



SE322 - INŽENJERSTVO ZAHTEVA

Upravljanje promenama zahteva

Lekcija 14

PRIRUČNIK ZA STUDENTE

SE322 - INŽENJERSTVO ZAHTEVA

Lekcija 14

UPRAVLJANJE PROMENAMA ZAHTEVA

- ✓ Upravljanje promenama zahteva
- ✓ Poglavlje 1: Proces upravljanja zahtevima
- ✓ Poglavlje 2: Kontrola verzija zahteva
- ✓ Poglavlje 3: Status zahteva
- ✓ Poglavlje 4: Upravljanje zahtevima
- ✓ Poglavlje 5: Upravljanje promenama
- ✓ Poglavlje 6: Vežba
- ✓ Poglavlje 7: Domaći zadatak
- ✓ Poglavlje 8: Projektni zadatak
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Uvodne napomene

Ako imate velike zahteve, dobijate samo deo rešenja; njima se mora da dobro upravlja i i da se efikasno komuniciraju među učesnicima projekta. Kontrola verzije pojedinačnih zahteva i skupa zahteva jedna je od glavnih aktivnosti upravljanja zahtevima.

Većina programera naišla je na izgled jednostavnu promenu koja se ispostavila daleko složenijom nego što se očekivalo. Programeri ponekad ne proizvode ili ne mogu da proizvedu realne procene troškova i drugih pogodnosti predložene promene softvera. Uz to, kada se programeri koji žele ugone kupcu pristanu da dodaju poboljšanja koja korisnici zahtevaju, promene zahteva prolaze kroz zadnja vrata umesto da ih odobre odgovarajuća akteri. Ovakva nekontrolisana promena uobičajen je izvor haosa projekta, rasporeda, problema sa kvalitetom i teških osećaja. Ova lekcija opisuje i formalne prakse kontrole promena.

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

▼ Poglavlje 1

Proces upravljanja zahtevima

VIDEO PREDAVANJE ZA OBJEKAT "PROCES UPRAVLJANJA ZAHTEVIMA"

Trajanje video snimka: 37min 50sek

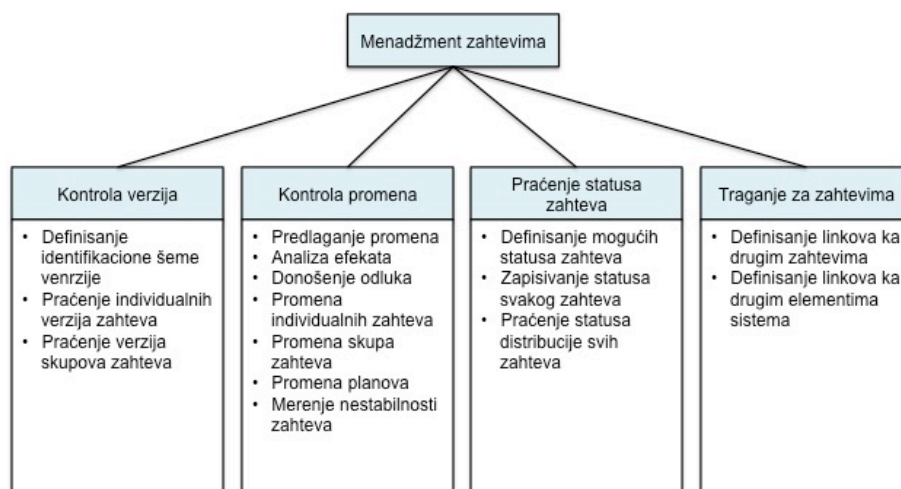
Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

GLAVNE AKTIVNOSTI PROCESA UPRAVLJANJA ZAHTEVIMA

Upravljanje zahtevima obuhvata sve aktivnosti koje održavaju celovitost, tačnost i valutu sporazuma u toku projekta

Upravljanje zahtevima obuhvata sve aktivnosti koje održavaju celovitost, tačnost i valutu sporazuma u toku projekta. Na slici 1 prikazane su osnovne aktivnosti upravljanja zahtevima u četiri glavne kategorije: kontrola verzije, kontrola promena, praćenje statusa zahteva i traženje zahteva.

Vaša organizacija treba da definiše aktivnosti za koje se očekuje da će projektni timovi obavljati radi upravljanja svojim zahtevima.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 1.1 Glavne aktivnosti upravljanja zahtevima

Dokumentovanje ovih aktivnosti i vežbača u njihovom izvođenju omogućava članovima organizacije da ih dosledno i efikasno sprovede. Razmotrite sledeće teme:

- Skup alata, tehnika i konvencija za razlikovanje verzija pojedinačnih zahteva i zahteva
- Način na koji se setovi zahteva usvajaju i zasnivaju (videti Poglavlje 2, „Zahtevi iz perspektive kupca“)
- Načini na koje se predlažu, ocenjuju, pregovaraju i prenose novi zahtevi i promene postojećih
- Kako proceniti uticaj predložene promene
- Atributi zahteva i postupci praćenja statusa zahteva, uključujući status zahteva koje ćete koristiti i koji ih mogu menjati
- Ko je odgovoran za ažuriranje zahteva, prati informacije i kada
- Kako pratiti i rešavati probleme sa zahtevima
- Kako će planovi i obaveze projekta odražavati promene zahteva
- Kako efikasno koristiti alat za upravljanje zahtevima (RM)

OSNOVNA LISTA ZAHTEVA

BA postavlja mehanizme skladištenja zahteva, definisati attribute zahteva, koordinisati status zahteva i ažurirati podatke u tragovima, a po potrebi nadgledati i promeniti aktivnosti.

Možete uključiti sve ove informacije u jedan opis procesa upravljanja zahtevima. Alternativno, možda biste radije napisali odvojene kontrole verzija, kontrolu promena, analizu uticaja i procedure praćenja statusa. Ovi postupci bi trebalo da se primenjuju u vašoj organizaciji jer predstavljaju zajedničke funkcije koje bi svaki projektni tim trebao da obavlja. Poglavlje 31, „Poboljšanje procesa zahteva“, opisuje nekoliko korisnih procesnih sredstava za upravljanje zahtevima.

U opisima procesa treba da se identifikuje uloga tima koji poseduje svaku od aktivnosti upravljanja zahtevima. Poslovni analitičar projekta obično ima glavnu odgovornost za upravljanje zahtevima. BA će postaviti mehanizme skladištenja zahteva, definisati attribute zahteva, koordinisati status zahteva i ažurirati podatke u tragovima, a po potrebi nadgledati i promeniti aktivnosti. Opis postupka takođe treba da ukaže ko ima ovlašćenja da modifikuje proces upravljanja zahtevima, kako treba da se postupa sa izuzecima i put eskalacije zbog prepreka.

Razvoj zahteva uključuje aktivnosti za dobijanje, analiziranje, preciziranje i potvrđivanje zahteva softverskog projekta. Rezultati razvojnih zahteva uključuju poslovne zahteve, zahteve korisnika, funkcionalne i nefunkcionalne zahteve, rečnik podataka i različite modele analiza. Nakon pregleda i odobrenja, svaki definisani podskup ovih stavki predstavlja osnovnu osnovu zahteva.

Kao što je opisano u Poglavlju 2, osnovna lista zahteva (requirements baseline) je skup zahteva sa kojima su se akteri složili, često definišući sadržaj određenog planiranog izdanja ili iteracije razvoja. Projekat može imati dodatne sporazume u vezi sa ishodima, ograničenjima, rasporedom, budžetom, zahtevima tranzicije i ugovorima; one leže izvan okvira programa ovog predmeta.

U vreme kada je set zahteva baziran - obično nakon pregleda i odobrenja - zahtevi se stavljaju pod upravljanje konfiguracijom (ili promenom). Naknadne promene mogu se izvršiti samo kroz definisani postupak kontrole promena projekta. Pre baziranja, zahtevi se i dalje razvijaju, tako da nema smisla nametati nepotrebne prekomerne procese tim izmenama. Bazna linija (osnovna lista zahteva) može se sastojati od nekih ili svih zahteva u određenom SRS-u (bilo za ceo proizvod ili za jedno izdanje), ili određenog skupa zahteva koji se čuvaju u RM alatu ili dogovorenog skupa korisničkih priča za jednu iteraciju na agilnom projektu.

Ako se opseg izdanja promeni, ažurirajte zahteve u skladu s tim. Razlikovati zahteve u određenoj osnovnoj liniji od ostalih koji su predloženi, ali nisu prihvaćeni, dodeljuju se drugoj osnovnoj liniji ili ostaju nerazvrstani u zaostatku proizvoda. Ako su zahtevi navedeni u obliku dokumenta kao što je SRS, jasno ga identifikujte kao osnovnu verziju da biste ga razlikovali od prethodnih nacрта.

NAKNADNE PROMENE ZAHTEVA

Bez obzira kako odgovarate na promenljive zahteve, prihvatite realnost prilagođavanja očekivanja i obaveza kada je to potrebno.

Smeštanje zahteva u RM alatu olakšava identifikaciju onih koji pripadaju određenoj osnovnoj liniji i upravljanje promenama na toj osnovnoj liniji.

Razvojni tim koji prihvati predložene promene ili dopune zahteva možda neće biti u mogućnosti da ispuni svoj postojeći raspored i obaveze o kvalitetu. Rukovodilac projekta mora da pregovara o izmenama tih obaveza sa pogođenim menadžerima, kupcima i drugim zainteresovanim stranama. Projekat može prilagoditi nove ili izmenjene zahteve na različite načine:

- Određivanjem zahteva sa nižim prioritetom za kasnije iteracije ili njihovim smanjenjem u potpunosti
- Pribavljanjem dodatnog osoblja ili podučavanjem nekog dela
- Proširenjem rasporeda isporuke ili dodavanjem iteracija agilnom projektu
- Žrtvovanjem kvaliteta za isporuku do prvobitnog datuma

Nijedan pojedinačni pristup nije univerzalno ispravan, jer se projekti razlikuju po fleksibilnosti funkcija, osoblja, budžeta, rasporeda i kvaliteta. Izbor se treba zasnivati na poslovnim ciljevima projekta i prioritetima koje su ključni akteri utvrdili tokom pokretanja projekta. Bez obzira kako odgovarate na promenljive zahteve, prihvatite realnost prilagođavanja očekivanja i obaveza kada je to potrebno. To je bolje nego zamisliti da će sve nove funkcije na neki način biti ugrađene u originalni datum isporuke bez prekoračenja budžeta, izgaranja članova tima ili kompromisa kvaliteta.

