



SE322 - INŽENJERSTVO ZAHTEVA

Praćenje zahteva i alati za praćenje zahteva

Lekcija 15

PRIRUČNIK ZA STUDENTE

SE322 - INŽENJERSTVO ZAHTEVA

Lekcija 15

PRAĆENJE ZAHTEVA I ALATI ZA PRAĆENJE ZAHTEVA

- ✓ Praćenje zahteva i alati za praćenje zahteva
- ✓ Poglavlje 1: Sledljivost zahteva
- ✓ Poglavlje 2: Matrica sledljivosti zahteva
- ✓ Poglavlje 3: Alati za praćenje zahteva
- ✓ Poglavlje 4: Procedura za praćenje zahteva
- ✓ Poglavlje 5: Pобоljšanje procesa zahteva
- ✓ 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

Izmene softvera koje se čine jednostavnim često imaju dalekosežne posledice, pa je potrebno modifikovanje mnogih delova sistema. Teško je pronaći sve elemente sistema na koje može uticati izmenjeni zahtev. U lekciji 13 naveli smo važnost izvođenja analize uticaja kako bi se uverilo da tim zna šta treba da pređe pre nego što se obaveže na sprovođenje predložene promene. Analiza uticaja promena je lakša ako imate mapu puta koja pokazuje gde su u softveru implementiran svaki zahtev ili poslovno pravilo.

U ovoj lekciji razmatraćemo praćenje, tj. traženja zahteva (*requirements tracing*) ili sledljivost zahteva. Informacija o sledljivosti zahteva dokumentuje zavisnost i logičke veze između pojedinih zahteva i ostalih elemenata sistema. Ovi elementi uključuju druge zahteve različitih vrsta, poslovna pravila, arhitekturu i ostale komponente projektnog rešenja, module izvornog koda, testove i datoteke pomoći. Informacije o sledljivosti olakšavaju analizu uticaja pomažući vam da identifikujete sve radne proizvode koje ćete možda morati da izmenite da biste implementirali predloženu promenu zahteva.

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

▼ Poglavlje 1

Sledljivost zahteva

VIDEO PREDAVANJE ZA OBJEKAT "SLEDLJIVOST ZAHTEVA"

Trajanje video snimka: 18min 35sek

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

ŠTA JE SLEDLJIVOST ZAHTEVA?

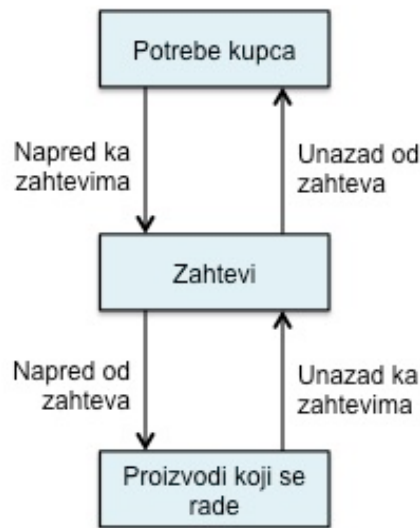
Sledljivost zahteva omogućava utvrđivanje veza sa potrebama korisnika i sa funkcijama proizvoda.

Ulaženje u trag veza zahteva omogućavaju vam da pratite život nekog zahteva , prema napred i nazad, od nastanka do implementacije. Sledljivost (*treacability*) je jedna od karakteristika odličnih zahteva. Imajte na umu da sledljivost - koja ima svojstva za olakšavanje traganje - nije isto što i biti praćen, tj. imati zapisanu logičku vezu između zahteva i ostalih elemenata. **Da bi zahtevi bili sledljivi, svaki od njih mora biti jedinstveno i trajno obeležen takoda možete dase na njega nedvosmisleno pozivate tokom celog projekta.** Zapisujte zahteve tako da budu kratki i jednostavni ("sitnozrnati" , "*fine-grained*"), umesto da stvarate velike paragrafe koji sadrže mnogo pojedinačnih funkcionalnih zahteva koje čitaoci moraju da razotkriju.

Slika1 prikazuje četiri vrste veza za traženje zahteva. Potrebe korisnika prate se prema zahtevima, tako da možete reći na koje će zahteve uticati promena potreba tokom ili nakon razvoja. Potrebe kupaca mogu se artikulirati u obliku poslovnih ciljeva, zahteva tržišta i / ili zahteva korisnika. Kompletan set tragova unapred vam takođe daje sigurnost da su postavljeni zahtevi odgovorili na sve navedene potrebe kupca. Suprotno tome, možete pratiti zahteve prema potrebama kupaca da biste identifikovali poreklo svakog softverskog zahteva. Ako odlučite da predstavljate potrebe korisnika u obliku slučajeva korišćenja, gornja polovina slike 1 ilustruje praćenje između slučajeva korišćenja i funkcionalnih zahteva.

Donja polovina slike 1 pokazuje da, kako se zahtevi povezuju sa razvoja, možete da pratite trag napred definisanjem veza između pojedinih funkcionalnih i nefunkcionalnih zahteva i specifičnih elemenata sistema. Ova vrsta veze omogućava vam da utvrdite da ste zadovoljili svaki zahtev, jer znate koje komponente projektnog rešenja i koda odgovaraju svakoj od njih. Četvrta vrsta veze prati trag određenog elementa proizvoda unazad do zahteva, tako da znate zašto je svaki element kreiran. Većina aplikacija ukazuju indirektno na tu vezu preko

testova, ili direktno preko komentara u kodu na zahteve koje je odredio korisnik, ali trebali biste znati zašto je napisana svaka linija koda.



