Upravljanje softverskim projektima

GORAN D. KILIBARDA, Fakultet za projektni i
inovacioni menadžment, Beograd
VESNA M. ŠOBAJI, Fakultet za projektni i
inovacioni menadžment, Beograd
IVANA M. BERI, Fakultet za projektni i
inovacioni menadžment, Beograd
PETAR M. JOVANOVI, Fakultet za projektni i
inovacioni menadžment, Beograd

Pregledni rad
UDC: 005.8:004.4
DOI: 10.5937/tehnika1601145K

U današnje vreme, nemoguće je zamisliti uspešno poslovanje bez korišćenja informaciono - komunikacionih tehnologija. Primena projektnog menadžmenta u upravljanju ICT projektima poštuje iste osnovne principe i koncepte projektnog menadžmenta, kao i kod drugih vrsta projekata, međutim, potrebno je da se analiziraju pojedine vrste IT projekata i njihove specifične karakteristike i da se izabere i prilagodi odgovarajuća metodologija za upravljanje pojedinim vrstama IT projekata. U ovom radu posebna pažnja je posvećena projektima razvoja softvera, njihovim specifičnostima, izazovima koji se nalaze pred učesnicima u tom procesu, kao i metodologijama za upravljanje projektima, koje se mogu koristiti za prevazilaženje poteškoća i postizanje boljih rezultata.

Ključne reči: upravljanje projektima, razvoj softvera, izazovi, metodologije

1. UVOD

Primena informacionih tehnologija je, u savremenom svetu, nezaobilazni faktor efikasnog poslovanja, rešavanja složenih problema i realizacije složenih poduhvata. Sve savremene naučne discipline koriste informacione tehnologije u rešavanju raznovrsnih problema kojima se bave. Menadžment i specijalizovane discipline menadžmenta ne mogu se danas zamisliti bez korišćenja informacionih tehnologija. Kada govorimo o projektnom menadžmentu i korišćenju informacionih tehnologija u ovoj specijalizovanoj disciplini menadžmenta, potrebno je da se posebna pažnja posveti realizaciji IT projekata i primeni metodologije projektnog menadžmenta u upravljanju realizacijom IT projekata.

Živimo u globalnom informacionom društvu i činjenica je da su ranari znatno promenili život ljudima. Uprkos uspešnoj primeni softvera u skoro svim mogućim oblastima, projekti razvoja softvera imaju reputaciju da su neuspešni - na primer, često kašnje, premašuju budžet, ili ne ispunjavaju potrebe kupa-

ca [1], [2]. Broj javno dostupnih studija neuspešnih projekata razvoja softvera je mali, jer većina neuspešnih softvera, posebno onih u organizacijama u privatnom sektoru, ostaje neprijavljen [3]. Ova situacija je razumljiva, jer ni organizacije, ni pojedinci koji su uključeni, ne žele da se detalji tih projekata pojave u bilo kojim medijima, što čini njihovo istraživanje teškim, tj. nemoguće im [4]. Većina projekata o kojima je moguće pronaći podatke su projekti javnog sektora, ali ne zato što oni greše već ili više od privatnog sektora, već zato što im pravila i propisi otežavaju da prikriju svoje greške. Privatna preduzeća su ipak uspešnija u tome da što manje ljudi sazna za njihov neuspeh.

Od danas pa do 2025. godine, sposobnost kompanija i njihovih proizvoda, sistema i usluga da budu konkurentni, da se prilagode i opstanu zavisiće od velikog i kontinuiranog rasta obima softvera koji će koristiti [5]. Neophodnost primene softvera je opšteprihvaćena činjenica, ali ono što donosi probleme jeste sam razvoj softvera i upravljanje projektima razvoja softvera.

U ovom radu će biti prikazane specifičnosti softverskih projekata, kratak prikaz metodologija za upravljanje softverskim projektima, faze u upravljanju softverskim projektima i neke od najčešćih grešaka prilikom upravljanja ovom vrstom projekata.

Adresa autora: Goran Kilibarda, Fakultet za projektni menadžment, Beograd, Bože Janković 14

Rad primljen: 30.11.2015.

Rad prihvaćen: 15.12.2015.

2. IT PROJEKTI I SPECIFIKOSTI SOFTVERSKIH PROJEKATA

Upravljanje IT projektima korišćenjem projektnog menadžmenta je dosta složena materija koja nije dovoljno obrađena u literaturi. Od IT projekata se najčešće dosta očekuje. Izvršni menadžeri u kompanijama smatraju da će uvođenje IT-a doprineti poboljšanju poslovnih procesa i efikasnijem radu celokupne organizacije [6].

