

## Razvoj softvera primenom agilnih metodologija

ACA D. JOVANOVI , Fakultet za projektne i inovacioni menadžment, Beograd  
 FILIP P. JOVANOVI , Fakultet za projektne i inovacioni menadžment, Beograd  
 LJILJANA Z. MILETI , Fakultet za projektne i inovacioni menadžment, Beograd  
 IVANA M. BERI , Fakultet za projektne i inovacioni menadžment, Beograd

*Originalni nau ni rad*  
 UDC: 005.5:[659.2:004  
 DOI: 10.5937/tehnika1606896J

*U ovom radu su prikazane mogu nosti razvoja softvera primenom agilnih metodologija. Posebno su razmatrane mogu nosti i prednosti primene Scrum metodologije u razvoju softvera i povezanost primene agilnih metodologija sa uspehom projekata razvoja softvera.*

**Key words:** *softver, razvoj, agilne metodologije*

### 1. UVOD

Zna ajne specifi nosti koje odlikuju IT projekte, posebno projekte razvoja softvera, uslovile su potrebu za stvaranjem novih pristupa, procedura ili metodologija za efikasan rad na programiranju i razvoju softvera. Tradicionalni pristupi nisu bili pogodni, pre svega, zbog toga što se projekti razvoja softvera naj eš e zapo inju bez vrstih i nepromenljivih specifikacija i zbog toga što se naj eš e zahtevaju este promene, što tako e zahteva druga iji i fleksibilniji pristup. Kod razvoja softvera veza sa klijentom je veoma vrsta i klijent esto nije siguran šta ta no o ekuje kao rezultat, te kroz zahteve za promenama ide ka svom cilju. Mogu e je da se rade faze ili delovi softverskog rešenja i kao takvi isporu uju klijentu, koji ocenjuje da li razvoj softvera ide u pravcu koji je za njega poželjan. Klijent ponekad zahteva da se proces vrati na prethodnu fazu i da se uvedu odre ene promene koje bi poboljšale softverski proizvod [1, 2, 3]

Da bi se na ovaj na in radilo, potrebno je da tim koji radi na razvoju softvera bude spreman na fleksibilan pristup u radu, na stalne promene i blisku saradnju sa klijentom. U pitanju je specifi an timski rad, neprekidna i brza komunikacija unutar tima i posebno bliska komunikacija sa klijentom kroz este sastanke i brzu i efikasnu razmenu mišljenja i informacija,

kako bi se usaglasile želje i mogu nosti postizanja rezultata. Sve ove specifi nosti dovele su do stvaranja novog pristupa u razvoju softvera, koji se naziva agilni pristup ili agilne metodologije, iji je glavni cilj bio da ubrzaju i poboljšaju proces razvoja softvera i omogu e efikasniji završetak ovog procesa i brže dolaženje do krajnjeg rešenja.

Agilne metodologije su nastale krajem devedesetih godina kada je grupa softverskih inženjera zaklju ila da dotadašnji pristupi i metodologije razvoja softvera nisu pogodne u uslovima turbulentnog okruženja i da nije mogu e vezivati se i posti i vrste rokove isporuke softverskih rešenja, uz zadovoljstvo klijenta. Oni su se sastali i kroz me usobnu razmenu mišljenja došli do osnovnih principa agilnih metodologija, koje su zapisali u tzv. Agilni Manifest. [4] To je po etak stvaranja agilnih metodologija koje bi trebalo da uvaže specifi nosti razvoja softvera i proces razvoja softvera u ine bržim, fleksibilnijim i efikasnijim.

### 2. OSNOVNI PRINCIPI I METODE AGILNIH METODOLOGIJA

Kao što je re eno, osnovni principi agilnih metodologija razvoja softvera su prezentirani u Agilnom manifestu [4] koji je u svojoj prvobitnoj verziji sa državao slede a principe:

- Vrhunski prioritet je zadovoljan klijent, što se postiže blagovremenom i kontinuiranom isporukom softvera.
- Spremnost da se prihvati promena zahteva, ak i u kasnijoj fazi razvoja softvera.