Slika 1.1 Četiri vrste sledljivosti zahteva [Wiegers]

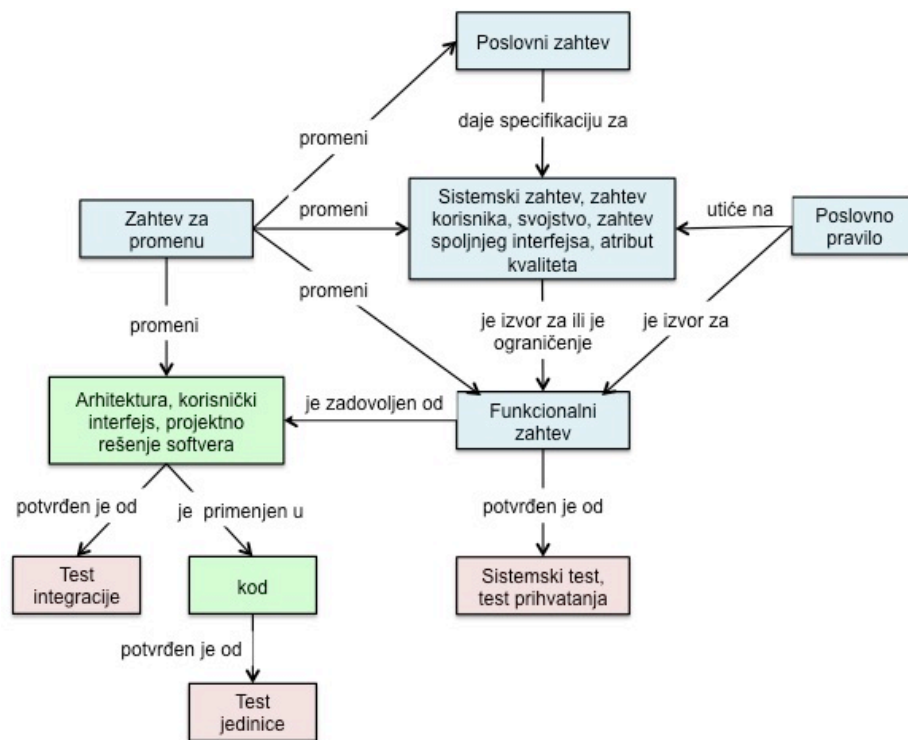
Veze u tragovima mogu vam pomoći da povežete zahteve i delove koda (funkcionalnosti), ako nisu ranije povezani i da napravite potpuniju sliku o tome kako se delovi vašeg sistema uklapaju zajedno. Veze u tragovima takođe vam pomažu da pratite roditeljsku povezanost, međusobne veze i zavisnosti između pojedinačnih zahteva. Olakšava širenje promena ako dođe do promene nekog zahteva.

VRSTE SLEDLJIVOSTI ZAHTEVA

Sleđenjem tragova, tj. veza, saznajemo sa kojim potrebama, s jedne strane, i sa kojim elementima softvera, s druge strane, je neki zahtev povezan.

Sledljivost zahteva vam omogućava da sledite njegov trag, tj. vezu, ili ka korisniku i njegovim potrebama, ili ka funkcionalnosti uobličenu nekim projektnim rešenjem softvera ili delom koda softvera. Praćenjem ovih "tragova", tj. veza, možete upravljati primenom zahteva, pratiti promene koje treba da izvršite ako njih menjate, ili ako menjate npr. kod, koji zahtev onda mora da promenite, ako je povezan sa tim delom koda. Sleđenjem ovih tragova, tj. veza, saznajemo sa kojim potrebama, s jedne strane, i sa kojim elementima softvera, s druge strane, je neki zahtev povezan

Slika 2 prikazuje mnoge vrste odnosa sledljivosti koje se mogu definirati na projektu. Naravno, ne morate da definišete i upravljate svim ovim vrstama veza u tragovima. Na mnogim projektima možete dobiti većinu prednosti sledljivosti koje želite samo za delić potencijalnog napora. Možda vam je potrebno samo da pratite sistemske testove do funkcionalnih zahteva ili zahteva korisnika. Uradite analizu troškova i koristite kako biste odlučili koje će veze doprineti uspehu vašeg projekta, kako u pogledu razvoja tako i dugoročnog napora na održavanju. Ne tražite članove tima da troše vreme na snimanje informacija, osim ako ne znate kako ih mogu koristiti.



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

Slika 1.2 Neke veze u tragovima zahteva

MOTIVACIJA ZA SLEĐENJE TRAGOVA ZAHTEVA

Sleđenje zahteva omogućava način da se pokaže usaglašenost sa specifikacijom, ugovorom ili regulativom

Ako ste prevideli neki zahtev je više nego sramota jer znači da kupac nije zadovoljan ili da proizvod nedostaje kritična funkcija.

Sleđenje tragova zahteva omogućava način da se pokaže usaglašenost sa specifikacijom, ugovorom ili regulativom. Na nivou organizacije, sleđenje tragova zahteva za implementacijom može poboljšati kvalitet vaših proizvoda, smanjiti troškove održavanja i olakšati ponovnu upotrebu.

Održavanje informacija o vezama tokom procesa razvoja i održavanja, zahteva disciplinu i vreme. Ako informacije o tragovima postanu zastarele, verovatno ih nikada nećete rekonstruisati. Zastareli ili netačni podaci o tragovima gube vreme šaljući programere i održavanje na pogrešan put, uništavajući svako poverenje koje su programeri mogli imati u podatke. Zbog ovih realnosti trebali biste koristiti tragove zahteva iz pravih razloga. Evo nekih potencijalnih prednosti sprovođenja traženja zahteva:

- **Pronalaženje nedostajućih zahteva** Potražite poslovne zahteve koji ne prate bilo koje korisničke zahteve, kao i zahteve korisnika koji ne prate bilo koje funkcionalne zahteve.
- **Pronalaženje nepotrebnih zahteva** Potražite sve funkcionalne zahteve koji ne odgovaraju zahtevima korisnika ili preduzeća i zato možda neće biti potrebni.

- **Potvrda i saglasnost** Informacije o tragovima možete koristiti prilikom sertifikacije proizvoda koji je kritičan za sigurnost, kako bi pokazao da su svi zahtevi implementirani - mada to ne potvrđuje da su pravilno implementirani! Informacije o tragovima pokazuju da su zahtevi za poštovanje propisa uključeni i adresirani, kao što je često potrebno za aplikacije za kompanije za zdravstvenu zaštitu i finansijske usluge.
- **Promenite analizu uticaja** Bez informacija o tragovima, postoji velika šansa da ćete zanemariti sistemski element na koji biste mogli uticati ako dodate, izbrišete ili izmenite određeni zahtev.
- **Održavanje** Pouzdane informacije o tragovima olakšavaju vašu sposobnost da ispravno i potpuno izvršite promene tokom održavanja. Kada se korporativne politike ili vladini propisi promene, softverski sistemi se često moraju ažurirati. Tabela koja pokazuje gde je svako primenjeno poslovno pravilo adresirano u funkcionalnim zahtevima, projektnom rešenju i kodu olakšava pravilno izvršenje potrebnih promena.
- **Praćenje projekata** Ako tokom razvoja snimate podatke u tragovima, imaćete tačan zapis o statusu implementacije planirane funkcionalnosti. Odsutni linkovi ukazuju na radne proizvode koji još nisu stvoreni.
- **Reinženjering**: Možete izlistati funkcije starog sistema koji menjate i nađite njihove zahteve u zahtevima i SW novog sistema

RAZLOZI ZA ČUVANJE INFORMACIJA O TRAGOVIMA ZAHTEVA

Uspostavljanje tragova nije mnogo posla ako informacije prikupljate u toku razvoja, ali naporno je i skupo uraditi na dovršenom sistemu.