IT projekti su takvi projekti koji koriste određeni hardver, softver i mrežu da bi se stvorio informatički proizvod, usluga ili rezultat [7]. Ovo je jedna od mogućih i uopštenih definicija IT projekata.

Kada govorimo o IT projektima i o upravljanju IT projektima treba imati u vidu veoma široku lepezu različitih IT projekata. Međutim, da bi govorili o procesu upravljanja IT projektima i korišćenju odgovarajuće metodologije projektnog menadžmenta, potrebno je izvršiti određenu klasifikaciju i kategorizaciju IT projekata u određene grupe, koje imaju iste ili slične karakteristike.

U praksi se najčešće govori o projektima razvoja i uvođenja različitih softverskih rešenja i o složenijim i krupnijim projektima uvođenja informacionih sistema. To bi mogla da bude neka osnovna klasifikacija, u smislu standardizacije procesa upravljanja IT projektima, koja pruža osnovu za dalje analize i razrade, a tako će i za definisanje procesa upravljanja IT projektima.

Prema Snedakeru [8], postoje sledeće vrste IT projekata:

- Strategijski ili operativni
- Dugoročni ili kratkoročni
- Hardver ili softver
- Razvojni ili implementacioni

Prema Rakoviću [9], ICT projekti se dele na:

- Projekte uvođenja informacionih sistema
- Softverske projekte
- Telekomunikacione projekte

Uzimajući u obzir procese projektovanja i implementacije IT projekata, može se reći da možemo razlikovati projekte koji se odnose na:

- Dizajn hardvera
- Dizajn i implementacija softvera
- Projektovanje i implementacija informacionih sistema
- Obuku korisnika

Ova oblast takođe nije ozbiljnije obrađena u literaturi koja najčešće daje sasvim uopštene podele. Međutim, za definisanje i korišćenje metodologije za efikasno upravljanje IT projektima neophodne su preciznije podele koje bi izrazile specifične karakteristike

ovih projekata i tako omogućile bolje određivanje procesa i metodologije upravljanja IT projektima [6].

Za svakoga ko želi da se bavi ovom oblašću izuzetno je važno da dobro razume specifičnosti softvera kao proizvoda, karakteristike njegovog kvaliteta kao preduslova zadovoljstva korisnika i specifičnosti procesa razvoja softvera i upravljanja softverskim projektima [10].

Naime, samo pisanje programa stvar je tehnike koja realizuje korake koji su predviđeni u postupku za rešavanje posmatranog problema. Međutim, iste korake za rešavanje problema svako od nas bi realizovao na svoj način, čak i kada bi koristili isti programski jezik za pisanje programa. To znači da pisanje programa, tj. izrada softvera sadrži u sebi određenu kreativnu komponentu od koje će svakako zavisiti i veličina i složenost napisanog programa [10].

Prema ISO_2382 "Softver je intelektualna kreacija koja sadrži programe, procedure, pravila i odgovarajuću dokumentaciju koja se odnosi na rad nekog sistema za obradu podataka" [10].

Iako softverski projekti dele veliki broj zajedničkih atributa sa klasičnim projektima, ipak poseduju nekoliko karakteristika koji ih čine veoma različitim od drugih vrsta inženjerskih projekata [10], [11]

- Nevidljivost (proizvod je neopipljiv, tj. predstavlja logičku, a ne fizičku tvorevinu. Ne postoji fizički objekat sa kojim se radi. Ulaz u projekat predstavlja zamisao, tj. opis funkcionalnosti koje se očekuju kao izlaz iz projekta. Zbog navedenog, kao i zbog visokog nivoa dinamičnosti sredine u kojoj se sprovode projekti razvoja softvera, rezultat projekta je često teško predvidiv.
- Kompleksnost - Softverski projekti često mogu biti kompleksniji od drugih razvojnih projekata, imaju i u vidu veliki broj veza u samom predmetu projekta.
- Fleksibilnost - Imaju i u vidu dinamičnost sredine u kojoj se projekti elektronskog poslovanja realizuju, oni se moraju realizovati na veoma fleksibilan način, kako bi bilo moguće izvršiti adaptaciju u svakom trenutku realizacije projekta.
- Nedostatak iskustva - Softversko inženjerstvo je nova disciplina, pa jednostavno nemamo mnogo saznanja o načinima upravljanja softverskim projektima, naročito velikim softverskim projektima.
- Veliki softverski projekti su često napravljeni na osnovu zahteva određenog klijenta (custom – made). Većina velikih softverskih sistema je specifična, pa je iskustvo stečeno na jednom projektu od male pomoći u drugom.