---

Adresa autora: Aca Jovanovi , Fakultet za projektne i inovacioni menadžment, Beograd, Bože Jankovi a 14  
 Rad primljen: 01.11.2016.  
 Rad prihva en: 07.11.2016.

- Spremnost da se redovno isporu uju završeni delovi softvera u kra im vremenskim intervalima.
- Klijenti i developeri svakodnevno razmenjuju mišljenja u toku celokupnog trajanja projekta.
- Na razvoju softvera rade visoko motivisani stručnjaci, uz jaku podršku i pogodne uslove za rad.
- Kontakt licem u lice je najbolji i najefikasniji metod prenosa i razmene informacija.
- Primenljiv softver je osnovno merilo napretka.
- Svi u esnici – klijenti, developeri i pokrovitelji (sponzori) moraju biti spremni da zajedni ki kontinuirano rade uskla enim tempom.
- Agilnost se pospešuje kroz posve enost tehni kom kvalitetu i dobar dizajn.
- Od suštinske važnosti je jednostavnost u radu.
- Samoorganizovani timovi postižu najbolju arhitekturu, zahteve i dizajn.
- Samoorganizovani timovi redovno analiziraju mogu nosti da bolje i efikasnije rade i inoviraju svoje budu e postupke.

Agilni manifest je dopunjen Deklaracijom me u zavisnosti [5] koja promovise slede e principe agilnog razvoja softvera:

- Pove anje ROI kroz kontinuirano kreiranje vrednosti.
- Rezultati se isporu uju kroz stalne iteracije i podelu vrednosti sa korisnicima.
- Treba o ekivati neizvesnost i potrebu za predvi anjem, adaptacijom i upravljanjem.
- Pošto su zaposleni klju ni resurs u stvaranju vrednosti, stvaramo im pogodno okruženje i podsti emo kreativnost i inovacije.
- Performanse se pove avaju kroz timsku odgovornost za rezultate.
- Timska efektivnost se pove ava kroz specifi ne situacione strategije, procese i prakse.

Ovi osnovni principi, manje ili više, opisuju ideje na kojima se zasniva agilni pristup razvoju softvera, usmeren, pre svega, da uvažavaju i specifi nosti IT projekata, doprinese broj i efikasnijoj isporuci proizvoda uz zadovoljstvo klijenata. Nakon više godina primene agilnih metodologija, može se re i da se ovaj pristup bazira na slede im, osnovnim principima:

- Razvoj softvera treba realizovati u kontinuiranoj saradnji sa klijentom i uvažavaju i njegove zahteve.
- Razvoj softvera treba raditi u kratkim delovima, koje treba isporu ivati i razmatrati sa klijentom.
- Prihvatiti sve zahteve za promene od klijenta.
- Komunikacija sa klijentom, sponzorima, stejkholderima i razvojnim timom treba da bude pravovremena i neprekidna.

- Sastanci i razmena mišljenja i povratne informacije su klju ne za dalji efikasan rad na razvoju softvera.
- Tim za razvoj softvera treba da bude samoorganizovan i spreman za fleksibilno delovanje.
- Potreban je jak timski rad i zajedni ka odgovornost za rezultat.
- Tim za razvoj softvera podsticati na inovativnost i obezbediti pogodnu sredinu za rad.

S obzirom na to da se agilne metodologije razvijaju od kraja devedesetih godina, veliki broj istraživa a koji se bave ovom problematikom pokušavao je da pruži adekvatnu definiciju agilnih metodologija ili metoda. Agilne metodologije se definišu kao specifi ni upravlja ki pristupi razvoju softverskih proizvoda, koji uklju uju stalnu i aktivnu ulogu svih koji su uklju eni u ovaj posao (klijent, sponzor, razvojni tim, stejkholderi), prihvatanje zahteva klijenta i uvo enje traženih promena, jak timski rad i neprekidnu komunikaciju i razmenu informacija svih u esnika u razvoju softvera.