▼ Poglavlje 2

Kontrola verzija zahteva

KAKO KOTROLISATI VERZIJE ZAHTEVA?

Kada bazni dokument dokumentujete, prvo arhivirajte označenu verziju, zatim prihvatite sve revizije, a zatim spremite čistu verziju kao novu osnovnu liniju, spremnu za sledeći krug izmena.

Kontrola verzije - na jedinstven način identifikuje različite verzije predmeta - primenjuje se na nivou pojedinačnih zahteva i skupa zahteva, najčešće predstavljenih u obliku dokumenata. Započnite kontrolu verzija čim sastavite zahtev ili dokument kako biste mogli da zadržite istoriju promena.

Svaka verzija zahteva mora biti jedinstveno identifikovana. Svaki član tima mora biti u mogućnosti da pristupi trenutnoj verziji zahteva. Promene moraju biti jasno dokumentovane i saopštene svima koji su pogođeni. Da biste umanjili zabunu i pogrešno komuniciranje, dozvolite samo određenim pojedincima da ažuriraju zahteve i budite sigurni da se identifikator verzije menja kad god se izvrši ažuriranje. Svaka cirkulirana verzija dokumenta o zahtevima ili svaki zahtev u alatu treba da sadrži istoriju revizije koja identifikuje izvršene promene, datum svake promene, pojedinca koji je izvršio promenu i razlog svake promene.

Najbolji pristup kontroli verzije je čuvanje zahteva u alatu za upravljanje zahtevima. RM alati prate istoriju promena izvršenih za svaki zahtev, što je dragoceno kada se morate vratiti na stariju verziju. Takav alat omogućava komentare koji opisuju obrazloženje odluke o dodavanju, izmeni ili brisanju zahteva. Ovi komentari su korisni ako zahtev postane tema za raspravu u budućnosti.

Kada bazni dokument dokumentujete, prvo arhivirajte označenu verziju, zatim prihvatite sve revizije, a zatim spremite sada čistu verziju kao novu osnovnu liniju, spremnu za sledeći krug izmena. Skladištite dokumente sa zahtevima u alatki za kontrolu verzije, poput one koju vaša organizacija koristi za kontrolu izvornog koda putem procedura odjave i prijave. Ovo će vam omogućiti da se vratite na starije verzije ako je potrebno i da znate ko je promenio svaki dokument, kada i zašto.

Najjednostavniji mehanizam za kontrolu verzija je ručno označavanje svake revizije dokumenta prema standardnoj konvenciji. Šeme koje pokušavaju da razlikuju verzije dokumenata na osnovu datuma sklone su konfuziji. Koristim konvenciju koja prvu verziju bilo kojeg novog dokumenta označava svojim naslovom i „Nacrt verzije 1.0“. Sledeći nacrt zadržava isti naslov, ali je identifikovan kao „

Nacrt verzije 1.0 2. „Autor uvećava broj nacrtu sa svakom iteracijom sve dok dokument ne bude odobren i baziran. Tada se identifikator verzije menja u „Verzija 1.0 odobrena“, ponovo zadržavajući isti naziv dokumenta. Sledeća verzija je ili „Verzija 1.1 skica 1“ za manju reviziju ili „Verzija 2.0 skica 1“ za značajne promene. (Naravno, „glavni“ i „manji“ su subjektivni i

zavise od konteksta.) Ova šema jasno razlikuje nacrt i baznu verziju dokumenta, ali zahteva ručnu disciplinu od strane onih koji modifikuju dokumente.

▼ Poglavlje 3

Status zahteva

ATRIBUTI ZAHTEVA

Svaki zahtev se može predstaviti kao objekt sa atributima koji uspostavljaju kontekst i pozadinu zahteva. Preko atributa, alati mogu lako da pretražuju zahteve.

Zamislite svaki zahtev kao objekt sa svojstvima koja ga razlikuju od drugih zahteva. Pored svog tekstualnog opisa, svaki zahtev treba da sadrži prateće podatke ili attribute povezane sa njim. Ovi atributi uspostavljaju kontekst i pozadinu za svaki zahtev. Vrednosti atributa možete uneti u dokument, proračunsku tabelu, bazu podataka ili - najefikasnije - u alat za upravljanje zahtevima. Lako je koristiti i više od nekoliko atributa zahteva sa dokumentima.

RM alati obično nude nekoliko atributa sistema generisanih pored toga što vam omogućavaju da definišete druge, od kojih se neki mogu automatski popuniti. Alati vam omogućavaju da pretražite bazu podataka kako biste pregledali odabrane podskupove zahteva na osnovu vrednosti atributa. Na primer, mogli biste navesti sve zahteve visokog prioriteta koji su dodeljeni Shari-u za implementaciju u izdanju 2.3 i imaju status odobrenja. Sledi lista potencijalnih atributa zahteva koje treba uzeti u obzir:

- Datum kreiranja zahteva
- Trenutni broj verzije zahteva
- Autor koji je napisao zahtev
- Prioritet
- Status
- Poreklo ili izvor zahteva
- Obrazloženje zahteva
- Broj izdanja ili iteracija kojoj je zahtev dodeljen
- Zainteresovani za kontakt sa pitanjima ili za donošenje odluka o predloženim promenama
- Metoda validacije koja se koristi ili kriterijumi prihvatanja

Zahtevi planirani za izdanje menjaće se kako se dodaju novi zahtevi, a postojeći se brišu ili odlažu. Tim može žonglirati posebnim dokumentima o zahtevima za višestruka izdanja ili iteracije. Ostavljanje zastarelih zahteva u SRS-u može zbuniti čitaoce oko toga da li su ti zahtevi sadržani u toj osnovnoj liniji. Rešenje je skladištenje zahteva u RM alat i definisanje atributa Number Release. Odlaganje zahteva znači promenu njegovog planiranog izdanja, tako da jednostavno ažuriranje broja izdanja premešta zahtev u drugu osnovnu liniju. Izbrisati i odbiti zahteve pomoću atributa statusa, kao što je opisano u sledećem odeljku.

PRAĆENJE STATUSA ZAHTEVA

Praćenje statusa znači upoređivanje mesta gde se zaista nalazite u projektu razvoja u određenom trenutku u odnosu na očekivanja

Praćenje statusa svakog funkcionalnog zahteva tokom razvoja pruža preciznije merilo napretka projekta. Status je bio jedan od atributa zahteva koji su predloženi u prethodnoj sekciji. Praćenje statusa znači upoređivanje mesta gde se zaista nalazite u projektu razvoja u određenom trenutku u odnosu na očekivanja šta znači „kompletno“ za ovaj razvojni ciklus. Možda ste planirali da implementirate samo određene tokove slučaja upotrebe u trenutnom izdanju, ostavljajući potpunu implementaciju za buduće izdanje. Pratite status samo onih funkcionalnih zahteva koji su izvršeni za trenutno izdanje, jer to je skup koji bi trebalo da bude 100% završen pre nego što proglasite uspeh i otpremite izdanje.

Tabela na slici 1 navodi nekoliko mogućih statusa zahteva. Neki praktičari dodaju i druge, kao što su *Dizajnirani* (dizajnerski elementi koji ispunjavaju funkcionalne zahteve su kreirani i pregledani) i *Isporučeni* (softver koji sadrži zahtev je u rukama korisnika, kao za prihvatanje ili beta testiranje). Vredno je voditi evidenciju o odbijenim zahtevima i razlozima zbog kojih su odbijeni. Odbijeni zahtevi imaju mogućnost da se pojave kasnije tokom razvoja ili na budućem projektu. Status odbijenog omogućava vam da držite predloženi zahtev na raspolaganju za eventualnu buduću referencu, a da pritom ne skupljate set određenih zahteva izdanja. Ne morate da pratite sve moguće statuse u tabeli na slici 1; izaberite one koji dodaju vrednost vašim zahtevima.

Razvrstavanje zahteva u nekoliko kategorija statusa je smislenije od pokušaja praćenja postotka ispunjenosti svakog zahteva ili kompletne početne vrednosti izdanja. **Ažurirajte status zahteva samo kada su ispunjeni određeni prelazni uslovi.** Određene promene statusa takođe zahtevaju ažuriranja podataka o traženju zahteva da bi se naznačilo koji su elementi projektnog rešenja, koda i ispitivanja adresirani na zahtev.

Status	Definicija
Predložen	Zahtev je zahtevan od strane ovlašćenog izvora
U toku	Biznis analitičar aktivno radi na definisanju zahteva.
Predložen	Napisana je početna verzija zahteva.
Prihvaćen	Zahtev je analiziran, izvršena je procena njegovog efekta, i postavljen je u osnovnu listu zahteva za specifično izdanje. Ključni akteri su saglasni da ga uključe, a grupa za razvoj softvera je sada opredeljena da ga primeni.
Primenjen	Kod koji primenjuje zahtev je projektovan, napisan i jedinično testiran. Zahtev je povezan sa odgovarajućim elementima projektnog rešenja i koda. Softver koji primenjuje zahtev je sada spreman za testiranje, recenziju i druge provere.
Proveren	Zahtev je zadovoljio svoje kriterijume prihvatanja, što znači da je potvrđena primena zahteva. Zahtev je povezan sa odgovarajućim testovima. Zahtev se smatra sada kompletnim.
Odložen	Prihvaćen zahtev se sada planira za primenu u kasnijem izdanju.
Obrisan	Prihvaćen zahtev je uklonjen iz osnovne liste zahteva. Uključuje objašnjenje o razlogu brisanja i informaciju o tome ko je je doneo odluku o brisanju.
Odbijen	Zahtev je predložen, ali nikad nije prihvaćen i nije planiran za primenu ni u jednom budućem izdanju. Uključuje objašnjenje o tome zašto je odbijen i ko je o tome doneo odluku.