- **Informacije o ponovnoj upotrebi** Tragovi e olakšavaju ponovnu upotrebu komponenta proizvoda identifikovanjem paketa povezanih zahteva, dizajna, koda i testova.
- **Testiranje** Kada test ne uspe, veza između testova, zahteva i programera kodne tačke prema verovatnim područjima koja treba ispitati na kvar.

Mnoge od tragova imaju dugoročnu korist, smanjujući ukupne troškove životnog ciklusa proizvoda, ali povećavajući troškove razvoja naporima uloženim u akumuliranje i upravljanje informacijama u tragovima. Pogledajte zahteve koji prate kao investiciju koja povećava vaše šanse da isporučite održiv proizvod koji zadovoljava sve navedene zahteve kupca. Ova investicija isplatiće dividendu kad god morate modifikovati, proširiti ili zameniti proizvod. Uspostavljanje tragova nije mnogo posla ako informacije prikupljate u toku razvoja, ali naporno je i skupo uraditi na dovršenom sistemu.

▼ Poglavlje 2

Matrica sledljivosti zahteva

PRIMER MATRICE SLEDLJIVOSTI

Matrica sledljivosti prikazuje kako je svaki funkcionalni zahtev povezan unazad za određeni slučaj upotrebe i prosleđuje se jednom ili više elemenata

Najčešći način predstavljanja veza između zahteva i drugih elemenata sistema je u **matrici sledljivosti zahteva**, koja se takođe naziva matrica sledljivosti zahteva (requirements traceability matrix) ili tabela sledljivosti. Tabela na slici 1 ilustruje deo matrice za sledljivost zahteva, koja je izvedena iz sistema za praćenje hemikalija.

Zahtev korisnika	Funkcionalni zahtev	Element projektnog rešenja	Element koda	Test
UC-28	catalog.query.sort	Class catalog	CatalogSort()	search.7 search.8
UC-29	catalog.query.import	Class catalog	CatalogImport() CatalogValidate	search.12 search.13 search.14

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

Slika 2.1 Jedna vrsta sledljivosti zahteva

Tabela na slici 1 prikazuje kako je svaki funkcionalni zahtev povezan unazad za određeni slučaj upotrebe i prosleđuje se jednom ili više elemenata projektnog rešenja, koda i ispitivanja. Element projektnog rešenja može biti nešto poput arhitektonske komponente, tabele u relacionom modelu podataka ili klase objekta. Reference kodova mogu biti metode klase, uskladištene procedure, nazivi datoteka izvornog koda ili moduli unutar izvorne datoteke. Uključivanje više detalja u tragovima zahteva više posla, ali vam daje tačne lokacije povezanih softverskih elemenata.

Popunite podatke kada posao bude gotov, a ne onako kako je planirano. To jest, unesite CatalogSort () u kolonu „Element koda“ prvog reda u tabeli na slici 1 samo kada je zapisan kod u toj funkciji. Na taj način čitalac zna da popunjene ćelije u matrici sledljivosti zahteva ukazuju na posao koji je završen.

Drugi način predstavljanja informacija u tragovima je kroz skup matrica koje definišu veze između parova sistemskih elemenata, kao što su ove:

- Jedna vrsta zahteva za ostale zahteve iste vrste
- Jedna vrsta zahteva za zahtevima druge vrste
- Jedna vrsta zahteva za testovima

Možete koristiti ove matrice za definisanje različitih odnosa koji su mogući između parova zahteva, poput „specificira / je određeno od“, „zavisi od“, „je roditelj od“, i „ograničava / ograničava od“.

KARDINALNOST VEZA U MATRICI SLEDLJIVOSTI

Svaka matrice ćelija na preseku dve povezane komponente sadrži simbol koji označava vezu koje mogu da definišu odnose tipa jedan-na-jedan, jedan-na-mnogo mnogo-na-mnogo.

Tabela na slici 2 prikazuje matricu dvosmerne sledljivosti. Većina ćelija u matrici je prazna. Svaka ćelija na preseku dve povezane komponente sadrži simbol koji označava vezu. Tabela koristi strelicu da naznači da se iz određenog slučaja upotrebe izvodi određeni funkcionalni zahtev. Na primer, FR-2 je praćen od UC-1, a FR-5 i od UC-2 i od UC-4. Ovo ukazuje da se funkcionalni zahtev FR-5 ponovo koristi u dva slučaja upotrebe, UC-2 i UC-4.

Funkcionalni zahtev	Slučaj upotrebe			
	UC-1	UC-2	UC-3	UC-4
FR-1	↙			
FR-2	↙			
FR-3			↙	
FR-4			↙	
FR-5		↙		↙
FR-6			↙	

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

Slika 2.2 Matrica sledljivosti zahteva koja prikazuje veze između slučajeva upotrebe i funkcionalnih zahteva

Veze u tragovima mogu da definišu odnose jedan na jedan, jedan na mnoge ili između mnogih i drugih sistema. Format u tabeli na slici 1 obuhvata ove kardinalnosti omogućavajući vam unos nekoliko stavki u svaku ćeliju tabele. Evo nekoliko primera mogućih kardinalnosti veza:

