

Primjena LeSS pristupa za razvoj softvera za obračun vode

Marijana Dragičević

Univerzitet u Istočnom Sarajevu

Filozofski fakultet

Pale, BiH

marijana.dragicevic@ff.ues.rs.ba

Sladjan Babarogic

Univerzitet u Beogradu

Fakultet organizacionih nauka

Beograd, Srbija

sladjan.babarogic@fon.bg.ac.rs

Sažetak— Cilj ovog rada je da se opiše mogućnost primjene Less pristupa na konkretnom domenu, kao i da se prezentuju prednosti i mane samog pristupa. LeSS (Large Scale Scrum) je pristup razvoju softvera koji proširuje principe i prakse Scrum-a na veća i složenija okruženja.

Scrum je metoda razvoja softvera koja se u praksi odlično pokazala u primjeni na manjim projektima ili projektima gdje radi jedan razvojni tim. Međutim, kada više timova radi na jednom projektu, tradicionalni Scrum se može suočiti sa izazovima u koordinaciji, komunikaciji i održavanju jedinstvenog fokusa. Tada se preporučuje primjena LeSS-a, koji podrazumijeva uključivanje više timova koji rade zajedno na zajedničkom projektu.

U ovom radu je opisana mogućnost primjene Less-a na razvoju softvera za obračun vode.

Ključne riječi- Scrum, Large Scale Scrum, Softver za obračun vode

I. UVOD

Prilikom razvoja složenih softverskih proizvoda može se pojaviti niz izazova i problema koji nisu tehničke prirode. U razvoju kompleksnih projekata obično učestvuje više timova, pa s tim u vezi može doći do otežane komunikacije među timovima i teže dostupnosti informacija svima što vodi ka nesporazumima, nekonzistentnosti, neefikasnosti, zastojevima u razvoju, nepravilnom određivanju prioriteta zadataka, dupliranju posla itd. Zatim, složeni projekti često imaju zahtjeve koji se mijenjaju ili dopunjavaju. Stoga je neophodno koristiti određene metodologije razvoja softvera koje će omogućiti pravilno upravljanje projektom, usklađenu komunikaciju među razvojnim timovima, efikasnost i konzistentnost u izvršavanju pojedinačnih zahtjeva.

Metodologije razvoja softvera predstavljaju strukturiran pristup koji definiše način organizacije rada unutar timova, upravljanje zadacima i isporučivanje projekta. Različite metodologije razvoja softvera su namijenjene različitim softverskim proizvodima. Izbor odgovarajuće metodologije je od ključnog značaja.

Klasične rigidne metodologije koje teže perfekciji specifikacije zahtijevaju izuzetno obučene analitičare i projektante, kao i dosta vremena i truda posvećenog samoj metodologiji što često predstavlja barijeru u razvoju softvera uzevši u obzir da se često susrećemo sa ograničavajućim faktorima kao što su kratki rokovi za završetak projekta, ograničen budžet što implicira angažovanje manjeg broja informatičkog kadra.

Klasične metodologije razvoja softvera su vrlo strukturirane i predvidive. Svaka faza razvoja je sekvencijalna i dobro definisana. Ove metodologije se oslanjaju na opsežno planiranje unaprijed. Zatim, radi se često sa ograničenom interakcijom među ulogama, pa pojedinci moraju imati duboko razumijevanje svoje specifične faze.

Za razliku od klasičnih metodologija, agilne metodologije razvoja softvera više zahtijevaju promjenu u načinu razmišljanja, fokusirajući se više na saradnju, prilagodljivost, fleksibilnost, kontinuirano poboljšanje. Agilne metodologije naglašavaju saradnju među timovima ili članovima tima tako što se održavaju dnevni sastanci koji zagovaraju jasnu i otvorenu komunikaciju među članovima tima. Agilni timovi isporučuju male, funkcionalne dijelove softvera u kratkim iterativnim ciklusima (sprintovi), što omogućava konstantno dobijanje povratnih informacija, kao i prilagođavanje promjenama zahtjeva.