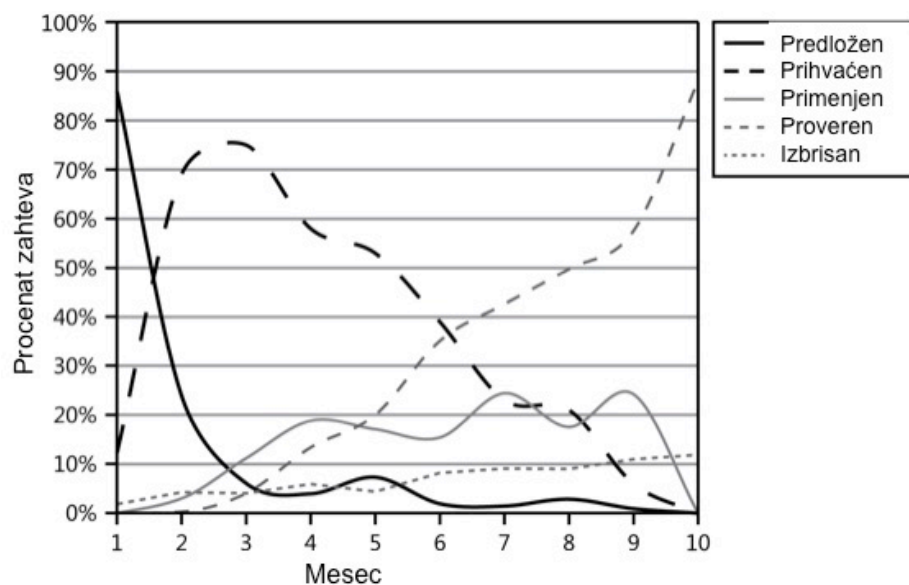
Izvor: Karl Wiegers, Joy Beatty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 3.1 Predloženi statusi zahteva

PRAĆENJE STATUSA ZAHTEVA TOKOM VREMENA

Broj zahteva se povećava kako se opseg izdanja dodaje i smanjuje kada se funkcionalnost ukloni sa osnovne linije.

Slika 2 ilustruje kako možete vizuelno nadgledati status skupa zahteva tokom hipotetičkog desetomesečnog projekta. Pokazuje procenat svih zahteva sistema. Svaki zahtev ima vrednost statusa na kraju svakog meseca. Praćenje distribucije u procentima ne pokazuje da li se broj zahteva u osnovnoj liniji vremenom menja. Broj zahteva se povećava kako se opseg izdanja dodaje i smanjuje kada se funkcionalnost ukloni sa osnovne linije. Krivulje prikazuju kako se projekat približava svom cilju potpune verifikacije svih odobrenih zahteva. Posao je obavljen kada svi zahtevi koji su mu dodeljeni imaju status *Verifikovani*, *izbrisani* ili *Odloženi*.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 3.2 Praćenje statusa zahteva tokom razvojnog ciklusa softvera

REŠAVANJE PROBLEMA SA ZAHTEVIMA

Skoro svi nedostaci prijavljeni rano u projektu povezani su sa pitanjima u zahtevima, kao što su traženje pojašnjenja o zahtevu, odluke o dometu, pitanja o dometu, i o izvodljivosti razvoja.

Tokom projekta će se pojaviti brojna pitanja, odluke i pitanja vezana za zahteve. Potencijalna pitanja uključuju stavke označene kao TBD, čekaju odluke, informacije koje su potrebne i sukobe koji čekaju rešenje. Lako je izgubiti iz vida ove otvorene probleme. Snimite probleme u alatu za praćenje problema tako da svi zainteresovani akteri imaju pristup njima. Proverite i rešavajte problem jednostavno i osigurajte da ništa ne prođe kroz pukotine. Neke od prednosti korišćenja alata za praćenje problema su:

- Pitanja iz višestrukih pregleda zahteva sakupljaju se tako da se nijedan problem nikada ne izgubi.
- Voditelj projekta može lako da vidi trenutni status svih pitanja.
- Za svaki broj može se dodeliti pojedinačni vlasnik.
- Istorija diskusija oko nekog pitanja može se zadržati.
- Tim može započeti razvoj ranije s poznatim setom otvorenih pitanja, a ne da čeka da se SRS popuni.

Rešite probleme sa zahtevima, tako da oni ne ometaju pravovremeno utvrđivanje visokokvalitetnih zahteva postavljenih za vaše sledeće izdanje ili ponavljanje. Grafikon koji prikazuje preostala izdanja i brzinu kojom se zatvaraju mogu vam pomoći da predvidite kada će se svi problemi zatvoriti, tako da možete ubrzati rešavanje problema ako je potrebno.

Kategorizacija problema će vam pomoći da odredite koji delovi zahteva su još potrebni. Malo otvorenih pitanja u odeljku može značiti ili da zahtevi još nisu pregledani ili da su otvorena pitanja uglavnom rešena.

Skoro svi nedostaci prijavljeni rano u projektu povezani su sa pitanjima u zahtevima, kao što su traženje pojašnjenja o zahtevu, odluke o dometu, pitanja o izvodljivosti razvoja i obavezama oko samih zahteva. Svi akteri mogu prijaviti pitanja dok pregledaju uslove. Tabela na slici 2 navodi nekoliko uobičajenih problema sa zahtevima koji se mogu pojaviti.

Tip problema	Opis
Pitanje zahteva	Nešto nije razumljivo ili nije odlučeno a vezano je za zahtev
Nepostojeći zahtev	Programeri su otkrili nedostajući zahtev za vreme projektovanja ili implementacije.
Netačan zahtev	Zahtev je loš. Treba ga korigovati ili isključiti zahtev.
Pitanje implementacije	Programeri primenjuju zahteve, ali oni imaju pitanja o tome kako bi nešto trebalo da radi, ili o nekom alternativnom projektnom rešenju.
Dupliranje zahteva	Dva ili više istovetna zahteva su otkrivena. Sve ih obrišite sem jednog.
Netraženi zahtev	Neki zahtev više nije potreban.

Izvor: Karl Wiegars, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 3.3 Uobičajeni tipovi problema sa zahtevima

▼ Poglavlje 4

Upravljanje zahtevima

PRAĆENJE I MERENJE ULOŽENOG RADA NA ZAHTEVIMA

Vreme provodeno na ove aktivnosti povezane sa zahtevima je ulaganje u uspeh projekta, a ne trošak.

Kao i kod razvoja zahteva, vaš projektni plan treba da sadrži zadatke i resurse za aktivnosti upravljanja zahtevima opisanim u ovoj lekciji. Ako pratite koliko truda ulažete u razvoj i upravljanje zahtevima, možete proceniti da li je bilo premalo, ispravno ili previše i u skladu sa tim prilagoditi svoje buduće planiranje. Merenje rada zahteva promenu kulture i individualnu disciplinu za evidentiranje svakodnevnih radnih aktivnosti. Praćenje uloženog rada nije dugotrajno koliko se ljudi ponekad boje. Članovi tima stiču dragocen uvid iz saznanja kako su zapravo proveli svoje vreme, u poređenju sa tim kako su mislili da provode vreme, u poređenju sa načinom na koji su trebali da provode vreme. Praćenje rada takođe pokazuje da li tim obavlja planirane aktivnosti vezane za zahteve.

Imajte na umu da uloženi rad nije isto što i proteklo kalendarsko vreme. Zadaci se mogu prekinuti; možda će im trebati interakcije sa drugim ljudima što dovodi do kašnjenja. Ukupni rad po zadatku, u jedinicama radnog vremena, možda se neće promeniti zbog takvih faktora (iako učestali prekidi smanjuju produktivnost pojedinca), ali trajanje kalendara se povećava.

Kada pratite napore na razvoju potreba, možda bi vam bilo dragoceno da odvojite vreme koje ljudi provode u ulozi BA i vreme koje provode drugi učesnici u projektu. Praćenje vremena BA pomoći će vam da isplanirate koliko BA treba u budućim projektima. Merenje ukupnog rada utrošenog na aktivnosti koje zahtevaju svi akteri daje vam osećaj ukupnih troškova aktivnosti na projektu.

Zabeležite broj sati provedenih u aktivnostima na razvoju potreba kao što su sljedeće:

- Aktivnosti planiranja u vezi sa zahtevima projekta
 - Održavanje radionica i intervjuja, analiza dokumenata i obavljanje drugih aktivnosti za izvlačenje
 - Pisanje specifikacija zahteva, kreiranje modela analize i određivanje prioriteta
 - Stvaranje i ocenjivanje prototipova namenjenih pomoći u razvoju zahteva
 - Pregled zahteva i obavljanje drugih aktivnosti validacije
- Trud posvećen sledećim aktivnostima računajte kao napor za upravljanje zahtevima:
- Konfigurisanje alata za upravljanje zahtevima za vaš projekat
 - Podnošenje promena zahteva i predlaganje novih zahteva
 - Procena predloženih promena, uključujući analizu uticaja i donošenje odluka
 - Ažuriranje skladišta zahteva

- Zahtevi za komuniciranje se menjaju zainteresovane strane
- Status zahteva za praćenje i izveštavanje
- Stvaranje informacija o tragovima zahteva

UPRAVLJANJE ZAHTEVIMA U PROJEKTIMA AGILNOG RAZVOJA SOFTVERA

Agilni projekti se prilagođavaju promenama tako što će proizvoditi proizvod kroz seriju razvojnih iteracija i upravljati dinamičkim zaostatkom proizvoda (backlog).

Agilni projekti se prilagođavaju promenama tako što će proizvoditi proizvod kroz seriju razvojnih iteracija i upravljati dinamičkim zaostatkom proizvoda (**backlog**) koji ostaje da se uradi. Kao što je opisano u lekciji 1, akteri u razvoju postižu dogovor o pričama koje će se implementirati u svakoj iteraciji. Nove priče koje kupci dodaju dok se ponavlja iteracija imaju prioritet u odnosu na preostale zaostale sadržaje i dodeljuju se budućim iteracijama. Nove priče mogu zameniti priče sa nižim prioritetom ako tim želi da zadrži originalni raspored isporuke. Cilj - kao što bi trebalo da bude za sve projekte - jeste da uvek radimo na pričama sa najvišim prioritetom kako bi što brže isporučili maksimalnu vrednost kupcima.