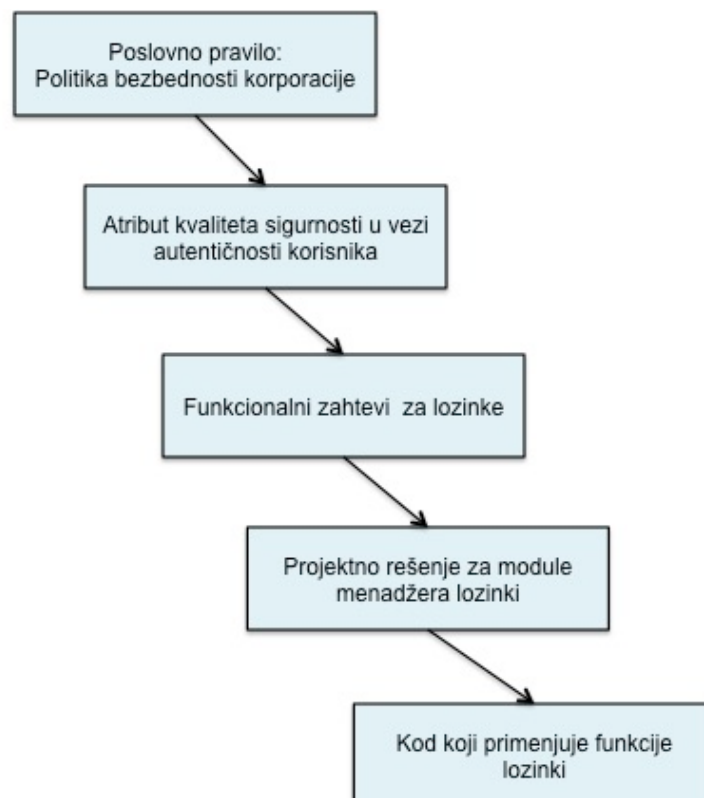
- **Jedan-na-jedan** Element projektnog rešenja implementiran je u jednom kodnom modulu.
- **Jedan funkcionalni zahtev** Jedan funkcionalni zahtev potvrđuje se više testova.
- **Mnogo-na-mnogo** Svaki slučaj upotrebe dovodi do više funkcionalnih zahteva, a određeni funkcionalni zahtevi su zajednički za nekoliko slučajeva upotrebe. Slično tome, zajednički ili ponovljeni element projektnog rešenja može zadovoljiti nekoliko funkcionalnih zahteva. U idealnom slučaju ćete snimiti sve ove međusobne veze, ali u praksi, odnosi mnogih u tragovima postaju složeni i teško se upravljaju.

LANAC SLEDLJIVOSTI KOJI UKLJUČUJE NEFUNKCIONALNE ZAHTEVE

Nefunkcionalni zahtevi ne prate direktno kod. Odgovarajuće funkcionalne zahteve možete pratiti unazad prema njihovim roditeljskim nefunkcionalnim zahtevima.

Nefunkcionalni zahtevi, poput atributa kvaliteta, često ne prate direktno kod. Zahtev za vreme odziva može diktirati upotrebu određenog hardvera, algoritama, struktura baza podataka i arhitektonskog pristupa. Zahtev za prenosivost može ograničiti jezične karakteristike koje programer koristi, ali možda neće rezultirati određenim segmentima koda koji omogućavaju prenosivost. Ostali atributi kvaliteta zaista se implementiraju u kod. Sigurnosni zahtevi za proveru autentičnosti korisnika vode do izvedenih funkcionalnih zahteva koji se mogu implementirati putem lozinki ili biometrijske funkcionalnosti. U tim slučajevima, odgovarajuće funkcionalne zahteve možete pratiti unazad prema njihovim roditeljskim nefunkcionalnim zahtevima i prosleđuju se dole nizvodnim proizvodima kao i obično.

Slika 3 ilustruje mogući lanac sledljivosti koji uključuje nefunkcionalne zahteve.



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

Slika 2.3 Primer lanca sledljivosti za zahteve koji se odnose na sigurnost aplikacije

KO DEFINIŠE VEZE U TRAGOVIMA?

Odredite uloge i pojedince koji bi trebalo da dostave svaku vrstu informacija u tragovima za vaš projekat.

Veze u tragovima treba da definiše onaj ko ima na raspolaganju odgovarajuće informacije. Tabela na slici 4 prikazuje neke tipične izvore znanja o vezama između različitih vrsta izvora i ciljnih objekata. Odredite uloge i pojedince koji bi trebalo da dostave svaku vrstu informacija u tragovima za vaš projekat. Očekujte neki povraćaj od zauzetih ljudi koje analitičar ili rukovodilac projekta traži da dostave ove podatke. Ti praktičari imaju pravo na objašnjenje traženja zahteva, zašto to daje vrednost i zašto se od njih traži da doprinesu procesu. Istaknite da su dodatni troškovi prikupljanja informacija u tragovima u vreme kada se posao obavlja mali; to je prvenstveno pitanje navike, discipline i uspostavljanja mehanizma za skladištenje.

Tip objekta izvora veze	Tip objekta cilja veze	Izvor informacije
Sistemska zahtev	Funkcionalni zahtev	Inženjer sistema
Zahtev korisnika	Funkcionalni zahtev	Biznis analitičar
Poslovni zahtevi	Zahtev korisnika	Biznis analitičar
Funkcionalni zahtev	Funkcionalni zahtev	Biznis analitičar
Funkcionalni zahtev	Test	Tester
Funkcionalni zahtev	Element arhitekture	Arhitekta sistema ili programer
Funkcionalni zahtev	Drugi elementi projektnog rešenja	Projektant ili programer
Element projektnog rešenja	Kod	Programer
Poslovno pravilo	Funkcionalni zahtev	Biznis analitičar

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

Slika 2.4 Uobičajeni izvori informacija za veze u tragovima

▼ Poglavlje 3

Alati za praćenje zahteva

SUMNJIVE VEZE U MATRICI SLEDLJIVOSTI ZAHTEVA

Neki alati automatski označavaju vezu traga kao sumnjivu kada se objekat na bilo kojem kraju veze menja. Sumnjiva veza prikazuje vizuelni indikator u odgovarajućoj ćeliji matrice sledljivosti.

Alati za upravljanje komercijalnim zahtevima često imaju moćne mogućnosti praćenja zahteva. Možete da smestite zahteve i druge informacije u bazu podataka alata i definišete veze između različitih vrsta sačuvanih objekata, uključujući vršnjačke veze između dva zahteva iste vrste. Neki alati vam omogućavaju da razlikujete odnos između traga i traga, automatski definišući komplementarne veze. To jest, ako naznačite da se zahtev R nalazi za ispitivanje T, alatka će takođe pokazati simetrični odnos u kojem je T praćen od R.

Neki alati automatski označavaju vezu traga kao sumnjivu svaki put kada se objekat na bilo kojem kraju veze modifikuje. Sumnjiva veza prikazuje vizuelni indikator (poput crvenog znaka pitanja ili dijagonalne crvene linije) u odgovarajućoj ćeliji u matrici sledljivosti zahteva. Na primer, ako ste promenili slučaj upotrebe 3, matrica sledljivosti zahteva u tabeli na slici 2 u poglavlju 2 može izgledati kao tabela 1 sledeći put kada je vidite. **Pokazatelji sumnjivih veza (u ovom slučaju upitnici) govore da proverite da li je potrebno promeniti funkcionalne zahteve 3, 4 i 6 da bi ostali u skladu sa modifikovanim UC-3.** Nakon što unesete sve potrebne izmene, ručno očistite indikatore sumnjivih veza. Ovaj postupak pomaže da se osigura da ste uzeli u obzir poznate efekte promene.