Scrum je agilna metodologija razvoja softvera. Koristi se za upravljanje razvojem složenih projekata tako što ih razlaže na manje iteracije koji se nazivaju sprintovi. Kao agilna metodologija, naglašava iterativni razvoj i fleksibilnost, uključujući uloge Product Owner-a, Scrum Master-a i razvojnog tima. Scrum se pokazao kao izuzetno dobar pristup razvoju manjih projekata gdje učestvuje jedan razvojni tim.

LeSS je agilna metodologija razvoja softvera koja skalira Scrum principe za veće, složenije projekte sa više timova koji rade zajedno na jednom projektu. LeSS proširuje Scrum na više timova koji rade na istom projektu. LeSS zadržava principe Scrum-a, ali dodaje pravila za koordinaciju i integraciju među timovima.

Scrum i LeSS se koriste za efikasno upravljanje i isporuku složenih projekata, ali zadovoljavaju različite razmjere i složenost. Da bi se učinio pravilan odabir, potrebno je

razumjeti potrebe projekta, kao i razliku između Scrum-a i LeSS-a.

U poglavlju 2 su opisani Scrum i LeSS, kao i razlike među njima. U poglavlju 3 je izdvojeno nekoliko naučnih radova na sličnu temu i opisano šta je u njima rađeno. U poglavlju 4 je opisana potencijalna primjena LeSS-a na softveru za obračun vode. Poglavlje 5 je zaključak. Poglavlje 6 je spisak korišćene literature.

II. O LESS-U

Tradicionalni razvoj softvera sa timovima sa jednom funkcijom, odloženim ili slabim povratnim informacijama, prediktivnim planiranjem unaprijed i uzastopnim tokom od analize do testiranja nije veoma uspješan u današnjem promjenljivom svijetu. [1]

Scrum i Large Scale Scrum (LeSS) se koriste u agilnom upravljanju projektima za poboljšanje produktivnosti i kvaliteta u razvoju softvera.

Scrum je razvojni okvir empirijske kontrole procesa u kome tim sa više funkcija koji upravlja sam sobom razvija proizvod na inkrementalni način. Svaki sprint sa vremenskim okvirom treba da se može isporučiti kupcu. Product Owner je jedan i odgovoran je za maksimizovanje vrijednosti proizvoda, određivanje prioriteta stavkama u Product Backlog-u, kao i adaptivno odlučivanje o cilju svakog sprinta zasnovano na konstantnim povratnim informacijama i učenju. Tim je odgovoran za postizanje cilja sprinta, a unutar tima ne postoje ograničavajuće specijalizovane uloge. Scrum Master obučava Product Owner-a, tim i organizaciju kako da koriste Scrum na najbolji način. Projekt menadžer ili tim lider ne postoje. [2]

LeSS (Large Scale Scrum) je proširenje Scrum-a za skaliranje agilnog razvoja na više timova koji rade na istom proizvodu. LeSS zadržava osnovne principe Scrum-a, ali dodaje dodatna pravila i smjernice za rješavanje složenosti razvoja velikih projekata sa većim brojem timova. Glavne karakteristike LeSS-a su opisane ispod.

Uloge:

- Product Owner – Postoji jedan Product Owner za cijeli proizvod bez obzira što u razvoju učestvuje više timova.
- Scrum Master – Slično kao u Scrum-u, ali svaki tim ima svog Scrum Master-a, a može postojati i LeSS Scrum Master za koordinaciju.
- Timovi – Postoji više međufunkcionalnih timova koji zajedno rade na istom proizvodu.

Artefakti:

- Product Backlog - Jedinstveni, prioritetni Product Backlog za cijeli proizvod koji dijele svi timovi.
- Sprint Backlog – Svaki tim ima svoj vlastiti Sprint Backlog izveden iz Product Backlog-a.
- Increment – Kombinovani inkrement svih timova koji integriše njihov rad i ispunjava definiciju gotovog.

Događaji:

- Sprint Planning - Sprovodi se u dva dijela – jedan dio za cjelokupno planiranje i drugi dio za planiranje specifično za tim.
- Daily Scrum – Svaki tim sprovodi svoj dnevni skram.
- Sprint Review – Zajednički sastanak sa predstavnicima svih timova radi pregleda kombinovanog inkrementa.
- Sprint Retrospective – Timovi sprovode pojedinačne retrospektive, nakon čega slijedi ukupna retrospektiva među timovima.

