

KRITIČNI FAKTORI USPEHA AGILNIH PROJEKATA CRITICAL SUCCESS FACTORS FOR AGILE PROJECTS

Pavel Kovač, Slobodan Morača, *Fakultet tehničkih nauka, Novi Sad*

Oblast – PROJEKTNI MENADŽMENT

Kratak sadržaj – *Kroz rad su prikazani kritični faktori uspeha na agilnim projektima. Pregledom literature iz oblasti agilnih projekata i kritičnih faktora došlo se do nekolicine faktora koji se povezuju sa uspehom agilnih projekata. Cilj istraživanja je da se identifikuju faktori koji dovode do uspeha kako bi se menadžerima dao set korisnih parametara za merenje, upravljanje i kontrolu softverskih projekata sa preciznošću.*

Ključne reči: *Kritični faktora uspeha, Agilni projekti, strategija isporuke softvera, velocity*

Abstract – *Throughout the paper, critical success factors on agile projects are outlined. A review of the literature on agile projects and critical factors revealed a number of factors related to the success of agile projects. The goal of the research was to identify the factors that lead to success in order to give managers a set of parameters to measure, manage and control software projects with precision.*

Keywords: *Critical Success Factors, Agile projects, Software delivery strategy, velocity.*

1. UVOD

Statistika ukazuje da IT projekti u proseku prekoračuju budžet za 45%, vreme isporuke softvera za 7% i isporučuju 56% manje vrednosti od planirane [1]. Razvojem IT industrije, projekti koji su vezani za IT postaju značajniji. Zbog velikog značaja ovih projekata potrebno je uticati na veliku stopu njihovog neuspeha.

Naučna literatura slaže se da se primenom agilnih metodologija na IT projekte pozitivno utiče na krajnji rezultat i njegov percipirani uspeh. Unutar agilnih metodologija i agilnih projekata potrebno je identifikovati koji faktori se mogu pratiti i meriti kako bi se osigurao željeni rezultat. Uticajem na kritične faktore osigurava se da projekat bude isporučen u planiranom vremenu, troškovima, obimu i da istovremeno zadovoljava poslovne zahteve koji su definisani od strane naručioca projekta. Jer se IT projekat preduzima kako bi se rešio neki poslovni problem a ne isporučio projekat u okviru trostrukog ograničenja.

2. AGILNE METODOLOGIJE

Dobro je poznata činjenica da se veliki procenat projekata unutar IT smatra neuspešnim. Jedan od glavnih razloga za kašnjenje jeste brz razvoj tehnologije koji sa sobom pov-

lači i promenu zahteva klijenta tokom trajanja projekta. Odnosno poslovni zahtevi klijenta na početku nisu isti kao tokom realizacije i na kraju projekta. Kako bi se projekti prilagodili čestim promenama na tržištu i nemogućnosti naručioca projekta da definiše svoje poslovne potrebe nastaju agilne metodologije.

Prednost agilne metodologije jeste što smanjuje inicijalnu potrebu za planiranjem projektnog posla koji će najverovatnije biti zastareo dok ne dođe na izvršavanje. To se ostvaruje iterativnim razvojem odnosno planira se samo onoliko posla koliko je potrebno da se na kraju iteracije korisniku isporuči funkcionalnost koja može da se testira.

Pored toga agilna metodologija se zalaže za samoorganizujuće timove koji su fleksibilni i prilikom svake iteracije rade na realizaciji najvažnije savremene potrebe. Ovakvim pristupom značajno se smanjuje rizik da poslovni zahtevi klijenta jednom kada dođu na red za isporuku budu zastareli. Rezultat jeste projektni tim koji kontinuirano na svakih nekoliko nedelja isporučuje softver korisniku. Međutim nije dovoljno samo implementirati neku od agilnih metodologija već je potrebno i pratiti kritične faktore kako se uspeh projekta ne bi prepustio slučaju.