Nemoguće je izvršiti ručno praćenje zahteva za bilo koje, ali vrlo male aplikacije.

Funkcionalni zahtev	Slučaj upotrebe			
	UC-1	UC-2	UC-3	UC-4
FR-1	↙			
FR-2	↙			
FR-3			↙?	
FR-4			↙?	
FR-5		↙		↙
FR-6			↙?	

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

Slika 3.1 Sumnjive veze u matrici sledljivosti zahteva

Možete koristiti proračunsku tablicu za održavanje podataka u tragovima do nekoliko stotina zahteva, ali veći sistemi zahtevaju robusnije rešenje. Praćenje traženja zahteva ne može biti u potpunosti automatizovano, jer znanje o vezama potiče iz misli članova razvojnog tima. Međutim, nakon što utvrdite veze, alati vam mogu pomoći u upravljanju ogromnom količinom informacija u tragovima

▼ Poglavlje 4

Procedura za praćenje zahteva

VIDEO PREDAVANJE ZA OBJEKAT "PROCEDURA ZA PRAĆENJE ZAHTEVA"

Trajanje video snimka: 28min 8sek

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

KORACI PROCEDURE ZA PRAĆENJE ZAHTEVA

Kada dodajete poboljšanja ili menjate postojeći sistem, zapišite šta ste otkrili o vezi između koda, testova, projektovanja i zahteva, da bi držali matricu sledljivosti ažurnom.

Preporučuju se sledeći koraci za primenu traženja zahteva (preko tragova) u nekom projektu:

1. Obučite tim i vaše rukovodstvo o konceptima i značaju praćenja zahteva, vašim ciljevima ove aktivnosti, gde se čuvaju podaci u tragovima, i tehnikama za definisanje veza. Zamolite sve učesnike da se obavežu na svoje odgovornosti.
2. Izaberite vezu odnosa koju želite da definišete iz mogućnosti prikazanih kao na slici 2 u poglavlju 2. Ne pokušavajte da odradite sve ovo odjednom.
3. Odaberite vrstu matrice sledljivosti koju želite koristiti: jednosmerni stil prikazan u tabeli na slici 1 poglavlja 2 ili nekoliko matrica poput one prikazane u tabeli na slici 2 poglavlja 2. Izaberite mehanizam za čuvanje podataka: tabelu u tekstualnom dokumentu, proračunsku tablicu (Excell) ili (mnogo bolje) alat za upravljanje zahtevima.
4. Utvrdite delove proizvoda za koje želite da zadržite informacije o sledljivosti. Započnite sa kritičnim osnovnim funkcijama, visokorizičnim delovima ili delovima za koje očekujete da će proći najviše održavanja i evolucije tokom života proizvoda.
5. Odredite pojedince koji će dostaviti svaku vrstu informacija o vezama i osobu (najverovatnije BA) koji će koordinirati aktivnosti praćenja i upravljati podacima.
6. Izmenite svoje razvojne procedure tako da podsetite programere da ažuriraju veze nakon primene zahteva ili odobrene promene. Podaci o tragovima bi trebalo da se ažuriraju ubrzo nakon što neko izvrši zadatak koji stvara ili menja vezu u lancu zahteva.
7. Definišite konvencije o označavanju koje ćete koristiti da svakom elementu sistema date jedinstveni identifikator kako bi se elementi mogli povezati.

8. Kako razvoj traje, svaki učesnik treba da obezbedi tražene informacije u tragovima dok završavaju sitnice. Naglasite prednost stalnog prikupljanja podataka o tragovima u odnosu na njihovo sastavljanje na važnoj prekretnici ili na kraju projekta.

9. Povremeno proverite informacije o tragovima da biste bili sigurni da se aktuelni. Ako je zahtev prijavljen kao implementiran i verifikovan, a podaci o njegovim tragovima su nepotpuni ili netačni, postupak praćenja zahteva ne funkcioniše kako treba. Ovaj postupak počinje da se primenjuje na početku novog projekta. Ako održavate postojeći sistem, verovatno nemate dostupne podatke u tragovima. Počnite odmah da sakupljate ove informacije. Sledeći put kada dodate poboljšavate ili izvršite modifikaciju, zapišite šta ste otkrili o vezi između koda, testova, projektovanja i zahteva.

DA LI JE PRAĆENJE ZAHTEVA SPROVODLJIVO? DA LI JE POTREBNO?

Odlučite da koristite bilo kakvu poboljšanu inženjersku praksu zasnovanu kako na troškovima primene tehnike, tako i na riziku njenog korišćenja.

Možete zaključiti da je sakupljanje podataka u tragovima sa zahtevima skuplje nego što vredi ili da to nije izvedivo za vaš projekat. To je potpuno moguće. Dobijanje alata sa potrebnim mogućnostima, njegovo postavljanje, unošenje podataka i održavanje aktuelnosti je skupo i dugotrajno. Možda neće trebati da konstruišete grupnu memoriju poput ove ako članovi vašeg tima poseduju potrebno znanje i dele ga sa drugima kada je to potrebno. Samo vaš tim može odlučiti da li praćenje zahteva - bilo da se radi samo o zahtevima za testovima ili nečem složenijim - dodaje vrednosti vašem projektu iznad njegove cene. Ipak, uzmite u obzir sledeći primer. *Učesnik jedne konferencije koji je radio kod proizvođača aviona rekao je da je SRS za deo njegov najnoviji avion bio stub papira visine dva metra. Imali su kompletnu matricu sledljivosti zahteva. Kaže da je leto na tom modelu aviona i bio srećan kad je čuo da su programeri tako pažljivo upravljali njihovim softverskim zahtevima.* Upravljanje tragovima na ogromnom proizvodu, sa mnogo međusobno povezanih podsistema, zahteva mnogo rada. Ovaj proizvođač vazduhoplova zna da je to neophodno. Američka Federalna uprava za vazduhoplovstvo se slaže: za sertifikaciju vazduhoplovnog softvera potrebna je sledljivost od zahteva do projektnog rešenja. Slično tome, američka administracija za hranu i lekove zagovara da proizvođači medicinskih proizvoda pokazuju sledljivost potreba proizvoda u proizvodima sa nižom cenom kao deo postupka provere valjanosti uređaja.