Razlike između Scrum-a i LeSS-a:

- Skaliranje – Scrum je dizajniran za pojedinačne timove koji obično imaju 3-9 članova. LeSS je dizajniran za skaliranje Scrum-a na više timova koji rade na istom projektu.
- Koordinacija – LeSS uključuje dodatne mehanizme koordinacije za rješavanje zavisnosti i integracijskih izazova koji se javljaju sa više timova.
- Product Owner – U LeSS-u postoji jedan Product Owner za cijeli projekat bez obzira na broj timova i on osigurava jedinstveno određivanje prioriteta i vizije.
- Događaji – Dok su Scrum događaji specifični za tim, LeSS događaji uključuju i događaje specifične za tim i događaje među timovima kako bi se osiguralo usklađivanje i integracija.
- Smjernice i pravila – LeSS pruža detaljnije smjernice i pravila za upravljanje razvojem velikih razmjera, dok je Scrum više minimalistički i fokusira se na operacije jednog tima.

III. POVEZANI RADOVI

Praksa potvrđuju primjenu LeSS-a u izradi velikih softverskih projekata.

U radovima [3] i [4] je opisan način primjene LeSS-a u razvoju softvera u Nokiji, globalnoj telekomunikacijskoj kompaniji, kao i izazovi sa kojima su se susreli prilikom prelaska na LeSS. U pitanju je projekat gdje radi 20 timova koji su raspoređeni u 4 zemlje. U radu [5] je opisana primjena LeSS-a u projektu za Vladinu organizaciju u Norveškoj.

Iako se Less uveliko koristi u razvoju velikih projekata, postoje i specifični problemi sa kojima se kompanije suočavaju prilikom upotrebe LeSS-a, što je opisano u radu [6] na primjeru upotrebe LeSS-a u dvije velike kompanije.

Scrum-of-Scrum sastanak se spominje u literaturi kao mehanizam za rukovanje međutimskom koordinacijom unutar LeSS-a. U radu [7] je predstavljena višestruka studija slučaja o tome kako su Scrum-of-Scrum sastanci primijenjeni u dva velika, globalno distribuirana Scrum projekta koji obuhvataju

po oko 20 Scrum timova. Zaključak je da Scrum-of-Scrum sastanci loše funkcioniraju kad ima previše učesnika sa nepovezanim interesima, a dobro funkcioniraju među timovima koji imaju zajedničke ciljeve i interese.

U radovima [8], [9], [10], [11] i [12] su opisane osnovne karakteristike LeSS-a, niz izazova, praksi, principa, alata i metrika koji se odnose na LeSS, kao i poređenje LeSS-a sa tradicionalnim i drugim agilnim metodama razvoja softvera.

IV. STUDIJA SLUČAJA

Voda se obračunava svakom mjernom mjestu koje ima vodomjer (ili koje nema vodomjer ali ima fiksni način obračuna vode). Svako mjerno mjesto pripada nekom potrošaču. Mjerno mjesto i potrošač se povezuju ugovorom gdje se definiše način obračuna mjernog mjesta. Sav energetska dio se veže za mjerno mjesto, a finansijski dio za potrošača. Dakle, mjerno mjesto se obračunava, a faktura se ispostavlja potrošaču. Ukoliko potrošač ima više mjernih mjesta, ispostavlja se jedna faktura sa pojedinačno iskazanim vrijednostima obračuna za svako mjerno mjesto.