3. DEFINISANJE KRITIČNIH FAKTORA NA AGILNIM PROJEKTIMA

Kritične faktore uspeha na agilnim projektima možemo posmatrati kao ograničen broj oblasti u kojima kada se ostvaruje zadovoljavajući rezultat dovodi do konkurentne prednosti osobe, departmana ili organizacije. U kontekstu rada, kritični faktori uspeha mogu biti definisani kao faktori koji moraju biti prisutni kako bi agilni projekat bio uspešan.

Chow i Cao su kroz svoje istraživanje ispitivali kritične faktore uspeha agilnih projekata. Od ukupno 36 faktora za koje su pretpostavili da utiču na uspeh projekta, suzili su izbor na 12 faktora. Uspeh projekta su merili u odnosu na kvalitet softvera, obim projekta, planirano vreme i troškove. Zatim su kroz on-line upitnik prikupljali podatke od strana članova *Agile Alliance* koja predstavlja neprofitnu organizaciju koja je posvećena promovisanju agilnih metodologija i principa koji su definisani u agilnom manifestu. Ispitanici su odgovarali prema Likertovoj skali kako posmatraju uspeh na agilnim projektima i koji faktori mogu da doprinosu uspehu. Rezultati su pokazali da se kao najznačajniji kritični faktori uspeha na agilnim projektima izdvajaju [2]:

1. Strategija isporuke softvera
2. Agilne tehnike za razvoj softvera
3. Sposobnost tima

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Slobodan Morača, red. prof.

4. Procesi menadžmenta projekta
5. Timsko okruženje
6. Uključenost klijenta

Kada se povuče paralela između dobijenih rezultata istraživanja i agilnog manifesta svaki od faktora uspeha odgovara nekom od principa definisanog u manifestu.

Strategija isporuke softvera zadovoljava 1. i 3. princip agilnog manifesta: *zadovoljenje kontinuiranom isporukom vrhunskog softvera i redovna isporuka primenljivog softvera u periodu od nekoliko nedelja.*

Agilne tehnike za razvoj softvera se odnose na 9. i 10. princip agilnog manifesta: *stalna posvećenost vrhunskom tehničkom kvalitetu i jednostavan dizajn.*

Sposobnosti tima se odnose na 5. princip agilnog manifesta: *ostvarivanje projekata uz pomoć motivisanih pojedinaca.*

Procesi menadžmenta projekta se odnose na 6. i 8. princip agilnog manifesta: *komunikacija licem u lice i održavanje kontinuiranog tempa isporuke i negovanje kontinuiranog tempa isporuke nezavisno od trajanja projekta.*

Timsko okruženje ima veze sa 11. principom: *najbolje rezultate ostvaruju samoorganizujući timovi.*

Uključenost klijenta odgovara 1. principu agilnog manifesta: *prioritet je zadovoljan klijent.*

Kao najznačajniji kritični faktori uspeha na agilnim projektima se ističu strategija isporuke i agilne tehnike za razvoj softvera iz razloga što utiču na najveći broj postavljenih dimenzija uspeha. Na osnovu istraživanja se može zaključiti da dokle god se odabere visoko kompetentan tim koji poštuje stroge agilne inženjerske prakse i principe i izvršava ih u agilnom maniru sa odgovarajućom strategijom isporuke verovatno je da će projekat biti uspešan.

Postavlja se pitanje kako menadžeri na projektu da prate ove kritične faktore uspeha i osiguraju da agilni tim poštuje strategiju isporuke i agilne tehnike za razvoj softvera. Kupiainen, V. Mäntylä i Itkonen su kroz svoje istraživanje došli do nekolicine faktora koji se prate na agilnim projektima kako bi osigurali uspešnost. U pojedinačne faktore do kojih su došli između ostalog spadaju: velocity, procena vremena, burndown grafik, burnup grafik, vreme trajanja ciklusa, throughput, rad u progresu, dijagram kumulativnog toka, upravljanje ostvarenom vrednošću (EVM), tehnički dug, running tested feature (RFT), broj defekata na projektu, zadovoljstvo korisnika, procena korisničke priče i drugi [2]. Pojedinačni kritični faktori čije vrednosti se mere i prate na projektu se takođe nazivaju i ključni indikatori uspeha (KPI).