Neki agilni timovi, posebno veliki ili raspoređeni timovi, koriste agilni alat za upravljanje projektima da bi pratili status ponavljanja i priče dodeljene njemu. Priče i njihovi povezani kriterijumi prihvatanja i testovi prihvatanja mogu se postaviti u kategoriju "preostalog posla" (**backlog**) ili ga uneti u alat za upravljanje pričama. Status priče se može nadgledati korišćenjem statusa analognih onima opisanim u prethodnim sekcijama. Evo nekih mogućih statusa:

- Zaostatak (priča još nije dodeljena ponavljanju)
- Definisano (detalji priče su diskutovani i shvaćeni, a testovi prihvatanja su napisani)
- U toku (priča se realizuje)
- Završeno (priča je u potpunosti implementirana)
- Prihvaćeno (prijemni testovi su prošli)
- Blokirano (programer ne može da nastavi dok se nešto drugo ne reši)

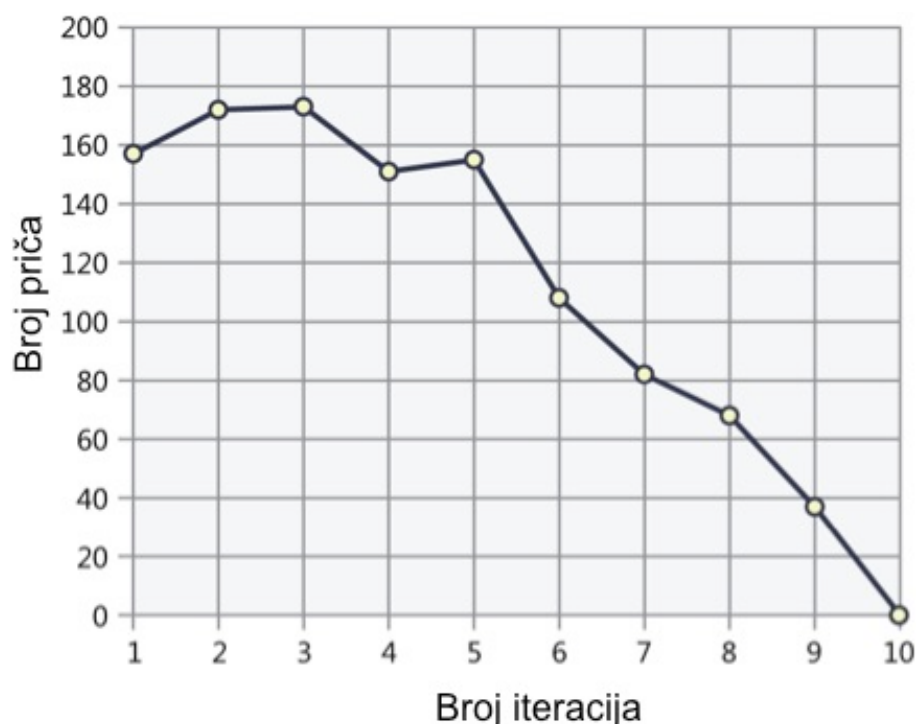
Agilni projekti obično prate njihov napredak grafikom ponavljanja. Tim procenjuje ukupnu količinu posla koji treba obaviti na projektu, često veličine u jedinicama tačke priče, koje su izvedene iz razumevanja korisničkih priča u zaostatku proizvoda (**backlog**). Ukupan broj tačaka priče proporcionalan je količini napora koji tim mora potrošiti da bi sproveo zahteve. Tim dodeljuje određene korisničke priče svakoj iteraciji na osnovu njihovog prioriteta i njihove procenjene veličine u pričama. Prošla ili prosečna brzina tima diktira broj tačaka priče koje tim planira da objavi u iteraciji određenog trajanja kalendara.

PRIKAZ BROJA PRIČA PO ITERACIJI PRI AGILNOM RAZVOJU SOFTVERA

Dijagram pomaže timu da izbegne sindrom „urađenog 90 odsto“, tako što je vidljiva količina preostalog posla, za razliku od obavljenog posla.

Razvojni tim dijagramski prikazuje broj priča korisnika koje ostaju da se rade u preostalim iteracijama, tj. u formi " zaostatka proizvoda" (backlog) na kraju svake iteracije. Ovaj se ukupan broj menja kako se posao završava, jer se trenutne priče bolje razumeju i preispituju, kako se dodaju nove priče i kako kupci uklanjaju čekanje posla iz zaostalog dela. To jest, umesto da prati broj i status pojedinačnih funkcionalnih zahteva ili karakteristika (koje mogu biti različitih veličina), tabela dijagram pokazuje ukupan posao koji treba da se uradi u određeno vreme.

Na slici 1 prikazana je dijagram broja priča po iteracijama hipotetičkog projekta. Imajte na umu da je opseg koji je ostao, mereno broja priča, zapravo povećan u iteracijama 2, 3 i 5. To ukazuje da je razvoju softvera dodato više novih funkcija nego što je dovršeno ili uklonjeno tokom iteracije. Dijagram pomaže timu da izbegne sindrom „urađenog 90 odsto“, tako što je vidljiva količina preostalog posla, za razliku od obavljenog posla, koji ne odražava neizbežne povećanja obima. Nagib dijagrama takođe otkriva predviđeni datum završetka projekta, tačku na kojoj nijedan posao ne ostaje u zaostatku.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 4.1 Prikaz broja priča po iteracijama pri agilnom razvoju

ZAŠTO UPRAVLJATI ZAHTEVIMA?

Upravljanje zahtevima pomaže da se osigura da se rad koji ulažete u razvoj zahteva ne prosipa.

Bilo da vaš projekat sledi sekvencijalni razvojni životni ciklus, jedan od različitih okretnih životnih ciklusa ili bilo koji drugi pristup, upravljanje zahtevima je suštinska aktivnost. Upravljanje zahtevima pomaže da se osigura da se rad koji ulažete u razvoj zahteva ne prosipa.

Efikasno upravljanje zahtevima smanjuje jaz u očekivanjima tako što su svi akteri na projektu informisane o trenutnom stanju zahteva u toku razvojnog procesa. Omogućuje vam da znate kuda ste krenuli, kako putuje i kada ste stigli na svoje odredište.

▼ Poglavlje 5

Upravljanje promenama

VIDEO PREDAVANJE ZA OBJEKAT "UPRAVLJANJE PROMENAMA"

Trajanje video snimka: 32min 59sek

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

ZAŠTO UPRAVLJATI PROMENAMA?

Nekontrolisana promena zahteva je uobičajen izvor haosa projekta, promene u rasporedu, problema sa kvalitetom i stresnih situacija.

Većina programera naišla je na izgled jednostavnu promenu koja se ispostavila daleko složenijom nego što se očekivalo. Programeri ponekad ne (ili ne mogu) da proizvedu realne procene troškova i drugih pogodnosti predložene promene softvera. Uz to, kada se programeri koji žele da pristanu da dodaju poboljšanja koja korisnici zahtevaju, promene zahteva prolaze kroz zadnja vrata umesto da ih odobre odgovarajući akteri. Ovakva nekontrolisana promena uobičajen je izvor haosa projekta, rasporeda, problema sa kvalitetom i stresnih situacija. Ovo poglavlje opisuje i formalne prakse kontrole promena i kako agilni projekti uključuju promene.

Promena softvera nije loše; u stvari, to je neophodno. Praktično je nemoguće definisati sve zahteve proizvoda unapred. Svet se menja kako razvoj napreduje: nastaju nove tržišne mogućnosti, menjaju se propisi i politike, a poslovne potrebe se razvijaju. Efikasan softverski tim može brzo reagovati na neophodne promene tako da proizvod koji grade pruža pravovremenu vrednost za kupca. Organizacija koja se ozbiljno bavi upravljanjem softverskim projektima mora da osigura da:

- Predložene promene zahteva promišljeno se procenjuju pre nego što se obavežu.
- Odgovarajući pojedinci donose informisane poslovne odluke o traženim promenama.
- Aktivnost promene postaje vidljiva pogođenim zainteresovanim stranama.
- Odobrene promene saopštavaju se svim učesnicima zahvaćeni promenom.
- Projekat uključuje izmene zahteva na dosledan i efikasan način.

Projekat uključuje izmene zahteva na dosledan i efikasan način.

Ali promena uvek ima cenu. Revizija jednostavne veb stranice može biti brza i laka; promena projektnog rešenja integrisanog kola može koštati desetine hiljada dolara. Čak i odbijeni

zahtev za promenu zahteva vreme potrebno da ga podnesete, procenite i odlučite da ga odbacite. **Ako akteri na projektu ne upravljaju promenama tokom razvoja, oni zaista neće znati šta će im se dostaviti, što u konačnici dovodi do nedostatka očekivanja.**

Problemi mogu nastati i ako programer implementira promenu zahteva direktno u kodu bez komunikacije sa ostalim članovima tima. Dokumentirani zahtevi tada postaju netačni prikaz onoga što proizvod čini. Kod može postati krhak ako se izvrše promene bez poštovanja arhitekture i strukture dizajna

UPRAVLJANJE "PUZEĆIM" PROMENAMA OKVIRA PROJEKTA

Problem nije u tome što se zahtevi menjaju, već što kasne promene mogu imati veliki uticaj na već obavljeni posao.

U idealnom svetu dokumentovali biste sve zahteve novog sistema pre početka izgradnje, a oni bi ostali stabilni tokom razvojnih napora. To je pretpostavka iza modela razvoja vodopada, ali to ne funkcioniše dobro u praksi. **U nekom trenutku morate zamrznuti zahteve za određenom iteracijom izdanja ili razvoja ili je nikad nećete dovršiti.** Međutim, gušenje promene prerano ignoriše stvarnosti da kupci nisu uvek sigurni šta im treba, poslovne potrebe se menjaju, a programeri žele da odgovore na te promene.

