim intervalima razmatraju načine kako da postanu efikasniji, zatim se usklađuju i na osnovu tih zaključaka prilagođavaju dalje postupke.

## Vrednosti metode

Vrednosti koje donosi metodologija su izložene u delu koje predstavlja i manifest iste, i predstavljaju esenciju pristupa. Deo manifesta koji govori o vrednostima, ima sledeći izgled:(Agile Alliance, 2015)

„Otkrivamo bolje načine razvoja softvera razvijajući softver sami i pomažući drugima pri njegovom razvijanju. Kroz taj rad naučili smo da više vrednujemo:

* pojedince i interakcije od procesa i alata,
* primenljiv softver od detaljne dokumentacije,
* saradnju sa klijentima od ugovornih aranžmana,
* reakciju na promenu od pridržavanja plana.

Drugim rečima, iako cenimo značaj činilaca predstavljenih na desnoj strani, stavke prikazane na levoj strani vrednujemo više.

## Vrste Agile pristupa

Kako bi se stekao potpuni uvid u ovu metodologiju, potrebno je navesti vrste pristupa koje ona obuhvata. Pristupi su sledeći: (McLaughlin, 2015)

* Scrum-koji predstavlja oblast ovog rada, i čiji ćekoncept biti izložen u narednim poglavljima.
* Lean i Kanban razvojna metodologija – osnovni principi su:
  + eliminacija otpada,
  + unapredjenje procesa ucenja,
  + odlucivanje sto kasnije tokom procesa,
  + brza isporuka,
  + osposobljavanje tima,
  + izgradnja integriteta,
  + holisticki pristup.

Primenom Lean-a stavlja se naglasak na brzinu i efikasnost razvoja toka procesa, a počiva na brzoj i pouzdanoj dvosmernoj komunikaciji izmedju programera i korisnika programa.

Kanban pristup se bazira se na tri principa:

* + vizualizacija rada na dnevnoj bazi (šta je urađeno danas),
  + ograničenje količine rada koji je u toku (sprečava nastanak situacije u kojoj tim preuzima preveliki posao na sebe),
  + povećanje toka procesa (kada je nešto završeno, prvi sledeći zadatak po prioritetu se uzima iz backloga i stavlja u rad).

Kanban ohrabruje timski rad i kolaboraciju uz aktivno učenje i determinisanje optimalnog toka procesa.

* Extreme Programming (XP) - najpoznatija ali i najkontroverznija Agile metoda. Predstavlja sveukupan pristup brzoj i kontinuiranoj isporuci softvera visokog kvaliteta. Potencira visok procenat učesća korisnika u razvoju, brže petlje povratnih informacija, kontinualno testiranje, planiranje i posvećen timski rad na isporuci vrednosti funkcionalnog softvera u intervalima od jedne do tri nedelje, dakle veoma frekventno.

Pored ovih metoda, tu su još i Crystal, Dynamic Systems Development Method (DSDM) I Feature-Driven Development (FDD) i sl.

## Agile versus Waterfall metoda

Kako bi se istakla inovativnost Agile metodologije u u odonosu na tradicionalni pristup organizovanju iupravljanju projektima u organizaciji, tzv. Waterfall metodi, potrebno je izvršiti komparaciju istih.

Metodologija Waterfall je sekvencijalni proces, i sastoji se od osam faza:(Base36 - Smart Slutions, 2012)

* koncepcija,
* inicijacija,
* analiza,
* projektovanje,
* konstrukcija,
* testiranje,
* implementacija i
* održavanje.

Kako je proces sekvencijalan, to podrazumeva da prelazak na naredni korak, podrazumeva kompletni završetak prethodnog, jer ne postoji mogućnost povratka. Nema meste za promene I greške, jer bi to značilo povratak na početak. Dakle, neophodno je veoma pažljivo planiranje.

Prednosti metode:

* Waterfall metodologija naglašava preciznu evidenciju. Takva evidencija pruža mogućnost poboljšanja postojećeg programa u budućnosti.
* U ovom pristupu klijent zna šta da očekuje. On će imati ideju o veličini, ceni i rokovima projekta, kao iodređenu predstavu o tome šta ćekakav ćebiti na kraju.
* U slučaju fluktuacije zaposlenih ,kvalitetna dokumentacija u Waterfall-u omogućava minimalan uticaj na projekat.

Negativne strane metode:

* Kada je jedan korak završen,ne postoji mogućnostpovratka na prethodnu fazu i eventualnihpromena.
* Oslanjanjena početne uslove u velikoj meri, može dovesti do problema i prekida projekta, jer ako početni uslovi nisu dobri, čitav proces se remeti i nastaju poteškoće.
* Ako je došlo do greške u određenim zahtevima ili je potrebno izrvšenje određene promene, projekat mora da počne od početka sa kompletno novim uslovima.
* Ceo proizvod se testiran samo na kraju. Ako postoje greškena početku koje su otkrivene kasno, to utiče na sledeću fazu. Do kasnog ispitivanja dolazi zato što se periodična odlažu, kako bi se ispoštovao vremenski tok projekta.
* Proces ne uzima u obzir rastuće potrebe klijenta. Ako klijent iznese nešto više od prvobitnih zahteva, to će se odraziti na vremenski tok i budžet projekta.

Waterfall metoda se korisiti:

* Kada postoji jasna slika o tome kakav proizvod treba da bude na kraju.
* Kada korisnici neće biti u mogućnosti da menjaju obim projekta jednom kada je započet.
* Kada je definicija, ne brzina, ključ uspeha.

Kada je u pitanju Agile metodologija, počinje se od jednostavnijeg projektovanja proizvoda, a zatim se radi na malim modulima. Rad podrazumeva određene vremenske interval ili sprintove, na kraju kojih se projekat testira i ocenjuje. Ovde se mogu otkriti greške i obezbediti povratne informacije od klijenata, kako bi se nastavilo dalje, u novi sprint.Dakle, postiže se određeni stepen fleksibilnosti, koji tradicionalna metoda ne dozvoljava.

Prednosti:

* Agile metodologija dozvoljava promjne nakon inicijalnog planiranja. Promene su sastavni deo procesa, i očekuju se po zahtevu klijenta.
* Kako postoji mogućnost izmena tokom procesa, to olokšava dodavanje karakteristika koje će proizvod činiti kompatabilnim sa a najnovijim dešavanjima u nekoj industriji.
* Na kraju svakog sprinta se vrši ocenjivanje projekta, što omogućava klijentima da iznesu svoje mišljenje i dobiju proizvod po sopstvenoj meri.
* Testiranje omogućava pronalaženje grešaka tokom razvojnog ciklusa, koje ne bi bile primećene u završnoj fazi.
* Kako se vrši detaljno testiranje na kraju svakog ciklusa, proizvod bi mogao da pređe u komercijalnu fazu upravo tada. Ovim se povećava verovatnoća ranijeg datuma lansiranja proizvoda.

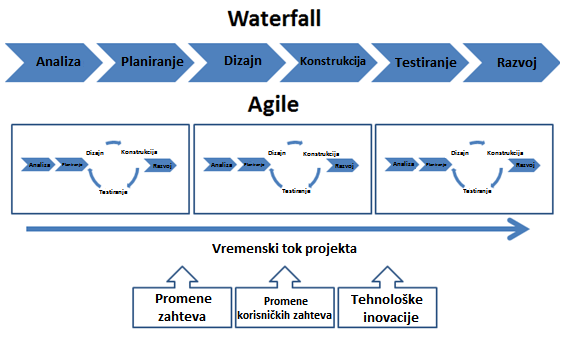
Nedostaci:

* Sa manje uspešnim projektnim menadžerom, projekat može postati serija sprintova sa neupotrebljivim elementima, npr. kod razvoja softvera, serija neupotrebljivih kodova. Ako se to desi dolazi do prekoračenja vremenskog toka i budžeta.
* U nedostatku plana inicijalnog projekta, finalni proizvod može biti veoma različit od njegove prvobitne zamisli i namene.

Agile metodologija se koristi:

* Kada je brzina proizvodnje od veće važnosti nego kvalitet proizvoda.
* Kada klijenti mogu da procene obim projekta.
* Kada ne postoji definitivna slika o tome kakav bi trebalo da bude finalni proizvod.
* Kada postoje obučeni ljudi koji su sposobni da samostalno rade i donose odluke.
* Kada je u pitanju proizvod u oblasti u kojoj se standardi brzo menjaju.

Slika 1 daje uporedni prikaz pomenutih metodologija. (Neo Globe Consulting, LLC, 2014)



Slika 1 Uporedni prikaz Waterfall i Agile metodologije

# SCRUM METODOLOGIJA

## Poreklo i definicija metodologije

Kada je u pitanju poreklo termina "Scrum", on potiče iz članka koji su 1986. godine objavili HirotakaTakeuchi and Ikujiro Nonaka u Harvard Business Review-u , pod nazivom „Nova nova igra za razvoj proizvoda“ („The New New Product Development Game “). Takeuchi i Nonaka su kroz holističku perspektivu opisali ovaj koncept u kome se projektni timovi sastoje od malih kros-funkcionalnih timova, koji rade na dostizanju zajedničkog cilja. Kao primer ovakvog tima autori su naveli ragbi tim. (Pham & Pham, 2011)

Dalji doprinos ovom pravcu dali su Jeff Sutherland i Ken Schwaber. Tokom rada na kreiranju Objektno orijentisane analize i dizajna („Object Oriented Analysis and Design“)Jeff Sutherland, tada na poziciji potpredsednika inženjeringa u Easel koroporaciji, shvatio je da njegov softverski tim ima potrebe za ubrzanim razvojem aplikacija. Želeo je proces koji će omogućiti demonstraciju funkcionisanja kodova, na kraju svake iteracije kratkog trajanja, radije nego prikaz Gantograma, a sve ovo po zahtevu generalnog direktora pomenute korporacije.

U istom periodu, Ken Schwaber je razmišljao o tome da pomogne svojoj kompaniji, Advanced Development Methods, kroz poboljšanje softverskih procesa, u cilju povećanja produktivnosti angažovanih timova. Nakon daljeg analiziranja drugih nezavisnih proizvođača softvera, shvatio je da su njihovi razvojni procesi slični po tome što svi koriste empirijske procese, uz konstantnu inspekciju i prilagođavanje.

Udruženim snagama, na zahtev konzoricijuma pod nazivom „Object Management Group“, pomenuta dva autora su sumirali dugogodišnji rad u ovoj oblasti, stvorivši novu metodologiju koju su nazvali Scrum. Navedeno znanje je dobilo obličje 1995. godine, u članku pod nazivom „Scrum i savršena oluja“ („Scrum and the Perfect Storm“). (Pham & Pham, 2011)

Tvorci ovog koncepta su definisali Scrum kao „radni okvir u kojem ljudi mogu da otkriju kompleksne adaptivne probleme, istovremeno isporučujući proizvode najveće moguće vrednosti na produktivan i kreativan način“. (Schwaber & Sutherland, 2013)

Naredni primer nam na ilustrativan način opisuje scrum u kratkim crtama: (Kniberg & Skarin, 2010)

1. Podelite zaposlene u svojoj organizaciji u nekoliko kros-funkcionalnih, samo-organizovanih timova. Slika 2 prikazuje navedeno.



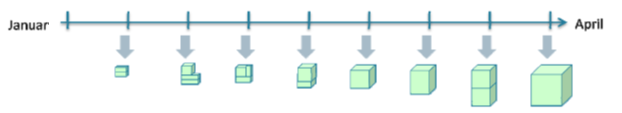
Slika 2 Scrum: Podela zaposlenih u timove

1. Podelite posao na male, konkretne zadatke i rezultate. Sortiraje njihovu listu po prioritetu i procenite relativne napore za realizaciju svake stavke. Slika 3 daje prikaz:



Slika 3 Scrum: Podela posla

1. Podelite vreme na kratke iteracije fiksnog trajanja (obično 1 - 4 nedelje), sa proizvodom čije će karakteristike potencijalno moći da zadovolje potrebe tržišta u tom trenutku, i koji će biti spreman na kraju svake iteracije. Slika 4 prikazuje podelu vremena:



Slika 4 Scrum: Podela vremena

1. Optimizujte plan lansiranja proizvoda i ažurirajte prioritete uz tesnu saradnju sa korisnikom zasnovanu na uvidu i analizi rezultata svake iteracije.
2. Optimizujte proces vršenjem retrospektive na kraju svake iteracije.

Dakle, umesto da velika grupa provodi mnogo vremena radeći na velikim stvarima, imamo mali tim koji kraće vremena radi na mala stvarima, ali se redovnom integracijom postiže prvobitna celina.

## Vrednosti

Kada je reč o vrednostima Scrum metode, one su od značaja jer transformišu način na koji tim funkcioniše i ostvaruje rezultate,i stvara poverenje između različitih internih i eksternih strana. Svako može da ispoštuje tehnički deo koncepta, ali vrednosti su te koje su neopiljive i koje se ne mogu kupiti. Tabela 1 prikazuje vrednosti i antiteze SCRUM metode.

Tabela 1 Vrednosti i antiteze SCRUM metode

|  |  |
| --- | --- |
| **Vrednost** | **Antiteza** |
| Odgovornost | Angažovanje rukovodioca u ime drugih |
| Fokus | Suboptimizacija |
| Otvorenost | Tajnovitost |
| Poštovanje | Položaj, moć i kontrola |
| Hrabrost | Oprez i ugovori |

Iako su vrednosti povezane, često je nemoguće da sve imaju isti udeo I balans u rohanizaciji. Ipak, ako je selekcija neophodna, treba otpočeti process implementacije Scrum-a sa otvorenošću i hrabrošću. Takođe, pomenute vrednosti se ne moraju slagati sa ličnim vrednostima pojedinaca, i to predstavlja poteškoću u primeni, upravo zbog različitosti.

U organizaciji koja se zasniva na saradnji i kreativnosti, implementacija pristupa ne predstavlja problem, jer se obe strane vode načelima fleksibilnosti, pa zaposleni ne bi trebalo da pružaju otpor. Sa druge strane, tamo gde su osnovni principi moć i kontrola, ovaj pristup predstavlja izazov za rigidnu strukturu, pa je implementacija otežana. Ljudi koji su navikli na određene obrasce i šeme, koji se nikada nisu dovodili u pitanje, teško prihvataju nove načine rada, u kompletno drugačijem obliku. Obično tvrde da promena nije neophodna, pa nove zahteve u poslovanju pokušavaju da ukolpe u postojeći sistem, što dovodi do sveukupnog disbalansa. (Viscardi, 2013)

SCRUM vrednost #1: **Hrabrost**

Hrabrost predstavlja mogućnost da se sučimo sa sopstvenim strahovima, i ona prožima svaki aspekt života. Nisu svi svi sposobni da se suoče sa svojim strahovima, pa je takav slučaj i u organizaciji. Previše uplašeni zaposleni čiji cilj što manja odgovornost za različite procese I zadatke se skrivaju iza različtih dokumenata i ugovora, samo da bi neko drugi mogao da odgovara za iste. Tako se rizik prebacuje između zaposlenih, i dolazi do stagnacije, jer se usporava normalan rad, i sveopšti progres u obavljanju zadataka.

U ovom pristupu, rešenje za pomenuti problem je suzbijanje straha od odgovornosti, svesnom odlukom da se svaki dan, malo po malo, zaposleni suočavaju sa onim čega su se plašili. Potrebno je da preuzmu odgovornost za deo procesa, tačnije sprint, a zatim uspomoć nadređenih da uklone prepreke koje su im otežavale rad, kakve god one bile. (Viscardi, 2013)

SCRUM vrednost #2: **Posvećenost**

Odgovornost i posvećenost obavezama su bitan segment u ovom pristupu. Kada se obavežemo da ćemo završiti određeni zadatak, to zahteva angažovanje i doslednost. Scrum timovi su dizajnirani sa posvećenošću tako da članovi tima rade zajedno kako bi isporučili najbolji mogući proizvod i doprineli rastu prodaje.

Članovi SCRUM tima se obavezuju na sledeće: (Viscardi, 2013)

1. Ceo tim se obavezuje na zajednički rad kako bi se postigli ciljevi postavljeni u sprintu, i to zahteva veliku odgovornost i posvećenost. Oni moraju da postignu konsenzus u planiranju često bez kompletne slike proizvoda, svaki dan planiraju akcije za naredni, I prezentuju svoj rad u dnevnom pregledu.
2. Vlasnik proizvoda se zauzvrat obavezuje da će kreirati viziju proizvoda i razjašenje stavki u product backlog-u, kao i da će dati timu smernice za neka pitanja. Pored toga, on će biti prisutan pri pregledu sprinta, kao i da će imati uvid u funkcionalnosti tokom sprinta, ne tek na kraju.
3. Scrum Master se obavezuje da će štiti tim i njegove članove, vodeći računa da im obezbedi sve što je potrebno za ispunjenje njihovi obaveza. On void računa da vlasnik proizvoda i tim budu srećni, motivisani i puni energije zbog njihovog posla.

SCRUM vrednost #3: **Otvorenost**

Tajnovitost je barijera svake promene, i pokretač problema. Ćutanjem i prikrivanjem informacija se remeti komunikacija, a ona je osnov funkcionisanja. U ovoj situaciji, najvažnija je borba za otvorenost. Umesto da nakon menadžer drži sastanak u jednoj sobi sa zatvorenim vratima, trebalo bi se odlučiti za razgovor u otvorenom radnom prostoru. Ako Vaš tim nije  
završio sve zadatke do kraja sprinta, treba biti otvoren po tom pitanju, i izneti stav. Ako vlasnici proizvoda nemaju jasnu predstavu o funkcionisanju tima, trebalo bi ih uključiti u dnevne sastanke. Ako postoji percepcija da tim nije mnogo produktivan, potrebno je kreirati i postaviti sistem merenja produktivnosti. Otvorenost daje mogućnost u uvid u realnost kapaciteta i brzinu rada. Zaključak: potrebno je deliti informacije, dobre, loše i ružne. Ljudi donose dobre odluke kada znaju istinu. (Viscardi, 2013)

SCRUM vrednost 4#: **Fokus**

Fokus predstavlja sposobnost da se kompletno usmerimo na zadatke i posao. To je naša lična odlukam i najbolji rezultati se dobijaju kada se pažnja usmerava na zadatke motivisanošću i voljom. Problem nastaje kada menadžeri pokušavaju da previse organizuju i suboptimizuju rad, izazivajući problem u obavezama i narušavajući prirodan način na koji se zaposleni fokusiraju na zadatke. Niko ne može da opiše način na koji jedna osoba uspostavlja fokus na bitne stvari.

Kada se tim fokusira na stvaranje increment proizvoda u kratkim ciklusima, kao što je slučaj u ovom pristupu, sve što se nađe na putu tome, smatra se otpadom. Vreme čekanja, handoffs, defektie , zastareli alata i slično, predstavljaju prepreke koje treba zahtevaju fokus i donošenje odluka, tako da tim može da smanji vreme i povećanje kvaliteta funkcija u svakom sprintu. Dakle, fokus otkriva neusaglašenosti u malim koracima (Viscardi, 2013)

SCRUM vrednost 5#: **Poštovanje**

Poštovanje predstavlja odraz divljenja nekoj osobi zbog različitih osobina, kvaliteta i veština koje ona poseduje. Antiteza poštovanju je moć, položaj i kontrola, te menadžer ima izbor ili da  
motiviše zaposlene poštujući ih, ili da ih kontroliše koristeći moć i svoju poziciju.  
dobar lider će najverovatnije izabrati prvu opciju. Scrum Master će naići na prepreke vezane za moć i kontrolu, u organizacijama koje postizanje dobrih performansi nagrađuju unapređenjem ljudi. Pošto niko ne želi da ispusti tu šansu, i šansu za narednim unapređenjima, u cilju zaštite svog položaja, oni će koristiti moć i kontrolu. Ovo na kraju gradi odbojnost, pa čak i  
prezir među članovima tima.

Kako bi promovisao i usadio koncepte poštovanja i autonomije, Scrum Master ima nekoliko mogućnosti od kojih je jedna i dnevni sastanak tima, zatvorenog tipa, tj. sa “zatvorenim vratima”. Iako je potrebno negovati otvorenost, on mora bude na visini svog zadatka, te ponekada mora i da zamoli određene osobe da napuste sastanak ili da se suprotstavi menadžeru, a sve to u cilju postizanja određenih rezultata i doslednosti u domenu poštovanja.

# SCRUM model

## Uloge

Scrum model podrazumeva tri vrste uloga: vlasnika proizvoda, Scrum Mastera i tim za razvoj, i one su sastavni deo svakog pristupa u ovoj oblasti. Pored toga, ne manje važan značaj imaju i menadžement, menadžer za promene i Scrum konsultant. U narednom delu rada će biti detaljno opisane sve navedene uloge, i njihovo mesto u modelu.

### Vlasnik proizvoda – Product Owner

Vlasnik proizvoda je krucijalna i najvažnija uloga u Scrum modelu. To je osoba u čijem je vlasništvu proizvod, i on je potpuno odgovoran za donošenje svih odluka koje će uticati na proizvod u budućnosti. Njegovi zadaci se mogu sagledati i iz perspektive menadžera zaduženog za proizvod i oni podrazumevaju: (Maximi, 2015)

* analizu tržišta i konkurencije,
* unapređenje i održavanje proizvoda,
* utvrđivanje zahteva vezanih za proizvod i provera adekvatnosti njihove formulacije,
* kreiranje koncepta za uvođenje na tržište,
* uvođenje proizvoda na tržište,
* podršku i edukaciju u domenu prodaje,
* brigu o kupcima,
* razvoj strategije proizvoda,
* planiranje, koordinaciju i implementaciju strateških mera,
* izradu studija izvodljivosti,
* procenu očekivanog obima prodaje,
* kreiranje mape putanje proizvoda,
* oblikovanje procesa uvođenja i životnog ciklusa proizvoda,
* eliminaciju proivoda,
* optimizaciju povraćaja na investicije,
* redukciju ukupnih troškova vlasništva,
* optimizaciju vrednosti za kupce.