4. KRITIČNI FAKTORI USPEHA

Cilj primenjivanja softverskih metrika je da se softverskim menadžerima i profesionalcima da set korisnih i primenljivih podataka za merenje, procenu, upravljanje i kontrolu softverskih projekata sa strogom preciznošću. Bez ključnih indikatora učinka (KPI) teško se može pratiti napredak projekta prema definisanom cilju. Agilni KPI su indikatori koji pomažu da se prate performanse projekta. Prema Kelvinu: *“Ukoliko ne možeš*

da izmeriš, ne možeš da unaprediš“ je izjava koja ima za cilj da motiviše menadžment na projektu da prate ključne indikatore učinka projekta [4]. Cilj je identifikovanje kritičnih faktora koji utiču na strategiju isporuke softvera i agilnih tehnika za razvoj softvera.

Kritični faktori se mogu grupisati prema oblasti:

1. Planiranja iteracije
2. Praćenje napretka iteracije i projekta
3. Kvalitet softvera

4.1. Planiranje iteracije

Ključni indikatori uspeha koji se mere i posmatraju kod planiranja iteracija i projekta se odnose na: obim, određivanje prioriteta i resurse. Faktori planiranja se koriste tokom planiranja iteracije kada tim treba da odredi prioritete prilikom razvoja softvera i uskladili svoj rad i kako bi održali kontinuirani tempo isporuke.

Velocity

KPI velocity se može izračunati na osnovu parametara story points (SP), projektnih zadataka ili potrebnog vremena za realizaciju zadatka izraženog u satima ili danima. Velocity predstavlja ukupan projektni posao koji agilni tim može da završiti tokom trajanja iteracije. Kada se koristi story points kao polazna osnova na kraju svake iteracije agilni tim sabira korisničke priče koje je isporučio i broj bodova koji su sa njima povezani. Sabrani zbir story points-a na kraju iteracije predstavlja velocity tima. Nakon nekoliko iteracija tim može da izračuna vrednost ovog parametra tj koliko story points-a može da isporuči tokom iteracije.

Prema prosečnoj vrednosti dalje mogu da planiraju koliko korisničkih priča mogu da isporuče tokom svake iteracije. Ukoliko su sve korisničke priče unapred definisane i procenjene, pod pretpostavkom da će velocity ostati ujednačen tokom vremena trajanja projekta može se predvideti kada će projekat biti gotov.

Takođe velocity se koristi kako bi ograničio posao u okviru iteracije. Na ovakav način se osigurava da programeri odaberu samo onoliko posla koliko mogu da završe tokom iteracije.

4.2. Praćenje napretka iteracije i projekta

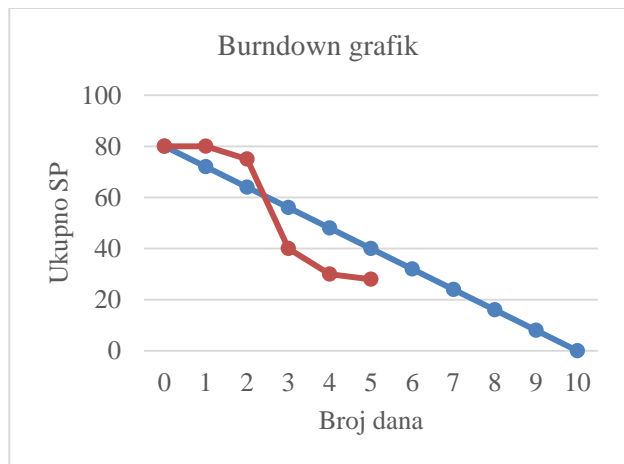
Pomoću korišćenja ključnih indikatora uspeha povećava se transparentnost projekta, omogućava se lakše ostvarivanje ciljeva i balansiranje rada. Burndown i burnup grafici pomažu u prikazivanju trendova projekta kao i mogućnost predviđanja rokova za isporuku softvera.