Kilibarda i dr. [1] smatraju da se agilne metodologije razlikuju od tradicionalnih zbog toga što zahtevaju razvoj softverskih proizvoda kroz kra e razvojne cikluse, ijim završetkom je mogu e isporu iti jedan deo softverskog proizvoda klijentu i zajedno sa njim uvoditi potrebne izmene, kako bi se brže i efikasnije išlo ka kona nom rezultatu. To potvr uje osnovne principe agilnih metodologija koji ukazuju na stalnu i tesnu saradnju sa klijentom, zatim mogu nost neprekidnog preispitivanja i uvo enja promena i na jak timski rad samoorganizovanih timova koji imaju slobodu u radu i delovanje u povoljnom okruženju koje sponzor treba da omogu i.

Fokusiranje na rad na manjim delovima softverskog proizvoda i njihovu isporuku klijentu i zajedni ko preispitivanje omogu ava da se do e do novih zahteva i promena koje poboljšavaju proizvod i da se fleksibilnim pristupom i inovativnim angažovanjem celog razvojnog tima i drugih u esnika do e do željenog kona nog rezultata – primenljivog softvera kojim je zadovoljan klijent.

Posebna specifi nost agilnih metodologija je u delovanju i vo enju tima koji radi na razvoju softvera. Kao što je ve re eno, to su samoorganizovani, samovo eni timovi koji imaju posebna ovlaš enja u pogledu samostalnosti u radu. IT stručnjaci koji rade na razvoju softvera su jaki individualci koji ne podnose jake i krute principe vo enja tima koji im smanjuju mogu nost kreacije i inovativnosti u nalaženju najboljih rešenja. Oni traže zna ajnu samostalnost i slobodu u radu i specifi ne uslove sredine u kojoj rade. Zato je uloga vo e ovih razvojnih timova izuzetno zna ajna da, zajedno sa sponzorom (menadžment firme),

obezbede najbolje uslove za rad i za komunikacije unutar i izvan tima.

### 3. SCRUM METODA

U praksi primene agilnih metodologija za razvoj softverskih proizvoda danas se predlaže i koristi veći broj metoda. U najpoznatije spadaju Scrum, Extreme Programming (XP), Crystal Clear, DSDM, i dr. Scrum metoda spada u najpopularniju, i u praksi najviše korišćenu, metodu agilnog upravljanja razvojem softvera. Ova metoda se bazira na osnovnim principima koji karakterišu agilni pristup i pogodna je u praksi jer je veoma jednostavna za korišćenje. Ova metoda sugerira da se rad na razvoju softvera odvija u kratkim ciklusima koji se nazivaju sprintovi, zatim da se obavljaju neprekidne konsultacije sa klijentom i da se nakon određenog ciklusa vrši analiza i preispitivanje i eventualno uvedu željene i potrebne izmene. To uključuje obavezne sastanke pre i nakon svakog sprinta, razmatranja da li je sve urađeno prema zahtevima i da li je potrebno uvoditi neke promene. U određenoj situaciji moguće je vratiti se unazad i realizovati određeni sprint u skladu sa novim zahtevima. [1, 2, 6, 7]

Razvojni ciklusi koji se nazivaju sprintovi su vremenski intervali koji mogu da traju mesec dana, najčešće traju dve ili više nedelja. Prednost rada u razvojnim ciklusima je u tome što je moguće da se na kraju sprinta klijentu isporu i deo softverskog proizvoda i da se na osnovu njegovih sugestija deo proizvoda prilagodi novim zahtevima u narednom ciklusu. Na taj način se omogućava da se u toku, a ne na kraju posla, u skladu sa iskustvom i uvode promene i poboljšanja koja daju bolji krajnji rezultat, odnosno dobar softverski proizvod sa kojim je klijent biti zadovoljan.