Vlasnik proizvoda bi trebalo da poseduje sedam ključnih kvaliteta: (Pham & Pham, 2011)

1. Uspešno upravljanje očekivanjima zainteresovanih strana i konfliktnim prioritetima- poseban značaj predstavlja odnos sa stejkholderima koji su zainteresovani za postojeći projekat, i imaju veliki uticaj na njega.
2. Posedovanje znanja o proizvodu i jasna vizija o istom- jasna predstava o proizvodu je zvezda vodilja tima.
3. Mogućnost transofrmacije vizije o proizvodu u kvalitetno skladište proizvoda (product backlog), kroz zahteve vezane za proizvod- lista dobro formiranih korisničkih priča (user story) koja se smešta u skladište proizvoda je krucijalna.
4. Potpuna dostupnost za aktivno uključivanje u timski rad, ne samo tokom sprint-a, već i tokom planiranja i startovanja sprinta – ovo podrazumeva rad sa timom za razvoj na dnevnoj bazi, podrazumevajući pregled postignuća.
5. Dobar organizator koji može da koordinira više aktivnosti, zadržavajući ciljanu perspektivu i prisebnost – organizacija je, kao i uvek, neophodna.
6. Komunikacija i širenje vizije proizvoda, ne samo u timu, već i kroz poslovanje, tako da poverenje u timu ostaje nepromenjeno tokom životnog ciklusa proizvoda-neophodno je razmišljanje naprednije od prosečne osobe.
7. Dobar lider, sposoban za vođstvo, obuku i podršku timu, paralelno vodeći računa o dostizanju o vrednosti koja se očekuje od IT sektora - bez efektivnog lidera tim je samo anarhična vorevina.

Dakle, vlasnik proizvoda je osoba koja najbolje razume potrebe klijenata i može da ih izrazi kroz zahteve vezane za proizvod. On nije ni tehnički ekspert, stariji developer ni IT projektant. Svako ima drugačiji pristup viziji proizvodia, ali postoji tehnika „5W“ koju koriste vlasnici proizvoda kako bi definisali istu: (Pham & Pham, 2011)

* Whom? (Kome?): ciljani korisnik.
* Why? (Zašto?): na koji način je ovaj proizvod specijalan?
* What? (Šta?): šta proizvod nudi?
* Where? (Gde?): lokacija.
* When? (Kada?): vremenska zona.

I ako je njegovna glavna uloga rad na korisniničkim zahtevima, i ažuriranju stavki u skladištu podataka, Postoji mogućnost da bude propterećen velikom količinom zadataka, i to radeći na i na nekim nepotrebnim detaljima. Upravo ovo direktno uključivanje u razvoj, može omesti njegovu sposobnost obavljanja primarne uloge vlasnika proizvoda. Tada vlasnik proizvoda postaje “zastupnik” vlasnik proizvoda (Proxy Product Owner). (Maximi, 2015) Na kraju, bitna stvar koja razlikje vlasnika od menadžera proizvoda je što vlasnik nema mogućnost da odustane od proizvoda, i u obavezi je da ga dovede do željenog oblika, do kraja.

### Scrum Master

Scrum Master je uloga koja je zadužena za upravljenja procesom Scrum-a u organizaciji. Dakle, on se stara da se učesnici u procesu pridržavaju Scrum pravila, preciznije on je zadužen za tim. On je često više cenjen zvog svoje ličnosti i postignuća, čak i kada nema formalnu podršku upravljačke strukture. Njegove glavne odgovornosti su: (Pries & Quigley, 2010)

1. uklanjanje prepreka,
2. rešavanje sukoba u timu,
3. sihnornizacija timskih i Scrum pravila,
4. nabavka resursa,
5. održavanje fokusa tima.

Zadaci Scrum Mastera su: (Maximi, 2015)

* povećanje produktivnosti Scrum tima uklanjanjem prepreka,
* pomoć pri razumevanju i poštovanju Scrum procesa,
* obuka učesnika Scrum projekta,
* fokusiranje učesnika na ciljeve projekta,
* omogućavanje realizacije događaja vezanih za proces,
* komunikacija sa svim osobama koje su povezane sa Scrum procesom.

Jedna od najvažnijih odgovornosti Scrum Master je da stvori timsko okruženje pogodno za ostvarenje zadatih ciljeva i da se uveri da su svima jasni zadaci. Umesto da vrši direktnu kontrolu aktivnosti, trebalo bi da razgraniči uslove u kojima se one vrše, i time da određenu autonomiju članovima tima. Kada se uspešno završi faza formiranja radnog okvira, i kada se usvoje potrebna znanja, obično sledeči problem nastaje u domenu odnosa sa funkcijama koje su povezane sa Srum procesom. To su obično segmenti prodaje, kupovine, pa čak i menadžmet, i opet je Scum Master zadužen za balansiranje i uspostavljanje korektnih odnosa. Stoga, sedam ključnih kvaliteta koje bi trebalo da poseduje svaki Scrum Master su : (Pham & Pham, 2011)

1. Veliko teorijsko i praktično znanjeo Scrum-u – koliko je važno i sertifikovano znanje o Scrum meodi, toliko je relevantna i iskustvena komponenta, s obzirom na specifičnosti procesa.
2. Velike sposobnosti za rukovođenje, ali i istovremenu pomoć- pored klasične pomoći u vidu vođenja kroz proces, važno je otklanjanje prepreka tokom rada, kao i održavanje okruženja neometanog od strane eksternih činilaca.
3. Snažne organizacione sposobnosti – posebno kada su i vlasnik proizvoda i tim novi, na Scrum Masteru je organizacija svih vrsta sastanaka.
4. Dobre komunikacione veštine – zbog mnogo ljudi i mnogo zadataka.
5. Veštine prezentacije – podrazumeva se rad u Power Point alatu ili sličnom.
6. Sposobnost rešavanja konflikata – jer ne postoji tim koji funkcioniše bez konflikata, pogotovo u novom okruženju.
7. Sposobnost za rad na razvoju ljudskih veština - motivacija i podrška članovima tima da se razvijaju kroz ovaj proces je ključna, zbog ukupnih performansi tima.

Na kraju, Scrum Master nije developer, ili stručnjak iz oblasti upravljanja proizvodom. Njemu su potrebne meke veštine, koje prikazuje Tabela 2. (Maximi, 2015) Mišljenja različitih autora se preklapaju, ali ovde je akcenat na mekim veštinama.

Tabela 2 Meke veštine Scrum mastera

|  |  |
| --- | --- |
| Veštine komunikacije | Dobro upravljanje vremenskim rasporedom |
| Veštine rešavanja konflikata | Disciplina |
| Pregovaračke i diplomatske veštine | Veštine refleksije (za sebe i za druge) |
| Sposobnost izgradnje i održavanja kvalitetne mreže u kompaniji | Prihvatanje kritike |
| Otvorenost | Afinitet prema procesima |
| Hrabrost | Poznavanje timske dinamike |
| Veštine modernizacije i podrške | Poznavanje uticaja okruženja na motivaciju i produktivnost |
| Ekspertsko Scrum znanje | Osnove psiholoških veština |
| Vladanje development terminologijom | Osnove socioloških veština |
| Motivacione veštine | Težnja i sposobnost stalnog učenja i usavršavanja |

### Tim za razvoj

Razvojni tim je odgovoran za transformaciju zahteva vezanih za proizvode u gotov proizvod. Ovo se odvija u iteracijama i inkrementalno, pa tim svakih par nedelja mora da isporuči neki rezultat. Nije moguće planirati zadatke i analize na mesečnoj osnovi. Scrum Master je tu da u tome pomogne, ali tim se samostalno organizuje i uklanja prepreke, da ne bi došlo do neuspeha. Rezultat toga je samopouzdanje i iskustvo koje pomaže timu da ispravno odredi količinu posla za svaki sprint, i da kaže NE u situacijama kada je to opravdano.

Potrebne kompetencije tima variraju od situacije do stiuacije. Ekspertiza u pojedinoj oblasti i lični aspekti su podjednako važni, te veliki značaj u procesu selekcije novih developera predstavlja učešće Vlasnika proizvoda, Scrum Mastera i postojećih članova tima. Ovo se obično odnosi na programere, testere, eksperte u polju dokumentacije, analitičare, dizajnere korisničkog interfejsa, itd. Preporuka za većinu timova je podjednak broj programera i testera. Ukoliko su developeri već radili u Scrum okruženju, ili bili deo nekog Scrum tima, time je ovaj proces znatno olakšan. U ovom slučaju, organizaciona kultura je već bazirana na otvorenosti, što daje slobodu članovima tima da izraze svoje mišljenje, pa i nezadovoljstvo jedni drugima. Razvojni tim može da odluči kome je mesto u timu, a kome nije, kroz sledeće savete za odabir: (Maximi, 2015)

Savet Br. 1. Osnovni sastav softver tima (Maximi, 2015)

Ako ne znate od kojih potrebnih tehničkih veština da krenete, primenite prvo pravilo: Angažujte jednak broj programera i testera u timu, podržanih projektantom. Za početak, uznite projektanta sa iskustvom, dva programera i dva testera. Na Sprint Retrospective sastancima identifikujte dalje potrebe za kadrovima.

Savet Br.2. Metod selekcije (DeMarco & Lister, 2013)

Prilikom pozivanja kandidata na razgovor za posao, dajte im prostora da naprave kratku prezentaciju svog prethodnog rada, u trajanju od 5 minuta. Pritom, dozvolite im da ovo predstave pred potencijalnim budućim kolegama i podstaknite ih na diskusiju sa kandidatom. Kada se razgovor završi, povedite raspravu sa timom i ozbiljno razmotrite utiske o kandidatu. Ovo je korisno za bolju percepciju veština kandidata i, u slučaju da ga zaposlite, osigurava bržu integraciju člana u tim.

### Menadžment

Velika je greška posmatrati menadžment organizacije kao ulogu koja je nebitna, a još veća drskost smatrati je nepotrebnom. Postoji veliki broj primera u kojima dominira zanemarivanje upravljačke strukture, što je rezultiralo velikim problemima. Osnovni zadatak menadžmenta je da upravlja organizacijom, da je unapredi, i da osigura uspeh u poslovanju. Ovo naravno uključuje oblikovanje psolovnih procesa. Stoga, on predstavlja podršku i saveznika pri rešavanju problema i prolagođavanju zaposlenih procesu primene Scrum metode.

Menadžeri su obično vrlo zauzeti ljudi, te može izgledati da ne posvećuju dovoljno pažnje različitim zahtevima. Trebalo bi ih podržati dobrom pripremom odluka, izbegavajući nepotrebne sastananke (izbegavati neprecizne ciljeve i loše osmišljene programe), kao i eksplicitnim naglašavanjem koristi od angažovanja za organizaciju, kako na grupnom tako i na individualnom nivou. Potrebno je uključiti mendažment, jer na kraju krajeva svima treba sigurnost, pomoć i podrška. Lep odnos i poštovanje doneće velike pobede Scrum metodi. (Maximi, 2015)

### Menadžer promena

Ova uloga nema nimalo lak zadatak. Već sam naziv pozicije budi strah u ljudima. On mora da u njima promeni ono što je u svima nama donekle duboko ukorenjeno: osećaj sigurnosti u poznatim uslovima. Dakle, recimo da je osnovna misija menadžera promene da pozitivno uzdrma atmosferu u organizacijij i time stvori bazu za inovacije u strukturi i kulturi. Promene se ne sprovode bez cilja. Cilj je povećanje konkurentske prednosti.

Uloga menadžera promena je uloga nekoga ko treba da Scrum-uje organizaciju, vrlo je slojevita, i zahteva veoma izdiferenciran pristupe i načine razmišljanja. Naravno, i čelične živce. Menadžer promena mora da bude spreman na mnogo pitanja, na koje mora da sam odgovara ili da to delegira nekoj poziciji kao što je, naprimer, Scrum coach (Scrum trener). Izazov u Scrum promenama je postignuće da vas ljudi čuju. Ne samo da vas slušaju. Drugim rečima, da poruka stigne do njih. Za to je potrebno izgraditi autoritet. Formalni autoritet se stiče samom pozicijom, dok se neformalni mora samostalno izgraditi, i to najviše stavivši se u gore navedenu ulogu psihologa. Na prvom mestu, zaposleni treba da shvate da su ove promene dobre za njih, da će im olakšati život u organizaciji, a i da je to za dobrobit kompanije kao celine. Drugo, moraju da steknu poverenje u menadžera i njegove sposobnosti da sprovede adekvatne promene, da one zaista zažive, i da ljudi zaključe sami da je došlo do promena. (Maximi, 2015)

### Scrum konsultant

Scrum konsultant je pozicija mača sa dve oštrice. Njegov zadatak je da utemelji svoje Scrum znanje u organizaciju, ali i da otkriva probleme i prepreke koje mogu da se pojave u njegovoj primeni, kao i da istu olakša koliko je to moguće. Ljudi generalno imaju urođen otpor prema promenama i svemu što je novo i što može da ugrozi njihovu dosadašnju rutinu. Kada se u firmi pojavi neko od spolja, zaposleni su skloni da ga posmatraju kao pretnju za sopstvene pozicije. Ipak, iz istog razloga, mogu da imaju više obzira prema direktnom pristupu Scrum konsultanata u njihovom ispitivanju, shvatajući da oni ne poznaju još uvek dovoljno kompaniju. Scrum konsultant je stručnjak u svojoj oblasti koji dobro naplaćuje svoje usluge, i kada se to uzme u obzir, neće doći do prevelikog trošenja njegovog vremena. Zato kompanije angažuju eksterne saradnike samo u slučaju velike potrebe, a i tada gledaju da što efikasnije iskoriste to znanje. (Maximi, 2015)

Potrebno je biti pažljiv u odabiru Scrum konsultanta. Ovo nije kratkoročni angažman u trajanju od dve nedelje. Zapanjujuća činjenica je da je za pravilnu i potpunu primenu Profound Scrum-a potrebno i do deset godina rada. Prema tome, ako postoji želja da ova eksterna pomoć urodi plodom, potreban je mudar izbor. Ovde se sudaraju efektivnost i troškovna efikasnost. Iako je Scrum na prvi pogled usko specijalizovana oblast, to je zaista samo na prvi pogled. Postoji mnogo različitih veština i specijalizacija. Dakle, prvo se mora determinisati kakav je to stručnjak potreban, i šta se želi konkretno postići njegovim angažovanjem, jer ovih konsultanata zaista ima mnogo, dobrih, loših, najboljih, kao u svakoj struci. Provera spiska govornika na skorijim Scrum konferencijama je uvek odličan način prve selekcije. To može biti prvi krug. U drugom krugu već mora doći do poklapanja potreba i zahteva firme sa kvalifikacijama konsultanta. Takođe, ako postoji neko ko može da preporuči eksperta, tim bolje. Ipak, treba uzeti u obzir da je svaka organizacija specifičan entitet, i neće svaki savetnik odgovarati svakom sistemu. Kada na to dodamo nijanse u organizacionoj klimi i kulturi, možemo govoriti o jedinstvenosti koja krasi svaki živi organizam. Preduzeće, to i jeste, organizam, živ zbog ljudi koji ga čine. Kada se napravi uži izbor, treba zakazati intervjue. Doza skepse je uvek poželjna u ovakvim razgovorima za angažovanje. Preporučuje se upotreba literature kako bi se proverilo osnovno znanje kandidata, uz poigravanje sa nazivima pojmova i koncepata. Poželjno je malo konfuzije u kojoj će se snaći samo najbolji.. Pogrešan korak u inicijalnoj fazi odobira ima efekat domina, svaka naredna će biti pogrešna, i ceo proces je obesmišljen.

Odmah zatim, treba razmotriti odnos cene i primljene usluge. Nije poenta plaćati previše, ali ni da graditi Scrum na polovičnim temeljima. Kao i u svakoj trgovini, cenjkanje i traženje najboljeg odnosa cene i kvaliteta je poželjno. Plan angažmana bi trebalo u početku da pokriva nekoliko meseci, i da uključuje mesečne posete. Kao i lekar, Scrum konsultant mora da dodje u povremenu vizitu i proveri da li se njegovo znanje prima na organizacijsko tle. „Scrumovanje“ može da potraje, to je dugotrajan proces, možemo ga nazvati i poduhvatom. Tako, u trećoj godini Scrumovanja, što je neka zaključna faza, treba organizovati ozbiljne radionice u dužim vremenskim intervalima. (Maximi, 2015)

### Pile i prase

Na kraju, poznata anegdota koja predstavlja Scrum metodu u neformalnom smislu: priča o piletu i prasetu koji žele da otvore restoran. (Schwaber & Beedle, Agile software development with Scrum , 2002)

Pile: „Hajde da otvorimo restoran! “

Prase: „Kako ćemo ga nazvati? “

Pile: „Šunka i jaja! “

Prase: „Ne, hvala. Ja ću da budem posvećen, ali ti ćeš jedini da budeš uključen! “

U prenesenom značenju, postoje dve vrste uloga: *pile*, koje je *uključeno*, i *prase* koje je samo *posvećeno*. Alijas, tu su oni koji su odgovorni za projekat i koji imaju ovlaščenje da rade sve što je potrebno za uspeh istog, i oni koji nisu odgovorni, pa ne moraju nepotrebno da budu uključeni u iste poslove. U prvu grupu spadaju korisnici, stejkholderi i menadžeri, a u drugu Product Owner, Scrum Master i tim. (Pries & Quigley, 2010)

## Artefakti

Još jedna komponenta Scrum modela jesu artefakti, i napominju se tri najvažnija: (Maximi, 2015)

* Korak u razvoju proizvoda(Product Increment);
* Skladište proizvoda(Product Backlog);
* Skladište sprint-ova (Sprint Backlog).

Njihov cilj je smanjenje dokumentacije u samom procesu Scrum-a kao, i napora u tom domenu na određeni, prikladan nivo, kroz konstatnu proveru i adaptaciju svih zahteva i zadataka. Još jedna njihova odlika je da moraju biti dostupni i transparentni u svakom trenutku, svim učesnicima u procesu. Dakle, nema skrivenih informacija.

### Korak u razvoju proizvoda (Product Increment)

Skup razvijenih funkcionalnosti, zadataka, aktivnosti na projektu, čiji krajnji rezultat ima svoju vrednost u smislu kvaliteta koji zadovoljava zahteve klijenta, i može se isporučiti, naziva se inkrement proizvoda. Tim “izbacuje” inkrement proizvoda periodično, na kraju svake iteracije. Uloga Product ownera je da na kraju svake iteracije ispita da li on ispunjava kriterijume uspeha i prema tome da usmerava dalji proces, u dogovoru sa stejkholderima. (Viscardi, 2013)

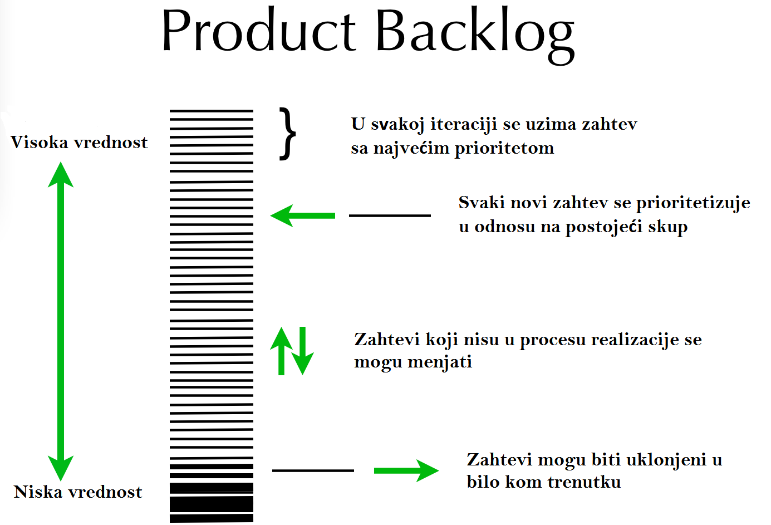
Jedna od Scrum tekovina je obaveza da tim nakon svake iteracije ima isporučiv proizvod, na kome je potrebno veoma malo dorade, ili ona nije potrebna uopšte. Prednost ovog pristupa leži u tome što je proces sve vreme transparentan,iI nema iznenađenja. Ako proizvod ima nedostataka, svi su upućeni koji su to nedostaci i kako će se oni nadoknaditi. Multifunkcionalni Scrum tim snosi punu odgovornost za proizvod, s obzirom na princip samoupravljanja. (Maximi, 2015)

### Skladište proizvoda (Product Backlog)

Product Backlog je artefakt koji se vezuje za Product Ownera i njegove želje, i predstavlja listu želja. To je spisak svih karakteristika proizvoda koje on i ostali stejkholderi žele da vide kao sastavni deo proizvoda. Product owner je zadužen za ažuriranje liste, kao i za prioritetizaciju najvažnijih ili najvrednijih stavki, ali ostale zainteresovane strane imaju pravu na nove predloge, kao i na vizuelni pristup listi. Lista predstavlja osnovno sredstvo za rad Development team-a, i alat za kolaboraciju sa Product Ownerom. (Maximi, 2015)

Stavke u listi su sortirane po vrednosti, ili hitnosti realizacije, i kako se jedna stavka implementira, na red dolazi druga. Kada se tim odabere određenu stavku za realizaciju u sprint-u, ona i i njen prioritet su nepromenjeni, za razliku od narednih, gde postoji mogućnost izmene vrednosti i detalja. (Viscardi, 2013) Product Owner može da vrši prioritetizaciju na više različitih načina koje odgovaraju specifičnoj organizaciji, i može u nju dodavati neke dodatne informacije, opet po želji. Slika 5 daje prikaz izgleda navedenenog opisa. (Viscardi, 2013)

Slika 5 Product Backlog



**Korisničke priče - User Story**

Kako bi se formirao dobar Product Backlog, potrebno je zahteve vezane za proizvod formulisati u vidu korisničkih priča (User Story), koje će prestavljati njegove stavke (Product Backlog Items). User story predstavlja koncizan način objašnjenja zahteva vezanih za proizvod, i to često u vidu određene kartice. Kada je upitanju opis zahteva, perspektiva koja objašnjava isti je perspektiva interakcije korisnika sa proizvodom.