- Tehnologija se menja veoma brzo - Na ve ini velikih softverskih projekata koriste se nove tehnologije; za mnoge projekte, ovo je ak i *raison d'etre* (razlog postojanja, smisao)

Svako ko se bavi razvojem softvera zna da je to složen i rizi an posao, a njegovi u esnici su uvek u potrazi za novim idejama koje e ih dovesti do boljih rezultata, boljeg softvera. Sre om, softverski inženjering je još uvek mlada profesija koja se razvija i kod koje su svake godine o igledne inovacije i poboljšanja u najboljim praksama.

3. RAZLOZI ZA NEUSPEH PROCESA RAZVOJA SOFVERA

Na po etku se proces razvoja softvera svodio na to da se napiše softverski kod. Unapre enja koja su se dogodila u industriji su dovela do toga da proces postane kompleksniji i uklju i i arhitekta, analiti are, programere, testere i korisnike u razvoj koda. Iako se o procesu razvoja softvera mnogo govori i zna, ti projekti i dalje imaju najve i procenat neuspeha u realizaciji. Razlozi su mnogobrojni, ali naj eš i su:

1. Projektni ciljevi nisu potpuno definisani: veoma je važno znati koje su potrebe i o ekivanja; šta je to što bi softver trebalo da obezbedi. Precizno definisani rezultati dovode do boljeg definisanja ciljeva i mogu minimizirati šanse za neuspeh.

2. Loše planiranje i procene: veoma je važno proceniti budžet i vremenski okvir koji je realisti an i u skladu sa definisanim ciljevima. To je mogu e jedino ako postoji razumevanje funkcija koje je potrebno da softver obezbedi krajnjim korisnicima.

3. Neadekvatna ili nedostatak projektno metodologije: u procesu razvoja softvera ne sme se potceniti uloga projektnog menadžera. Uspešan razvoj softvera je teško posti i bez pravog vo stva, jer se radi o procesu u kom je neophodno koordinirati razli ite aspekte projekta. Projektni menadžer je zadužen za nadgledanje procesa, predvi anje prepreka i usmeravanje stvari na željeni put.

4. Uzimanje zahteva – naj eš i problem koji ima kriti an uticaj na realizaciju projekta. Ukoliko se ne odvoji dovoljno vremena za razumevanje potreba korisnika, dolazi do neuspeha, a u nekim situacijama i potreba za nemogu im korekcijama. Najbitnije je prvo razumeti i usvojiti zahteve klijenata, pa tek nakon toga, pristupiti prilago avanju softverskog rešenja. Jedan od saveta za izbegavanje neuspeha u ovoj fazi je i angažovanje poslovnog analiti ara, koji e prikupiti zahteve klijenta uzimaju i u obzir i netehni ke aspekte.

5. Razumevanje jezika – naj eš i problem je to što IT stru njaci i klijenti misle da pri aju istim jezikom, što uglavnom nije ta no. To dovodi do situacije da IT stru njak isporu i proizvod koji klijent u stvari nije

tražio. Ove probleme u komunikaciji je najteže rešiti, jer se identifikuju tek nakon završenog procesa. Zbog toga bi bilo poželjno da u projektnom timu postoji bar jedna osoba koja jednako dobro razume informacione tehnologije i poslovne procese.

6. Upravljanje o ekivanjima – naj eš e se od IT stru njaka o ekuje nemogu e, jer ljudi koji ne razumeju dovoljno proces, razmišljaju o softveru kao o magi nom štapi u koji e rešiti sve probleme. Ovde je najvažnija uloga projektnog menadžera, jer bi on trebalo da upravlja o ekivanjima klijenata i vra a ih na razuman nivo [12].

Zbog kompleksne prirode softverskih projekata i brzog razvoja tehnologije koja je uklju ena u proces, razvoj novog softverskog proizvoda zahteva sistematski pristup. Faze uspešnog projekta razvoja softvera su: analiza, dizajn, implementacija, testiranje, rasporeivanje, održavanje, merenje i pra enje progressa, razvoj, automatsko adresiranje, testiranje. Imaju i u vidu da je razvoj i primena softvera kontinuiran proces, važno je angažovati profesionalce koji idu u korak sa promenama u tehnologiji. Pored toga, neophodno je angažovati stru an projektni tim, na elu sa projektnim menadžerom, i koristiti metodologije projektnog menadžmenta, kako bi se pove avala verovatno a uspeha ovih projekata.