Rast zahteva uključuje novu funkcionalnost i značajne modifikacije koje su predstavljene nakon što je set zahteva utvrđen kao polazna baza. Što duže traje projekat, to je i veći rast. Zahtevi za softverskim sistemima obično rastu između 1 procenta i tri procenta po kalendarskom mesecu. Razvoj nekih zahteva je legitiman, neizbežan, pa čak i povoljan. Međutim, stalne male promene okvira projekta, kada projekat kontinuirano uključuje više funkcionalnosti bez prilagođavanja resursa, rasporeda ili ciljeva kvaliteta je štetan. **Problem nije u tome što se zahtevi menjaju, već što kasne promene mogu imati veliki uticaj na već obavljeni posao.** Ako se svaka predložena promena odobri, akterima se može činiti da softver nikada neće biti isporučen - i zaista, možda i neće.

Prvi korak u upravljanju okvirom projekta je dokumentovanje poslovnih ciljeva, vizije proizvoda, obima projekta i ograničenja novog sistema, kao što je opisano u lekciji 3, „Postavljanje poslovnih zahteva.“ **Neophodno je proceniti svaki predloženi zahtev ili karakteristiku prema poslovnim zahtevima.** Angažovanje kupaca u izazivanju zahteva smanjuje broj zahteva koji se previde. Primena prototipova pomaže u kontroli puzanja okvira projekta pomažući programerima i korisnicima da dele jasno razumevanje potreba korisnika i mogućih rešenja. Korišćenje kratkih razvojnih ciklusa pri inkrementalnom razvoju novih izdanja softvera, omogućava česta doterivanja softvera sa novim zahtevima.

Najefikasnija tehnika za kontrolu puzanja okvira projekta je sposobnost da se kaže „ne“. Ljudi ne vole da kažu „ne“, a razvojni timovi mogu biti pod jakim pritiskom da uvek kažu „da“. Filozofije poput „kupac je uvek u pravu“ ili „postići ćemo ukupno zadovoljstvo kupaca“ su dobre na kratko, ali platiš cenu za njih. Ignorisanje cene ne menja činjenicu da promena nije besplatna. Predsednik jednog proizvođača softverskog alata navikao je da čuje kako menadžer razvoja kaže „ne sada“ kada predloži novu funkciju. „Ne sada“ je ukusnija od jednostavnog odbacivanja. Ona drži obećanje da će funkciju uključiti u narednu verziju.

PROMENA POLITIKE KONTROLE PROMENE ZAHTEVA

Politike imaju smisla samo ako su realne, dodaju vrednost i ako se primenjuju.

Rukovodstvo bi trebalo da komunicira o politici koja utvrđuje opredeljenja po kojima koja će projektni timovi postupati sa predloženim promenama u zahtevima i svim drugim značajnim artefaktima projekata. Politike imaju smisla samo ako su realne, dodaju vrednost i primenjuju se. Sledeća *opredeljenja o politici kontrole promena* mogu biti od pomoći:

- Sve promene moraju da prate proces. Ako zahtev za promenu nije podnesen u skladu sa ovim postupkom, neće se razmatrati.
- Na neodobrenim izmenama neće se izvoditi nikakvi drugi projektantski ili implementacioni projekti osim istraživanja izvodljivosti.
- Jednostavno traženje promene ne garantuje da će ona biti izvršena. Nadzorni odbor za promene projekta (CCB) odlučiće koje promene će primeniti.
- Sadržaj baze podataka o promenama mora biti vidljiv svim učesnicima u projektu.
- Za svaku promenu mora se izvršiti analiza uticaja.
- Svaka ugrađena promena mora se pratiti do odobrenog zahteva za promenom.
- Obrazloženje iza odobrenja ili odbijanja zahteva za promenom mora se zabeležiti.

Naravno, sitne promene teško će uticati na projekat, a velike promene će imati značajan uticaj. U praksi ćete možda odlučiti da određene odluke prepustite odlučivanju programerima, ali **nikakva promena koja utiče na rad više pojedinaca ne bi trebalo da zaobiđe vaš postupak**. Uključite „brzi put“ da biste ubrzali zahteve za promenom niskog rizika i ulaganja u komprimirani ciklus odlučivanja.

OSNOVNI KONCEPTI PROCESA KONTROLE PROMENA ZAHTEVA

Proces kontrole promena vam omogućava da pratite status svih predloženih promena i pomaže da se osigura da predložene promene ne budu izgubljene

Razuman postupak kontrole promena omogućava vođama projekta da donose informisane poslovne odluke koje će pružiti najveću vrednost kupca i poslovne vrednosti, istovremeno kontrolišući troškove životnog ciklusa proizvoda i raspored projekta. Proces vam omogućava da pratite status svih predloženih promena i pomaže da se osigura da predložene promene ne budu izgubljene ili zanemarene. Nakon što odredite skup zahteva, trebalo bi da sledite ovaj postupak za sve predložene izmene u toj osnovnoj liniji.

Akteri projekta se ponekad pozivaju da prate novi postupak, ali postupak kontrole promena nije prepreka unošenju potrebnih izmena. To je mehanizam toka i filtriranja koji se osigurava da projekat ubrzano uključuje najprikladnije promene. Ako predložena promena nije dovoljno bitna da bi zainteresovanim stranama trebalo samo nekoliko minuta da je pošalju putem

standardnog, jednostavnog kanala, tada nije vredno razmatranja za uključivanje. **Proces vaše promene treba da bude dobro dokumentovan, što je jednostavnije i pre svega efikasno.**

Upravljanje promenama zahteva slično je procesu prikupljanja i donošenja odluka o izveštajima o kvarovima. Isti alati mogu podržati obe aktivnosti. Imajte na umu, međutim: alat nije zamena za dokumentovani proces, a nijedan nije zamena za odgovarajuće diskusije. **Razmotrite i alat i pisani postupak kao načine podržavanja ovih kritičnih razgovora.**

Kada je potrebno da ugradite promenu, počnite na najvišem nivou apstrakcije koji promena dotiče i kaskadira promene kroz komponente sistema. Na primer, predložena promena može uticati na potrebe korisnika, ali ne i na poslovne zahteve. Izmena sistemskih zahteva na visokom nivou mogla bi da utiče na brojne softverske i hardverske zahteve u više podsistema. Neke promene odnose se samo na interni sistem sistema, kao što je način na koji se implementira komunikaciona usluga. To nisu promene zahteva koje su vidljive za korisnike, već promene projektnih rešenja ili koda.

OPIS PROCESA KONTROLE PROMENA ZAHTEVA

Izveštaj o procesu promene zahteva treba da sadrži kriterijume za unos, definisane zadatke za realizaciju promene, način potvrde realizovane promen

Slika 1 ilustruje obrazac za opis procesa kontrole promene za rukovanje izmenama zahteva. .Ako je ovaj obrazac previše detaljan za vaše okruženje, smanjite ga za više neformalnih projekata. Bilo bi korisno da u sve opise procesa uključimo sledeće četiri komponente:

- Kriterijumi za unos, uslovi koji moraju biti ispunjeni pre nego što započne izvršenje procesa
- Različiti zadaci uključeni u proces, projektna uloga odgovorna za svaki zadatak i drugi učesnici u zadatku
- Koraci za potvrdu da su zadaci pravilno izvedeni
- Izlazni kriterijumi, uslovi koji ukazuju na to da se proces uspešno dovršava

1. Svrha i okvir
 2. Uloge i odgovornosti
 3. Stanja zahteva za promenama
 4. Ulazni kriterijumi
 5. Zadaci
 1. Ocenjivanje zahteva za promenom
 2. Donošenje odluke o promeni
 3. Primena promene
 4. Overavanje promene
 6. Izlazni kriterijumi
 7. Izveštavanje o promeni kontrole statusa
- Dodatak: Atributi smešteni za svaki zahtev

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 5.1 Obrazac za sadržaj izveštaja o procesu promene zahteva

UPUTSVO ZA PRIPREMU IZVRŠTAJA O PROCESU UPRAVLJANJE PROMENAMA-1

Uputstvo za poglavlja: Namena i delokrug, Uloge i odgovornosti i Promena statusa zahteva

1. Namena i delokrug

Opišite svrhu ovog procesa i organizacioni opseg na koji se odnosi. Navedite da li su izuzete neke posebne vrste promena, kao što su promene privremenih radnih proizvoda. Definišite sve uslove koji su neophodni za razumevanje ostatka dokumenta.

2. Uloge i odgovornosti

Navedite uloge projektnog tima koji učestvuju u aktivnostima kontrole promena i opišite svoje odgovornosti. Tabela na slici 2 predlaže neke važne uloge; prilagodite ih svakoj projektnoj situaciji. Za svaku ulogu nisu potrebni različiti pojedinci. Ista osoba može ispuniti nekoliko - možda svih - uloga na malom projektu.

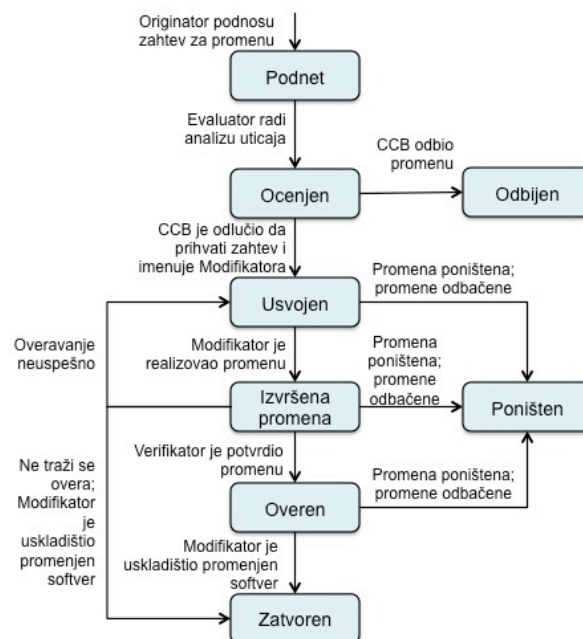
Uloga	Opis i odgovornosti
CCB predsednik	Predsednik odbora za kontrolu promena (Change Contrl Board – CCB); ima odgovornost donošenja konačne odluke ako u CCBu nema saglasnosti; utvrđuje Evaluatora i Modifikatora za svaki zahtev za promenu.
CCB	Grupa odlučuje i daje saglasnost ili odbija predloženu promenu u projektu.
Evaluator	Osoba koja analizira uticaj predložene promene.
Modifikator	Osoba koja je odgovorna za realizaciju promena u softveru kao odgovor na prihvaćeni zahtev.
Originator	Osoba koja podnosi novi zahtev za promenom zahteva.
Primač zahteva	Osoba koja prima na početku nove zahteve za promenama zahteva.
Verifikator	Osoba koja određuje da li je promena tačno realizovana.