Cilj je smanjenje dokumentacije, koja se gomila paralelno sa razvojem proizvoda, i promenom korisničkih zahteva. Vrednost User Story-ja je više od običnog zahteva; to je diskusija sa korisnikom koja je nezamenljiva, i koja je osnova za razvoj proizvoda. (Viscardi, 2013)

Formulisanje User Story-ja je posao koji zahteva određeni nivo kreativnosti, ali i proceduralno znanje. Kako bi se navedeno olakšalo, predlaže se pristup u dva koraka: (Pham & Pham, 2011)

***Korak 1 : Identifikacija stejkholdera i njihovih ciljeva***

Kada je u pitanju projekat razvoja novog softvera, neophodno je identifikovati sve stejkholdere koji su uključeni. Zatim, potrebno je saznati njihove ciljeve kroz pitanja: „Koji su Vaši poslovni ciljevi ili postignuća? “, „Zašto želite novi softver? “ i „Na koji način merite svoja postignuća? “.

Identifikacija ciljeva se može realizvati kroz SMART pravilo, tj. Ciljevi bi trebalo da budu:

* **Specific** – Specifični;
* **Measurable** – Merljivi;
* **Achievable** - Dostižni;
* **Realistic** - Realistični;
* **Time-based** - Vremenski orijentisani.

***Korak 2 : Prikupljanje zahteva za Product Backlog***

U ovom koraku predstavnici različitih stejkholdera iznose zahteva vezane za proizvod, pa je neophodno dobro razumevanje zbog transformacije u User Story-je. Predlaže se tehnika „Drveće i šuma“: počevši od šume, tj. od nivoa ukupnog proizvoda, potrebno je zapitati se od čega on treba da se sastoji. Preciznije, koliko drveća treba šuma da ima. Dalje, neophodna je podela na grane, a zatim na lišće, što ujedno predstavlja najveći nivo detalja, upravo User Story. Može se ići i obrnutim redom, gde se od korisnika zahteva lista svih zahteva, pa da se grupisanje istih realizuje navedenim principom.

Verifikacija User Story-ja se može izvršiti CUTFIT pravilima, koje daju potvrdu o njihovoj pravilnoj formulaciji, spremnosti za procenu Development team-a, i na kraju razvoj: (Pham & Pham, 2011)

* **Consistent (Konzistentan)**: konzistentan zahtev nije u suprotnosti sa drugim zahtevom.
* **Unambiguous (Nedvosmislen)**: bez obzira na tumača zahteva, on treba da ima samo jedno značenje.
* **Testable** **(Podložan testiranju)**: potrebno je testirati svaki zahtev, ali ako on nije u formi za testiranje, posao je uzaludan.
* **Feasible (Izvodljiv)**: zahtev mora biti implementiran u postojećem okruženju, sa svim svojim mogućnostima i ograničenjima.
* **Independent (Nezavisan)**: nijedan User Story ne bi trebalo da zavisi od drugo User Story-ja.
* **Traceable (Upariv)**: svaki User Story mora da poseduje mogućnost povezivanja sa korisnikom i njegovim ciljevima.

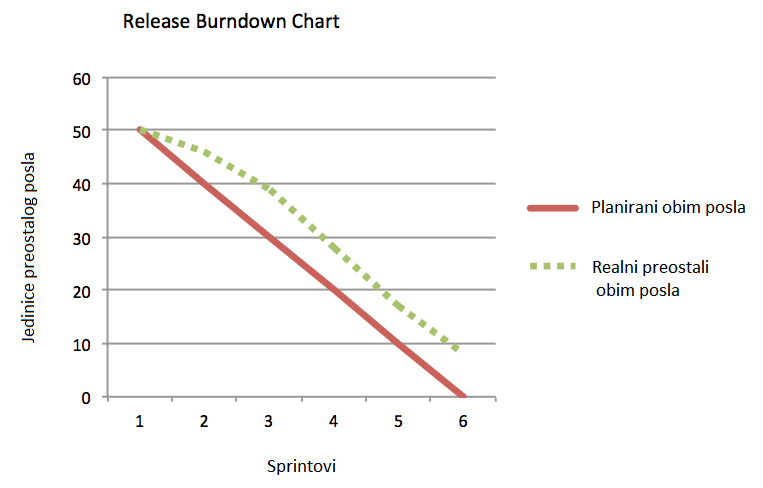
Kada je kraj pisanju User Story-ja? Odgovor mogu dati sledeći redovi: (Pham & Pham, 2011)

1. Kada korisnik ne može da ga razloži na više povezanih priča, tj. kada su uzeti u obtir svi delovi aplikacije.
2. Kada se mogu rasčlaniti na taskove (zadatke), koje tim rangira prema vremenu potrebnom za njihovo kompletiranje (4 do 8 sati). Tada rad na razvoju može da počne.
3. Kada tim može da počne sa procenom poente zahteva, kako bi mu dodelio određeni broj bodova, o čemu će biti reč u narednom delu rada.

Deo Product Backloga, podskup, koji se definiše za jedan poseban cilj, naziva se ***Release backlog***. Iako se Release Backlog može odrediti unapred, Product Owner ima pravo da premešta i menja raspored stavki u njemu, pregovara o obuhvatu, u skladu sa njegovom predstavom domena i vremenskog horizonta projekta. Naravno, on zahteva periodično ažuriranje. (Viscardi, 2013)

Alat za vizualizaciju (**Release burndown chart**) pokazuje koliko posla je preostalo u Product Backlog-u nakon svakog Sprinta. Product Owner iz ovoga može da dobije informacije na osnovu koji donosi odluke o obimu, troškovima i vremenu projekta. Slika 6 daje detaljniji prikaz. (Viscardi, 2013)

Slika 6 Release burndown chart



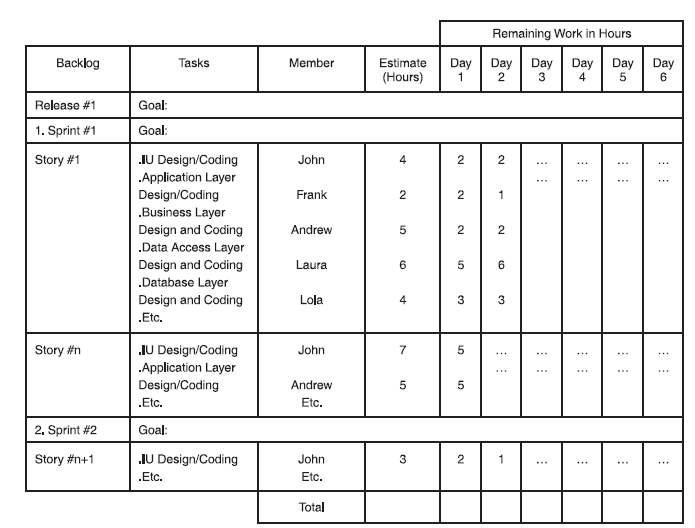
### Skladište Sprintova (Sprint Backlog)

Sprint Backlog predstavlja nešto nalik skladištu Srpintova, preciznije, on se sastoji od stavki u Product Backlogu koji je predviđen od strane tima za razvoj u tekućem Sprintu. Takođe, čine ga i razloženi zadaci zarad prikaza kako ove stavke označiti kao finalizovan inkrement proizvoda (Product Increment). Kreira se tokom Sprint Plannig Meeting-a, od strane Development Tima, koji ima isključivu odgovornost u ovom domenu. Članovi tima vrše svakodnevno ažuriranje, kako bi pokazali koliko vremenskih jedinica rada je ostalo do završetka njihovih zadataka, a takođe mogu uklanjati, dodavati ili menjati stavke. (Viscardi, 2013) Ljudi van tima nemaju pravo na učešće u njegovom stvaranju i korišćenju. (Maximi, 2015)

Kako Scrum posmatra svaki Sprint kao poseban projekat, može se izvršiti poređenje sa projektnim planom. Sprint backlog nije realan ako se ne menja u vremenu. Informacije koje dolaze iz ovog backloga imaju neutralnu vrednost, što znači da niko ne mora da daje odgovor do ovog trenutka, bez obzira na ritam obavljanja posla. O korenu nekog problema je moguće naučiti samo nakon analize neke situacije, i da li ciljevi Sprinta mogu biti postignuti ili ne. (Maximi, 2015)

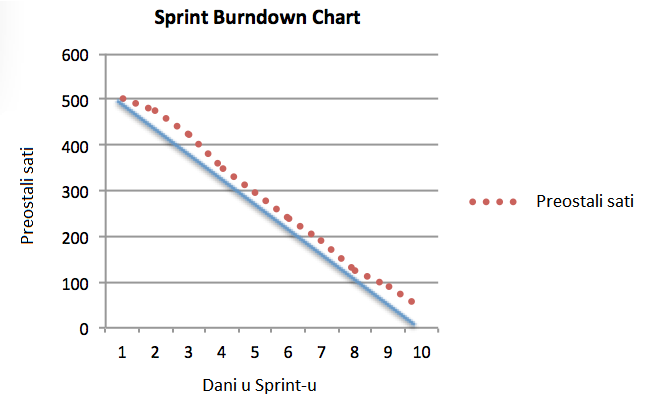
Slika 7 daje prikazuje primer izgleda Sprint Backloga-a. (Pham & Pham, 2011)

Slika 7 Sprint Backlog



Kao i u slučaju Product Backlog-a, i ovde imamo alat za vizualizaciju - ***Sprint Burndown***. Tokom određenog Sprint-a, kada se izvrši ažuriranje preostalih sati po zadatku na dan, može se videti realna mogućnost završetka zadatka do kraja Sprint-a. Zajedno sa dnevnim sastankom (Daily Scrum), Sprint backlog i i ovaj alat pomažu timu da sagledaju da li postoje određena odstupanjam i da se fokusiraju na korekciju istih. Sprint Burn down “spaljuje” sate po danima u Sprint-u. (Viscardi, 2013) Slika 8 prikazuje opisani alat.

Slika 8 Sprint Burndown chart



Na kraju, sumaran prikaz navedenih artefakata daje Slika 9.

Slika 9 Sumaran prikaz artefakata



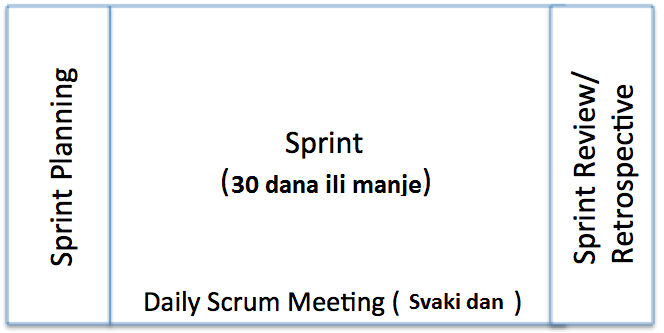
## Vremenska osnova

Kako bi model bio potpun, neophodno je predstaviti i vremenski okvir Scrum metode. Njeha čine sledeće komponente: (Viscardi, 2013)

* Vremenska kadenca – Sprint;
* Planiranje Sprinta – Sprint Planning;
* Dnevni sastanak – Daily scrum meeting;
* Sastanak za pregled – Sprint review meeting;
* Retrospektiva – Sprint Retrospective;
* Planiranje lansiranja proizvoda – Release Planning;

Vremenski okvir prikazuje modela prikazuje Slika 9. (Viscardi, 2013)

Slika 10 Vremenski okvir Scrum metode



### Sprint

Sprint predstavlja vremensku iteraciju, definisanu fiksnim početkom i završetkom, koja traje 30 dana ili manje. (Viscardi, 2013) Jeff Sutherland Sprint naziva ciklusom i opisuje ga sledećim rečima: „*Na početku svakog ciklusa se održava sastanak na kome se planira Sprint. Tim odlučuje o količini posla koju može da završi u toku naredne dve sledeće nedelje. Uzimaju zadatke koji moraju biti završeni sa prioritetizovane liste, i često, zapisuju ih na samolepljivi papir i stavljaju na zid. Nakon toga, donosi se odluka o količini posla koja može da se završi tokom tog Sprint-a. Na kraju Sprint-a, tim prezentuje svoja postignuća i vrši uvid u broj obavljenih zadataka. Da li ih je bilo previše u sprint-u i nisu svi završeni? Da li h je bilo premalo? Tada razmatraju na koji način mogu bolje da rade u narednom sprint-u, da li ih je bilo i koje su prepreke koje su uticale na brzinu njihovog rada.*“ (Sutherland, 2014)

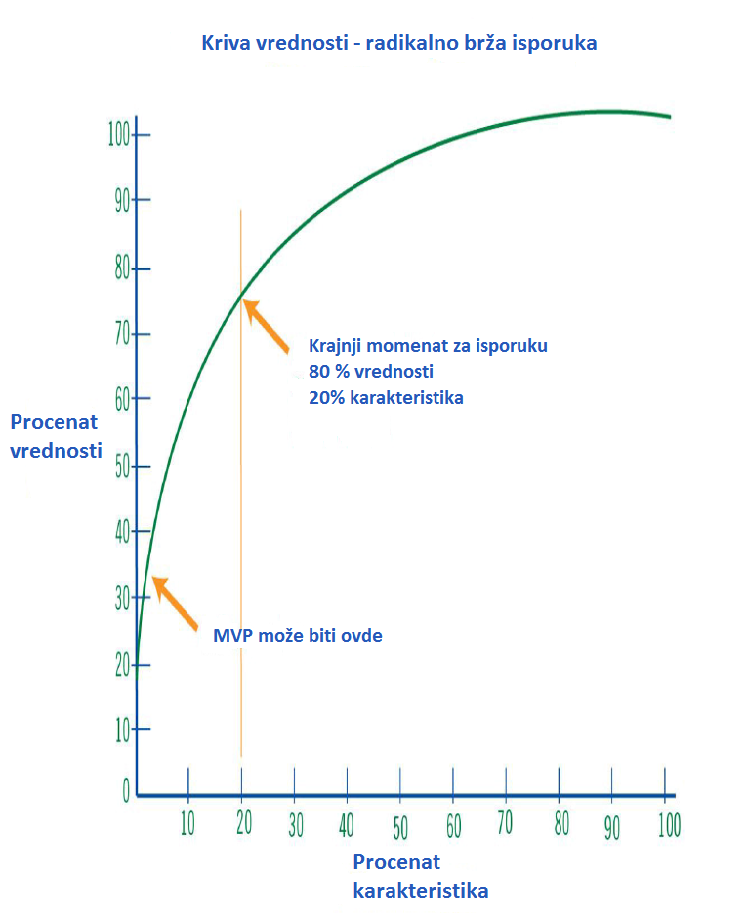
Potrebno je navesti određene smernice vezane za rad u Sprint-u: (Maximi, 2015)

* Ne postoji slobodno vreme između Sprint-ova, oni prate jedan drugog.
* Maksimalno trajanje Sprint-a je četiri nedelje, dok minimum nije defiisan, ali se u praksi obično uzima jedna nedelja kao najkraće moguće trajanje istog.
* Preporuka je da se počne sa dvonedeljnim Sprint-om, a ako postoji problem, postoji mogućnost povratka na jednonedeljni, kako bi se uvideli nastali problemi.
* Potrebno je osluškivati predloge i sugestije samog tima koji učestvuje u ovom procesu, kako bi se poboljšalo samo funkcionisanje tokom Sprint-a.
* Konačno, na kraju svakog Sprint-a je potrebno imati spreman proizvod sa najmanjim brojem karakteristika, koji u datom momentu koji može da zadovolji potrebe tržišta, o čemu će biti reč.

**Minimum Viable Product - MVP**

Scrum metodologija toliko obiluje raznim terminima, od kojih jedan i MVP (Minimum Viable Product). MVP je nešto kao nebrušeni dijamant, sirovije, fleksibilno, podložno izmenama, elastično, nadogradivo i doradivo. MVP je proizvod sa najmanjim brojem karakteristika u datom momentu koji može da zadovolji potrebe tržišta, odnosno klijenta, i koji ima dovoljnu vrednost za isporuku. Dakle, ključne reči su da je isporučiv i to u datom momentu. Isporučiv automatski znači da ima vrednost. Ali to nije ona vrednost koja je cilj u Watterfall metodi, ili proizvod ubica , već ona koja je dovoljna da proizvod služi svrsi, sa razvijenim brojem funkcionalnosti. Postavlja se pitanje, koji je to broj minimalan da bi proizvod bio MVP? Neki univerzalni, zlatni presek, nalaže da to bude 20 odsto od ukupnog skupa funkcionalnosti koje su definisane na samom početku projekta. (Slika 10) (Sutherland, 2014) Istina je da upravo ovaj pristup diktira naporan i brz tempo, informacije stalno pristižu, i malo je vremena da se one pravilno upotrebe, a rad mora da bude tačan i produktivan.

Slika 11 MVP koncept



Ipak, MVP može biti isporučen i pre, odnosno bilo gde u razmaku od 0 do 20 procenata. Ipak, poslednji voz za isporuku na koji tim treba da utovari svoj nebruseni dijamant je trenutak dostignuća ove zlatne petine.

Druga bitna stvar je dati momenat. Kako projekat odmiče, sprintovi se gomilaju, tim će lansirati više ovih proizvoda. Ali, svaki sledeći će biti drugačiji, savršeniji i sa više ugrađenih informacija.

Koje funkcionalnosti će upasti u ovaj ekskluzivni “krug petine”, zavisi od prioretitezacije koja je učinjena još u inicijalnim fazama planiranja, a dogovorena je između tima Ii klijenta. I ovde važi dobro poznato Paretovo pravilo: 20 % funkcionalnih karakteristika nosi sa sobom 80 % vrednosti.

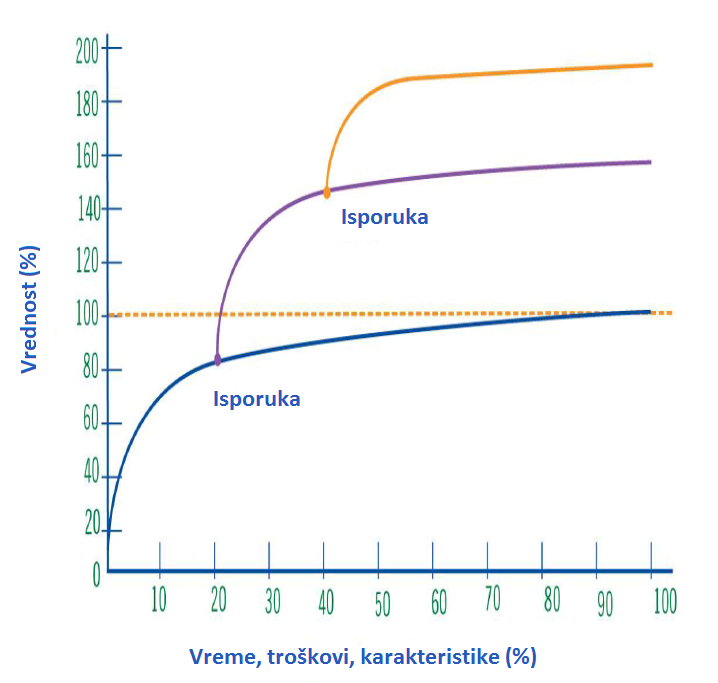
Šta je osnovna vrednost ovog pristupa?

1. Brzina
2. Povratna informacija

Dakle, MVP je jedna baza u vremenu koja se stalno puni novim podacima, a sve na osnovu zahteva i povratnih informacija od klijenta. Znanje se prenosi iz sprinta u sprint, što je glavni princip Scruma. To je suština MVP koncepta. (Sutherland, 2014)

Sa razvojem sve većeg broja karakteristika, raste vrednost. Već je napomenuto da je najpogodniji momenat za isporuku 20 odsto. Isporuka se naravno može učiniti i kasnije, ali ne bez posledica po vreme i novčanik. Naime, troškovi rastu sa vremenom. Navedeno prikazuje Slika 11. (Sutherland, 2014)

Slika 12 Odnos vrednosti i vremena, troškova i karakteristika proizvoda



### Planiranje Sprint-a (Sprint Planning)

Tokom ovog sastanka, Product Owner i tim diskutuju o stavkama najvišeg prioriteta iz Product Backloga, kao i mogućnostima implementacije istih. Odbrane stavke, kao i njihovi prateći zadaci čine Sprint Backlog. Sastanak prodrazumeva dva dela: prvi, kada Product Owner diskutuje o najvažnijim stavkama Product Backlog-a, uz prateće modele, crteže i sl., kao i u razlozima za njihovu realizaciju i drugi kada Tim za isporuku razmatra načine za realizaciju stavki. Početak drugog dela sastanka zapravo početak Sprint-a. Ovaj sastanak traje osam sati, ali se srazmerno dužini Sprint-a može smanjivati. (Viscardi, 2013)

### Dnevni sastanak (Daily scrum meeting)

Dnevni sastanak za svrhu ima prezentaciju napretka tima, kako bi njegovi članovi mogli da izvrše pregled i adaptaciju postavljenih ciljeva, i to na dnevnom nivou, jer se promene dešavaju velikom brzinom. Učesnici na ovom sastanku su Scrum Master i tim, ponekada i Product Owner, a svako drugi može biti samo u ulozi posmatrača. Tokom ovog petnaestominutnog sastanka članovi tima diskutuju šta su uradili do jučerašnjeg sastanka, šta će uraditi do sutrašnjeg, i koje su prepreke na putu k tome.