4. UPRAVLJANJE SOFTVERSKIM PROJEKTIMA

IT projekti predstavljaju svojevrsan izazov za projektno menadžere, jer podrazumevaju koordinaciju sa mnogo stejkholdera i integraciju razli itih tehnoloških kapaciteta. U svakoj fazi životnog ciklusa IT projekata veoma je važna uloga projektnog sponzora, projektnog menadžera i projektnog tima.

Uspešni timovi za razvoj softvera treba da uspostave ravnotežu između brze isporuke kvalitetnih softverskih sistema, ispunjenja zahteva stejkholdera, rešavanja rizika, uvo enja promena i poboljšanja na ina na koji rade svoj posao [13]. Omogu avanje efikasne komunikacije kroz odgovaraju e kanale i sastanke jedan je od klju nih poslova menadžera softverskih pro.

Proces upravljanja IT projektima se sastoji iz ve eg broja podprocesa. Faze mogu biti razli ito definisane, od strane razli itih autora, ali, bez obzira na razli te definicije, moglo bi se re i da se proces upravljanja softverskim projektima sastoji iz slede eg:

1. Koncipiranje – obuhvata: izbor i finansiranje projekata, identifikaciju klju nih stejkholdera, utvr i vanje svrhe i sadržaja projekta, pripremu projektno povelje

2. Definisanje zahteva - podrazumeva identifikaciju i jasno izražavanje zahteva klijenata, razdvajanje tehni kih i funkcionalnih zahteva, koriš enje

različitih metoda za prikupljanje zahteva, razvoj metodologija za pravljenje zahteva.

3. Planiranje – sastoji se od definisanja ključnih komponenti projektnog plana i procesa planiranja, izrade WBS-a, izrade vremenskog rasporeda, procene trajanja, resursa i troškova, planiranja rizika i odgovora na rizike, izrade pomoćnih planova, uključuju komunikacije, nabavku, kvalitet i sl.

4. Dizajn - preliminarne i detaljne aktivnosti dizajna, tipičan sadržaj dokumenata koji sadrže tehničku specifikaciju, tehnike dizajniranja koje se koriste u razvoju tehnika kao rešenja, odlučivanje o opcijama kupiti ili napraviti

5. Izrada - razvoj projektnog tima koji će izraditi i isporučiti proizvod, aktivnosti kontrole i obezbeđenja kvaliteta, testiranje i nadgledanje, ocena performansi projekta, razvoj i upotreba metodologija za upravljanje promenama zahteva, razvoj strategija za odgovor na rizike

6. Isporuka - ključne aktivnosti isporuke, razvoj strategija za proveru obuhvata i prihvatanje od strane klijenta.

Dosadašnja iskustva u ovoj oblasti su pokazala da aktivnosti pravljenja rasporeda i plana projekta i upravljanje rizikom predstavljaju kritične tačke u projektima razvoja softvera. Problem pravljenja rasporeda projekta sastoji se u odlučivanju ko šta radi tokom životnog ciklusa softverskog projekta. Ovo je ključno pitanje u praksi softverskog inženjerstva, jer se ukupnim budžetskim i ljudskim resursima mora optimalno upravljati, u cilju uspešnog okončanja projekta.

Aktuelni softverski projekti obično zahtevaju kompleksno upravljanje koje obuhvata pravljenje rasporeda, planiranje i nadgledanje. Postoji potreba da se nadgledaju ljudi i procesi, kao i da se efikasno alociraju resursi, kako bi se postigli konkretni ciljevi u okviru različitih ograničenja. Ciljevi su obično smanjenje trajanja projekta, smanjenje troškova projekta i povećanje kvaliteta proizvoda. U stvarnom projektu, projektni menadžer želi plan koji će pomiriti, koliko god je to moguće, ova tri suprotstavljena cilja [14].

Efikasno upravljanje softverskim projektima je ključno za postizanje uspešnog ishoda. Ciljevi projekta treba da budu dogovoreni na samom početku. Prilikom određivanja ciljeva treba uzeti u obzir njihove implikacije, u smislu stvarnih rezultata i uticaja koji će ti rezultati imati. Takođe, postoji potreba da se razmotri uticaj samog procesa razvoja. Projektni tim treba da bude dobro informisan o ovim pitanjima i da ima priliku da ih u potpunosti raspravi kako bi mogli da donesu svoje zaključke. Tim bi trebalo da razmotri sve implikacije tog plana, uključujući i etičke. To će možda zahtevati dodatna sredstva unutar i izvan organizacije. Pogrešno je postaviti preuski okvir za

razgovore, u pokušaju da se uštede novac i vreme. Široki skup pitanja je neminovno nastati tokom projekta. Ukoliko su članovi tima nepripremljeni, nedostaje im usmerenje i slabije obavljati posao. Zato je potrebno da projektni sponzor poseduje viziju i autoritet, kako bi obezbedio podršku projektnom timu, kao i da bude obavešten da razmatra i tehničke i etičke pitanja.