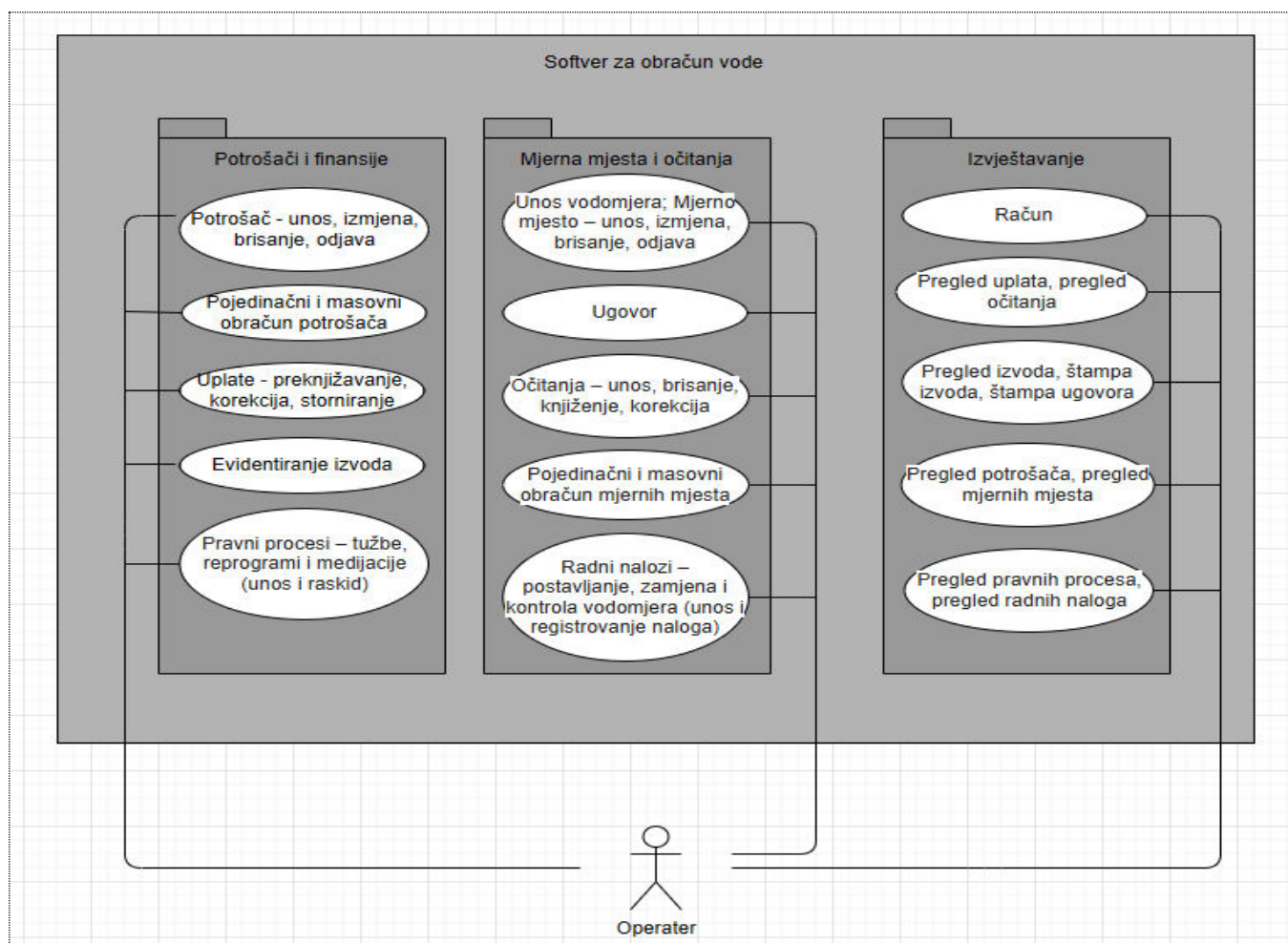
Dakle, početna karika je mjerno mjesto koje je ugovorom vezano za potrošača i koje u ugovoru ima definisan način

obračuna i tarifu odnosno cijenu po kojoj se vrši obračun. Za mjerno mjesto se veže vodomjer (ukoliko se mjeri potrošnja vode, tj. ukoliko nije fiksna način obračuna vode). Sa vodomjera se očitavaju vrijednosti potrošnje mjesečno koje sačinjavaju karticu očitavanja (energetsku karticu) mjernog mjesta, a što nazivamo energetskim dijelom sistema. Na osnovu očitavanja se pravi obračun mjernog mjesta, a na osnovu obračuna mjernog mjesta se pravi obračun potrošača odnosno faktura koja se izdaje potrošaču. Vrijednosti iskazane na računu u novčanoj vrijednosti sačinjavaju finansijsku karticu potrošača. Račun se izdaje potrošaču, a potrošač se duži za iznos računa, što čini dugovnu stranu finansijske kartice potrošača. Sve uplate potrošača čine potražnu stranu finansijske kartice potrošača.