Uloga Scrum Mastera u ovom slučaju je da olakša funkcionisanje tima, i uzme u obzir prepreke probleme koji članovi ne mogu da reše, kao i da konverzaciju usmerava što dalje od rešavanja problema, koje će on rešiti nakon sastanka. Sastanak se održava uvek na istom mestu, u isto vreme. (Viscardi, 2013)

### Sastanak za pregled (Sprint review meeting)

Još jedan u nizu sastanaka koje Scrum podrazumeva je Sprint review, čija je svrha razmena informacija između Scrum Mastera, Product Ownera i članova tima o isteklom Sprint-u. Revidira se sve što je urađeno, analizira napredak razvoja proizvoda, koje funkcionalnosti je potrebno promeniti, koje odluke su donete, itd. Svaki sprint review treba da uključuje kratak demo o ispunjenju siljeva Sprinta. Vremensko trajanje sastanka je oko četiri sata za Sprint od 30 dana. (Viscardi, 2013)

### Retrospektiva (Sprint Retrospective)

Tokom završnog sastanka, tj. retrospektive, tim vrši retrospektivu svih događaja tokom Sprint-a, i šta je bilo dobro, i šta nije tokom njihove realizacije. Pored toga, preduzimaju se aktivnosti koje će rezultirati promenama koje tim želi da sprovede u narednom Sprint-u. Scrum Master se brine za prepreke koje tim ne može samostalno da prevaziđe, i izveštava tim o napretku u tom domenu, u narednim Sprintovima. Preoblem su često različita organizaciona pitanja, koja predstavljaju poteškoće za tim. Pitanja koja se postavljaju u ovom momentu su: (Anonymos, 2008)

* Objektivna pitanja (Šta se dogodilo?).
* Reflektivna pitanja (Kako se osećamo povodom toga?).
* Interpretativna pitanja (Šta to znači?).
* Pitanja odluke (Šta ćemo da radimo povodom toga?.)

Trajanje ovog sastanka je do tri sata. (Viscardi, 2013)

### Planiranje lansiranja proizvoda (Release Planning)

Planiranje lansiranja je opcioni sastanak u Scrumu, obično se održava na samom početku projekta. Product Owner i tim raspravljaju o svim stavkama Product Backlog-a, a posebno o onim koje se odnose na rok lansiranja. Prema tome, product backlog ima svoj podskup koji se naziva release backlog, koji omogućava timu da se fokusira na bitne stvari u datom momemntu. Neretko, kako vreme odmiče, timovi prolaze sprintove i stiču nova znanja i iskustva, što im daje više samopouzdanja da odlože planirano lansiranje, i da razvoj posmatraju dugoročnije. Koliko će ovaj sastanak trajati, zavisi od mnogo činioca, broja timova, njihove alokacija i koliko je Backlog razrađen. Ovaj sastanak se periodično ponavlja u toku trajanja projekta. (Viscardi, 2013)

# IMPLEMENTACIJA SCRUM METODE U PREDUZEĆU

Kako bi se izvršila određena poboljšanja u preduzetničkom tipu organizacija, i uspostavili mehanizmi koji će omogućiti razvoj i prelazak u stabilno stanje poslovanja, sam proces implementacije Scrum metode mora biti pažljivo realizovan. Detaljno razmatranje potreba same organizacije i posvećenost, rezultovaće dobrom osnovom, koja će omogućiti da preduzetnička, mlada organizacija, dođe do tačke transformacije u organizaciju sa razvijenim, stabilnim poslovanjem. Svkak organizacija je različita, te su i njene mogućnosti raznolike, pa postoje različiti oblici i pristupne tačke implemetacije. No, koraci implementacije su univerzalni, i mogu se voditi sledećom procedurom, koja može biti i okvir za rešavanje drugih tipova problema: (Kotter, 2012)

1. Uspostaviti osećaj hitnosti;
2. Formirati vodeću koaliciju;
3. Razviti strategiju, uz prethodno kreiranje vizije;
4. Komunicirati promenu vizije;
5. Osnažiti zaposlene za preduzimanje različitih akcija;
6. Ostvariti kratkoročne pobede;
7. Konsolidovati postignuća i sprovoditi više promena;
8. Učvrstiti nove pristupe u kulturi.

Potrebno je proći kroz sve faze, linearno, ali to ne isključuje mogućnost istovremene aktvinosti u nekoliko faza. Takođe, periodi tranzicije između faza su drugačiji u svakoj organizaciji, i promene same organizacije mogu trajati od tri do deset godina. Na kraju, kao i uvek, sve zavisi od od same organizacije. (Maximi, 2015)

## Pristupi

Postoji nekoliko mogućnosti za implementaciju metode, što sve zavisi od željenog oblika. U narednom delu rada biće prikazane različite pristupne tačke, kao i njihove prednosti i nedostaci. Postoje tri različita pristupa: (Maximi, 2015)

* Top-Down – odozdo na dole,
* Bottom-Up – odzgo na dole,
* Submarines – “Podmornica”.

### Top-Down pristup

Ovaj pristup podrazumeva podršku upravljačke strukture od samog početka, i odatle sam njegov naziv. Dakle, kreće se od najviše tačke koju predstavlja menadžment, spuštajući se na dole, na niže hijerarhijske nivoe. Na duže staze, uspeh je moguć samo ako postoji podrška menadžmenta, iako je poznato da bitku dobija vojska, ne generali. Ovim pristupom se može implementirati bilo koji oblik Scrum-a. (Maximi, 2015)

Koristi ovog pristupa su očigledne: jasan mandat, budžet, pripremljeno osoblje, jasan cilj i pordška menadžmenta će implementaciju Scrum-a voditi napred, u progres. Tu je i pristup obuci koju omogućava upravljačka struktura, zatim spoljnim konsultantima i drugim korisnim alatima. Sa druge strane, stvara se pritisak menadžmenta, jer investicija u gore navedene benefiti mora biti oopravdana, jer niko ne pristupa projektu ako ne očekuje neku korist. Takođe, potrebno je ubediti tim za razvoj metode i ostale zaposlene da je čitav postupak koristan i za njih. Sledeći problem ovog pristupa je uverenje da postoji “srebrni metak” za neuspešne projekte, što dovodi do prevelikih očekivanja po pitanju efektivnosti i uspešnosti implementacije. Krivci za ovo su nerealna očekivanja zaposlenih, ali i konsultanti koji daju obećanja i garancije za povećanje produktivnosti, samo da bi privukli mušterije. Navedeno je pogrešno, jer sama metoda daje transparentnost, ali za produktivnost je potrebno otkloniti različite prepreke. (Maximi, 2015)

### Bottom-up pristup

Najčešće, implementacija Scrum-a se vrši odozdo na dole. Inicijatori ovoga su iskusni programeri ili menadžeri za razvoj, koji metodu vide kao rešenje za njihove probleme. Dakle, ide se od nižih hijerarhijskih nivoa, što daje totalno drugačiji karakter celokupnom pristupu. (Maximi, 2015)

Prednost ovog pristupa je visoka motivacija tima i “mi to možemo stav”, za razliku od drugih pristupa koji podrazumevaju iscrpne razgovore i ubeđivanja zaposlenih. Pored toga, početna faza uvođenja metode je uspešna dok proces ne dosegne svoje prirodne granice. Kada se određeni proces jednom nađe izvan domašaja tima, organizacija umanjuje entuzijazam zaposlenih i sprečava dalje promene. Ovde dolazi do merenja koristi između zadržavanja postojećih pristupa i implementacije Scrum-a. Određeni procesi su dobro služili u prošlosti, i dolazi do straha da će njihova promena ili eliminacija biti manje vredna nego potencijalne koristi koje Scrum donosi.Takođe, rivalstvo između menadžera na može ugroziti proces.

Jedna zanimljiva pojava u ovom pristupu su i hibridni procesi, kada se određeni projekat vođen Scrum pristupom implementira u tradicionalan pristup organizovanju, što se naziva “Water-Scrum” ili “Water-Scrum-Fall”. Odlike ovog pristupa su timovi koji su podeljeni tako da rade ne na proizvodima ili njihovim karakteristikama, već po kriterijumima tehničkih tema, sa mogućnošću outsourcinga nekih faza.

Na kraju, ovaj pristup omogućava implementaciju Scrum PRN, Faҫade Scrum i Virtual Scrum Studio oblika. (Maximi, 2015)

### Podmornica

Svaka dobra stvar ima svoje problematike, pa tako i Scrum. Jedna od njih su takozvane “podmornice”, simbolična predstava za sve ono što se odvija ispod površine u jednom Scrum timu. Generalno, ljudi imaju težnju da rade ono što smatraju da je ispravno, prvenstveno iz razloga što previše veruju u sebe i skloni su da “teraju po svom“, što često opravdavaju mišljenjem da je to najbolje za tim. Scrum tim je sastavljen od stručnjaka iz različitih sfera, pa se ovde upliće i takozvana ekspertska moć. Dakle, postoji opasnost da u okviru firme imate problem sa malom Scrum “sektom”. (Maximi, 2015)

Ovi submehanizmi nisu nimalo naivni, i lako mogu ugroziti poslovanje. Gledano sa pozicije top menadžmenta, snosite odgovornost za dobrobit kompanije, a pre svega vaših zaposlenih. Pored svakodnevnih problema na koje se nailazit u borbi za opstanak biznisa, potrebno je izaći na kraj sa timom koji radi na važnim projektima, a otima se kontroli. Pored toga što su ovo podmornice (submarines), tim se može pretvoriti u pravu subkulturu u organizaciji. Ovi buntovnici mogu koštati živaca i novca ako se na vreme ne zauzmet duboko kritički stav prema ovoj vrsti ponašanja i obavljanja posla. Ne bi trebalo da uzimaju stvari u svoje ruke kada za to nisu nadležni, i da pozivaju na autonomiju i otcepljenje.

Osnivanjem biznisa, postavlja se i neka stabilna i jasnu mreža procesa sa međusobnom uslovljenošću. Podmornice čine da u jednom trenutku shvatite da deo sistema radi potpuno nezavisno od prvobitnog sistema procesa.

Na sreću, kao i prave podmornice, funkcionišu sve dok ne budu razotkrivene, tada su im šanse za opstanak ravne nuli. Preporuka je izbegavati ovaj pristup, u globalu. Podmornice su opravdane samo u slučaju kada su preko potrebno sredstvo za međusobno upoznavanje tima i brži uvod u projekat i uklopanje u organizaciju, kao i kada ponestanu sve druge opcija. Ipak, i tada, treba se truditi da one isplivaju na površinu što je brže moguće, kako bi se predupredili svi probleme koji su prethodno navedeni, i to alarmiranjem top menadžmentu. Još jedan od razloga koji ide u prilog tvrdnji da izbegavanju ovi podvodnih i “podlih” mehanizama, leži u činjenici da na taj način jedini oblik Sruma koji se može postići je Scrum PRN. Prema tome, potrebno je koristiti “transparentnija” prevozna sredstva. (Maximi, 2015)

## Oblici

### Scrum PRN

Scrum PRN predstavlja oblik metode koja se koristi u organizacijama bez formalne podrške, i koji predvođen ljudima koji su na nižim hijerhijskim nivoima. PRN predstvlja skraćenicu od “pro re nata”, i u prevodu znači “uzeti koliko je potrebno”. Sama implementacija uglavnom funkcioniše nesvesno ili viralno, uspehom tima za razvoj, koji svoje rezultate postižu upravo ovom metodom, pa ose ostali timovi ugledaju na njega. U drugim situacijama, upravljačka struktura može biti svesna postojanja ovog oblika,iIi može zahtevati njegovo korišćenje u kritičnim situacijama

Organizacione promene su ograničene na nivo tima. Ipak, sa Scrum PRN-om, postoje teškoće zadržati se na već stečenom znanju. Jednom završen PRN projekat, mukom stečeno znanje se teško zadržava na istom nivou, i ljudi se obično nanovo raspoređuju na druge projekte. Na kraju dana, kompanija ne dobija dodatnu vrednost, na procesnom nivou posmatrano. Takođe, još jedna problematika leži u tome što svaka implementacija Scruma unutar organizacije se razlikuje od ostalih, a ne postoji sumaran, već pojedinačan uvid u svaku od njih. Svi timovi nezavisno jedni od drugih prave greške, stvaraju sopstvenu dobru praksu i postižu uspeh ili neuspeh. Kompanije koje koriste Scrum PRN troše značajan potencijal, ali bez sumnje mogu biti veoma uspešne. (Maximi, 2015)

### Façade Scrum

Ovaj oblik Scrum metode je prvi čija implementacija obuhvata preduzeće u celini, za razliku od prethodnih koji su bili implementirani samo u određenim segmentima. Formiranje osnove ovog oblika je inicirano od strane upravljačke strukture. Koncept se zasniva na tome da se postojećim procesima u organizaciji daju novi “agilni” nazivi, što rezultira niskom vrednošću poslovanja. Dakle, postoji fasada kreirana od naziva, a ispod nje su procesi koji ne poštuju vrednosti i praksu agilnosti. Iteracije koje su osnova rada se samo formalno nazivaju sprintovima, ali ovde ne postoje ljudi zaduženi za testiranje i analizu, već samo programeri, i ne postoji definicija završenog proizvoda. Ne postoji proizvod koji je nakon svake iteracije potencijalno spremna za lansiranje, već se iteracije koriste u maniru |Waterfall metode, kao sledeća faza nedovršenog proizvoda.

Što se tiče uspeha, timovi korišćenjem ovog pristupa ne postižu povećanje produktivnost i kvaliteta, a ponajmanje lično zadovoljstvo. Potrebno je naglasiti da Façade Scrum može nastati kao posledica korišćenja Virtualnog Scrum Softver oblika, i ako je cilj bio postizanje agilnosti procesa, te treba biti pažljiv pri implementaciji. (Maximi, 2015)

### Scrum Softver Studio

Scrum Softver Studio predstavlja oblik čija je funkcija razvoj organizacije kroz učenje, i to u vidu institucionalne forme pravnog lica, koja pripada matičnoj kompaniji, i koja podržava projekte u istoj. Pravila koja ovde važe su inovacije i eksperimenti, uz podršku materijala za rad, različitih metrika, pa čak i radnih mesta za timove. Za svaki projekat organizacija donosi odluku da li će ga realizovati sa tradicionalnim procesima, ili Scrum metodom. U ovom drugom slučaju projekat se realizuje u studiu, koji je fizički odvojen od matične strukture. Studio je formiran tako da ne utiče na postojeći poredak, što je smanjuje uticaj na zaposlene koji tada ne pružaju otporu, jer se pravi distinkcija između struktura.

Za projekte koji će biti realizovani u okviru studija, važe određena pravila: (Maximi, 2015)

1. Svaki projekat će biti podržan Scrum procesima, njegovim empirijskim principima,  
   inteligencijom sa dna ka vrhu i samo-organizacijom.
2. Svaki projekat će imati Scrum razvojni tim sa vlasnikom proizvoda,  
   Scrum Masterom, i ne više od devet programera.
3. Scrum Master mora biti iskusan u upravljanju projektima u oblasti Scrum metode, uz prihvatanje smernica od Scrum trenera iz studija.
4. Vlasnik proizvoda će aktivno raditi sa timom na formulisanju zahteva, pregledu rada i zadataka i empirijskom prilagođavanju u cilju optimizacije vrednosti projekta i ostvarenja njegove vizije.
5. Razvojni tim će se sastojati od programera sa svim veštinama  
   neophodnim za stvaranje poboljšanja potencijalno upotrebljivih funkcionalnosti, baziranih na zahtevima vlasnika proizvoda.
6. Tokom projekta, prethodni sistem izveštavanja će privremeno biti neaktivan.
7. Svaki napredak će zadovoljiti definiciju “transparentnog” i “potpunog”, koju nalaže Scrum metoda.
8. Scrum tim će koristiti savremene inženjerske prakse i alate koje pruža Studio, i ako je neophodno, realizovaće se obuka za korišćenje.
9. Projekat će biti u skladu sa standardima organizacije, kao i sa politikama, procedurama i standardima Studija.
10. U najvećoj mogućoj meri Scrum tim će biti povezan sa Studijom. Zaposleni će raditi puno radno vreme na projektu.
11. Scrum tim će korisititi metrike koje nudi Studio, u cilju poboljšanja svog rada.
12. Članovi tima će dati doprinos formiranju osnovama znanja koje nudi Studio, koji se bazira na iskustvu stečenom kroz rad na projektu.

Prihvatanjem navedenih pravila, pruža se mogućnost formiranja značajne baze podataka i znanja, korisnih za buduće projekte. Dalje, promene u kulturi, načinu razmišljanja i pristupu su vezane samo za Studio, što ne remeti ostatak organizacije. Što se tiče nedostataka, dolazi do situacija da projekti koji bi bili pogodni za korišćenje Scrum-a isti ne primenjuju, jer je pristup Studiju dobrovoljan. Departmani kao što su tehnička podrška ili kontrola ne učestvuju uopšte u ovom pristupu, pa dolazi do neiskorišćenog potencijala. Na kraju, kako se način razmišljanja ne menja u organizaciji, dolazi do konstantnih konflikata između postojeće i nove, opisane prakse. (Maximi, 2015)

### Virtualni Scrum Softver Studio

Ovaj oblik Scrum metode predstavlja podvrstu Scrum Softver Studija, i njegova primena u organizaciji podrazumeva slaganje sa već ustanovljenim pravilima poslovanja. Kompletna organizaciona struktura matične kompanije ostaje u svom prvobitnom obliku, i implementacija Scrum-a se realizuje kroz dodavanje segmenta za upravljanje projektima (npr., kancelarija za upravljenje projektima) postojećoj strukturi i osoblju. Pri svakom projektu donosioci odluka imaju izbor između Scrum i tradicionalnih metoda za upravljanje projektima. Kada se započinje projekta korišćenje Scrum-a, uspostavljaju se standardni procesi uz dodatak korisnih alata kao što su liste za proveru, baze podataka i različiti tipovi obuka.

Inicijativa za primenu ovog oblika dolazi obično od strane par ljudi koji su imali prilike da sagledaju vrednost istog, ali nisu uspeli da prenesu na celu organizaciju. Prednost ovog pristupa je činjenica da on ne zahteva velike izmene u organizaciji, što je pogodno sa aspekta ljudskog faktora. Sa druge strane, tu su problemi: pravila koja prate ovaj pristup ne mogu biti prisilno primenjena, te tokom tog procesa dolazi do smanjenja produktivnosti i smanjenja zadovoljstva zaposlenih. Prostor za rad i ostali resursi su često u deficitu. Kako projekti normalno teku, i koristi se drugačija procesna dokumentacija, koju vrlo verovatno neki projektni menadžeri nisu ni pročitali, transparentnost kroz organizacione procese se takođe ne povećava. Dakle, ne postoje prečice za Scrum Studio menadžera. Promene u naporima za postizanje uspeha nailaze na jak otpor u ovom slučaju, i, pod uslovom da je sve izvodljivo, dugo traju.

Kada se sagleda navedeno, ovakav oblik može biti nepogodan, te se prelazak na realni Scrum Studio oblik treba planirati ispočetka. Naravno, i ovde postoji rizik od povratka na staru praksu, i gubitak koristi koje su stećene primenom virutalnog oblika Scrum-a. (Maximi, 2015)

### Temeljni Scrum

Temeljni oblik Scrum-a podrazumeva implementaciju obuhvata kompaniju u svim poljima poslovanja. Svaki zaposleni je edukovan i poznaje koncepte, vrednosti i praksu Scruma koji je u ovom slučaju shvaćen u potpunosti. Način razmišljanja je fokusiran, orijentisan na korisnike i rezultate, empirijski, iteracioni, i okrenut ljudima.. Kompaniji je potrebno oko pet do deset godina ulaganja u proces promene, da bi postigla ovo stanje, koje rezultira niskom stopom fluktuacije zaposlenih. Najčešće jedan poseban lider pokreće ovaj proces.

Sve dok Temeljni Scrum ne postao Održivi Temeljni Scrum, za ovaj celokupan proces promene je zadužen jedan “šampionski” izvršilac. Ukoliko on napusti kompaniju ili dobije unapređenje, kompanija biva vraćena u pređašnje stanje. Ovo je obično bolje nego prvobitno stanje, ali nije u istoj meri produktivno kao kod Održivog Scrum-a. Dodatno, postoji rizik da organizacija izgubi na fleksibilnosti prema eksternim promenama, jer odlična situacija danas, sutra može predstavljati problem. Pogotovo kada se promene neadekvatno utemelje, što čini da održivost ostane samo nedostižna vizija. (Maximi, 2015)

### Održivi temeljni Scrum

Kao i sa prethodnim oblikom Scruma-a, uspeh je dokazan i stopa fluktuacija je veoma niska. Razlika u odnosu na Temeljni Scrum je što primena koncepta ne zavisi od jedne osobe koja samostalno upravlja procesom, već to sada čini veći broj ljudi. Kompanija nastavlja da se menja i ostaje otvorena za nove ideje, čak i ukoliko lider napusti organizaciju. Koncept promena je duboko usađen u svačiju svest.

Organizacija koja ima ustanovljen ovaj pristup može da se suoči sa bilo kojim izazovom i razvije adekvatna rešenja, uz način razmišljanja koji je fokusiran na proveru iIi adaptaciju. Kroz transparentnost i precizan fokus, blagovremena identifikacija problema i produktivno razmišljanje su osigurani. Svi zaposleni su visoko motivisani, proseci se prilagođavaju novim okolnostima, ali vrednosti ostaju netaknute, što pruža značajnu konkurentsku prednost. Menadžment organizacije više nije u obliku višeslojne piramide, već se “obavija” oko nezavisnih, ali međusobno povezanih jedinica, koje imaju zajedničku viziju. Generalno, ovim oblikom se postižu najveće benefiti za preduzeće, te se ističe kao najuspešniji među prethodno opisanim. (Maximi, 2015)

### Povezanost ciljeva i različitih oblika Scrum metode

Pre nego što se započne proces implementacije Scrum-a u preduzeću, potrebno je doneti odluku o željenom obliku. U zavisnosti ko je donosilac ove odluke, i na kom je položaju, i kakva je podrška upravljačke strukture i sveopšti cilj, rezultati mogu da variraju. Potrebno je postaviti sledeća pitanja: (Maximi, 2015)

1. Šta želim da postignem u mojoj trenutnoj situaciji uvođenjem  
   Scrum-a?
2. Šta kompanija želi da postigne sa uvođenjem Scrum-a?
3. Ko je predvodnik implementacije Scrum-a i na kom položaju je u organizaciji?
4. Ko je podrška poduhvatu implemetacije?
5. Koliko timova i osoba je pod direktnim uticajem procesa?
6. U kom vremenskom periodu bi rezultat implemetacije trebalo da bude vidljiv?
7. Koliki je budžet na raspologanju?
8. Na koje se eksperte može računati?