Burndown grafik

Popularan način za praćenje napretka iteracije i parametra velocity je burndown grafik. Ovaj grafik pomaže menadžeru i projektnom timu da prikaže koliko posla je završeno do momenta posmatranja i koliko posla je ostalo da se završi do kraja iteracije. X-osa na grafiku predstavlja vremensku liniju iteracije, Y-osa predstavlja posao koji treba da bude završen tokom trajanja iteracije, ciljani burndown predstavlja sumu ukupnog posla koji treba da bude završen i ostvareni burndown predstavlja sumu ukupnog posla stvarno završenog.

Kroz grafik 1. se može videti da iteracija tima traje 10 dana i da im je planiran velocity od 80 SP. Ukoliko tim želi da bude u skladu sa strategijom isporuke softvera on nekoliko nedelja, dnevno mora da završe 8 SP što predstavlja ciljani burndown označen plavom linijom na grafiku.

Ostvareni burndown se predstavlja crvenom linijom. Sa grafika se može videti da agilni tim prva dva dana iteracije ne isporučuje skoro nikakvu vrednost, dok između drugog i trećeg dana isporučuju 20 SP.



Grafik 1. Burndown grafik iteracije prema Story Points

Ukoliko tim nastavi ovakvim tempom isporuke završiće projektni posao pre kraja iteracije. Međutim poželjno je da ostvareni burndown uvek bude približan planiranom, da nema velikih odstupanja.

Dok se kroz grafik manifestuje situacija kada tim odjednom isporučuje 20 SP. Menadžer treba da teži i osigurava kontinuirani tempo isporuke odnosno da ciljani i ostvareni burndown budu približne vrednosti.

Dijagram kumulativnog toka CFD

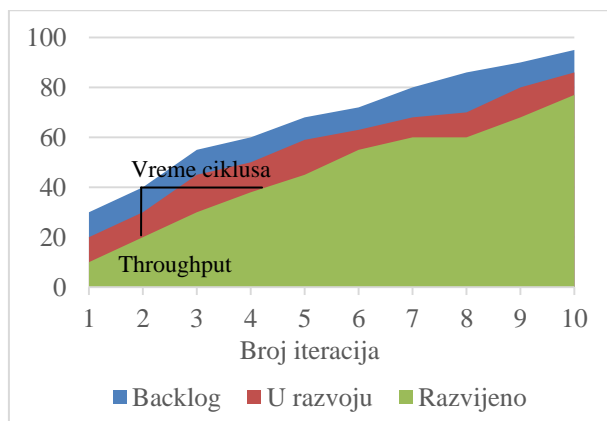
Na agilnim projektima poželjno je pratiti koliko jedna funkcionalnost ili korisnička priča provede u određenoj fazi. Takvo stanje se prati kada se korisnička priča iz statusa uraditi (*To Do*) pređe u status rad u progresu (*WIP*) i završava se kada pređe u status gotovo (*Done*). Kako bi se predstavio status svih korisničkih priča koje se nalaze u backlog-u i koliko vremena provedu u svakoj od faza koristi se dijagram kumulativnog toka (CFD).

Kroz grafik 2. je prikazan proces agilnog projekta u periodu od 10 nedelja. Backlog odgovara statusu uradi, u razvoju odgovara status rad u progresu i razvijeno odgovara status gotovo.

Ukoliko menadžer želi da pogleda koje je vreme ciklusa za 20 korisničkih priča treba da uporedi oblasti prikazane na grafiku. Vrednosti ukazuju da je agilni tim 20 korisničkih priča (throughput) isporučio za nešto više od 2 iteracije (vreme ciklusa).

Throughput predstavlja ukupan broj funkcionalnosti, korisničkih priča, zadataka ili bagova završenih u posmatranom vremenskom periodu koji su spremni da budu isporučeni klijentu.