Energetski dio sistema podrazumijeva i funkcionalnosti radnih naloga (postavljanje vodomjera, zamjena i kontrola vodomjera) što utiče na očitavanja mjernog mjesta.

Finansijski dio sistema podrazumijeva i funkcionalnosti pravnih procesa (medijacije, tužbe i reprogrami).

Na Sl. 1. je prikazan dijagram slučajeva korišćenja navedenih funkcionalnosti.



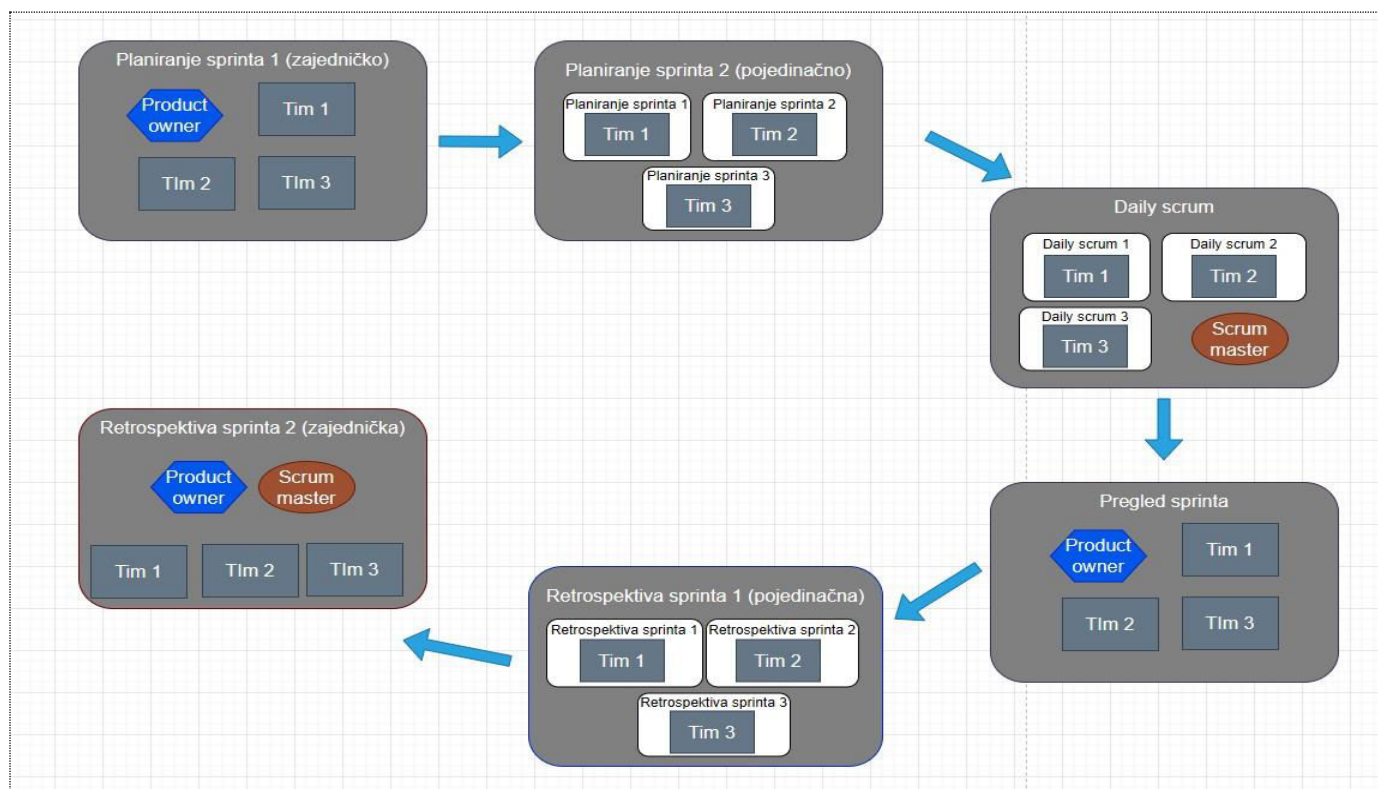
Slika 1 Dijagram slučajeva korišćenja softvera za obračun i naplatu potrošnje vode