Scrum PRN se upotrebljava kada je cilj brzo spašavanje veoma kritičnog projekta. Iako se smanjuje potencijal za dugoročni dobitak, ostvaruje se uspeh na trenutnom projektu. Čak i ako ne postoji ovalšćenje za uvođenje drugog oblika Scrum-a, uvek se može početi sa Scrum PRN oblikom.

Virtualni Scrum Software Studio se koristi u situacijama kada već postoji određeno iskustvo sa Scrum-om, ali menadžment ne želi da omogući kulturne promene ili ne daje svoju punu podršku. Iako omogućava transparentnost, strkturirano sakupljanje iskustva i produktivnost, ovde dolati do konstantnih sukoba bazične i nove kulture. Ovo se rešava primenom Scrum Software Studija, gde postoji izdvojeni entitet u kojem postoji mogućnost primene pravila i standarda koji su različitih od bazičnih.

Na kraju, ako postoji određeno iskustvo u ovom domenu i podrška upravljačke strukture, stvara se potencija za upotrebu Temeljnog Scrum-a. To znači nov način razmišljanja i povećanje prdokutivnosti, ali je neophodno strpljenje, često od pet do deset godina. Neophodna je podrška i razumevanje celokupne hijerarhije.

## Metode

U zavisnosti od konkretne situacije i projekta, timovi koji učestvuju u primeni Scrum-a, koriste različite metode za obavljanje svojih savkodnevnih aktivnosti. U narednom delu rada biče predstavljenje neke koje se najčešće koriste, ali lista nije konačna, i podrazumeva mogućnost dopune.

### Sastanak za procene

Ovaj sastanak ne bi trebalo da potraje duže od dva sata. Održava se kako bi Product Owner i članovi tima izložili svoje procene o vremenskom trajanju svake pojedinačne aktivnosti, odnosno zadatka, iz Product backloga. Svi moraju da daju svoje mišljenje o tome koliko će potrajati razvoj svake funkcionalnosti. Učestvovanjem u procenama, svi članovi dele odgovornost, i izbegava se situacija u kojoj neko može da prigovori kasnije da je za neki zadatak trebalo manje ili više vremena. Ovo s obzirom da se sve procene zajednički analiziraju i sumiraju u jedinstvenu za svaku stavku backloga. Kada Product Owner sazove sastanak, pretpostavka je da su tehnička pitanja već rešena od strane developera. Zatim se koristi jedna nekonvencionalna metoda planiranja i odlučivanja – Poker igra planiranja. (Maximi, 2015)

Sastanak za procene nema utvrđenu učestalost, već se zakazuje po potrebi. Scrum Master ima pravo da prekine sastanak za planiranje sprinta ukoliko se Product Owner pojavi sa nedovršenim procenama backloga. Na kraju krajeva, ova predviđanja su značajna za utvrđivanje mogućih tačnih datuma završetka proizvoda i troškovima koji su s tim povezani. (Maximi, 2015)

### Poker – igra planiranja

Poker u ovom kontekstu predstavlja alat za planiranje, tj. procenu trajanja zadataka koje obavlja Scrum tim. U ovu igru je uključen ceo tim, ali Product Owner može biti samo slušalac ili posmatrač.

Prva runda igre podrazumeva sledeće: na početku igre svaki učesnik dobija špil karata, u kom je na svkoj karti napisan set određenih procena, kao na primer Fibonačijev niz - 1, 2, 3, 5, 8, 13, 21, 34, 55 ili više. (Pries & Quigley, 2010) Scrum Master vodi beleške i priprema ovaj špil karata pre igre.

Dalje, Scrum Master usmeno prezentuje sadržaj svakog User Story-ja koji zahteva procenu, i odgovara na sva pitanja članova tima. Nakon toga, svaki član tima vrši odabir karte koja sadrži procenu koja se slaže sa njenim ili njegovim mišljenjem o dužini trajanja zadatka. Dužina trajanja zadatka se izražava u bodovima za estimaciju (Story Points). Kada su svi odabrali kartu, one se prikazuju u isto vreme. Ako procene značajno varijaju, i devijacija je prevelika, članovi tima sa najvećom i najmanjom vrednošću procene daju usmene argumente za iste. Ideja je da se stvori atmosfera slobodnog razmišljanja, jer su možda oni videli što drugi članovi tima nisu. Sa druge strane, ako ne dolazi do varijacija u proceni kroz igru, može značiti da je User Story nekoncizan i neupotrebljiv (Pries & Quigley, 2010)

Ako se nakon tri runde igre ne postigne konsenzus, kao pobednik se uzima procena najdužeg trajanja, a često i ona sa najvećim brojem ponavljanja. (Maximi, 2015) Igra podstiče dijalog, eliminiše grupno mišljenje i omogućava mišljenje stručnjaka.

### Poker – igra planiranja kroz apsolutne vrednosti

Osvrnuvši se na prethodnu igru, koncept ostaje isti, s razlikom da se ovde procena vrši u apsolutnim brojevima. Dakle, nema više poređenja među procenama, već je cilj izvršiti reaalnu procenu, u definisanim vremenskim jedinicama. Vremenska jedinica može biti na primer *sati po osobi*, i nju definiše tim, koji nakon toga koristi kartice za procenu trajanja implementacije određenog zahteva.

Navedeno funkcioniše samo kada je u pitanju procena kratkoročnih zahteva, kao što su zadaci u Sprintu, nikako procena stavki iz Product Backlog-a. Svrha korišćenja ove varijacije igre je sinhronizovanost, i činjenica da članovi tima formiraju konačnu procenu istovemeno, uz aktivno ušešće svih. I ovde, kao i u prethonom slučaju, svi imaju pravo na mišljenje, čim se postiže bolji brainstoming. (Maximi, 2015)

### Vremenska kutija – Timebox

O ovom konceptu je već bilo reči, ali iz iz perspektive podloge vremenskog okvira Scrum-a. Timebox je specifični period vremena, određenog trajanja, koji ne podržava produžetak trajanja. Predstavnici ovoga su dobro poznati Scrum sastanci i svaki Sprint, pa ča i individualne aktivnosti tokom sastanaka, kao na primer radionice od tri do pet minuta. Timebox ne podrazumeva produžetke, jer bi to poremetilo discipline, ali zato postoji mogućnost kreiranja novih. (Maximi, 2015)

### Brzina

Brzina Sprint-a predstavlja raspoložive sate tokom definisanog vremenskog perioda, i zapravo predstavlja tempo tima. (Pries & Quigley, 2010) Ona predstavlja sumu vremena koje je bilo potrebno da se realizuju određene funkcionalnosti tokom Sprint-ova, dok je prosečna brzina aritmetička sredina navedene sume. Merna jedinica za brzinu može biti sat ili Story Point.

Na primer: Na početku Sprinta, tim procenjuje da će razviti tri funkcionalnosti: A, B i C. Ove estimacije su učinjene uz pomoć Planning pokera. Za A funkciju dodeljena su tri, za B osam i za C - 21 Story point-a. Po završetku prvog Sprint-a, tim isporučuje A i B stavke Product Backlog-a u potpunosti, ali zato C nije završena. Nezavršen posao se ne uključuje čak ni delimično/proporcionalno u brzinu Sprinta. Prema tome, brzina se svodi na 11 u ovom Sprintu. U narednom, tim uspeva da razvije C do kraja, ali ništa drugo pored toga. Dakle: (Maximi, 2015)

Brzina tima: Sprint 1: 3+8=11; Sprint 2: 21.

Prosečna brzina tima: (11+21) / 2 = 16.

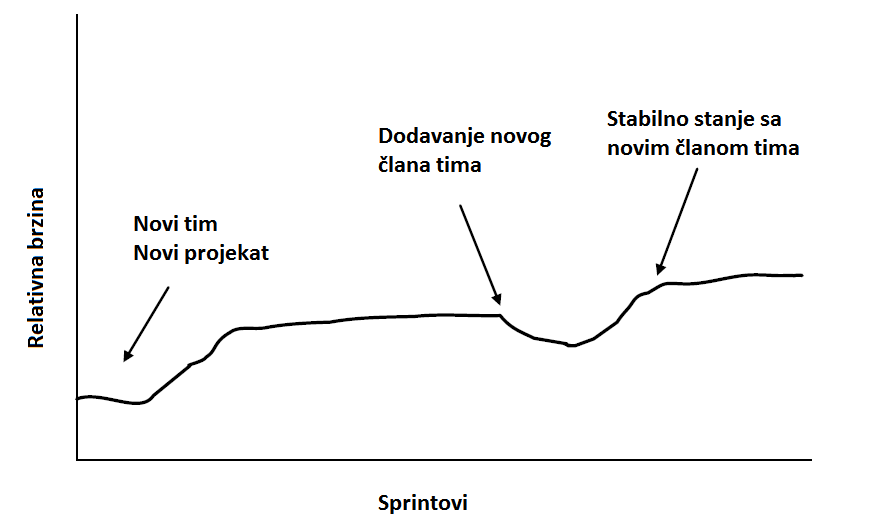
Navedeni pristup omogućava da se odredi u koje vreme i sa kojom verovatnoćom će određena funkcionalnost iz Product Backlog-a biti gotova, koristeći prosek, najbolju i najgori tempo rada. Obično se stavke Product Backloga prema prema brzini, i to nakon što su oba elementa (i stavke Backloga i brzina za svaku od njih pojedinačno) potpuno definisana, i zatim se generiše poslovna vrednost. Bug-ovi itehnički zahtevi, kao i sastanci i organizacioni poslovi se ne ubrajaju u poslovnu vrednost.

Brzina sprinta se menja kako sprint napreduje, pogotovo na početku projekta, kada su neki gubici u efikasnosti uobičajeni, zbog krive učenja. Razlozi za ove gubitke su obično: (Pries & Quigley, 2010)

1. nepoznavanje tehnologije,
2. nepoznavanje ciljeva dizajna,
3. slab timski razvoj.

Scrum Master može da očekuje davremenom tim povećava svoju brzinu rada – pod pretpostavkom da tim napreduje i da nema većih tehnoloških problema. Prema ovim tvrdnjama, funkcija očekivane brzine će vremenom izgledati kao što prikazuje Slika 13. (Pries & Quigley, 2010) Takođe, dovođenje novih ljudi u tim ima momentalan uticaj na brzinu, preko integracije novog igrača, njegovog upoznavanja sa tehnologijom i obimom projekta. (Pries & Quigley, 2010)

Slika 13 Funkcija očikivane brzine tokom Sprint-ova



## Finansijski aspekti

Korišćenje Scrum-a podrazumeva određenu finansijsku podlogu, koja mora biti pažljivo razmotrena, kako bi ključne strane stekle uvid u benefiti upotrebe. U narednom delu biće predstavljeni određeni finansijki koncepti koji čine komponente pomenute podloge.

### Troškovi projekta

Ključni pojam pri izračunavanju troškova projekta je *brzina tima*, koja se ovde može definisai kao broj user Story-ja ili stavki Product Backloga, izraženih u poenima (Story Points), koje tim mora da isporuči tokom jednog Sprint-a. U ovom kontekstu je relevantna zbog izračunavanja troškova ljudskih resursa na projektu. Naredni primer daje detaljnije objašnjenje.

Pretpostavlja se da je brzina tima 20 poena, Sprint traje 4 nedelje, i procena trajanja projekta je 160. Stoga, timu će biti potrebno oko 8 Sprint-ova ili 32 nedelje do završetka projekta. Ako su troškovi plata i bonusa na godišnje nivou 150 000 $. Dakle, troškovi ljudskih resursa iznose ([150,000$ × 32 nedelje]/52 nedelje)= $92,308 $.

Kada se pored navedenih troškova u obzir uzmu i troškovi opreme koja se koristi, mogu se sagledati ukupni troškovi koje jedan projekat zahteva. (Pham & Pham, 2011)

### Investicije

Odluka u ulaganje u određeni projekat zahteva razmatranje određenih pokazatelja, što je i slučaj kod upotrebe Scrum metode pri nekom projektu. U nastavku će biti predstavljeni određeni relevantni pokazatelji, i to u kontekstu razvoja softvera.

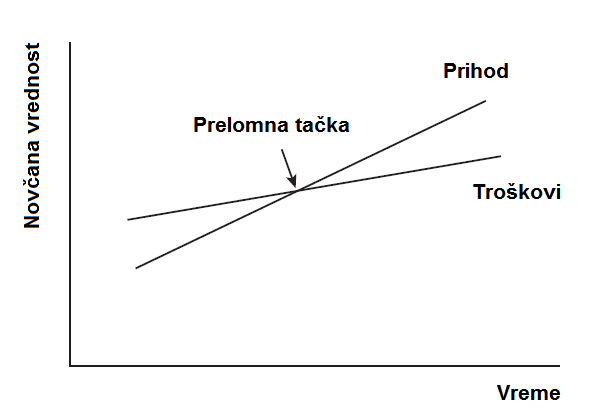
Period povraćaja

Period povraćaja predstavlja dužinu vremenskog perioda potrebnog da kompanija povrati početnu investiciju. Ovaj pokazatelj koji se još naziva i *prelomna tačka* i koji je naročito koristan pri internom razvoju softvera, sučeljava vrednost ivesticije u projekat, sa očekivanim novčanim tokom prihodima ili tokom životnog ciklusa proizvoda. (Slika 14)(Pham & Pham, 2011)

Primer: Vrednost inicijalne investicije je 400 000 $, a procena je da će prihodi iznositi 25 000 $ tokom perioda od tri meseca. Dakle,

Period povraćaja = 400 000 $ / 25 000 $ = 8 tromesečja, ili 24 meseca.

Slika 14 Prelomna tačka



Kupi ili izgradi (Buy versus Build)

Ova tehnika podrazumeva izračunavanje sledećih elemenata: (Pham & Pham, 2011)

* Razlika u ceni = Fiksna cena samostalnog razvoja – Fiksna cena kupovine proizvoda.
* Razlika u mesečnoj naknadi = Mesečna zarada pri kupovini – Mesečna zarada pri samostalnom razvoju.
* Broj meseci = Razlika u ceni / Razlika u mesečnoj naknadi.

Dalje, postoje tri mogućnosti:

1. Fiksna cena samostalnog razvoja > Fiksna cena kupovine proizvoda

Mesečna zarada pri samostalnom razvoju > Mesečna zarada pri kupovini,

*odluka: kupovina proizvoda.*

1. Fiksna cena samostalnog razvoja < Fiksna cena kupovine proizvoda

Mesečna zarada pri samostalnom razvoju < Mesečna zarada pri kupovini,

*odluka: samostalni razvoj, ako kompanija namerava da koristi sofvter manje od izračunatog broja meseci.*

1. a) Fiksna cena samostalnog razvoja > Fiksna cena kupovine proizvoda

Mesečna zarada pri samostalnom razvoju < Mesečna zarada pri kupovini,

*odluka: kupovina proizvoda,* ako kompanija namerava da koristi sofvter manje od izračunatog broja meseci.

b) Fiksna cena samostalnog razvoja < Fiksna cena kupovine proizvoda

Mesečna zarada pri samostalnom razvoju > Mesečna zarada pri kupovini,

*odluka: kupovina, ako kompanija namerava da koristi sofvter manje od izračunatog broja meseci.*

Neto sadašnja vrednost

Neto sadašnja vrednost projekta predstavlja razliku između sadašnje vrednosti budućih prihoda i sadašnje vrednosti budućih rashoda vezanih za projekat. Sadašnja vrednost projekta se izračunava pomoću sledeće formule: (Pham & Pham, 2011)

*SV =*

BV – buduća novčana vrednost projekta

i – diskontna stopa

n – vremenski period

Dakle: *NSV = SV prihoda – SV rashoda*

Opcije su sledeće:

* NSV > 0 – projekat se prihvata.
* NSV < 0 – projekat se odbacuje.
* Ako postoji više projekata u opticaju, odabira se onaj sa najvećom NSV.

Procenat povraćaja investicije

Pri izračunavanju ovog pokazatelja, relevantna su dva pojma: brzina i profitna margina. Kada je u pitanju *brzina*, ona pokazuje kojim tempom kompanija može da ostvari zaradu koja će pokriti investiranu vrednost, nadajući se i određenoj sumi profita. *Profitna margina* je vrednost koju kompanija ostvaruje nakon što izmiri troškove kreiranja i prodaje proizvoda, troškove poreza i kredita, i ostale troškove. (Pham & Pham, 2011)

*Procenat povraćaja investicije = Brzina ᵡ Profitna margina*

Primer: ako je u određeni projekat uloženo 5 000 000 $, a investicija je iznosila 2 000 000 $, pri čemu je profitna margina 10%, sledi:

Procenat povraćaja investicije = ᵡ 10% = 25 %

Kada je u pitanju komparacija više projekata, odabira se onaj sa navjećim procentom povraćaja investicije.

### Nadgledanje performansi projekta

Kako bi u svakom trenutku postojala mogućnost da Scrum tim prati performanse projekta za koji se koristi Scrum metoda, neophodno je napomenuti određene pokazetelje. Za početak, relevantan je pojam ***zarađene vrednosti***.

Zarađena vrednost predstavlja ukupan posao koji je završen do određenog datuma, i za koji je odobren budžet. Dakle, ako je projekata kao osnovu ima budžet od 200 000 $, i završeno je 40% posla, zarađena vrednost je 80 000 $.

U daljem delu će biti prikazani ostali pokazatelji koji kao osnovu u svom proračunu imaju zarađenu vrednost.

Performanse troškova

Cilj ovih pokazatelja je sagledavanje troškovne efikasnosti, kao i predviđanje trendova budućih performansi troškova: (Pham & Pham, 2011)

* **Troškovna varijansa:**

**Troškovna varijansa = Zarađena vrednost – Stvarni troškovi projekta,**

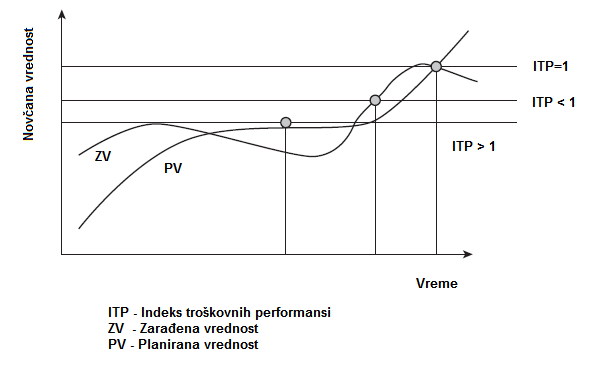
gde su stvarni troškovi projekta deo budžeta koji je potrošen na projektu do određenog trenutka. Pozitivna troškovna varijansa ukazuje da se troškovi kreću u dozvoljenom okviru, dok negativna znači prekoračenje budžeta.

* **Indeks troškovnih performansi:**

**Indeks troškovnih performansi = Zarađena vrednost /Stvarni troškovi projekta,**

čije se vrednosti kreću u opsegu od 0 do 2. Vrednost koja je veća od 1 znači da je zarađena vrednost veća od onoga što je uloženo u projekat, a obrnuto da tim troši više novca nego što stvara vrednost. (Slika 15)

Slika 15 Indeks troškovnih performansi



Performanse rasporeda

Cilj pokazatelja je sagledavanje performansi projekta, iz aspekta njegovog rasporeda, kao i predviđanje trendova u tom domenu: (Pham & Pham, 2011)

* **Varijansa rasporeda:**

**Varijansa rasporeda = Zarađena vrednost – Planirana vrednost,**

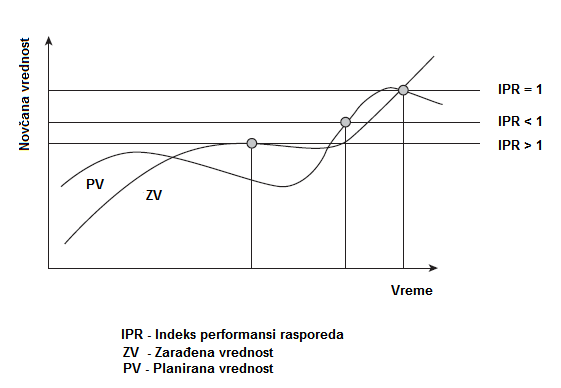
gde je planirana vrednost količina posla koja je planirana i za koju je odobren budžet. Ako je varijansa negativna, to znači da je projekat kasni u odnosu na predviđeni raspored, a ako je pozitivna, važi obrnuto.

* **Indeks peformansi rasporeda:**

**Indeks peformansi rasporeda = Zarađena vrednost / Planirana vrednost,**

čije se vrednosti kreću u opsegu od 0 do 2. Vrednost koja je veća od 1 znači da tim ostavaruje rezultate brže nego što je predviđeno rasporedom, a ako je manja od 1, važi suprotno. (Slika 16)

Slika 16 Indeks peformansi rasporeda



Predviđanje budžeta

Predviđanje budžeta ima za cilj sticanje uvida u stvarne troškove projekta, koristeći pokazatelj zarađene vrednosti: (Pham & Pham, 2011)

* **Procena na završetku:**

**Procena na završetku = Originalan budžet / Indeks Troškovnih performansi,**

dakle, ako je poznat indeks troškovnih performansi, moguće je predvideti konačne troškove.

* **Procena do završetka:**

**Procena do završetka = Procena na završetku – Stvarni troškovi,**

koja pokazuje koliko novca je potrebno da bi se projekat završio.

* **Varijansa na završetku:**

**Varijansa na završetku = Originalan budžet – Procena na završetku,**

koja pokazuje razliku između originalnog budžeta i onoga što je procenjeno.