Tim koji razvija softver korišćenjem Scrum metode ima posebna ovlašćenja u pogledu organizovanja i delovanja, a poseban član ili vođa tima (product owner) ima određena ovlašćenja i odgovornosti u vezi rada razvojnog tima i isporuke željenih rezultata klijentu. Pored članova tima koji rade na razvoju softvera Scrum metodologija predviđa i dve posebne uloge vezane za delovanje tima. To su product owner (vlasnik proizvoda) i scrum master (moderator ili posrednik).

Za product ownera se kaže da je to ličnost koja ima posebnu ulogu u razvoju softvera s obzirom na to da je odgovoran da proizvod bude razvijen i isporučen u skladu sa zahtevima klijenta. Zbog toga vlasnik proizvoda ima posebna ovlašćenja i odgovornost u slučaju primedaba koje daje klijent. Pošto omogućava stalnu vezu i komunikaciju između razvojnog tima i klijenta, vlasnik proizvoda je takođe odgovoran da ova komunikacija teče na obostrano zadovoljavajućem nivou. S

obzirom na ovlašćenja i odgovornosti koje ima, vlasnik proizvoda ima i specifičnu menadžersku poziciju.

Scrum master predstavlja specifičnu poziciju u razvoju softvera koja se odnosi na uspostavljanje posredništva između vlasnika proizvoda i članova razvojnog tima. Kao posrednik ili moderator, on sa jedne strane pomaže razvojnom timu da savlada teškoće i prepreke koje se javljaju u radu, a sa druge strane on daje preporuke i sugestije vlasniku proizvoda kako bi se na najbolji način u razvoju softvera priveo kraju. [6, 7]

Za Scrum metodologiju, kao i za ostale agilne metodologije, od izuzetne važnosti je da se uspostavi sistem komuniciranja putem direktne i povratne veze. Pored elektronskih i pismenih komunikacija posebno se insistira na uspostavljanju sistema direktnih sastanaka svih učesnika u procesu razvoja softvera. Pored svakodnevnih, operativnih sastanaka članova projektnog tima i posrednika, od izuzetne važnosti su sastanci, pre početka i po završetku svakog razvojnog ciklusa, kada klijent treba da, na osnovu isporučenog dela proizvoda, kaže svoje mišljenje, predloge i sugestije. Rezultati ovih sastanaka utiču na planiranje i odvijanje sledećeg sprinta.

Iako se smatra da je Scrum metodologija jednostavna i da se zbog toga dosta primenjuje u praksi, u literaturi se pominje da je moguće koristiti je zajedno sa nekom drugom agilnom metodologijom i da takav pristup može doneti bolje rezultate. U tom smislu, Scrum metoda se pominje kao pogodna za primenu zajedno sa nekom drugom agilnom metodom.

### 4. AGILNE METODOLOGIJE I USPEŠNOST SOFTVERSKIH PROJEKATA

S obzirom na velika očekivanja od uvođenja IT rešenja u poslovanje određene kompanije, nisu retka mišljenja da informacioni sistem koji je uveden ili softver koji je primenjen nisu dali očekivane rezultate. U literaturi se ističe da je procenat neuspešnih projekata razvoja softvera veoma veliki i da često u realizaciji ne donose željene rezultate. S obzirom na to da je proces razvoja softvera izuzetno složen i rizičan nije neobično da postoji veliki broj mogućih razloga za neuspešne softvere.

Spenser [8] i Kilibarda i dr. [1] navode najčešće razloge za neuspeh procesa razvoja softvera:

- Projektni ciljevi nisu potpuno definisani – ako nije precizno definisano šta softver koji se razvija treba da obezbedi, onda su velike šanse da će razvoj softvera doživeti neuspeh.
- Loše planiranje i procene – veoma je važno da se okvirno proceni vreme potrebno za razvoj softvera i potreban budžet. Ako ovi elementi nisu realno procenjeni i ako su nejasni ciljevi koje softver treba da ispuni, šanse za neuspeh su velike.