Izrada softvera za obračun vode prema LeSS metodi bi mogla biti organizovana na sljedeći način. U izradi bi učestvovala 3 tima: tim koji bi radio na funkcionalnostima finansijskog dijela softvera (tim 1), tim koji bi radio na funkcionalnostima energetskog dijela softvera (tim 2) i tim koji bi radio izvještavanje i preglede svih funkcionalnosti sistema (tim 3).

kako bi razgovarali o problemima i potencijalnim poboljšanjima među timovima.

Scrum master bi bio jedan za sve timove.

Na Sl.2 je prikazan dijagram ovih događaja i uloga. Prikazan je niz koraka koji bi se odnosio na jedan sprint. Za svaki pojedinačni sprint bi se ponavljao ovaj niz koraka.



Slika 2 Prikaz događaja i učesnika sprinta u softveru za obračun i naplatu potrošnje vode

Product Owner bi bio jedan koji bi nadgledao izradu cijelog projekta, davao prioritete Product Backlog-ovima i osiguravao da su timovi usklađeni sa vizijom projekta.

Sprint Planning bi se izvodio u dva dijela. U prvom dijelu bi se sastajali timovi kako bi razumjeli opšti cilj sprinta i koordinisali zavisnosti. Product Owner bi bio tu da odredi prioritete taskova iz Product Backlog-a. U drugom dijelu bi svaki tim imao pojedinačno planiranje, podjelu dodijeljenih taskova i kreiranje Sprint Backlog-a.

Svaki tim pojedinačno bi imao Daily Scrum-ove kako bi razgovarali o progresu, problemima i planiranju radnog dana.

Sprint Review bi bio zajednički sastanak svih timova na kome bi predstavnici svih timova predstavljali šta su uradili. Ovdje bi se razmatrao napredak razvoja cijelog projekta.

Sprint Retrospective bi se izvodio u dva dijela: retrospektiva tima gdje bi se svaki tim osvrtao na svoje procese i identifikovao moguća poboljšanja kao i opšta retrospektiva gdje bi se sastajali predstavnici svakog tima i Product Owner

Spintovi i izrada pojedinih funkcionalnosti Billing sistema za vodovod su predstavljeni u tabeli 1.

Tabela 1 Funkcionalnosti Billing sistema za vodovod

	Tim 1	Tim 2	Tim 3
Sprint 1	Potrošač – unos, izmjena, brisanje, odjava	Unos vodomjera; Mjerno mjesto – unos, izmjena, brisanje, odjava	Pregled potrošača, pregled mjernih mjesta
Sprint 2	Evidentiranje izvoda	Ugovor	Pregled izvoda, štampa izvoda, štampa ugovora
Sprint 3	Uplate – preknjižavanje, korekcija, storniranje	Očitavanja – unos, brisanje, knjiženje, korekcija	Pregled uplata, pregled očitavanja
Sprint 4	Pojedinačni i masovni obračun potrošača	Pojedinačni i masovni obračun mjernih mjesta	Račun
Sprint 5	Pravni procesi – tužbe, reprogrami i medijacije (unos i raskid)	Radni nalozi – postavljanje, zamjena i kontrola vodomjera (unos i registrovanje naloga)	Pregled pravnih procesa, pregled radnih naloga

Potrošač – unos, izmjena, brisanje, odjava – Omogućavanje unosa novih potrošača, kao fizičkih ili pravnih lica koji su potrošači vode i na koje stižu računi. Takođe, omogućavanje izmjene podataka o potrošaču, brisanja samo ako već nije u vezi sa mjernim mjestom i odjave ako je u vezi sa mjernim mjestom.