Obuka budućih stručnjaka u upravljanju softverskim projektima, tema je od velikog značaja [15].

Za eksperte na putu do uspeha u razvoju softverskih projekata od velike važnosti je znanje iz oblasti upravljanja softverskim projektima. Međutim, dovođenje stručne prakse u ulogu je težak zadatak. Isto budućim softverskim stručnjacima nemaju praktičnu obuku sa stvarnim životnim situacijama. Pored toga, izrazito teoretsko okruženje za učenje, poput onog u trenutnim nastavnim planovima za upravljanje softverskim projektima, prilično je nezanimljivo za buduću profesionalce [16]. Shodno tome, savremeni profesionalci stiču svoje iskustvo radom na realnim projektima, gde efekti pogrešnog plana ili donošenja odluka mogu dovesti do neuspeha projekta ili gubitka koristi za kompaniju za koju rade [15].

Kvalitet, rizik i uspešan razvoj softverskih projekata su tri koncepta koja su neosporno isprepleteni jedni sa drugim [17]. Dakle, upravljanje softverskim projektima je skup tehnika koje se koriste za razvoj i isporuku različitih vrste softverskih proizvoda. Ova disciplina u razvoju tradicionalno uključuje tehnička pitanja kao što su:

- izbor metodologije razvoja softvera,
- kako proceniti veličinu i raspored projekta,
- kako osigurati bezbednost,
- koje resurse je moguće ponovo upotrebiti
- koju razvojno okruženje koristiti za programiranje.

Disciplina takođe uključuje pitanja upravljanja kao što su: kada obaviti osoblje, koji su rizici za uspeh projekta, i kako projekat održavati na planiranom putu. Ovi izbori su zatim otelotvoreni u planu upravljanja softverskim projektom.

Razvoj softvera je često komplikovan i uključuje mnoge ljude iz različitih oblasti, različitih veština, iskustava i društvenih stavova. Postoje mnoge operativne odluke koje treba preduzeti tokom ove dugotrajne aktivnosti. Takođe, postoji mnogo različitih pristupa za kontrolu složenosti ove aktivnosti. Svim ovim metodama glavna svrha je pokušaj da se predvide i izbegnu sve mogućnosti koje mogu negativno uticati na softverski projekat. Negativne mogućnosti su one koji bi odložile isporuku softvera (koji obavlja željene funkcije) na blagovremen i isplativ način. Dodatna

svrha je da se izbegnu kasne promene u sistemu, jer što je kasnija promena, to je skuplja.

5. UPRAVLJANJE RIZIKOM NA PROJEKTIMA RAZVOJA SOFTVERA

Kada je u pitanju upravljanje rizikom na softverskim projektima, veliki broj neuspešnih projekata je iznedrio bazu naučnih lekcija sa smernicama o kritičnim tačkama softverskih projekata.

Boehm [18] je identifikovao deset najčešćih stavki rizika na koje se mora obratiti pažnja u projektima razvoja softvera, kao i tehnike koje se primenjuju za njihovo sprečavanje:

- Nedostatak kadrova - Potreban je dovoljan broj kadrova sa odgovarajućim iskustvom i znanjem u potrebnim oblastima.
- Nerealni raspored i budžet - Detaljno istraživanje i određivanje rokova i budžeta, inkrementalni razvoj, ponovna upotrebljivost softvera.
- Razvoj pogrešnih funkcija i svojstava – Organizaciona analiza, analiza cilja, formulacija operacija i konceptata, uključivanje klijenata u proces razvoja, analize kvaliteta.
- Razvoj pogrešnog korisničkog interfejsa – Pravljenje prototipa, različitih scenarija, uključivanje klijenata u proces razvoja.
- Pozlađivanje (preterivanje, dodavanje više funkcionalnosti/karakteristika nego što je potrebno) - Pravljenje prototipa, analize troškova i dobiti.
- Neprekidni niz izmena u zahtevima – Ograničavanje broja dozvoljenih izmena u zahtevima, skrivanje informacija, inkrementalni razvoj (odlaganje izmena za kasnije faze).
- Nedostaci u eksterno nabavljenim komponentama (eng. Benchmark) testovi, inspekcije, analiza kompatibilnosti, provera referenci.
- Nedostaci eksterno realizovanih poslova - Provera referenci, ugovori o nagradama, konkurentski dizajn ili prototipi (najbolji dobija posao), izgradnja timskog duha.
- Slabe performanse u realnom radu - Simulacije, benchmark testovi, pravljenje modela, pravljenje prototipova.
- Prevazilaženje mogućnosti računarske nauke - Tehnike analize, analize troškova i dobiti, pravljenje prototipova, provera referenci.