Čak i ako vaši proizvodi neće prouzrokovati gubitak života ili zdravlja ako ne uspeju, trebalo bi da shvatite ozbiljno traženje zahteva. U najmanju ruku, razmislite o pronalaženju između poslovnih zahteva i zahteva korisnika da potražite usklađivanje, propuste i nepotrebne zahteve.

Direktor velike korporacije je na konferenciji upitao: „Zašto to ne biste uradili za svoje strateške poslovne sisteme?“ To je odlično pitanje. Odlučite da koristite bilo kakvu poboljšanu inženjersku praksu zasnovanu kako na troškovima primene tehnike, tako i na riziku njenog korišćenja. Kao i kod svih softverskih procesa, **donesite ekonomsku odluku da uložite svoje dragoceno vreme tamo gde očekujete najveći povraćaj.**

▼ Poglavlje 5

Poboljšanje procesa zahteva

VEZE ZAHTEVA SA DRUGIM PROCESIMA U PROJEKTU

Promene koje unesete u svoje zahteve odraziće se na ostale procese u projektu i obrnuto

Zahtevi su srce svakog dobro vođenog softverskog projekta, koji podržavaju i omogućavaju ostale tehničke i upravljačke aktivnosti. Promene koje unesete u svoje zahteve odraziće se na ostale procese u projektu i obrnuto. Slika 1 ilustruje neke veze između zahteva i drugih procesa na projektu.



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

Slika 5.1 Veze zahteva sa drugim procesima u projektu

Planiranje projekta: Zahtevi služe kao temelj procesa planiranja projekata. Planeri biraju odgovarajući model razvoja softvera i životni ciklus. Planiranje projekta može da pokaže da nije moguće urediti ceo skup željenih svojstava zbog ograničenih resursa i rokova. Planiranje može da dovede do smanjenja opsega projekta ili do izbora inkrementalnog razvoja ili do postepene isporuke delova softvera. Radi pružanja funkcionalnosti u fazama. Na agilnom projektu, opseg je definisan setom korisničkih priča o proizvodu ili verziji i postupno se

implementira u svakoj iteraciji. Domet za buduće iteracije zasnovan je na merenju brzine iz prethodnih iteracija.

Praćenje i kontrola projekata: Praćenje projekta uključuje nadgledanje statusa projekta tako da rukovodilac projekta može da vidi da li konstruisanje softvera i verifikacija se realizuju po planu. Ako oni to nije slučaj, menadžment, kupci ili drugi akteri možda će morati da promene opseg kroz proces planiranja. Ovo bi promenilo zahteve koji se rade. Na agilnom projektu, opseg se definiše pomeranjem stavki nižeg prioriteta za buduće iteracije, ako je potrebno da biste dovršili svaku iteraciju prema rasporedu.

Kontrola promene: Nakon uspostavljanja niza zahteva, sve naredne promene i dopune treba da se izvrše kroz definisan proces kontrole promene. Zahtevi menjaju zaostatak toka rada i prioritete radnih predmeta u zaostatku. Praćenje zahteva vam pomaže da procenite uticaj promena opsega. Obaveštavaju se odgovorni radi odobrenja promena.

DOPRINOSI RAZVOJU ZAHTEVA DRUGIH AKTERA GRUPI ZA RAZVOJ SOFTVERA

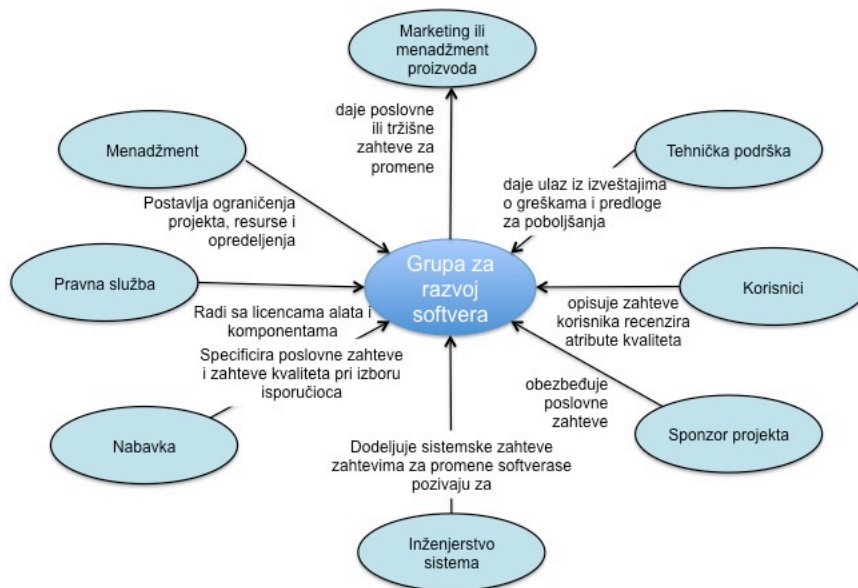
Ako ste poslovni analitičar ili rukovodilac projekata, morate naučiti aktere šta od njih očekujete da bi razvoj proizvoda uspeo.

Prihvatanje i testiranje sistema. Zahtevi korisnika i funkcionalni zahtevi su bitan ulaz u testove prihvatanja. Ako očekivano ponašanje softvera u različitim uslovima nije jasno definisano, tester će biti pod pritiskom da potvrde da je funkcionalnost softvera ostvarena u celosti.

Konstruisanje: Zahtevi su osnova za projektovanje i konstruisanje (kodiranje) softvera, te su u bliskoj vezi sa poslovima konstruisanja. Koristite preglede projektnog rešenja a kako biste osigurali da je ona pravilno ispunjavaju sve zahteve. Jedinično testiranje može odrediti da li softver zadovoljava specifikacije projektnog rešenja i zahteve. Praćenjem zahteva možete doći do elemenata projektnog rešenja i koda koji su dobijenu na osnovu svakog zahteva.

Korisnička dokumentacija: Zahtevi proizvoda su ulaz za pripremu korisničke dokumentacije. Ako su zahtevi slabo napisani ili ako kasne, to će se odraziti na kvalitet korisničke dokumentacije. Zbog toga pisci korisničke dokumentacije su podržavaoci dosledne primene pravila inženjerstva zahteva.

Na slici 2 prikazani su akteri na projektu koji bi mogli komunicirati sa grupom za razvoj softvera i dati svoj doprinos aktivnostima na projektu. Ako ste poslovni analitičar ili rukovodilac projekata, morate naučiti aktere šta od njih očekujete da bi razvoj proizvoda uspeo. Složite se o komunikacionim interfejsima između grupe za razvoj i ostalih funkcionalnih jedinica, kao što su interfejsi vezani sa specifikaciju zahteva sistema, dokument o zahtevima tržišta, ili skup korisničkih priča.