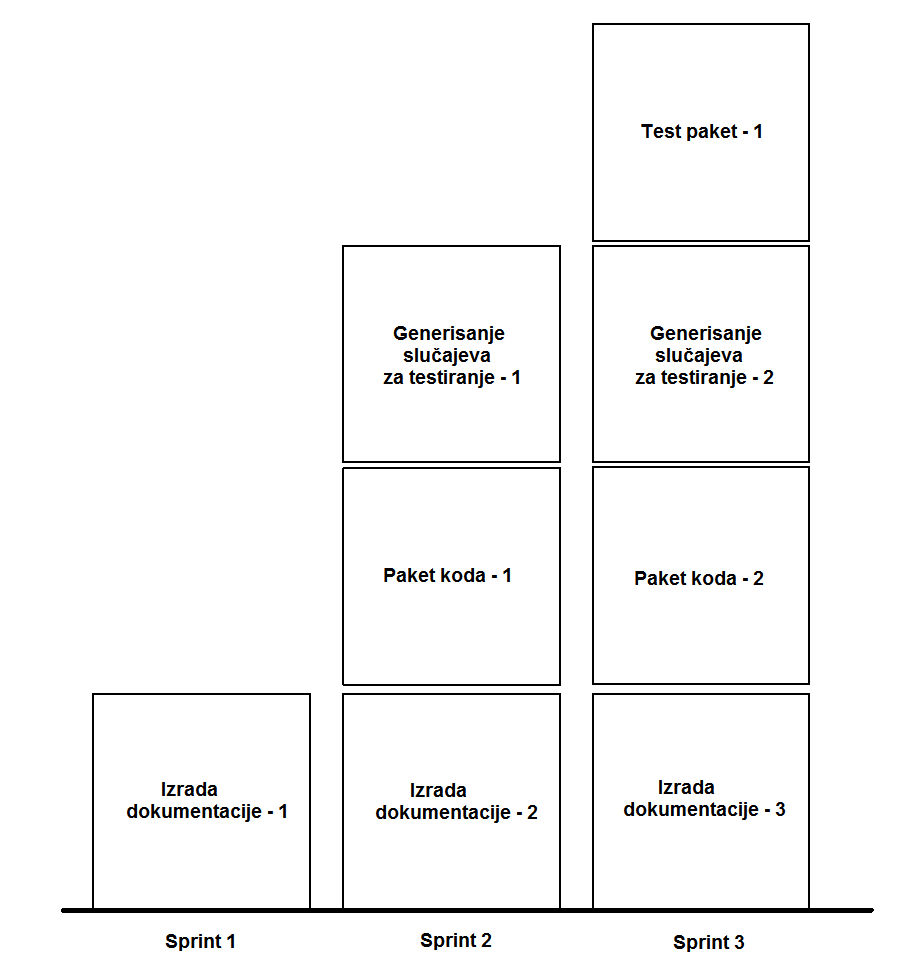
## Dokumentacija

Jedan od najvećih mitova u okviru ovog koncepta je da primena Scrum-a ne podrazumeva formiranje odgovarajuće dokumentacije, tj. da Scrum znači nepostojanje dokumentacije. Naravno, ovo nije istina, ma koliko se teži agilnosti i inovacijama. U tradicionalnom smislu, kada je reč o razvoju softvera, dokumentacija je podrazumevala različite planove projekta, specifikacije zahteva, funkcionalni dizajn, planove testiranja, i slično.

Kada je u pitanju primena Scrum-a, najvažniji document je   
Product Backlog. Srum namerno skraćuje razvojni ciklus specifikacije funkcionalnosti u vidu user story-ja, razgovora sa product ownerom, koda, testiranja I demonstracije – I to u danima, pre nego mesecima.

Prvi Sprint se koristi ta dokumentovanje zahteva vezanih za dizajn, i svaki naredni Sprint se dodaje dokumentaciji, dok se ceo proizvod ne kompletira. Tokom sledećeg sprinta, do revizije dokumentacije, dokumentacija će biti dopunjena novim setom funkcionalnosti proizvoda. (Slika 17) (Pries & Quigley, 2010)

Slika 17 Dokumentacija kroz Sprint-ove



Pored Product Backlog-a postoji nekoliko vrsta dokumenata koji imaju značaja za Scrum tim, i koja mogu doprineti poboljšanju produktivnosti. (Hilton, 2012)(Tabela 3)

Tabela 3 dodatna Scrum dokumentacija

|  |  |
| --- | --- |
| **Funkcionalna dokumentacija** | **Tehnička dokumentacija** |
| Nefunkcionalni zahtevi | Pregled dijagrama arhitekture |
| Model podataka i rečnik podataka | Dijagram komponenti |
| Front-end dizajn | Vodič za primenu |
| Korisnička interfejs mapa | DTAP environment configurations |
| Eksterna specifikacija interfejsa | Vodič za upotrebu |

Pored Scrum tima, tu su i ostale zainteresovane strane. Naročito je značajno imati dobro pripremljenu dokumentaciju, kada se dodaju novi članovi tima. Transfer postojećeg znanja i uvid u celokupan projekat se ne mogu realizovati usmenim putem, već određenim dokumentima. Na kraju, tu su i stejkholderi poput sponzora, marketing osoblja i korisnika, kojima su potrebna sledeća dokumenta: (Hilton, 2012)

* dokument o viziji i ciljevima projekta,
* registar rizika na projektu,
* raspored lansiranja proizvoda,
* sveukupni pregled funkcija,
* marketing brošure.

## Testiranje

Testiranje se u tradicionalnom pristupu nekom proizvodu ili softveru vrši u poslednjoj fazi. Kako je osnova Watterfall metodologije fazni proces, u nedostatku vremena, novca i drugih resursa može doći do prekida u određenoj fazi. Obično se to dešava u poslednjoj fazi, tj. testiranju, i posledica toga je skraćivanje ili totalna elminacija testiranja, pa tako dobri projekti ni ne dospevaju u ovu fazu, na uštrb njihovog kvaliteta. (Rasmusson, 2015)

Kada je reč o Scrum-u, testiranje ima drugačiju ulogu, i realizuje se nakon svakog Sprint-a. Dakle, testiranje se vrši paralelno sa razvojem proizvoda. U narednom delu, kroz prikaz određenih etapa testiranja, biće predstavljenje određene pojedinosti: (Teryokhina, 2013)

* Planiranje testiranja – vrši se pre početka kao i tokom projekta, i recenzira se u svakom Sprint-u.
* Zahtevi – Product Owner i Tim vrše pripremu zahteva za funkcionalnostima, koji se često menjaju, pri čemu opisi istih ne mogu detaljni. Tim za testiranje učestvuje u izmeni zahteva.
* Tes dizajna – Jedinični testovi se koriste za ravijene funkcionalnosti.
* Test zadovoljenja (kriterijumi – tim za testiranje može da izvrši u svakoj iteraciji:
  + pre isporuke – biznis analitičar ili tim za testiranje,
  + posle isporuke - korisnik.
* Tim za testiranje – integrisan je sa razvojnim timom, tj. u njegovom je sklopu.

Važnost definicije “završenog” proizvoda

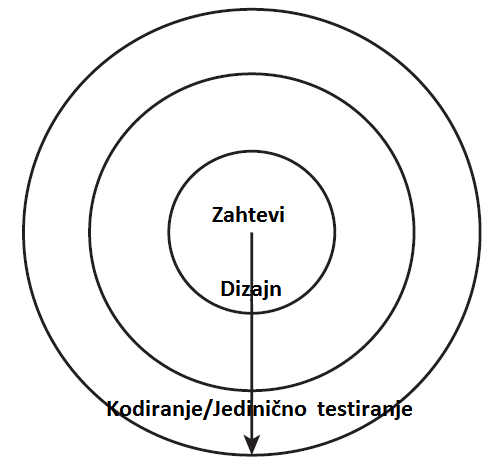
Cilj svakog projekta, tj. razvoja softvera je njegov uspešan završetak, stoga je timu neophodna definicija “gotovog” ili “završenog” proizvoda. Jeff Sutherland nevedeno iskazuje sledećim rečima: „Ono što je potpuno i kompletno završeno i može biti isporučeno bez ikakvog dodatnog rada. To možda nije kompletan proizvod, ali jeste kompletna jednna od njegovih kompletnih karakteristika.“ (Sutherland, 2014)

U sklopu ovog dela, pomenuta definicija je relevantna jer se njom određuje tip testiranja koji tim treba da izvrši, pri čemu se misli na testiranje od strane korisnika ili tehničko testiranje. Naravno, ona vraira od shodno različitim projektima, ali u daljem delu će biti predstavljenje neke od definicija zajedno sa ilustracijama. (Pham & Pham, 2011)

Scenario 1:

Tim donosi odluku da je njegov posao završen na kraju kodiranja i jediničnog testiranja. Ovaj slučaj važi kada tim radi paralelno na istoj proizvodnoj liniji, te vrši isporuku svojih rezultata nakon jediničnog testiranja, kako bi se izvršila integracija njihovog rada pre prelaska u narednu iteraciju. (Slika 14) (Pham & Pham, 2011)

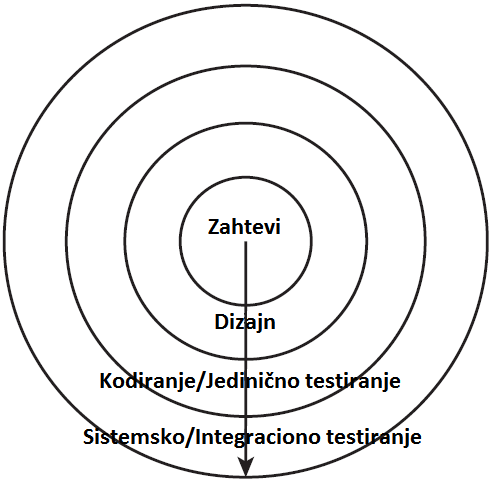
Slika 18 Scenario 1



Scenario 2:

Slika prikazuje uobičajeni scenario gde se na primeru vidi da je tim gotov sa radom samo ako su svi novi story-ji uneti i testirani pre demonstracije Sprinta (Slika 15) (Pham & Pham, 2011)

Slika 19 Scenario 2

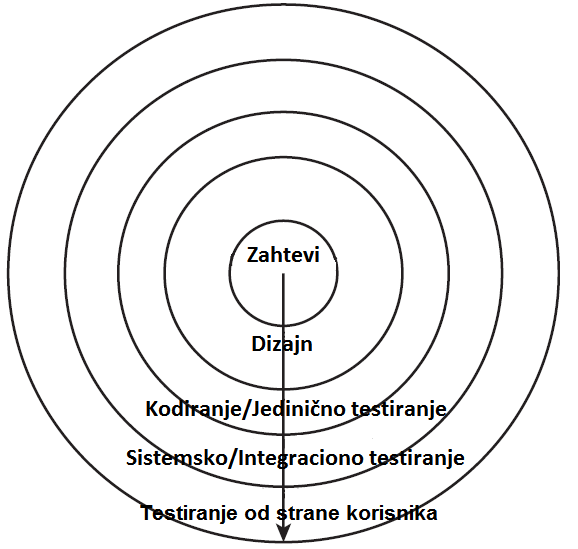
****

Scenario 3:

Ovo je slučaj kada si korisnici deo Scrum tima, i kada su oni odgovorni za realizaciju korisničkog testiranja pre nego što se potvrdi da je proizvod završen. (Pham & Pham, 2011)

(Slika 16)

Slika 20 Scenario 3



Najvažniji testovi

Kako je Scrum najvećim delom okvir rada za projektni menadžment, manje se govori o inženjerskim praksama vezanim za kodiranje i testiranje. Vrste testiranja koje slede trebalo bi da imaju svoje mesto u organizaciji u okviru Scrum metode su: (Pham & Pham, 2011)

1. ***Automatizovano testiranje*** *-* odmah nakon što tim počne da razvija regularne softverske inkremente, njegov tempo se ne može ispratiti bez upotrebe neke vrste automatizacije. Za razliku od manuelnog testiranja, koje zahteva proveru svakog slučaja ručno, automatizovano testiranje može da potraje satima, pa čak i danima, zavisno od formirane baze kodova. Jedna od glavnih koristi automatizovanog testiranja je da će softverski program za testiranje upravljati celim procesom. Naravno, kako bi se obavilo automatizovano testiranje, potreban je određeni trud oko alata za testiranje kao i testiranje svih uzoraka. Ali, to je samo deo vremena koje bi se potrošilo u manuelnom testiranju.
2. ***Kontinualno integrativno testiranje*** *-*ova vrsta testa je od kritičnog značaja za Scrum projekat, iz razloga što pokazuje nivo regularnosti proizvoda, koji bi u svakom trenutku trebalo da bude isporučiv.

## Izazovi i rizici primene Scrum metode

Kao i pri svakoj novoj ideji i podvigu, i u slučaju korišćenja Scrum metode se teži dostizanju maksimalnog uspeha, i sagledavanju celokupnog procesa iz pozitivne perspektive. Kako ništa nije savršeno, i u ovom slučaju se suočavamo sa različitim negativnim stranama i promenama koje nisu uvek u željenom smeru. U nastavku rada će biti izložena i druga strana korišćenja ove metode, sa ciljem što opreznije i uspešnije primene u preduzetničko orijentisanoj organizaciji.

### Promene i izazovi

Scrum metoda neminovno dovodi do određenih izazova i promena u organizaciji: (Schwaber, The Enterprise and Scrum, 2007)

* **Podela osoblja na "za i protiv"**- U najvećem broju slučajeva, gotovo je sigurno da će se 20 % zaposlenih suprotstaviti primeni metode, iz razloga što želje jednostavan način rada, koji podrazumeva najprostiji lanac komande i izvršenja. Rad u timovima, nove relacije, i slične situacije, dovode do problema. Tek nakon određenog perioda primene, zaposleni će biti motivisani da se priključr ovom procesu, sticanjem uvida u progres koji je postignut.
* **Konflikti** – konflikt je znak promene, i očekivana je pojava, jer ljudi imaju različita mišljenja o načinu obavljanja poslova. Neophodno je napraviti koncepciju novog načina funkcionisanja, uz paralelnu obuku zaposlenih, kako bi što uspešnije rešavali konflikte.
* **Promena uloge menadžera proizvoda** – u ovoj ulozi su sada korisinici i Product   
  Owner. Oni su odgovorni za upravljanje projektom, i odgovorni su višem rukovodstvu za uspeh ili neuspeh.
* **Inžinjering je odgovoran za kvalitet** – inžinjering organizacije je odgovoran za izgradnju i razvoj kvaliteta u svakom Sprint-u. Scrum Master neće dozvoliti kvalitet kranjeg proizvoda koji je ispod očekivanog standarda.
* **Promena sistema kompenzacije** – Scrum podrazumeva timski rad, ne izdvajanja individua. Ostavreni bonusi i stala podsticajna sredstva treba da budu namenjena svakom članu tima, i to na osnovu zajednički ostvarenih performansi.
* **Promena posla** – neminovno dolazi do promene i nestajanja radnih mesta, i ljudi dobijaju nove uloge. Štaviše, putevi karijere postaju mnogo bitniji nego doprinos timu i organizaciji.
* **Uloga mendažmenta se menja iz komandovanja u vođstvo** – filozofija i tehnike menadžmenta u slučaju Scrum-a se menjaju. Menadžeri će voditi zaposlene do željenih ciljeva, uklanjanjem prepreka, mentorstvom i podučavanjem. Uloga je roditeljska, jer je u pitanju odgoj, koji za cilj ima zrelost i svesnost, a sve to kroz učenje i iskustvo, ne kroz naređenja.
* **Preokreti na menadžerskim nivoima** – menadžment će takođe proći kroz određene promene, i u pitanju je određeni broj godina. U pitanju je težak i naporan rad, za koji određeni profenat sigurno neće biti zainteresovan, te će zasigurno odustati.
* **Povećanje broja ljudi neće biti rešenje** – uobičajena praksa je da kada postoji želja da se završi povećani obim posla, zapošljava veći broj ljudi. Ovaj potez dokazano nije dobar, iz razloga što dodavanje ljudi timovima, ili razređivanje postojećih stuktura kako bi se našlo novo mesto za svakog člana. Potrebno je fokusirati se na implementaciju metode, ne na dodavanje zaposlenih.

### Rizici

Rizicu u ovom domenu mogu biti različiti, i ne postoji konačan spisak istih. Ipak, timovi treba da obrate pažnju na sledeće činjenice, kada koriste Scrum metodu: (Fatulescu, 2014)

* **Veličina tima:** ona se obično kreće od 7 do 10 ljudi, a sve preko tog broja predstavlja potencijalni rizik. Razlog za to je neadekvatna organizacija sastanaka, koja utiče na kvalitet primene metode. Rešenje je za to je upotreba Scrum-a u Scrum-u, gde se projekat deli na više delova, od kojih svaki ima Scrum Master-a.
* **Mnogobrojni zahtevi:** kako postoji mnogo zahteva za funkcionalnostima na projektu, i kako se oni mogu prenositi na više kanala, njihova koordinacija može biti otežana. Oni mogu biti i kontradiktorni, što može usporiti momenat validacije proizvoda, i sam završetak istog.
* **Kvalitet razvoja:** rizici u ovom domenu se odnose na kvalitet koda i broj grešaka tokom integracije delova projekta. Neophodno je imati rigoznu politiku kvaliteta, kao i plan kvaliteta projekta koji precizno definiše očekivani nivo kvaliteta.

Upravljanje rizicima u ovom domenu omogućuje kombinacija sledećih tehnika: (Gilliland, 2013)

* Relativna estimacija upotrebom Fibonačijeve skale – suštinski daje odgovor na pitanje svih procena sa devijacijom od 50 procenata u okviru estimacionog ranga.
* Kratke iteracije/inkrementi/vreme ciklusa
* Deljenje posla na manje celine – smanjivanje rizika smanjenjem kompleksnosti
* Potrebe za istraživanjem – ako postoje nedoumice u vezi korišćenog pristupa, potrebno je organizujte istraživanje koje će imati za cilj iznalaženje načina kako nastaviti dalje.
* Arhitektura čeličnog niza – finalizacija malog dela neke funkcionalnosti što je brže moguće, kako bi se dokazala efektivnost pristupa.

Identifikacija, praćenje i kvantifikacija rizika pri primeni Scrum-a, ima sledeći izgled: (Gilliland, 2013)

**Ko:** Rukovodilac projekta, Scrum Master, Product Owner, analitičar, uspomoć tima, mogu biti odgovorni za identifikaciju, praćenje i upravljanje rizikom.

**Kada:** Regularni review sastanak treba da bude redovna stavka u agendi za road mapping i planiranje lansiranja (za identifikaciju, a ponekad i kvantifikaciju), planiranje sprinta (za praćenje I upravljanje), i retrospektivni sastanak (ograničavanje tipova rizika kroz proces promene).

**Gde:** Dokumentovano u Backlog-u projekta kao stavka koja se odnosi na rizik. Trebalo bi da bude povezana sa User Story-jima koji su uzročnici ili su najbliži predviđenom riziku.

**Kako:**

* **Tip stavke**: rizik.
* **Pregled:** rizik – rezime.
* **Opis:** opis rizika – uključuje vremenski raspored uzročnika.
* **Verovatnoća:** procenat verovatnoće.
* **Uticaj:** uticaj u smislu procenjenih dodatnih poena (Story Points) koji se dodeljuju user Story-jima ako je moguć rizik.
* **Odgovor:** Moguće planovi odgovora i aktivnosti u slučaju nastanka rizika.
* **Poeni:** uticaj \* verovatnoća.

**Zašto:**

* Smanjenje neplaniranog ili posla koji nema budžetsku podršku, koji se odnosi na različite delove projekta.
* Postavljanje realnih očekivanja u vezi ciljeva projekta, uz smanjenje iznenadjenja.

# PRAKTIČNA PRIMENA SCRUM METODE U START - UP PROJEKTNO ORIJENTISANIM ORGANIZACIJAMA

## Preduzetnička organizacija

Svaka organizacija je preduzetnička na neki način, ali konkretna definicija preduzetničke organizacije ima drugi značaj. Naime, preduzetnička organizacija ili Start – Up, predstavlja jednostavan organizacioni oblik koji obično uključuje jednu veliku operativnu jedinicu, sa jednim ili nekoliko pojedinaca u top menadžmentu. Uglavnom je nestrukturirana i fleksibilna. (Cross, 2015) Primer ovakve organizacije je Start – Up malog preduzeća, gde vlasnik koji je istovremeno i menadžer, vrši kontrolu nad poslovanjem. On se bavi usmeravanjem interpersonalnih odnosa, protoka informacija i sistema donošenja odluka.

Iako u ovom svetu važi koncept fleksibilnosti i fluidnosti, kako bi se formirao željeni oblik preduzetničke organizacije, neophodno je pridržavati se sledećih četiri koraka: (Chamorro-Premuzic, 2014)

1. Zapošljavanje preduzetnički orijentisanih ljudi – kreativni ljudi, sa mnoštvom ideja, spremni da rizikuju i budu proaktivni.
2. Rukovođenje zaposlenima – iako su oni i više nego spremni za rad i stvaralaštvo, potrebna je dobra koordinacija, jer raspoloženje i motivacija “kreativaca“ može varirati.
3. Formiranje preduzetnički orijentisanih timova – svaka inovacija će biti produkt zajedničkog rada, individualni učinak se ne može meriti sa grupnim.
4. Formiranje Start-up kulture – ključni faktor, predstavlja kulturu koja ohrabruje svakog zaposlenog da deluje u skladu sa preduzetničkom filozofijom, čak i ako nije za to prirodno predodređen.

## Osvrt na metode projektnog organizovanja na našim prostorima

Da bi se stekla jasna slika o primeni Scrum metode, potrebno je sagledati i onu suprotnu stranu medalje. Stoga, relevantan je osvrt na realno stanje i praksu preduzetništva u Srbiji po pitanju funkcionisanja upravljanja projektima, i generalno, oblicima ovog segmenta.

Počevši od *uloga* koje su definisane u Scrum-u, na slične nailazimo u našim preduzećima, i njihovoj praksi upravljanja razvojem proizvoda i usluga. Vlasnik proizvoda može biti sama organizacija, ali eksterni vlasnik. Tim za razvoj predstavlja jednostavno tim koji se bavi tehničkim aspektima i i samom izradom proizvoda ili pružanjem usluga. Da razmimoilaženja dolazi kada su u pitanju Scrum Master, menadžer promena i Scrum konsultant, jer su to uloge koje u našim uslovima može obavljati jedno lice, tj. menadžer za upravlajnje projektima. On može biti i ekspert u željenoj oblasti, koordinator aktivnosti, lider kada su u pitanju međuljudski odnosi i podrška, i još mnogo uloga, što podrazumeva mnogo posla za jednu osobu.