Pored toga, Jones [19] je dodatno predstavio tri ključna faktora rizika za softver, a koji se tiču rukovodilaca i menadžera softvera.

- Rizici povezani sa netačnim procenama i planiranjem rasporeda.
- Rizici povezani sa nepravilnim i optimističkim izveštavanjem o statusu.

- Rizici povezani sa spoljnim pritiscima, koji negativno utiču na softverski projekat.

Međutim, većina programera i projektnih menadžera doživljavaju procese i aktivnosti upravljanja rizikom kao dodatni posao i trošak. Procesi upravljanja rizikom su prva stvar koja se ukloni iz projektnih aktivnosti kada raspored projekta "isklizne". Kultura slobodnog duha u mnogim firmama za razvoj softvera je u sukobu sa količinom kontrole, što je neophodno za razvoj složenih softverskih sistema na disciplinovan način. Jones [19] napominje "osobnost IT je da se veoma kompleksni sistemi mogu graditi sa veoma niskim nivoom kontrole od strane pametnih, motivisanih ljudi". Mnogi stručnjaci za razvoj softverskih projekata smatraju da upravljanje rizikom i kontrola inhibiraju, odnosno koče kreativnost.

U tome se možda i kriju propusti i razlozi neuspeha softverskih projekata. Mnogi procesi upravljanja rizikom su kreirani kako bi pomogli organizacijama, ali je integracija tih procesa u organizacijama bila neuspešna [20]. Za upravljanje kompleksnim pitanjima vezanim za razvoj softvera, preporučuju se formalni procesi upravljanja rizikom, uz preporuku da se teoretski aspekti procesa prilagode praktičnim izazovima, kako bi implementacija procesa bila uspešna.

6. METODOLOGIJE ZA UPRAVLJANJE SOFTVERSKIM PROJEKTIMA

Da bi se efikasno upravljalo IT projektima potrebno je imati odgovarajuću metodologiju prilagođenu specifičnostima IT projekata i običajima kada je sposoban da primeni metodologiju za upravljanje konkretnim projektom. Razmotrimo najpre, sasvim kratko, koje se metodologije, u teoriji i praksi, koriste za upravljanje IT projektima. U krajnjoj analizi osvrnemo se na sledeće metodologije [6]:

- PRINCE 2
- PMI
- IPMA
- YUPMA
- Agilne metodologije

PRINCE 2 je interesantna i složena metodologija koja je najpre formirana za informatičke projekte, ali je kasnije doradana i proširena i za druge vrste projekata. Metodologija obuhvata osam osnovnih elemenata (organizacija, planiranje, kontrola, kvalitet, rizik, i dr.) i osam osnovnih podprocesa (startovanje, iniciranje, planiranje, kontrolisanje, i dr.). Obzirom na složenost metodologije, ona zahteva solidno prethodno znanje iz projektnog menadžmenta, da bi bila adekvatno primenjena. Što se tiče informatičkih projekata, može se konstatovati da je ona pogodna za ovu vrstu projekata.

PMI metodologija je dosta poznata, obimna i složena procesna metodologija, koja obuhvata devet podprocesa ili funkcionalnih oblasti, kao što su upravljanje obimom, upravljanje vremenom, upravljanje troškovima, upravljanje rizikom, upravljanje kvalitetom, itd. Ova metodologija je sveobuhvatna i teška za konkretnu primenu za određenu vrstu projekata, pošto ne daje preciznija objašnjenja na ina primene, već samo okvir koji sadrži ulaz, proces i izlaz. Ona se može primeniti za informatičke projekte, ali je neophodna dalja razrada i prilagođavanje [21].

IPMA metodologija ne predstavlja procesnu metodologiju koja bi definisala osnovne podprocese upravljanja projektom, već se bazira na kompetencijama koje treba da poseduje projektni menadžer da bi mogao efikasno da upravlja određenim projektom. To su tehničke, biheviorističke i kontekstualne kompetencije određenog projektnog menadžera, na osnovu kojih IPMA vrši internacionalnu sertifikaciju projektnih menadžera, čime se potvrđuje sposobnost za upravljanje projektima.