Vreme ciklusa ukazuje koliko je vremenski potrebni da se završi određeni zadatak.



Grafik 2. Dijagram kumulativnog toka (CFD)

Na osnovu pokazatelja vremena ciklusa i throughputa, menadžer može da planira ostatak projekta. Ukoliko želi isporučiti više od 20 korisničkih priča u periodu od 2 nedelje, mora da utiče da smanji vreme ciklusa. Povećanje se može postići povećanjem ljudi ili povećanjem produktivnosti.

4.3. Kvalitet softvera

Kvalitet softvera opisuje parametre koji pomažu razumevanju kvaliteta softvera pre i nakon isporuke korisniku. Kod ključnih indikatora uspeha koji se odnose na kvalitet potrebno je koristiti parametre koji nagoveštavaju kako napreduje rad agilnog tima i kako napreduje projekat.

Tehnički dug

Tehnički dug predstavlja parametar koja opisuje rezultat koji nastaje kad programeri koji rade na softveru u želji da isporuče softver na vreme, zanemare inženjerske prakse i pravila kako bi ispoštovali terminski plan. Tim hvata prečice prilikom programiranja softvera. Odnosno zanemaruje kritičan faktor uspeha agilne tehnike za razvoj softvera. Ovaj parametar ne mora samo da se odnosi na programiranje već može da se odnosi na celokupan životni ciklus razvoja softvera. Tehnički dug može da bude nameran i nenameran. Nameran dug može da bude rezultat taktičkih ili strateških odluka, dok nenameran nastaje kao rezultat niskog kvaliteta koda.

Upravljanje tehničkim dugom je moguće i pomoću jednostavne analize gde se analiziraju koje su prednosti a koji nedostaci plaćanja ovog duga. Procena može da bude gruba ili se može preciznije kvantifikovati kroz potrebne sate ili dane za ispravljanje funkcionalnost [5].



Slika 1. Prilagođen prikaz kvadrata tehničkog duga [4].

Boje na kvadratu tehničkog duga ukazuju na upozorenje (crvena i narandžasta) i željeni rezultat (zeleno i plavo). Nepromišljeno i slučajno ponašanje je najmanje poželjan rezultat tehničkog duga, jer ukazuje da tim uopšte ne prepoznaje tehnički dug i jednom kada nastane ne mogu da ga ispravu. Zatim sledi nepromišljena i namerna odluka nastaje kada tim poseduje kapacitete da donese ispravnu odluku i ispoštuje inženjerske principe razvoja softvera ali se odlučuju za bržu i često jednostavniju varijantu. Rezultat ovakvog ponašanja je softver koji ima puno grešaka i koji treba kasnije dorađivati ili softver koji možda nema grešaka ali nije u skladu sa željama klijenta. Zeleni kvadrat obuhvata obazrivu i promišljenu odluku i najpoželjniju kod tehničkog duga.

Tim je analizirao i svesno doneo odluku kako bi ispoštovao rok na projektu. Plavi kvadrat predstavlja obazrivu i slučajnu odluku. Tehnički dug koji predstavlja ovaj kvadrat se odnosi na proces učenja iz grešaka koje su nastale u tom procesu.

5. ANALIZA KRITIČNIH FAKTORA USPEHA NA AGILNIM PROJEKTIMA

Kada neki od parametara koji se povezuje sa uspehom na projektu odstupa od planirane vrednosti, menadžer na projektu treba da preduzme odgovarajuće mere kako bi se projekat vratio na put uspeha.

Ukoliko parametar velocity pokazuje ispod planirane vrednosti ili se putem dijagrama kumulativnog toka zapazi da je vreme ciklusa veće od planiranog kao jedan od glavnih uzroka za kašnjenje je loše planiranje. Programeri na projektu nisu isplanirali funkcionalnost do dovoljnog nivoa detalja. Kako bi se rešio taj problem pored sastanka planiranja iteracije potrebno je uvesti sastanak preciziranja korisničkih priča. Na taj način programeri mogu detaljnije da razlože kompleksne zadatke na jednostavnije celine kako bi lakše isplanirali projektni posao. Korisničke priče koje su obrađene na tom sastanku ne moraju nužno da uđu u narednu iteraciju. Time se smanjuje rizik lošeg planiranja.