Kada su u pitanju *artefakti*, korak u razvoju proizvoda kao forma funkcionalnog proizvoda se može javiti u različitim deltnostima proizvodnje, ne samo u oblasti razvoja softvera, te se može navesti primer izrade nameštaja po meri, koji će biti prikazan u daljem delu rada. Dalje, skladište porizvoda nalazi paralelu u našem okruženju kroz različite nacrte i planove vezane za karakteristike proizvoda, a skladište Spint-ova kroz skup aktulenih zahteva proizvoda na kojima se radi u određenom vremenskom periodu ili fazi.

Načela *vremenska osnove* Scrum metode se u najvećoj meri preklapaju sa praksom upravljanja vremenom i rasporedom u našim uslovima. I ako se ne koristi sam pojma Sprint-a kao određenog vremenskog perioda, za faze različitih procesa i aktivnosti je definisan početak, trajanje, i kraj. Na ovo se nadovezuje planiranje Sprint-a koje se de facto javlja kao neminovna potreba, i nužnost definisanja skupa zahteva koji će se ostvariti u određenom vremenskom periodu. Pored navedenog, različiti tipovi sastanaka, počev od dnevnih, retrospektivnih i revizornih svakako nalaze svoje mesto, shodno potrebama konkretnog preduzeća, uz naglasak da da postoje i varijacije u sadržajima i načinu izvođenja istih. Tako, na primer, dnevni sastanak u nekoj organizaciji može trajati svega petnaest minuta, dok u drugoj može zahtevati sat vremena.

Poseban osvrt u ovom delu je na eksternu stranu koja može doprineti unapređenju preduzetničke organizacije, gde se naglašava *uloga klijenata ili korisnika*. U oblasti razvoja softvera i generalno IT sektora, primarni značaj je u incijalnoj fazi, tokom definisanja zahteva proizvoda, a zatim pri testiranju istog. Dakle, uključiti korisnike je ključno. Kada je u pitanju Scrum metoda, kao što je već napomenuto, *testiranje* se vrši nakon svake iteracije, ili Sprint-a, što omogućava da se proizvod što bolje kreira, shodno zahtevima korisnika.

Ako bismo pažnju obratili na sektore koji nisu vezani za IT industriju, može se napraviti poređenje sa kastomizovanim proizvodima i uslugama. Napomena je da se poređenje sa serijskom i masovnom proizvodnjom eliminiše, jer je u tom slučaju u pitanju jedan proizvod čija se specifikacija kreira na osnovu istraživanja potreba velikog broja korisnika, i testiranje se vrši na uzorku, koji ne bi mogao uvek da bude isti, te testiranje dobija drugačiji značaj. Dakle, i u ovom slučaju, može se koristiti Scrum metoda, gde se u fazi kontrole kvaliteta uključuju korisnici, te se upravo naglašava značaj eksterne kontrole kvaliteta. Kao i u Scrum-u, i ovde je vlasnik proizvoda jedini, i krajnji korisnik, bez obzira koji tip proizvoda je u pitanju.

Prvi primer je *nameštaj po meri.* Počev od samog modela, kupac bira dizajn, boju, vrstu i kvalitet drveta, definiše mere shodno potrebama prostora. Majstor pristupa izradi, deo po deo, gde na kraju određenih etapa izrade i sklapanja kupac vrši proveru kvaliteta iskazanih zahteva, čime je direktno uključen u proizvodnju. Ovaj primer ima najviše sličnosti sa Scrum konceptom, jer je vlasnik, odnosno korisnik, konstantno uključen.

Uzmimo u obzir primer *automobila*. Kada je u pitanju model koji se prilagođava ukusu pojedinačnog kupca, može se napraviti delimični pandam Scrumu. To može biti restauracija starog modela automobila, po željama kupca. Ovde se osvrćemo na inostranog proizvođača. (Slika 21) (Guide, 2015) i (Slika 22) (Anonymous, 2015)

Slika 21 Pre restauracije



Slika 22 Nakon restauracije



U pitanju je Chevrolet Chevelle, model američkog proizvođača General Motors koji se proizvodio od 1964. do 1977. godine. To je jedan od prvih sportskih automobila iz zlatnih šezdesetih godina američke auto-moto industrije. Kao što se može videti, izmenjeni su točkovi, sedišta, motor, različite komponente i sistemi, pri čemu je zasigurno vlasnik imao uvid u svaku fazu, i proveravao je proces restauracije. Dakle, bio je uključen u čitav proces, kao i Product Owner pri korišćenju Scrum metode.

## Primer primene Scrum metode u Start-Up organizaciji

Korišćenje konvencionalnih obrazaca za opisivanje preduzetničke organizacije ne bi bilo pogodno, jer su u pitanju veoma različiti tipovi, te ćemo se poslužiti primerom. Pored toga, rad se bazira na primeni Scrum metode, te je logično da je u pitanju organizacija koja se bavi ***razvojem softverskih rešenja i konsultantskim uslugama u IT domenu.***

Kada je reč o opštim informacijama, pored osnovne delatnosti, organizacija o kojoj je reč je formirana pre nešto manje od godinu dana, kada je na početku zapošljavala svega četiri osobe, pored zakonskih vlasnika kojih je pet, i od kojih su dva aktivno uključena u funkcionisanje. U datom momentu*, svega dve osobe su bile zadužene za core delatnost, dok su se ostali bavili različitim non* - core aspektima formiranja organizacije, počev od biznis plana, finansijske podrške, formiranja kontakata, i opšte, izgradnje fundamentalnih komponenti. Kako se posao razvijao u željenom smeru, otvorila se mogućnost zapošljavanja dodatnog osoblja, i gde se nakon svega šest meseci došlo do statusa u kome firma zapošljava 10 zaposlenih, i struktura ima sledeći izgled: (Tabela 4) (Ondo - SETI, 2015)

Tabela 4 Hijerarhijska struktura

|  |  |  |  |
| --- | --- | --- | --- |
| **Naziv pozicije** | **Hijerarhija** | **Status** | **Broj** |
| Chief Executive Officer | Top Management | Vlasnik – aktivno uključen | 1 |
| Chief Operating Officer | Top Management | Vlasnik – aktivno uključen | 1 |
| Chief Strategy Officer | Top Management | Vlasnik | 1 |
| Chief Financial Officer | Top Management | Vlasnik | 1 |
| Chief Technology Officer | Top Management | Vlasnik | 1 |
| Office manager | Bottom Management | Zaposleni | 1 |
| Senior Software Developer | Bottom Management | Zaposleni | 3 |
| Junior Software Developer | Bottom Management | Zaposleni | 1 |
| Project Manager | Middle Management | Zaposleni | 2 |
| Bussines Developer | Bottom Management | Zaposleni | 1 |

Iz navedenog prikaza hijerarhijske strukture se jasno uočava da srednja linija menadžmenta ne postoji, što direktno implicira na vrstu re*lacija koje postoje između najvišeg i najnižeg nivoa.* Dakle, ne postojiautokratski tip *rukovođenja*, već svako ima veliku individualnu odgovornost za sopstvene učinke. Ne postoji naredbovni stav, već je *komunikacija* fleksibilna i neformalna, gde top menadžment direktno komunicira sa na primer, Junior Software developer-om, u cilju pomoći pri obavljanju zadataka. Titule nisu presudne, te možemo zateći i najviše pozicije u obavljanju određenih zadataka koji više odgovaraju nižim nivoima u hijerarhiji. Svi su istoj kancelariji, jedni drugima pomažu i odmažu u isto vreme! *Problemi* se rešavaju u hodu, i bitno je rešiti problem, na ovaj ili onaj način, jer procedure nisu krucijalne. I ovako tim uzima zasluge, ne individua.

Oblik koji se koristi u našem Start- Up je Façade Scrum, dakle u celom preduzeću, iniciran od starne upravljačke strukture. Scrum-ovanje počinje već pri *prvom susretu* sa Product Owner-om, kada se definiše tip proizvoda, i sve relevantne karakteristike. Nakon toga, održava se *Sprint Planning* koji je baza za dalji rad. Najrelevantija je interakcija između Product Owner-a i Senior Software Developer-a, dok Top Management i mi Project Manageri imamo ulogu pažljivog slušaoca i ponekad komentatora. Uobičajena trjajanje Sprint-a kod nas je dve nedelje, te je neophodna optimalna alokacija svih zahteva, što ponekad izaziva glavobolju programera!

Najveću zabavu predstavlja *sastanak za procene*, gde svi učestvujemo u proceni trajanja aktivnosti definisanih u User Story-jima, što će detaljnije biti objašnjeno u daljem delu rada. Na dnevnom redu je Poker Plannig, tu su Top Management, Softwer Developer-i, i mi Project Manager-i. Koriste se kartice za procenu, sa različitim vrednostim bodova za estimaciju, i ponekad može doći do velikih varijacija, s obzirom da svi imaju mogućnost da se izjasne koliko će trajati neka aktivnost. Ovo može potrajati dugo, ali na kraju se sve završi u najboljem redu, i svaka aktivnost dobije svoje vremensko trajanje.

Kada je u pitanju *izveštavanje*, ono se vrši na dnevnom nivou, kroz Daily meeting, kao i na mesečnom nivou, kroz pregled postignutih rezultata (Sprint Retrospective i Sprint Review). Zanimljivo je i tada, da u sali za sastanke, ne postoji formalni protokol sedenja, obraćanja i sl., jer svi rade za istu ideju, sa istom revnošću i posvećenošću, i svi žele da vide šta su postigli. Naravno, i ovo za nekoga može izgledati kao strašni sud, jer u firmi sa 10 zaposlenih, zaostajanje ili nedovoljan rad se uočava direktno, egzaktno, bez mogućnosti prikrivanja.

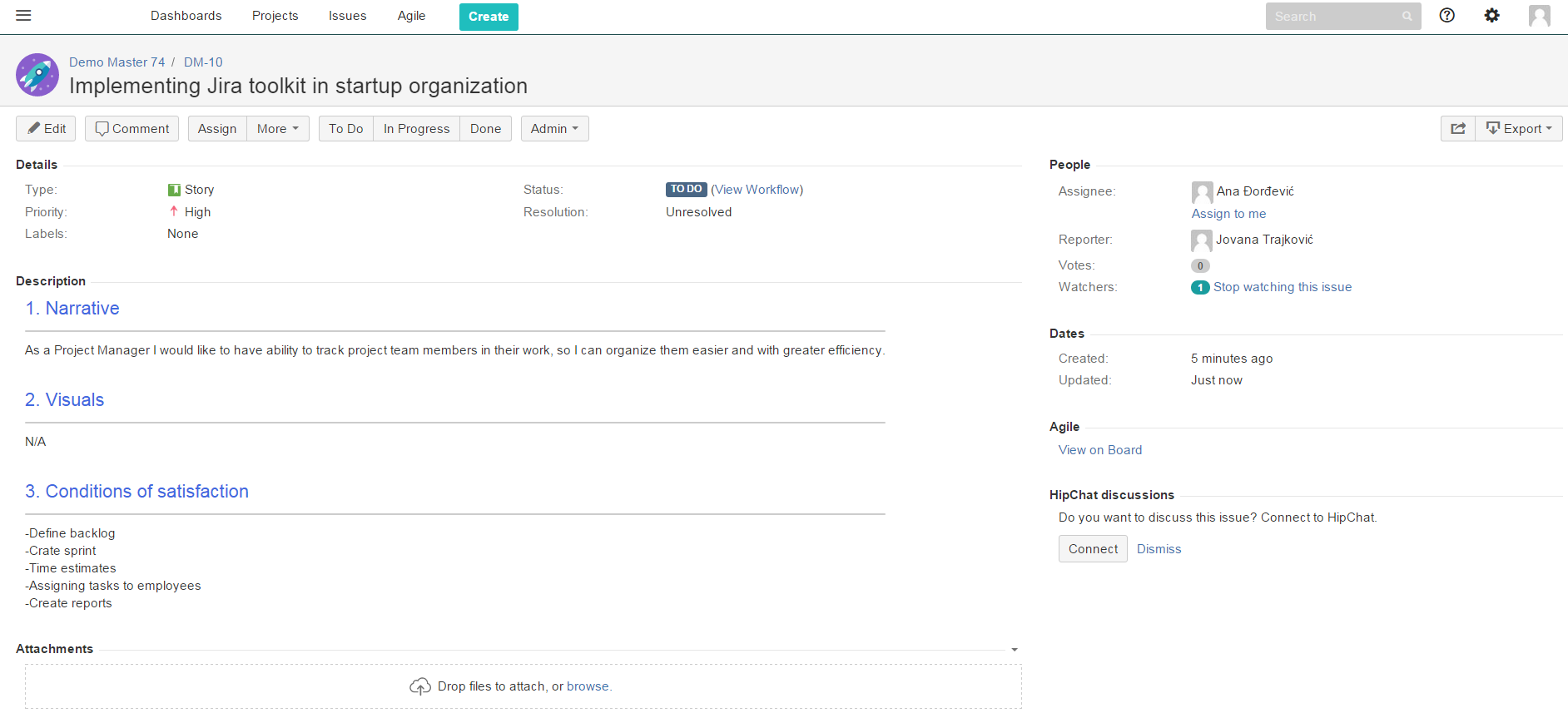
Specifičan segement u Scrum-u je *kontrola*, jer razvoj softverskih rešenja podrazumeva kontrolu od strane naručioca ili korisnika, te su neophodni česti sastanci istih sa developer-ima. Ovde se razmatraju postignuta rešenja, iznose dodatni zahtevi, vrše promene i korekcije. Dakle, opet postoji dirketna komunikacija između izvršnog, core, dela i klijenta, i taj vid kontrole je jednako relevantan kao i interna kontrola.

Na kraju dana, svi, i Senior-i, Developer-i i mi Project manager-i sedimo u dvorištu, u lazy bag foteljama, i pokušavamo samo da zamislimo kako će izgledati sledeći dan, jer u Start-Up to ne zna ni sam Start-Up. Ali preživimo!

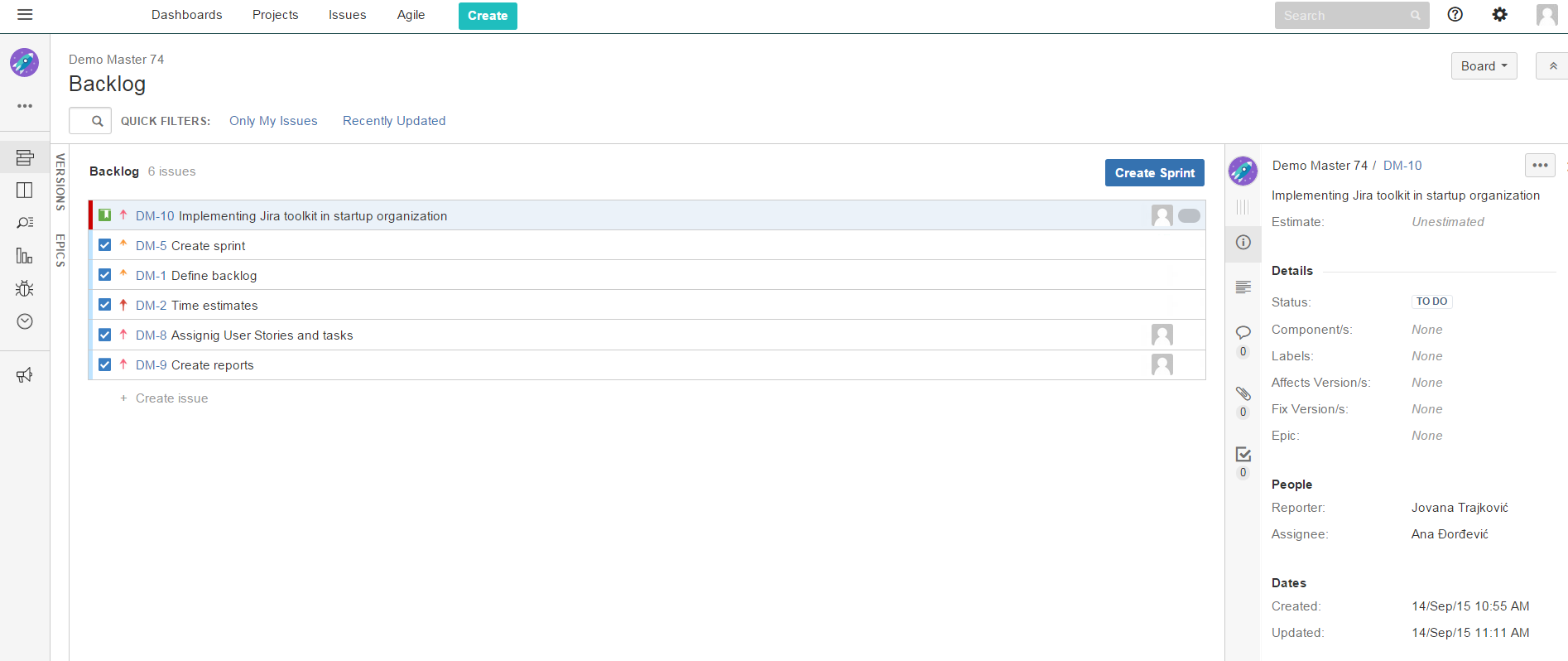
## Primena Scrum metode kroz Jira Agile software

Perspektiva Project Management-a Scrum-a se zasniva koordinaciji i komunikaciji sa odgovornima za realizaciju različitih zadataka, vezanih za određeni proizvod. Ideja je da se praktični aspekti rada prezentuju kroz prikaz upotrebe Jira Agile software, koji predstavlja alat koji koriste Project manager-i kako bi efikasno upravljali vremenom, aktivnostima zaposlenih i ostalim resursima.