YUPMA metodologija je razvijena u okviru udruženja YUPMA i predstavlja sublimaciju praktičnog rada na velikom broju projekata tokom zadnjih 30 godina. Metodologija je primenjena i proverena na nekoliko desetina projekata, koje su članovi YUPMA realizovali u zemlji i inostranstvu.

Navedena metodologija je dosta jednostavna i laka za primenu. Za informatičke projekte predlaže se i metodologija za takozvane biznis projekte, koja se može efikasno primeniti i za organizacione, konsultantske, marketinške i slične projekte.

Za IT projekte, posebno za projekte razvoja softvera, danas se predlažu Agilne metodologije, odnosno bolje agilne metode ili agilni pristupi. U početnom periodu, krajem devedesetih godina agilne metodologije su se razvijale i koristile za upravljanje IT projektima, pre svega za projekte razvoja softvera. Danas se sve više agilne metodologije pominju i kao efikasno sredstvo za upravljanje razvojem proizvoda u različitim oblastima poslovanja, mada ta ideja nije ozbiljnije raširena.

Kada se govori o agilnim metodologijama najčešće se vrši poređenje sa tradicionalnim pristupom upravljanja projektima koji mnogi nazivaju model vodopada. Ovaj pristup se izvodi iz činjenice, gde se upravljanje projektom obavlja kroz više faza koje se nadovezuju jedna na drugu. Kada se jedna faza završi prelazi se na sledeću i tako redom. Kod upravljanja projektima razvoja softvera počinje se sa fazom inicijalizacije, pa se ide na fazu definisanje zahteva, nakon ove faze ide se na razvoj, itd.

Kod projekata razvoja softvera može se pojaviti problem da se u toku rada uvode neke promene,

odnosno da se vrati na prethodnu fazu i uvedu potrebne promene, što tradicionalni pristup ne podrazumeva. Zbog toga se smatra da tradicionalni pristup nije dovoljno fleksibilan za IT projekte, koji najčešće nemaju vrsto definisanu strukturu, već se dešava da se u toku razvoja menjaju zahtevi i specifikacije, čime se menja i osnovna struktura projekta. Agilne metodologije se najčešće definišu kao upravljanje metodologije razvoja softvera koje koriste aktivnu ulogu naručioaca, uvođenje stalnih promena i specifičan timski rad, pri tome je značajno podržavati stalnu komunikaciju kroz redovne sastanke i aktivnu razmenu informacija. Za razliku od tradicionalnih metodologija, agilne metodologije rade kroz kratke razvojne cikluse, koji se zovu sprintovi, nakon kojih završetka je moguće isporučiti deo proizvoda klijentu na osnovu njegovog mišljenja može se dodatno usavršavati proizvod. Iz toga se može zaključiti da agilne metodologije podrazumevaju vrstu vezu i saradnju sa klijentima, mogućnost neprekidnog uvođenja promena i timski rad u kome su se članovi tima samo organizovali i slobodni u delovanju. Rad u manjim delovima projekta, odnosno sprintovima, omogućava bržu isporuku delova projekta klijentu, radi dobijanja njegovog mišljenja i eventualnih novih zahteva i fleksibilne mogućnosti uvođenja novih promena i zahteva radi dobijanja željenog rezultata.

Agilne metodologije zahtevaju i drugačiji pristup uvođenju projektnog tima, obzirom da su IT stručnjaci koji učestvuju u razvoju softvera individualci koji teško podnose krute principe i zahteve timskog rada. Za njih je potrebno da se koriste principi samoorganizovanih i samostalnih timova, kod kojih članovi timova imaju znatno veću samostalnost i slobodu u radu i posebne uslove u pogledu mesta u kome rade. Zbog svega toga treba reći i da su uloge projektnog menadžera kod upravljanja softverskim projektima različite od uloga koje imaju projektni menadžeri kod drugih vrsta projekata. Danas se predlaže već i broj agilnih metodologija, a među najpoznatijim su Scrum, Lean, Ekstremno programiranje i one obuhvataju najveći broj napred navedenih principa.