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 5.2 Opis uloga u procesu upravljanja promenama zahteva

3. Promenite status zahteva

Zahtev za promenu prolazi kroz definisan životni ciklus stanja. Ova stanja možete predstaviti koristeći dijagram tranzicije stanja kao što je prikazano na slici 3. Ažurirajte status zahteva samo kada su ispunjeni određeni prelazni kriterijumi.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 5.3 Dijagram stanja zahteva za promenu zahteva

UPUTSTVO ZA PRIPREMU IZVRŠTAJA O PROCESU UPRAVLJANJE PROMENAMA-2

Opisani su kriterijumi za prihvatanje unosa i dati su opisi vrste zadataka

4. Kriterijumi za ulazak

Osnovni kriterij za unos za vaš postupak kontrole promene je da je zahtev za promenu sa svim potrebnim informacijama primljen preko odobrenog kanala. Svi potencijalni inicijatori trebali bi znati kako podneti zahtev za promenom. Alat za promenu treba da dodeli jedinstveni identifikator svakom zahtevu i usmerava sve promene na Primalac zahteva.

5. Zadaci

Ovaj odeljak postupka opisuje zadatke koji se izvršavaju za obradu jednog zahteva za promenom.

5.1 Ocenite zahtev za promenom

Započnite ocenom zahteva za tehničkom izvodljivošću, troškovima i usklađivanjem sa poslovnim zahtevima i ograničenjima resursa projekta. Predsedavajući CCB-a može dodeliti evaluatora za obavljanje analize uticaja, analize rizika ili drugih procena. (Pogledajte odeljak „Analiza uticaja promena“ kasnije u ovom poglavlju.) Ovo osigurava da se razumeju posledice prihvatanja promene. Procenjivač i CCB takođe bi trebali razmotriti poslovne i tehničke implikacije odbacivanja zahteva, ako postoje.

5.2 Donesite odluku o promeni

Odgovarajući donosioci odluka, koji su se povezivali kao CCB, tada odlučuju da li će odobriti ili odbiti promenu. CCB daje svakoj odobroj promeni prioritetni ili ciljni datum implementacije, ili ona raspoređuje određenu iteraciju ili izdanje. To može jednostavno dodati novi zahtev zaostatku proizvoda na čekanju rada. CCB ažurira status zahteva i obaveštava sve pogođene članove tima.

5.3 Sprovedite promenu

Dodeljeni modifikator (ili modifikatori) ažurira pogođene radne proizvode po potrebi kako bi u potpunosti implementirao promenu. Informacije o tragovima koristite za pronalaženje svih delova sistema koji se promene dotiču i revidirajte podatke o tragovima, ako je potrebno da biste odrazili promene.

5.4 Proverite promenu

Promene zahteva obično se verifikuju putem stručnog pregleda kako bi se osiguralo da modifikovani rezultati ispravno adresiraju sve aspekte promene. Više članova tima može da proveriti promene koje su učinjene u različitim proizvodima za dalju upotrebu putem testiranja ili pregleda. Nakon završetka verifikacije, Modifikator skladišti ažurirane radne proizvode na odgovarajuće lokacije u skladu sa projektnim dokumentima i konvencijama o upravljanju kodom.

UPUTSTVO ZA PRIPREMU IZVRŠTAJA O PROCESU UPRAVLJANJE PROMENAMA-3

Uputstvo za poglavlja: Kriterijumi za izlazak, Promena izveštavanja o statusu kontrole i Dodatak

6. Kriterijumi za izlazak

Zadovoljavanje sledećih kriterijuma za izlazak pokazuje da je izvršenje vašeg postupka kontrole promene pravilno završeno:

- Status zahteva je odbijen, zatvoren ili otkazan.

- Svi modifikovani radni proizvodi se ažuriraju i čuvaju na ispravnim lokacijama.
- Za detalje o promeni i statusu zahteva za promenom obavešteni su relevantni akteri

7. Izveštavanje o statusu kontrole

Identifikujte grafikone i izveštaje koje ćete koristiti da sažete sadržaj baze promena. Ovi grafikoni mogu prikazati broj zahteva za promenu u svakoj državi kao funkciju vremena ili trendove u prosečnom vremenu kada je zahtev za promenu nerešen. Opišite postupke za izradu grafikona i izveštaja. Rukovodilac projekata koristi ove izveštaje prilikom praćenja statusa projekta.

ATRIBUTI ZAHTEVA ZA PROMENU

Svaki zahtev za promenu treba da ima određene atribute koji ga bliže određuju

Na slici 4 predloženi su atributi koje se mogu da se pamte sa svakim zahtevom za promenu. Neke atribute određuje Originator, a neke CCB. U postupku kontrole promene navedite koji su atributi obavezni, a koji neobavezni. Ne definišite više atributa nego što vam je stvarno potrebno. Vaš alat za promenu treba neke od njih da obrađuje (ID, datum predavanja, datum ažuriran) automatski.

Stavka	Opis
Izvor promene	Funikcionalna oblast koja zahteva promenu (marketing, menadžment, kupac, razvoj, testiranje)
ID zahteva za promenom	Jedinstveni identifikator dat zahtevu
Tip promene	Tip zahteva za promenu, npr. promena zahteva, predloženo poboljšanje, izveštaj od grešci.
Datum podnošenja	Datum kada je Originator podne zahtev za promenu
Datum promene	Datum poslednje podnesenog zahteva za promenu
Opis	Tekstualni opis zahtevane promene u slobodnom formatu
Prioritet primene	Relativni prioritet zahteva po mišljenju CCB: nizak, srednji i visoki
Modifikator	Osoba koja je zadužena za implementaciju zahteva
Originator	Osoba koja je podnela zahtev za promenu
Prioritet originatora	Relativni prioritet zahteva za promenu po mišljenu originatora: niski, srednji i visoki.
Planirano izdanje	Planirano izdanje ili iteracija za koju je se planira primena odobrenog zahteva.
Projekat	Naziv projekta u kome se zahteva promena
Odgovor	Odgovor zahtev u vidu slobodnog teksta. Bez brisanja prethodnog, mogući su i novi odgovori.
Status	Trenutni status zahteva za promenu, prema opcijama datim na slici 3
Naslov	Sažetni opis zahtevane promene u jednom redu
Verifikator	Osoba koja treba da potvrdi da je zahtev za promenu urađen tačno, tj. kako je propisano.

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 5.4 Predlog mogućih atributa zahteva za promenu softverskih zahteva

ODBOR ZA KONTROLU PROMENA

Odbor za kontrolu promena je telo ljudi - bilo da je reč o jednoj osobi ili raznovrsnoj grupi - koja odlučuje koje predložene promene i nove zahteve treba prihvatiti.

Odbor za kontrolu promena (Change Control Board - CCB) je telo ljudi - bilo da je reč o jednoj osobi ili raznovrsnoj grupi - koja odlučuje koje predložene promene i nove zahteve treba da prihvati, koje da prihvati revizijama, a koje da odbije. CCB takođe odlučuje koje prijavljene nedostatke treba ispraviti i kada ih ispraviti. Neke CCB su ovlaštene da donose odluke, dok druge mogu davati samo preporuke za odluku menadžmenta. Projekti uvek imaju određenu de facto grupu koja donosi odluke o promenama. Uspostavljanje CCB-a formalizuje sastav i ovlašćenja ove grupe i definiše njene operativne procedure.

Nekim ljudima izraz „Odbor za kontrolu promena“ predstavlja sliku rasipnih birokratskih pogrešaka. Umesto toga, mislite na CCB kao na dragocenu strukturu koja pomaže upravljanju čak i malim projektom. Na malom projektu ima smisla da samo jedan ili dvoje ljudi donose odluke o promenama. Veoma veliki projekti ili programi mogu imati nekoliko nivoa CCB-a, od kojih su neki odgovorni za poslovne odluke, kao što su promene zahteva, a neki za tehničke promene. Veliki program koji obuhvata više projekata uspostavio bi CCB na nivou programa i pojedinačnu CCB za svaki projekat. Svaki CCB projekta rešava pitanja i promene koje se tiču samo tog projekta. Pitanja koja utiču na više projekata i promene koje prelaze određeni trošak ili uticaj rasporeda eskaliraju se u CC na nivou programa.

Sastav CCB

Članstvo u CCB trebalo bi da predstavlja sve grupe koje treba da učestvuju u donošenju odluka iz okvira nadležnosti te CCB. Razmislite o izboru predstavnika iz sledećih oblasti:

- Upravljanje projektima ili programima
- Analiza poslovanja ili upravljanje proizvodima
- Razvoj
- Testiranje ili osiguranje kvaliteta
- Marketing, posao za koji se gradi aplikacija ili predstavnici kupca
- Tehnička podrška ili služba za pomoć

Donošenje odluka

Svaki CCB mora da definiše svoj postupak donošenja odluka koji bi trebao naznačiti:

- Broj članova CCB-a ili ključne uloge koje predstavljaju kvorum za donošenje odluka.
- Pravila odlučivanja koja se koriste (pogledajte poglavlje 2 za više informacija o pravilima odlučivanja).
- Može li predsedavajući CCB poništiti kolektivnu odluku CCB-a.
- Da li menadžment mora da ratifikuje odluke CCB

ALATI ZA KONTROLU PROMENA

Alati za kontrolu promena podržavaju prikupljanje, skladištenje i upravljanje promenama zahteva.