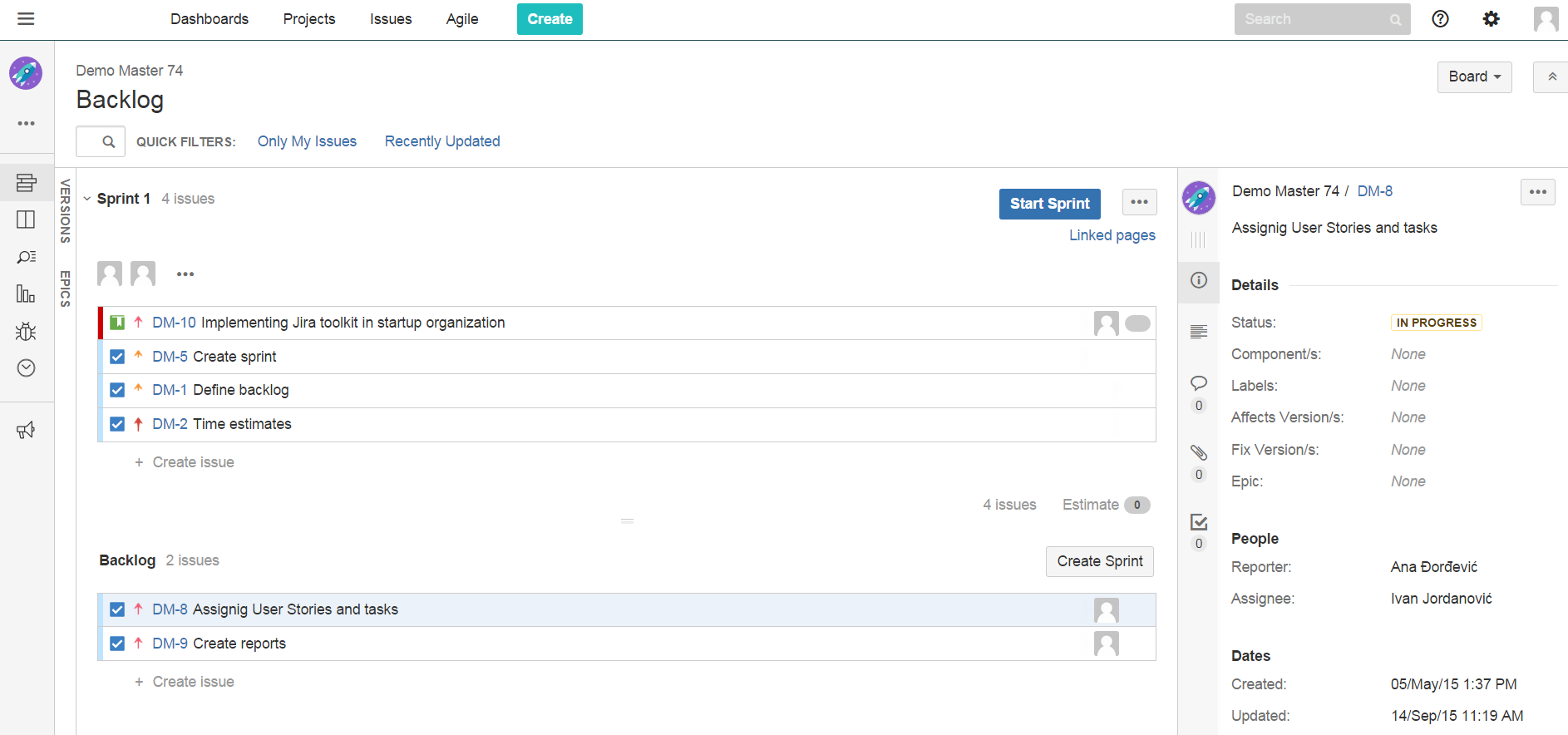
Slika 23 Kreiranje Epic-a



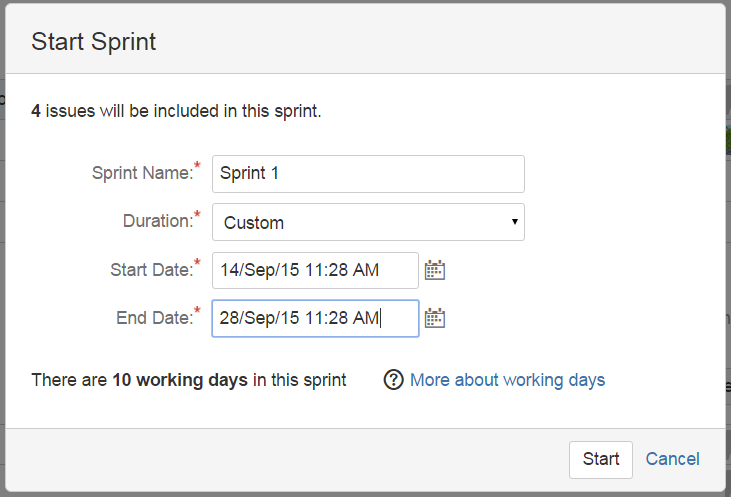
Slika 24 Backlog



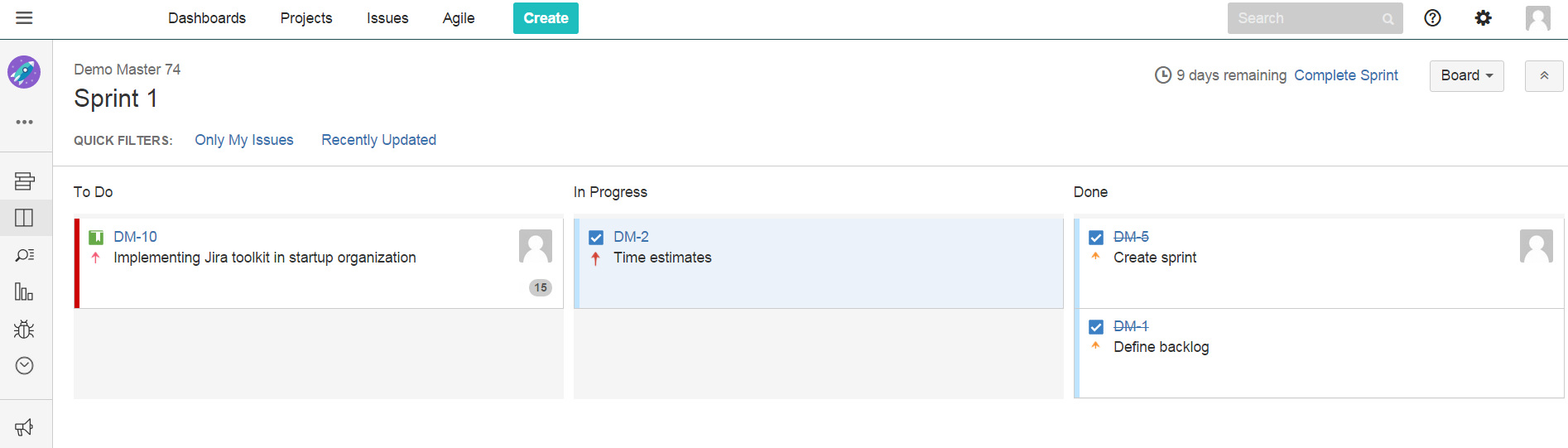
Slika 25 Backlog - Sprint



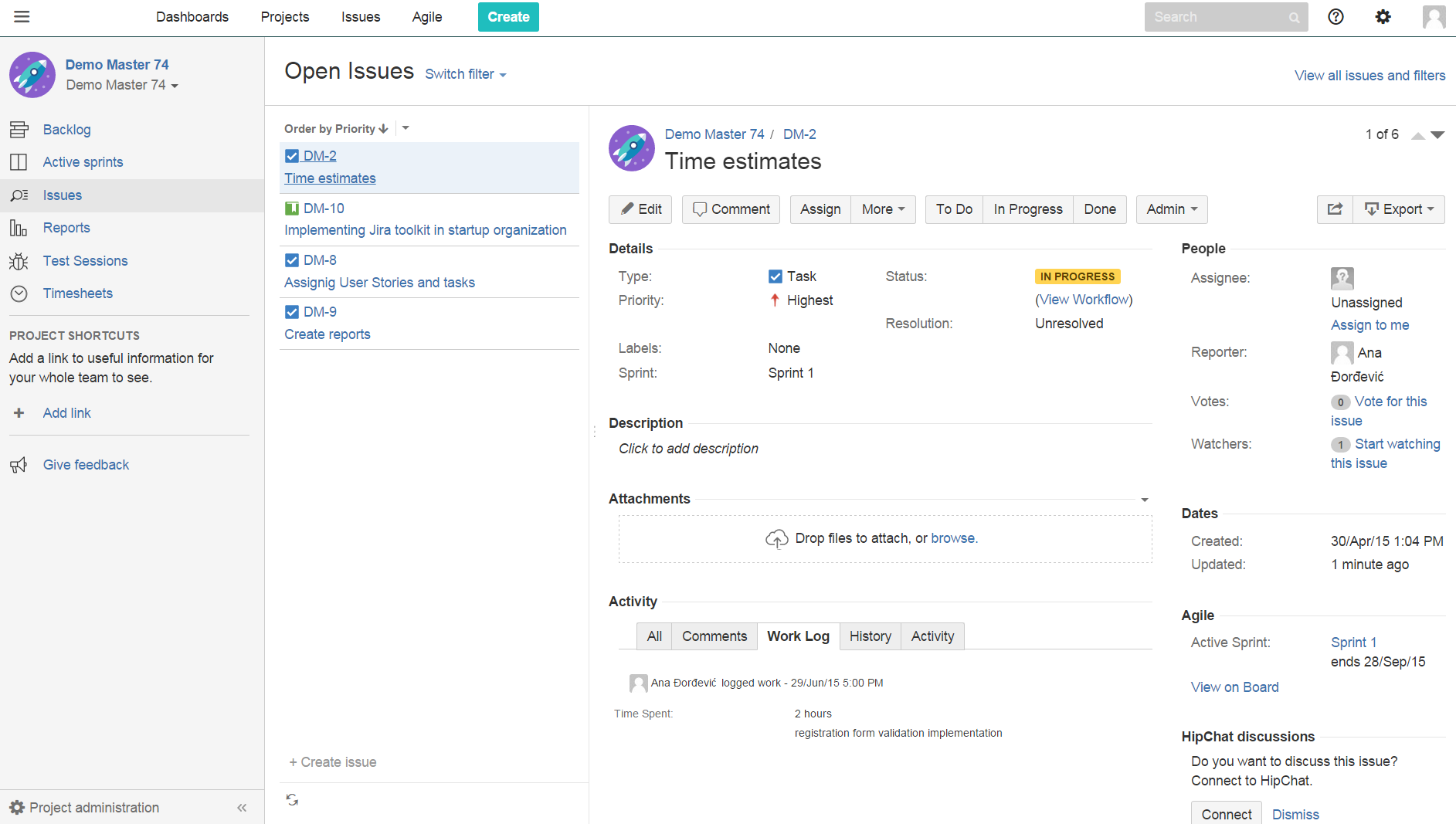
Slika 26 Startovanje Sprint-a



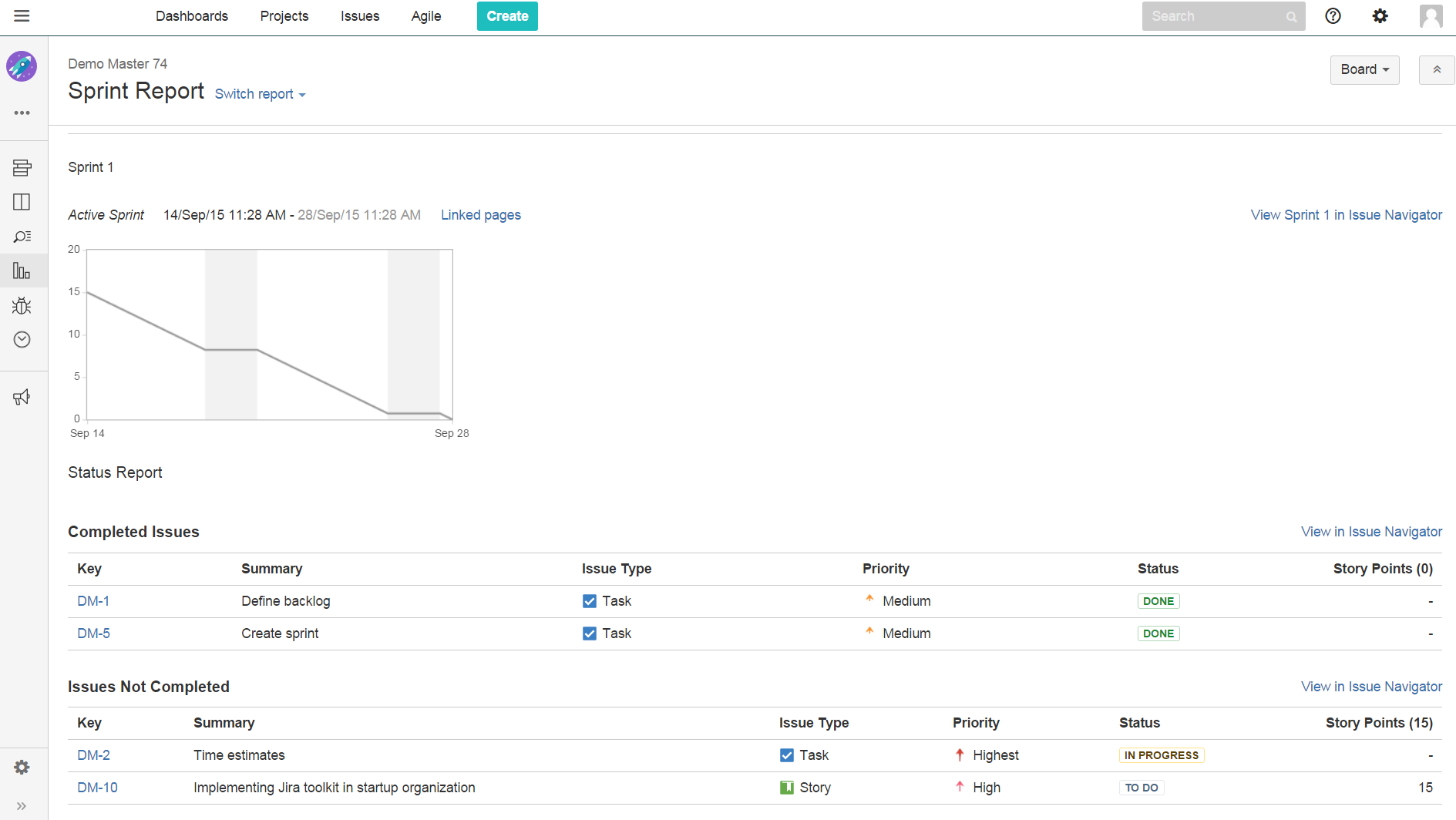
Slika 27 Sprint



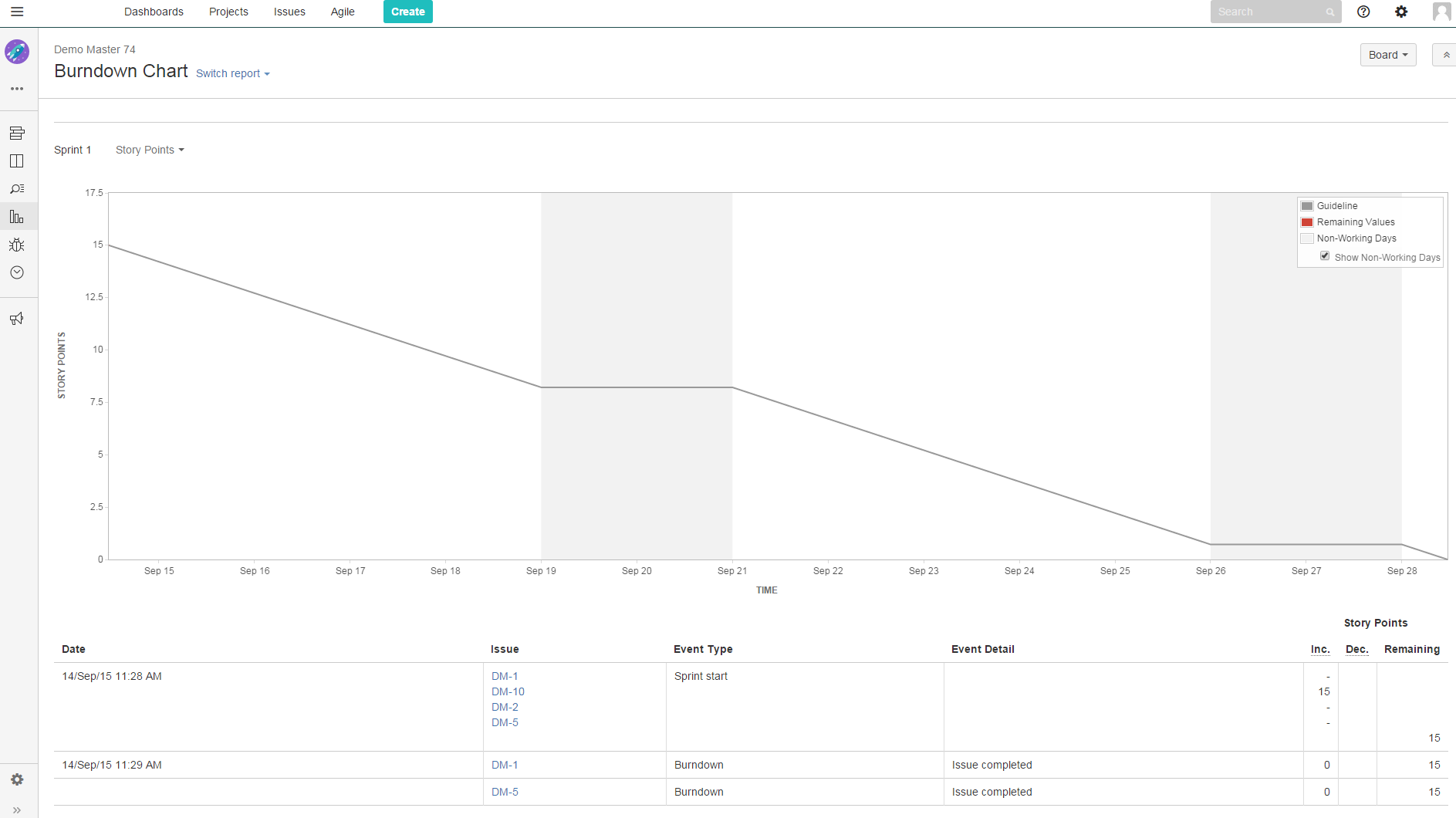
Slika 28 Detaljniji pregled zadatka u Sprint-u



Slika 29 Izveštaj o Sprint-u



Slika 30 Burndown Grafikon (Ondo - SETI, 2015)



# ZAKLJUČAK

Detaljniji pristup analizi Scrum metode pokazao je da je ovo oblast koja teži da ostale tradicionalne metode upravljanja projektima ostavi iza sebe, i koja može na raznim frontovima da doprinese boljem funkcionisanju različitih tipova organizacija. Postoji veliki potencijal za istraživanje i formiranje teorijskog okvira još neistražene praktične primene ove metode u organizacijama koje ne bave isključivo razvojem softverskih rešenja.

Analiza svih elemenata ove metode pokazala je da postoji sadržajniji, i pre svega efikasniji način organizovanja ljudi, procesa i aktivnosti, a da pri tom i korisnici i organizacija budu zadovoljni. Novi način pristupa do sada rigidnim i nefleksibilnim zadacima, menja perspektivu zaposlenih, podiže motivaciju i zainteresovanost za rad. Navedeno doprinosi lakoći obavljanja zadataka, smanjenju tenzije i uspehu u ostvarivanju ciljeva.

Takođe, primena metode uz opisani softverski alat olakšava koordinaciju svih uključenih na projektu, naglašavajući važnost uvida u rad i vremensku osnovu istog. U svakom trenutku postoji precizan prikaz odgovornosti za aktivnosti, kao i krucijalne tačke koje vode do cilja. Lično iskustvo u radu u ovom domenu pokazuje je da efikasna koordinacija ključ za dobru komunikaciju i izveštavanje, te se izbegavaju konflikti i nesuglasice.

Veliki značaj je imalo upoznavanje sa načinom razmišljanja i iskustvom autora korišćene literature, posebno sa načinom pristupa analizi svih oblasti. Kroz jednostavne primere i korišćenje jednostavnog stila pisanja, Scrum je postao oblast koja je lako razumljiv i dostupna svima.

Još jedan zaključak koji donosi upoznavanje sa ovom oblasti je da su organizacije u našoj zemlji u zaostatku, i da se još uvek teži upotrebi tradicinonalnih metoda, tj metoda “na papiru“. Neupućenost i nezainteresovanost stvaraju jaz između naše zemlje i svetskog standarda, te su edukacija i osvešćenost neminovni u ovoj oblasti. Princip „poznato je sigurnije“ treba da doživi preokret, što će dovesti do frekventnije upotrebe same metode. Primena metode u nekoj varijaciji ili pseudoobliku može dovesti to kompletne promene u funkcionisanju same organizacije, tako da čak i ti oblici dobijaju legitimitet na našim prostorima.

Pored neizbežnog razmatranja elemenata same metode i uloga zaposlenih, istakla bih važnost osobe odgovorne za uvođenje metode u organizaciju. Kada odluka o uvođenju potiče iz redova upravljačke strukture, kao što moje lično iskustvo pokazuje, svi su i više nego voljni da prihvate nove obrasce i načine. Prožimanje cele organizacije je ključno, i potrebno je da ista “diše“ Scrum. Razlike i separacije su nepoželjne, i one samo remete proces. Naravno, negodovanja su neizbežna, ali pravi pristup može da prebrodi sve.

I na kraju, poslužiću se starom latinskom poslovicom: *Periculum in mora* – opasnost je u odlaganju! Ne čekajte da Scrum došeta do vas, došetajte vi do njega – ako neće Muhamed bregu, breg će Muhamedu. Svi koji pročitaju ovaj rad bi tebalo barem da se zapitaju da li i koliko dobro rade bez Scrum-a, i da sagledaju šta je to najbolje što mogu da prihvate iz ovog pristupa. Ili to, ili barem jednom da pomenu ono što su pročitali, jer dobar glas daleko stiže!

# LITERATURA

1. Agile Alliance. (2015). *The Agile Manifesto*. Preuzeto sa http://www.agilealliance.org/the-alliance/the-agile-manifesto/
2. Anonymos. (2008). *Agile Methodology*. Preuzeto sa http://agilemethodology.org/
3. Anonymous. (2015). *1968 Chevrolet Chevelle custom coupe*. Preuzeto sa http://www.barrett-jackson.com/Archive/Event/Item/1968-CHEVROLET-CHEVELLE-CUSTOM-COUPE-117470
4. Base36 - Smart Slutions. (2012). *Agile & Waterfall Methodologies – A Side By Side Comparison*. Preuzeto sa http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/
5. Chamorro-Premuzic, T. (2014). *Four unavoidable steps to building an entrepreneurial organization*. Preuzeto sa http://www.fastcompany.com/3035437/hit-the-ground-running/4-unavoidable-steps-to-building-an-entrepreneurial-organization
6. Cross, V. (2015). *Definition of Entrepreneurial Organization*. Preuzeto sa http://smallbusiness.chron.com/definition-entrepreneurial-organization-19019.html
7. DeMarco, T., & Lister, T. (2013). *Peopleware: productive projects and teams.* Melbourne: Addison-Wesley.
8. Fatulescu, C. (2014). *Agile - Scrum Software Development Technology*. Preuzeto sa http://www.pentalog.com/approach/agile-scrum-methodology.htm
9. Gilliland, M. (2013). *Risk and Scrum*. Preuzeto sa https://www.scrumalliance.org/community/articles/2013/september/risk-and-scrum
10. Guide, A. E. (2015). *Chevrolet Malibu*. Preuzeto sa http://auto.howstuffworks.com/chevrolet-malibu5.htm
11. Handbook, T. P. (2013). *Stacia Viscardi.* Birmingham: Packt Publishing Ltd.
12. Hilton, P. (2012). *Scrum software project documentation*. Preuzeto sa http://blog.lunatech.com/2012/08/06/scrum-project-documentation
13. Kniberg , H., & Skarin, M. (2010). *Kanban and Scrum-making the most of both.* Raleigh: Lulu. com.
14. Kotter, J. P. (2012). Leading Change - Why Transformation Efforts Fail. *Harward Bussines Review*.
15. Maximi, D. (2015). *The Scrum Culture: Introducing Agile Methods in Organizations.* New York: Springer.
16. McLaughlin, M. (2015). *Agile Methodologies for Software Development*. Preuzeto sa http://www.versionone.com/agile-101/agile-development-methodologies-scrum-kanban-lean-xp/
17. Neo Globe Consulting, LLC. (2014). *Why Go Agile? Understanding the Benefits of the Agile Method of Development*. Preuzeto sa http://www.neoglobeconsulting.com/blog/2014/3/23/why-go-agile-understanding-the-benefits-of-the-agile-method-of-development
18. Ondo - SETI. (2015). Beograd.
19. Pham, A., & Pham, P.-V. (2011). *Scrum in Action: Agile Software Project Management and Development.* Boston: Cengage Learning.
20. Pries, K. H., & Quigley, J. M. (2010). *Scrum project management.* Boca Raton: CRC Press.
21. Rasmusson, J. (2015). *Agile In a Nutshell*. Preuzeto sa http://www.agilenutshell.com/agile\_vs\_waterfall
22. Schwaber, K. (2007). The Enterprise and Scrum. Microsoft Press. Preuzeto sa http://fabio.bressler.com.br/archives/public/scrum/The%20Enterprise%20and%20Scrum%20-%20Schwaber%20-%20Microsoft%20(2007).pdf
23. Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum .* Upper Saddle River: Prentice Hall.
24. Schwaber, K., & Sutherland, J. (2013). *The Scrum Guide.* Preuzeto sa www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf#zoom¼100
25. Shore, J. (2007). *The art of agile development.* Sebastopol: O'Reilly Media, Inc.
26. Stellman, A., & Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban.* Sebastopol: O'Reilly Media, Inc.
27. Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time.* New York: Crown Business.
28. Teryokhina, M. (2013). *Differences between Testing in Waterfall and Agile*. Preuzeto sa http://www.slideshare.net/ExigenServices/differences-between-testing-in-waterfall-and-agile
29. Viscardi, S. (2013). *The Professional ScrumMaster’s Handbook.* Birmingham: Packt Publishing Ltd.