7. ZAKLJUČAK

Kao i mnogi drugi upravljački zadaci, upravljanje projektima razvoja softvera je dinamičan proces koji se sastoji iz planiranja, realizacije, praćenja i kontrole realizacije zadataka. Zbog činjenice da se stvarni događaji na softverskim projektima skoro uvek razlikuju od onih definisanih u projektnim planovima, projektni menadžeri moraju konstantno preispitivati i prilagođavati angažovanje resursa, procene vremena i troškova promenljivom projektnom i organizacionom okruženju. Projekti razvoja softvera su kompleksni, multidimenzionalni poduhvati koji su, po svojoj

prirodi, podložni neuspehu. Rizik, kao mogućnost da se ostvare neželjeni događaji je imanentan ovim projektima, te je stoga neophodno upravljanje, u cilju minimiziranja negativnih uticaja na ishode projekta. Potencijalni gubici i troškovi su veliki i iznenađujući, pa je neophodno da svi učesnici u ovim projektima učestvuju od drugih i primenjuju najbolje prakse u realizaciji ovih projekata. Proces razvoja, naročito kada je softver u pitanju, je složen i nailazi na mnoge probleme. Iako se ovi problemi ne mogu potpuno eliminisati, mogu se kontrolisati primenom metoda za upravljanje projektima. U radu je dat sažet prikaz metodologija koje se mogu koristiti u upravljanju softverskim projektima, sa posebnim osvrtom na specifičnosti softverskih projekata, prepoznate razloge neuspeha i neželjene rizike koje se sa kojima se suoavaju projektni timovi i projektni menadžeri, odgovorni za realizaciju.

LITERATURA

- [1] Cerpa N, Verner J. M, Why did your project fail? Commun. ACM 52 (12), 130-134, 2009.
- [2] Glass R. I, The Standish report: does it really describe a software crisis? Commun. ACM 49 (8) 15-16, 2006.
- [3] Ewusi-Mensah, K, Software Development Failures: Anatomy of Abandoned Projects. MIT Press, USA, 2003.
- [4] Savolainen P, Ahonen J, & Richardson I. When did your project start? – The software supplier's perspective. Journal of Systems and Software, 104, June 2015.
- [5] Boehm B. W, Some future trends and implications for systems and software engineering processes. J. Syst. Eng. 9 (1), 1–19, 2006.
- [6] Jovanović P, Upravljanje projektom, VŠPM, Beograd, 2015.
- [7] Schwalbe K, Information technology project management, 5th edition, Thomson Course Technology, USA, 2007.
- [8] Snedaker S, How to Cheat at IT Project Management, Syngress Publishing Inc., Canada, 2005.
- [9] Raković R, Upravljanje ICT projektima, VŠPM, Beograd, 2010.
- [10] Raković R, Kvalitet softvera i softverski projekti, Energoprojekt InGraf d.o.o., Beograd
- [11] Brooks, F. P, No silver bullet essence and accidents of software engineering. IEEE Computer, 20 (4), 1987.
- [12] Spenser M. Understanding the Software Development Process, available at projectsmart.co.uk, 2010.
- [13] 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.
- [14] Alba E, & Chicano, F. J. (). Software project management with GAs. Information Sciences, 177(11), June, 2007.
- [15] Calderón A, & Ruiz M, A systematic literature review on serious games evaluation: An application to software project management. Computers & Education, 87, September, 2015.
- [16] Ibrahim I, Teaching project management for it students: methods and approach. In 2nd International Conference on Education and Management Technology (pp. 185-191), 2011.
- [17] Sarigiannidis L, & Chatzoglou P. D, Quality vs risk, An investigation of their relationship in software development projects. International Journal of Project Management, 32(6), August, 2014.
- [18] Boehm B. W, Software risk management principles and practices, IEEE Software 8 (1) 32-41, 1991.
- [19] Jones C, Minimizing the risks of software development, Cutter IT Journal 11 (6), 13-21, 1998,
- [20] Bjelica D, Todorović M, Mitrović Z, Risk appraisal for software projects in accordance with project management maturity models, Serbian Project Management Journal, Volume 3, Issue 2, December 2013.
- [21] A Guide to the Project Management Body of Knowledge, PMBoK Guide, PMI, USA, 2008.

SUMMARY

SOFTWARE PROJECT MANAGEMENT

At the present time, it is impossible to imagine a successful business without the use of information - communication technologies. The application of project management in the management of ICT projects respects the same basic principles and concepts of project management, as well as with other types of projects, however, it is necessary to analyse certain types of IT projects and their specific features, and to choose and adapt appropriate methodology for the management of certain types IT projects. In this paper, special attention was devoted to software development projects, their specificities, challenges that are in front of the participants in this process, as well as methodologies for managing projects, which can be used to overcome difficulties and achieve better results.

Key words: *project management, software development, challenges, methodologies*