Mnogi timovi koriste komercijalne alate za praćenje problema za prikupljanje, skladištenje i upravljanje promenama zahteva. Izveštaj o nedavno podnesenim zahtevima za izmene izdvojenim iz alata može poslužiti kao dnevni red sastanka CCB-a. Alati za praćenje izdataka mogu da izveštavaju o broju zahteva u svakom trenutku. Budući da se raspoloživi alati, njihovi

dobavljači i njihove karakteristike često menjaju, ovde ne dajemo posebne preporuke alata. Da biste podržali svoj proces promene, potražite alat koji:

- Omogućuje vam da definišete atribut koji čine zahtev za promenom.
- Omogućuje vam implementaciju životnog ciklusa zahteva za promenom sa više statusa zahteva za promenu.
- Utvrđuje model tranzicije stanja tako da samo ovlašćeni korisnici mogu da izvrše određene promene statusa.
- Zapisuje datum svake promene statusa i identitet osobe koja ga je napravila.
- Omogućava prilagodljivo, automatsko obaveštavanje putem e-pošte kada Originator podnese novi zahtev ili kada se status zahteva ažurira.
- Izrađuje i standardne i prilagođene izveštaje i grafikone.

Neki alati za upravljanje komercijalnim zahtevima imaju ugrađen sistem zahteva za promene. Ovi sistemi mogu da povežu predloženu promenu sa određenim zahtevom tako da se pojedinac koji je odgovoran za svaki zahtev obavesti e-poštom svaki put kada neko podnese odgovarajući zahtev za promenu.

Merenje aktivnosti promena

Merenje aktivnosti promena je način da se proceni stabilnost zahteva. Takođe otkriva mogućnosti za poboljšanje procesa koje mogu dovesti do manjih promena u budućnosti. Razmislite o praćenju sledećih aspekata aktivnosti na kojima se zahtevi menjaju:

- Ukupan broj primljenih zahteva za promenu, trenutno otvorenih i zatvorenih
- Kumulativni broj dodatih, izbrisanih i modifikovanih zahteva
- Broj zahteva koji su nastali od svake promene porekla
- Broj primljenih promena u odnosu na svaki zahtev od kada je osnovni
- Ukupni napor posvećen obradi i primeni zahteva za promenu

ANALIZA UTICAJA PROMENA

Analiza uticaja promena istražuje zahtev da se identifikuju komponente koje bi mogle da budu kreirane, modifikovane ili odbačene i da se proceni napor potreban za sprovođenje promene.

Potreba za analizom uticaja očigledna je za velika poboljšanja. Međutim, neočekivane komplikacije mogu vrebat i ispod površine čak i malih zahteva za promenom. Kompanija je jednom morala da promeni tekst jedne poruke o grešci u svom proizvodu. Šta može biti jednostavnije? Proizvod je bio dostupan u engleskoj i njemačkoj verziji. Nije bilo problema na engleskom jeziku, ali na nemačkom je nova poruka premašila maksimalnu dužinu znakova dodeljenu za prikazivanje poruka o grešci i u okviru sa porukama i u bazi podataka. Suočavanje s ovim na izgled jednostavnim zahtevom za promenom pokazalo se da je puno više posla nego što je programer predvideo kada je obećao brzi preokret.

Analiza uticaja je ključni aspekt odgovornog upravljanja zahtevima. Omogućava tačno razumevanje implikacija predložene promene, što pomaže timu da donese informisane poslovne odluke o tome koje predloge treba da odobri.

Analiza istražuje zahtev da se identifikuju komponente koje bi mogle da budu kreirane, modifikovane ili odbačene i da se proceni napor potreban za sprovođenje promene. Pre nego što programer kaže, „Naravno, nema problema“ u odgovoru na zahtev za promenu, trebalo bi da provede malo vremena na analizu uticaja.

Postupak analize uticaja

Predsedavajući CCB će zatražiti jednog ili više tehničkih ljudi (poslovni analitičari, programeri i / ili tester) da izvrše analizu uticaja za određeni predlog promene. Analiza uticaja uključuje tri koraka:

1. *Shvatite moguće implikacije promene.* Promena zahteva često proizvodi veliki efekat talasa, što dovodi do modifikacija u drugim zahtevima, arhitekturama, projektnim rešenjima, kodu i testovima. Promene mogu dovesti do sukoba s drugim zahtevima ili mogu ugroziti atribut kvaliteta, poput performansi ili sigurnosti.
2. *Identifikujte sve zahteve, datoteke, modele i dokumente* koji bi mogli biti modifikovani ako tim ugradi traženu promenu.
3. *Identificirajte zadatke potrebne za izvršenje promene* i procenite napor potreban za završetak tih zadataka.

VIDEO 30 - REQUIREMENTS MANAGEMENT GOOD PRACTICES - WIEGERS (VIDEO)

Trajanje: 4:34 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

VIDEO 32 - CHANGE CONTROL -WIEGERS (VIDEO)

Trajanje: 7:13 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

VIDEO 33 - VERSION CONTROL - WIEGERS (VIDEO)

Trajanje: 7:18 minuta

Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

▼ Poglavlje 6

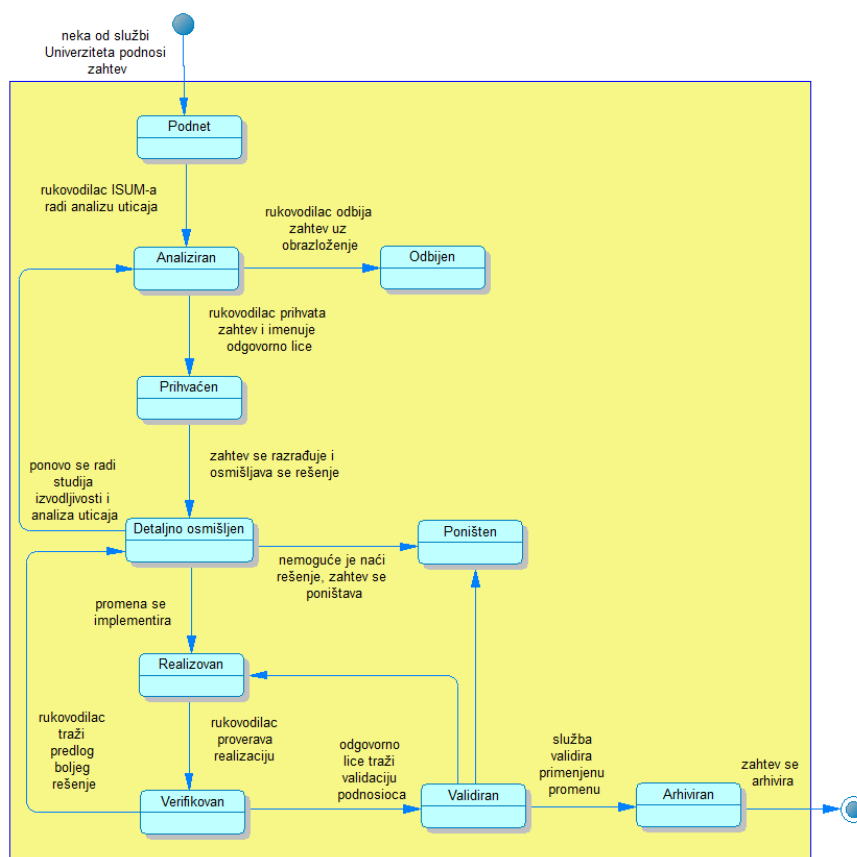
Vežba

UPRAVLJANJE ZAHTEVIMA ZA PROMENU

Upravljanje zahtevima za promenu u razvojnom timu Informacionog sistema Univerziteta Metropolitan

Upravljanje zahtevima za promenu se svakodnevno sprovodi u razvojnom timu Informacionog sistema Univerziteta Metropolitan. ISUM je sistem koji je u konstantnom razvoju i često prolazi kroz faze unapređenja ili modifikacija zbog promene poslovnih pravila. **Da bi se predstavio proces upravljanja zahtevima koji sprovodi ISUM tim, kreiran je dijagram stanja zahteva za promenom podnetih od strane drugih službi Univerziteta.** Primetićete sličnosti u vezi sa dijagramom datim na slici 3 u poglavlju 5.

Prepoznajte u okviru ISUM tima uloge projektnog tima koje su predstavljene u tekstu predavanja, a koje učestvuju u aktivnostima kontrole promena.

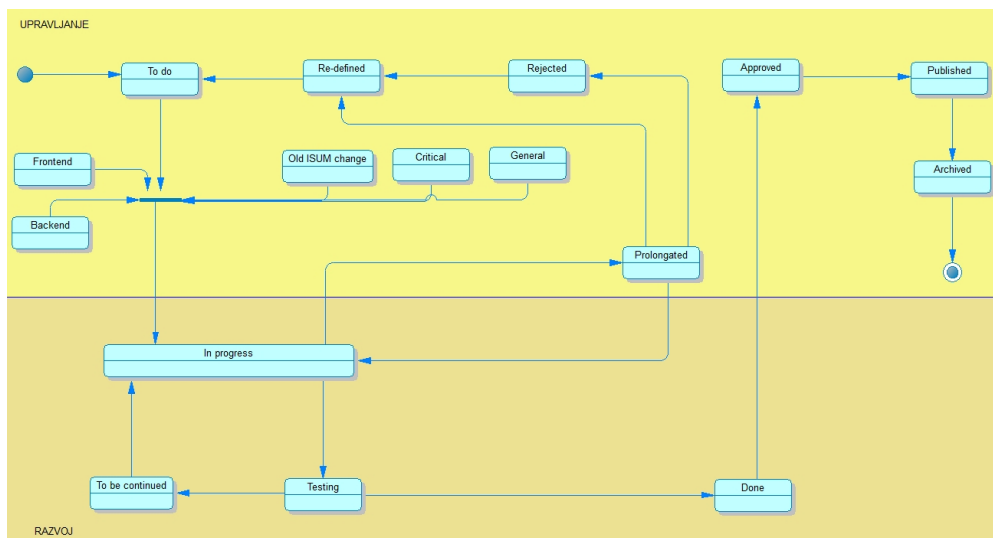


Slika 6.1 Upravljanje zahtevima za promenu na primeru ISUM tima [Izvor: Marina Damjanović]

UPRAVLJANJE ZADACIMA

Upravljanje radnim zadacima u razvojnom timu Informacionog sistema Univerziteta Metropolitan

Pored upravljanja zahtevima, ISUM tim sprovodi **poseban postupak upravljanja zadacima**. Nakon što ISUM tim dobije zahtev za izmenu sistema, zahtev se kao zadatak delegira nekom članu tima. Zadatak se unosi u alat za upravljanje zadacima sa inicijalnim statusom, nakon čega kreće njegovo rešavanje. Pravilnikom je unapred definisano koja stanja može da ima zadatak i koja stanja sme da mu dodeljuje određeni član tima. Članovi tima su podeljeni prema ulogama; član tima može imati upravljačku ili razvojnu ulogu. Na sledećem dijagramu su prikazana moguća stanja zadatka i ko ima pravo postavljanja zadatka u određeno stanje.