- Nedostatak valjane metodologije ili neadekvatna metodologija – za uspeh projekta razvoja softvera neophodno je koristiti dobru metodologiju. Veliki broj softvera je bio preskup i neuspešan zbog toga što nije korišćena adekvatna metodologija.
- Nerazumevanje ili neusvajanje zahteva – ovo je jedan od najčešćih uzroka za neuspešan razvoj softvera. Tim koji radi na razvoju softvera nije posvetio dovoljno vremena analizi zahteva klijenta. Zato je došlo do pogrešnog razumevanja zahteva klijenta i do neusvajanja njegovih zahteva. Potrebno je da se ovi zahtevi lagano i precizno analiziraju i da se na osnovu njih preciziraju softverska rešenja.
- Razumevanje jezika – veoma čest problem u razvoju softvera su teškoće u komunikaciji između razvojnog tima i klijenta. Mada IT stručnjaci i klijenti smatraju da govore istim jezikom i da se dobro razumeju, analize govore da postoje razlike u terminima koje koriste IT stručnjaci i poslovnim terminima koje koristi klijent. To dovodi do pogrešnog shvatanja šta treba uraditi i do razvoja pogrešnog proizvoda koji ne odgovara klijentu.
- Upravljanje očekivanjima – veoma često se događa u praksi razvoja softvera da se od razvojnog tima očekuje nemoguće. Kada klijent ne razume dobro proces razvoja softvera, on očekuje da dobije softver koji će da reši sve njegove probleme, te je neophodno da se u komunikaciji sa klijentom objasni šta je realno moguće ostvariti.

Detaljnim razmatranjem i analizom navedenih uzroka za neuspešne procese razvoja softvera, a u vezi sa primenom agilnih metodologija, može se zaključiti sledeće. [1, 8]

Kao jedan od glavnih razloga za neuspeh razvoja softvera navodi se nedostatak dobre metodologije ili primena neadekvatne metodologije. Ukoliko se primeni neka od poznatih agilnih metodologija, npr. Scrum ili XP, ovaj razlog može da se eliminiše u potpunosti, pošto ove metodologije omogućavaju efikasno odvijanje procesa razvoja softvera.

Kao jedan od značajnih razloga za neuspeh projekta razvoja softvera navodi se i nerazumevanje i neusvajanje zahteva korisnika. I ovaj razlog može biti eliminisan primenom agilnih metodologija, pošto one insistiraju na neprekidnoj komunikaciji razvojnog tima i klijenta. Tako se mogu otkloniti nerazumevanja i usvojiti zahtevi korisnika, kako u početku, tako i tokom procesa razvoja softvera.

Neprekidna komunikacija sa klijentom rešava i probleme navedene kroz nerealan očekivanja klijenta, jer je moguće i tokom procesa razvoja softvera objasniti klijentu šta je realno moguće postići i usaglasiti se sa njim u pogledu njegovih očekivanja. Pri tome je veoma

bitno da se ovakva komunikacija i usaglašavanja obave na početku procesa razvoja softvera.

Što se tiče korišćenja jezika i nerazumevanja između IT stručnjaka i klijenta, u pogledu korišćenja terminologije, agilne metodologije pružaju definisanje određenih termina koje bi koristili i IT stručnjaci i klijent. Ipak, u ovom slučaju neophodno je raditi na međusobnom objašnjavanju IT i poslovnih termina.

## 5. ZAKLJUČAK

Značajne specifičnosti IT projekata, posebno projekata razvoja softvera, zahtevaju potpuno drugačiju metodologiju u odnosu na tradicionalne. Kod projekata razvoja softvera nema vrstih i nepromenljivih zahteva, a često klijenti nisu u stanju da unapred daju precizne specifikacije, ni detaljan i promišljen plan realizacije koga se tima za razvoj softvera mora pridržavati.

Kod razvoja softvera veoma je važna uloga klijenta i stalna komunikacija sa klijentom, da bi se ispunili njegovi zahtevi i došlo do željenog krajnjeg rezultata – primenljivog softvera.

Zato je grupa inženjera sa solidnim iskustvom u razvoju softvera postavila principe tzv. agilnih metodologija razvoja softvera, koje uvažavaju najznačajnije specifičnosti projekata razvoja softvera.