Unos vodomjera; Mjerno mjesto – unos, izmjena, brisanje, odjava – Omogućavanje unosa novih vodomjera, unosa novih mjernih mjesta, izmjene podataka o mjernim mjestima, brisanja ako nisu u vezi sa partnerima i odjave ako su u vezi sa partnerom.

Pregled potrošača, pregled mjernih mjesta – Pregled postojećih potrošača po različitim parametrima: datumu prijave, datumu odjave, nazivu, šifri, adresi, kategoriji potrošnje. Pregled mjernih mjesta po različitim parametrima: datumu prijave, datumu odjave, nazivu, šifri, pridruženom vodomjeru.

Evidentiranje izvoda – Elektronski i ručni import izvoda banke, kao i unos uplata blagajne koji se knjiženjem izvoda pridružuju partneru i postaju vidljivi na finansijskoj kartici partnera kao potražna strana.

Ugovor – Ugovor je veza između partnera i mjernog mjesta. Jedan partner može imati više mjernih mjesta. Osim veze, ugovorom se definiše i način po kome će partneri biti obračunati (fiksna potrošnja ili potrošnja preko vodomjera).

Pregled izvoda, štampa izvoda, štampa ugovora – Pregled izvoda po različitim parametrima: datumu knjiženja, iznosima uplata, banci, blagajni, operateru. Omogućavanje štampe izvoda za arhivu. Omogućavanje štampe ugovora kao zvaničnog dokumenta.

Uplate – preknjižavanje, korekcija, storniranje – Omogućavanje preknjižavanja uplata sa partnera na partnera, kao i korekciju i storniranje uplata na partneru.

Očitanja – unos, brisanje, knjiženje, korekcija – Dok su finansije vezane isključivo za partnera, očitavanja su vezana isključivo za mjesto. U ovom sprintu bi se trebao omogućiti unos očitavanja sa vodomjera u sistem, knjiženje očitavanja pri čemu se sa čitačkih lista očitavanja spuštaju na energetske kartice mjernog mjesta, brisanje očitavanja koje još nije uvršteno u obračun ili korekciju očitavanja ako je već urađen obračun na osnovu tog očitavanja.

Pregled uplata, pregled očitavanja – Omogućavanje pregleda uplata i očitavanja po različitim parametrima: datumu unosa, iznosu.

Pojedinačni i masovni obračun potrošača – Omogućavanje procesa obračuna potrošača prilikom čega se uzima u obzir energetske obračun mjernog mjesta i na osnovu tarife potrošača i cijene se vrši finansijski obračun potrošača koji se iskazuje na finansijskoj kartici potrošača kao dug.

Pojedinačni i masovni obračun potrošača – Omogućavanje procesa obračuna potrošača prilikom čega se vrši energetske obračun mjernih mjesta na osnovu unesenih i proknjiženih očitavanja na mjernom mjestu.

Račun - Pravljenje računa partnera na kome se iskazuje potrošnja pripadajućih mjernih mjesta i dug (sumaran za sva pripadajuća mjerna mjesta) iskazan u odgovarajućoj valuti.

Pravni procesi – tužbe, reprogrami i medijacije (unos i raskid) – Sva redovna potrošnja partnera se iskazuje na finansijskoj kartici redovne potrošnje partnera. U ovom sprintu bi se omogućilo sprovođenje tužbi, reprograma i medijacije za neplaćeni dug redovne potrošnje. Finansijski to podrazumijeva prebacivanje duga sa kartice redovne potrošnje na karticu utuženja, reprograma ili medijacije, omogućavanje sklapanja rata i odgođenog plaćanja prema datumu valute rata u budućnosti.

Radni nalozi – postavljanje, zamjena i kontrola vodomjera (unos i registrovanje naloga) – Omogućavanje pravljenja radnih naloga za operatere za postavljanje, zamjenu i kontrolu vodomjera. Nakon povratka sa terena, omogućavanje zaključavanje otvorenih radnih naloga i eventualne izmjene na mjernom mjestu (npr. prilikom zamjene vodomjera).