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

Slika 5.2 Doprinosi razvoju zahteva drugih aktera grupi za razvoj softvera

DOBIJANJE PODRŠKE ZA PROMENE

Da biste umanjili strah od nepoznatog, obavestite o poboljšanju svog procesa. Objasnite prednosti koje će druge grupe dobiti od novog procesa.

Kada softverska organizacija promeni svoje zahteve, menjaju se i interakcije koje ona ima sa drugim zainteresovanim zajednicama. Ljudi ne vole da ih forsiraju iz svoje zone komfora, pa očekujte da se odupru promenama procesa koje predlažete. Shvatite poreklo otpora kako biste ga mogli i poštovati i ublažiti.

Veliki otpor dolazi iz straha od nepoznatog. Da biste umanjili ovaj strah, obavestite o poboljšanju svog procesa. Objasnite prednosti koje će druge grupe dobiti od novog procesa. Krenite sa ovog stanovišta: "Evo problema koje smo iskusili. Koja su pitanja iz vaše perspektive? Možemo li sklopiti glave na bolji način? Evo nekih oblika otpora na koje nailazite:

- Ljudi koji su već previše zauzeti da bi dobili svoj posao, ne misle da imaju vremena uložiti u usvajanje boljih praksi. Ali ako ne uložite to vreme, nema razloga da očekujete da će sledeći projekat biti uglađeniji od poslednjeg.
- Proces kontrole promena može se posmatrati kao barijera za rast sprovođenjem promena. U stvarnosti, to je struktura, a ne prepreka. To čini dobro informisane ljude da donose dobre poslovne odluke. Softverski tim mora osigurati da postupak promene zahteva zaista deluje. Ako novi procesi ne daju bolje rezultate, ljudi će prirodno pronaći načine kako da se snađu.
- Neki programeri i menadžeri vide pisanje i redizajniranje kao birokratske gubitnike vremena koji odgađaju "pravi posao" kodiranja. Ako možete objasniti visoku cenu neprestanog prepisivanja koda dok tim pokušava da shvati šta sistem treba da radi,

programeri i menadžeri će uvideti potrebu za dobrim zahtevima. Previđeni zahtevi mogu umanjiti profitabilnost tokom radnog veka softverskog proizvoda, jer se stalno moraju ulagati u proizvodnju nadogradnji.

Svaki put kada se od ljudi traži da promene način rada, prirodna reakcija je da pitaju: "Šta je to za mene?" Bolje pitanje - i ono koje zaslužuje dobar odgovor - je „Šta je tu za nas?“ Svaki proces se menja u jasnu korist projektnom timu, razvojnoj organizaciji, kompaniji i / ili kupcu. Akteri vide da se od kojih se traži da potroše više vremena, nego do sada.. Cilj je razumeti da se ova investicija može svesti na niži trošak i povećati vrednost za kupce. Ovo razumevanje moglo bi ih učiniti spremnijim da sada troše vreme.

Važan metod za sticanje posvećenosti procesu isticanje problema vidljivim na argumentovan i konstruktivan način.

PRINCIPI UNAPREĐENJA SOFTVERA:

Veliki napor za unapređenje procesa može se postići samo ako je osoba motivisana da angažuje resurse, postavi očekivanja i da da doprinos inicijativi za promene softvera.

1. **Poboljšanje procesa** treba da bude evolutivno i kontinuirano. Umesto da težite savršenstvu, razvijte nekoliko rešenja i započnite sa implementacijom. Prilagodite svoje pristupe timu sticanje iskustva sa novim tehnikama. Ponekad jednostavne i jednostavne promene mogu dovesti do znatnih dobitaka, zato potražite plodove koji više dobro.

2. **Ljudi i organizacije** se menjaju samo kada imaju podsticaj za to.

Najjači podsticaj za promenu je bol. Ne veštački izazvana bol, poput rasporeda nametnutog od strane menadžmenta kako bi se timovi trudili, već sasvim pravi kruh koji su ljudi iskusili na prethodnim projektima. Sledi nekoliko primera problema koji mogu ponuditi snažne upravljačke programe za promenu vaših zahteva :

- Projekat je promašio rokove jer su bili obimniji nego što se očekivalo.
- Programeri su radili puno prekovremenog rada zbog pogrešno shvaćenih ili dvosmislenih zahteva.
- Napori na testiranju sistema su izgubljeni jer testerji nisu razumeli šta proizvod treba da radi.
- prava funkcionalnost je bila prisutna, ali korisnici su bili nezadovoljni zbog sporog rada, loše upotrebljivosti ili drugih nedostataka kvaliteta.
- Organizacija je doživela visoke troškove održavanja zbog mnogih potreba koje su utvrđene tokom iznošenja zahteva.
- Potrebne promene nisu provedene na odgovarajući način tokom projekta, tako da isporučeno rešenje nije zadovoljilo potrebe klijenta.
- Izmene u zahtevima su izgubljene ili prebrisane od strane više BA koji rade na konkurentu sa procesom kontrole verzije.
- Kupcima nije bilo na raspolaganju da razjasne i razviju zahteve
- Pitanja vezana za zahteve nisu rešena blagovremeno, što je prouzrokovalo prepravku.

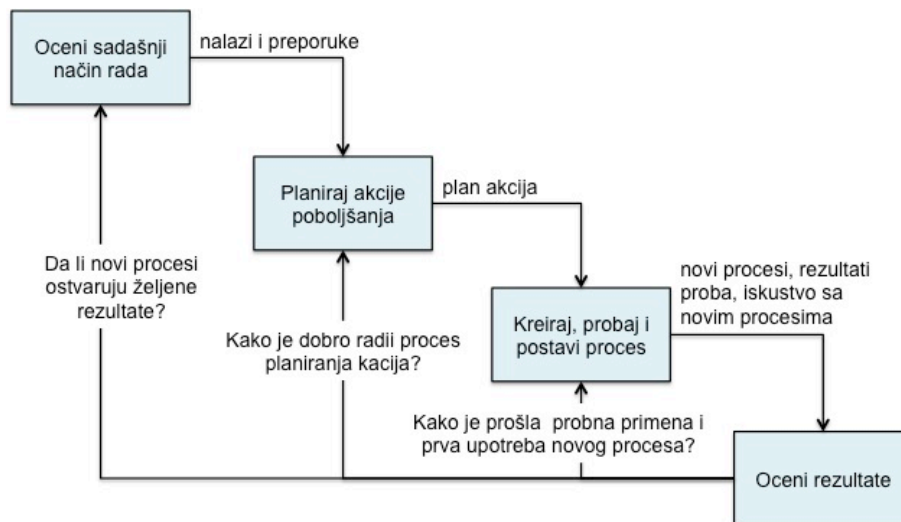
3. **Procesne promene** treba da budu ciljno orijentisane. Pre nego što započnete put ka vrhunskim procesima, proverite da li znate svoje ciljeve. Da li želite da smanjite količinu potrebnog posla? Da li želite da vas tokom primene zaobiđu? Želite li što prije smanjiti nepotrebne funkcije? Mapa puta koja definiše vaše šanse za uspeh.