Glavni principi agilnih metodologija, definisani u Agilnom manifestu i u Deklaraciji o međuzavisnosti su: iterativni i inkrementalni pristup, fleksibilnost u radu, stalna komunikacija sa klijentom, izrada i isporuka proizvoda u delovima, uvođenje promena, samorganizovan razvojni tim, itd.

Postoji značajan broj agilnih metodologija koje danas nudi teorija i praksa. U najjednostavnije i najpopularnije spada Scrum metodologija. Glavni principi Scrum metodologije su: rad u razvojnim ciklusima – sprintovima, postizanje vrednosti za klijente, fleksibilnost i prilagođavanje, transparentnost u radu i stalno preispitivanje i kontrola.

Posebno je značajan tim koji radi na razvoju softvera i pozicije product ownera (sponzora) i scrum mastera (posrednika ili moderatora). Razvojni tim radi kroz iterativni proces, gradeći deo po deo softvera, uz neprekidnu komunikaciju i transparentnost prema korisniku, koji ocenjuje isporučen delove softvera i zahteva eventualne promene radi postizanja povoljnijeg konanog proizvoda.

U radu su prikazane povoljne mogućnosti primene Scrum metode i ostalih agilnih metodologija, u razvoju različitih softverskih proizvoda. Posebno je analizirano i konstatovano da dosledna primena agilnih metodologija može da značajno smanji broj neuspešnih softverskih projekata koji se u literaturi ocenjuju kao prevelik.

## LITERATURA

- [1] Kilibarda G, Šobaji V, Beri I, Jovanovi P, Upravljanje softverskim projektima, *Tehnika 1/2016*, p. 145-153, 2016.
- [2] Jovanovi P, *Upravljanje projektom*, Visoka škola za projektni menadžment, Beograd, 2015.
- [3] Jovanovi P, Upravljanje projektima u IT okruženju, *XVIII Internacionalni simpozijum iz projektnog menadžmenta*, YUPMA 2014, Beograd.
- [4] *Principi na kojima se zasniva Agilni manifest*, 2001, preuzeto u oktobru 2016. sa <http://agilemanifesto.org/iso/sr/principles.html>
- [5] *Declaration of Interdependence*, 2005, preuzeto u oktobru 2016. sa <http://pmdoi.org>
- [6] Schwaber K, *Agile Project Management with Scrum*, Microsoft press, 2004.
- [7] Pichler R, *Agile Project Management with Scrum*, Addison-Wesley Professional, 2010.
- [8] Spencer M, *Understanding the Software Development Process*, 2010, dostupno na <https://www.projectsmart.co.uk/understanding-the-software-development-process.php>
- [9] Abrahamsson P, Salo O, Ronkainen J, Warsta J, *Agile software development methods: Review and analysis*, Oulu Finland, *VTT Publications*, 2002.
- [10] Mar K. and Schwaber K. Scrum with XP, Informit.com, (dostupno na <http://www.informit.com/articles/article.aspx?p=26057>), 2002.
- [11] Parsons D, Hokyoung R, Ramesh L, *The Impact of methods and techniques on outcomes from Agile software development projects*, Auckland, 2007.
- [12] Ewusi-Mensah, K, *Software Development Failures: Anatomy of Abandoned Projects*. MIT Press, USA, 2003.
- [13] Beri I, Jovanovi A, Popovi M, Radulovi A. Software for Business Operations in SME's, *Tehnika 5/2016*, p.756-760, 2016.
- [14] Jacobson I, Pan-Wei N, McMahon, P. E, Spence I & Lidman S, *The Essence of Software Engineering: Applying the SEMAT Kernel*. Pearson Education, Inc, 2013.

## SUMMARY

## APPLICATION OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT

*The paper presents the potentials for the development of software using agile methodologies. Special consideration is devoted to the potentials and advantages of use of the Scrum methodology in the development of software and the relationship between the implementation of agile methodologies and the software development projects.*

**Key words:** *software, development, agile methodology*