Pregled pravnih procesa, pregled radnih naloga – Omogućavanje pregleda pravnih procesa po različitim parametrima: datumu sprovođenja, broju rata, potrošaču, Omogućavanje pregleda radnih naloga po različitim parametrima: datumu kreiranja, datumu zatvaranja, statusu.

V. ZAKLJUČAK

Scrum i LeSS su agilne metode za razvoj softvera. Scrum je idealan za male projekte sa jednim timom. LeSS se koristi kada su projekti preveliki ili previše složeni za jedan tim. Prednosti scrum-a uključuju fleksibilnost, ranu isporuku dijelova proizvoda i visok kvalitet. LeSS prebacuje ove vrijednosti na veće projekte održavajući efikasnost i koherentnost među timovima.

U ovom radu je opisana moguća primjena LeSS-a na softveru za obračun vode. Softver za obračun vode obuhvata više modula pa je preporučljivo da radi više timova na razvoju. Dva najveća domenska modula su energetske i finansijske dio na kojima bi radila dva odvojena tima. Treći tim bi radio izvještavanje i preglede za oba modula, i energetske i finansijske. U radu je prikazan raspored razvoja funkcionalnosti po timovima.

Prednost ovakvog načina razvoja na navedenom primjeru se ogleda u mogućnosti da se istovremeno radi na funkcionalnostima koje nisu u međusobnoj vezi, a istovremeno je omogućeno da se svaki tim domenski fokusira na samo svoj dio.

LeSS metoda može donijeti i izazove zbog svoje složenosti u implementaciji, jer zahtijeva značajnu reorganizaciju i promjenu kulture u tradicionalnoj organizaciji. Neadekvatno iskustvo Scrum Mastera i Product Ownera mogu dovesti do neuspjeha u koordinaciji i donošenju odluka. Takođe, upravljanje međutimskim zavisnostima i balansiranje tehničkih praksi može biti izazovno.

LITERATURA

- [1] A Lightweight Guide to the Theory and Practice of Scrum, Version 2.0, P. Deemer, G. Benefield, C. Larman, B. Vodde, 2012
- [2] Large-Scale Scrum, More with LeSS, Craig Larman, Bass Vodde, 2016
- [3] Scaling Scrum in a Large Globally Distributed Organization: A Case Study, M. Paasivaara, C. Lassenius, IEEE 2016
- [4] Scaling Scrum in a Large Distributed Project, M. Paasivaara, C. Lassenius, IEEE 2011
- [5] Scaling Across Knowledge Boundaries: A Case Study Of A Large-Scale Agile Software Development Project, K. H. Rolland
- [6] Agile Scrum Scaling Practices for Large Scale Software Development, M. Girma, N. M. Garcia, ICISE 2019
- [7] Inter-team Coordination in Large-Scale Globally Distributed Scrum: Do Scrum-of-Scrum Really Work?, M. Paasivaara, C. Lassenius, V. T. Heikkilä
- [8] Large-Scale Agile Frameworks: A Comparative Review, F. Almeida, E. Espinheira
- [9] Analysing Large-Scale Scrum Practices with Respect to Quality Requirements Challenges, W. Alsaqaf, M. Daneva, R. Wieringa
- [10] Scaling Agile, C. Ebert, M. Paasivaara
- [11] Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review, H. Edison, X. Wang, K. Conboy
- [12] Scaling Agile in Large Organizations: Practices, Challenges and Success Factors, M. Kalenda, P. Hyna, B. Rossi

ABSTRACT

The aim of this paper is to describe the possibility of applying Less approach to a specific domain, as well as to present the advantages and disadvantages of the approach itself. LeSS (Large Scale Scrum) is an approach to software development that extends the principles and practices of Scrum to larger and more complex environments.

Scrum is a software development method that has proven to be excellent in practice when applied to smaller projects or projects where a single development team is working. However, when multiple teams work on a single project, traditional Scrum can face challenges in coordination, communication and maintaining a single focus. In this case, the application of LeSS is recommended, which involves the involvement of multiple teams working together on a common project.

This paper describes the possibility of applying Less to the development of water consumption billing software.

LESS APPROACH IN WATER CONSUMPTION BILLING SOFTWARE

Marijana Dragicevic, Sladjan Babarogic