Slika 6.2 Upravljanje radnim zadacima na primeru ISUM tima [Izvor: Marina Damjanović]

Šta mislite o predloženom modelu praćenja zadataka? Ocenite uspešnost primene ovakvog modela. Dopišite događaje za prelaskе stanja onako kako ih pretpostavljate.

DOKUMENT ZA UPRAVLJANJE ZAHTEVIMA ZA IZMENOM

Uzorak Dokumenta za upravljanje zahtevima za izmenom u razvojnom timu Innovative software technologies

Innovative software technologies je firma koja povremeno razvija interne softverske proizvode za Univerzitet Metropolitan. Mladi softverski inženjer je kreirao uzorak dokumenta za upravljanje zahtevima za izmenom (engl. *Requirements change control management*). Firma ima u planu da krene sa razvojem eksternih projekata. Proces rada sa službama Univerziteta je već ustaljen, ali razvojni tim je rešio da utvrdi poslovne procese i projektnu dokumentaciju koja će biti potrebna za rad sa drugim klijentima. U nastavku je dat sadržaj dokumenta za upravljanje zahtevima za izmenom.

Dajte svoju ocenu i predloge kako se može unaprediti ovaj uzorak za što efikasniji rad sa novim klijentima.

Sadržaj

1. Uvod	3
1.1 Svrha	3
1.2 Obim	3
1.3 Definicije, akronimi i pretpostavke	3
1.4 Reference	3
2. Zahtev za izmenu	3
2.1. <Id zahteva>-<Naziv zahteva>	3
2.1.1. Opis podnetog zahteva za izmenu	3
2.1.2. Role identifikovane u procesu izmene	4
2.1.3. Proces izmene	4

Slika 6.3 Sadržaj uzorka Dokumenta za upravljanje zahtevima za izmenom [Izvor: Marina Damjanović]

DOKUMENT ZA UPRAVLJANJE ZAHTEVIMA ZA IZMENOM - NASTAVAK

Uzorak dokumenta za upravljanje zahtevima za izmenom u razvojnom timu Innovative software technologies - nastavak

2. Zahtev za izmenu

Ovaj odeljak treba da sadrži zahteve za izmenom koji su primljeni i procesirani od strane tima na navedenom preioizvodu.

2.1. <Id zahteva>-<Naziv zahteva>

2.1.1. Opis podnetog zahteva za izmenu

Zvanje
Potreba/namena
Format
Dodatna objašnjenja I primeri

Formalno specificiran novi slučaj korišćenja (zasnovan na zahtevima u SRS-u)

Slika 6.4 Opis zahteva za izmenu [Izvor: Marina Damjanović]

2.1.2. Role identifikovane u procesu izmene

Direktor upravljanja izmenama – ko odobrava izmenu

Inicijator izmene – ko traži izmenu

Kreator izmene – ko dizajnira izmenu

Tester izmene – ko testira izmenu

Implementator izmene – ko implementira izmenu

Revizor izmene – ko proverava način sprovođenja izmene

Slika 6.5 Role identifikovane u procesu izmene [Izvor: Marina Damnjanović]

2.1.3. Proces izmene

Datum podnošenja – kada je stigao zahtev za izmenu

Prioritet – 1 (ekstremno hitno), 2 (hitno), 3 (važno), 4 (može da čeka), 5 (buduće ažuriranje)

Planiran datum isporuke – rok za isporuku

Rezervni plan – šta je plan u slučaju da nešto krene loše tokom implementacije izmene

Mera uspešnosti – kako će se meriti uspešnost ostvarene promene

Datum arhiviranja zahteva za izmenu – kada je zahtev konačno ispunjen

Slika 6.6 Proces izmene [Izvor: Marina Damnjanović]

ZADACI ZA VEŽBU

Tekst zadatka za vežbu

ZADATAK 1.

U tekstu predavanja su izneti predlozi za uobičajene statuse zahteva na tradicionalnim i na agilnim projektima. Nacrtajte dijagram prelaska stanja za oba pristupa, da biste prikazali uslove ili događaje koji pokreću promenu zahteva iz jednog statusa u drugi. (15 min)

ZADATAK 2.

Razmislite o tome šta biste voleli da promenite na E-Student modulu ili bilo kojoj aplikaciji koju često koristite i koju dobro poznajete. Stavite se u ulogu originatora i osmislite zahtev za promenu na toj aplikaciji koji biste uputili bordu za kontrolu promena. Iskoristite tabelu datu u lekciji koja identifikuje attribute koje treba da ima zahtev za promenu. Odaberite attribute koji su obavezni, a zaobiđite koji nisu. Pripremite obrazloženje. (15 min)

▼ Poglavlje 7

Domaći zadatak

DOMAĆI ZADATAK 14

Tekst domaćeg zadatka

Za sistem koji ste dobili da analizirate od DZ01, osmislite jedan zahtev za promenu koju bi trebalo da realizujete. Korišćenjem tabele date u lekciji, opišite attribute tog zahteva za promenu koji treba bliže da ga odrede. Ne morate se striktno držati predloženog templejta, odaberite shodno svom zahtevu koji su atributi obavezni, a koji nepotrebni.

Napomene:

Zadatak se rešava opisno i šalje kao .docx fajl.

Rešenje zadatka pošaljite na mejl adresu predmetnog asistenta. Rok za izradu je definisan Plan i programom predmeta.

▼ Poglavlje 8

Projektni zadatak

ZADATAK ZA RAD NA PROJEKTU

Tekst zadatka za rad na projektu

Ukoliko vam je asistent vratio projektni zadatak na doradu ili ispravak, primenite ono što je traženo i pošaljite unapređenu verziju dokumenta.

▼ Poglavlje 9

Zaključak

ZAKLJUČAK

1. Upravljanje zahtevima obuhvata sve aktivnosti koje održavaju celovitost, tačnost i valutu sporazuma u toku projekta.
2. BA postavlja mehanizme skladištenja zahteva, definisati attribute zahteva, koordinisati status zahteva i ažurirati podatke u tragovima, a po potrebi nadgledati i promeniti aktivnost.
3. Kada bazni dokument dokumentujete, prvo arhivirajte označenu verziju, zatim prihvatite sve revizije, a zatim spremite čistu verziju kao novu osnovnu liniju, spremnu za sledeći krug izmena.
4. Svaki zahtev se može predstaviti kao objekt sa atributima koji uspostavljaju kontekst i pozadinu zahteva. Preko atributa, alati mogu lako da pretražuju zahteve i da ih selektivno izdvajaju.
5. Praćenje statusa znači upoređivanje mesta gde se zaista nalazite u projektu razvoja u određenom trenutku u odnosu na očekivanja
6. Broj zahteva se povećava kako se opseg izdanja dodaje i smanjuje kada se funkcionalnost ukloni sa osnovne linije.
7. Skoro svi nedostaci prijavljeni rano u projektu povezani su sa pitanjima u zahtevima, kao što su traženje pojašnjenja o zahtevu, odluke o dometu, pitanja o izvodljivosti razvoja i obavezama
8. Vreme provedeno na aktivnosti povezane sa zahtevima je ulaganje u uspeh projekta, a ne trošak.
9. Agilni projekti se prilagođavaju promenama tako što će proizvoditi proizvod kroz seriju razvojnih iteracija i upravljati dinamičkim zaostatkom proizvoda (backlog) koji ostaje da se uradi.
10. Upravljanje zahtevima pomaže da se osigura da se rad koji ulažete u razvoj zahteva ne prosipa.
11. Nekontrolisana promena zahteva je uobičajen izvor haosa projekta, promene u rasporedu, problema sa kvalitetom i teških osećaja. Promene su neophodne, ali na kontrolisan način. Problem nije u tome što se zahtevi menjaju, već što kasne promene mogu imati veliki uticaj na već obavljeni posao.
12. Proces kontrole promena vam omogućava da pratite status svih predloženih promena i pomaže da se osigura da predložene promene ne budu izgubljene ili zanemarene
13. Izveštaj o procesu promene zahteva treba da sadrži kriterijume za unos, definisane zadatke za realizaciju promene, način potvrde realizovane promene i izlazne kriterijume
14. Odbor za kontrolu promena je telo ljudi - bilo da je reč o jednoj osobi ili raznovrsnoj grupi - koja odlučuje koje predložene promene i nove zahteve treba prihvatiti, a koje odbiti.

REFERENCE

Nastavi materijal pripremljen za studente se pravi s namerom da im omogući brži i skraćeni uvid u program lekcije, a na bazi jedne ili više referentnih udžbenika i drugih izvora . Nastavni materijal nije zamena za ove udžbenike, koje treba koristiti ako student želi da se detaljnije upozna sa nastavnim materijom. Očekuje se od studenta da poseduje bar jedan od navedenih udžbenika u Planu i programu predmeta.

Ova lekcija je urađena na bazi dela teksta datom **u poglavljima 27 i 28** knjige: **Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013**. Za detaljnije proučavanje i primere, studentima se preporučuje da pročitaju ovo poglavlje. Manji uticaj na sadržaj lekcije imaju ostale reference navedene u Planu i programu predmeta,