Još jedan od uzroka lošeg planiranja može biti nedostupnost vlasnika proizvoda (*Product Owner*). Ukoliko vlasnik proizvoda nije dostupan kako bi u detalje objasnio kako neka korisnička priča treba da se programira, moguće je da agilni tim potceni i prebaci u narednu iteraciju. Usled nedostupnosti adekvatnih detalja tim se obavezao na funkcionalnost koju neće moći da isporuči tokom iteracije. Pošto agilne metodologije ne priznaju parcijalno završen posao, to će se odraziti na planiran velocity i throughput. Odnosno terminski plan projekta će biti ugrožen. Uloga scrum gospodara (*Scrum Mastera*) jeste da osigura da vlasnik proizvoda uvek bude dostupan agilnom timu. Na taj način će tim moći redovno da isporučuje funkcionalnosti i osigura da imaju adekvatnu strategiju isporuke softvera koja dovodi do uspeha agilnih projekata.

Ukoliko se agilni tim fokusira na odgovarajuću strategiju isporuke softvera od nekoliko nedelja umesto da bude usmeren na terminski plan, može da se izbegne usko grlo u životnom ciklusu razvoja softvera kada velika količina koda odjednom dospe na testiranje. Vreme ciklusa

proizvodnje softvera se smanjuje i samim tim dolazi do češće isporuke softvera klijentu.

6. ZAKLJUČAK

Na osnovu prethodno iznesenih informacija može se zaključiti da su najznačajniji kritični faktori uspeha strategija isporuke softvera i agilne tehnike za razvoj softvera. Kako bi se osigurala primena ovih faktora na projektu, menadžeri treba da prate ključne indikatore uspeha poput velocity, throughput, burndown grafik, dijagram kumulativnog toka, tehnički dug i drugi. Cilj je kombinovati više različitih faktora uspeha kako bi menadžer napravio sistem upozorenja ukoliko se projekat ne kreće u željenom pravcu.

Jednostavnim pogledom na grafičke vrednosti menadžer može da zaključiti u kojoj oblasti se parametri ne kreću u skladu sa planiranom vrednošću i može da sprovodi korektivne mere. Nedostatak kritičnih faktora uspeha agilnih projekata je što ne uzimaju u obzir troškove i budžet projekta. Iz tog razloga poželjno je navedene faktore kombinovati sa ključnim indikatorom uspeha upravljanje ostvarenom vrednošću (EVM).

7. LITERATURA

- [1] Collyer, S., Warren, C., Hemsley, B. and Stevens, C., 2010. Aim, fire, aim—Project planning styles in dynamic environments. *Project Management Journal*, 41(4), pp.108-121.
- [2] Chow, T. and Cao, D.B., 2008. A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6), pp.961-971.
- [3] Kupiainen, E., Mäntylä, M.V. and Ikonen, J., 2015. Using metrics in Agile and Lean Software Development—A systematic literature review of industrial studies. *Information and Software Technology*, 62, pp.143-163.
- [4] Greening, D.R., 2010, January. Enterprise scrum: Scaling scrum to the executive level. In 2010 43rd Hawaii International Conference on System Sciences (pp. 1-10). IEEE.
- [5] Martin Fowler 2019, TechnicalDebt, MartinFowler, <https://martinfowler.com/bliki/TechnicalDebt.html>

Kratka biografija



Pavel Kovač rođen je u Novom sadu 1993. Osnove akademske studije je završio na Fakultetu Tehničkih Nauka. Tokom studija se opredelio za usmerenje projektni menadžment. Studije je završio 2016. Trenutno radi kao menadžer na IT projektu unutar telekomunikacione industrije.