4. Tretirajte svoje **aktivnosti poboljšanja** kao mini projekte. Mnoge inicijative za unapređenje zato što su loše planirane ili zato što se resursi nikada ne ostvaruju. Uključite resurse i zadatke za unapređenje procesa u opšti plan projekta. Izvršite planiranje, praćenje, merenje i izveštavanje koje biste radili za bilo koji projekat srazmerno veličini projekta poboljšanja.

CIKLUS POBOLJŠANJA PROCESA

Ciklus poboljšanja softverskog procesa odražava važnost saznanja gde ste sada, pre nego što krenete dalje i vodi vas ka promenama koje treba da poboljšaju proces.

Na slici 3 prikazan je efikasan ciklus poboljšanja procesa. Ovaj ciklus odražava važnost saznanja gde ste sada, pre nego što krenete dalje, potrebu da usmerite svoj dalji put i važnost od učenja na osnovu dosadašnjeg iskustva, kao deo kontinuiranog poboljšanja.



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

Slika 5.3 Ciklus poboljšanja softverskog procesa

Procenite sadašnju praksu rada

Prvi korak bilo kakvog poboljšanja trenutno se koristi za prepoznavanje njihovih snaga i nedostataka. Procena je osnova za odabir promena koje biste trebali izvršiti. To takođe donosi vidljivost prema procesima koji se koriste u organizaciji, a koji se često razlikuju od navedenih ili napisanih procesa. Otkrićete da različiti članovi tima rade različite stvari.

Možete da ocenite svoje trenutne potrebe na više načina. Dodatak A, „Vodič za rešavanje problema sa zahtevima“, nudi rešenja uobičajenih problema, zajedno sa mogućim uzrocima i mogućim rešenjima. Strukturirani upitnici mogu vam otkriti uvid u vaše trenutne procese

uz nisku cenu. Intervjui i diskusije sa članovima tima pružaju tačnije i sveobuhvatnije razumevanje od otkrivanja upitnika. Formalne evaluacije od strane spoljnih konsultanata daju listu nalaza-izjava o jakim i slabostima trenutnih procesa i preporuke za adresiranje mogućnosti za poboljšanje.

Za jednostavan pristup „uradi sam“, koristite upitnik iz Dodatka A, „Aktuelni zahtevi za samoprocenu“, da biste procenili trenutne prakse tehničke potrebe vaše organizacije. Ova samoocena vam pomaže da odlučite koji od vaših procesa je najpotrebniji za poboljšanje. Samo zato što sebi dajete nisku ocenu. Usmerite energiju na pokretanje poslovnih područja koja uzrokuju vaše projekte.

PLANIRANJE AKCIJA POBOLJŠANJA

Grupišite više akcija poboljšanja procesa razvoja zahteva u celokupni plan poboljšanja softverskog procesa.

U skladu sa filozofijom tretiranja projekata unapređenja procesa, napišite akcioni plan prateći vašu trenutnu procenu prakse. Taktički akcioni planovi ciljaju specifična područja poboljšanja, kao što su Svaki akcioni plan treba da identifikuje merljive ciljeve poboljšanja, učesnike i pojedine stavke akcije koje moraju biti dovršene kako bi se plan sproveo. Bez plana je lako zanemariti važne zadatke. Plan vam takođe omogućava da pratite napredak dok pratite završetak pojedinih stavki akcije. Slika 4 ilustrira obrazac akcionog plana za poboljšanje procesa koji smo koristili mnogo puta. U svaki akcioni plan ne uključite više od oko 10 stavki, tako da se plan može završiti za 2 ili 3 meseca. Kao primer, ovde se navode neke akcije:

1. Nacrtajte postupak kontrole promene zahteva.
2. Pregledajte i pregledajte postupak kontrole promena.
3. Pilot proces kontrole promene pomoću projekta A.
4. Revidirati postupak kontrole promena na osnovu povratnih informacija od pilota
5. Procenite alate za praćenje problema i odaberite jedan koji će podržati postupak kontrole promena.
6. Nabavite alat za praćenje problema i prilagodite ga da podržava proces kontrole promena.
7. Unesite novi proces i alat za kontrolu promene u organizaciju.

Dodelite svaku akciju određenoj osobi koja je odgovorna za uvid u to da je akcija završena. Ne dodeljujte „tim“ kao vlasnika akcije. Timovi ne rade; rade pojedinci. Ako vam treba više od oko 10 akcija, usredsredite se na početni ciklus aktivnosti na najvažnija pitanja. Zapamtite, promena procesa je postupna i kontinualna.

Akcioni plan za poboljšanje procesa razvoja zahteva					
Projekat:		Datum:;			
Ciljevi: <Navedite nekoliko ciljeva za uspešno izvršenje plana. Izrazite ciljeve u terminima poslovnih vrednosti, a ne promena procesa.>					
Mere uspeha: <Opišite kako ćete oceniti kada promene procesa ostvaruju željene rezultate>					
Opseg orahizacionog uticaja: <Opišite širinu uticaja promena procesa koje su opisane u ovom planu>					
Članovi tima i učesnici: <Identifikujte pojedince koji će primeniti plan akcija, njihove uloge, i broj sati nedeljno koje mora da odvoje>					
Praćenje i proces izveštavanja: <Opišite kako će biti praćen napredak relacije akcija, i kome s epodnose izveštaji o statusu i uspehu primene>					
Zavisnosti, rizici o barijere: <Utvrdite spoljne faktore koji su neophodni za uspeh plana ili koji mogu da spreče uspešnu primenu plana.>					
Procenite datum yavršetka svih akcija: <Kada očekujete da će plan biti u potpunosti iyvršen?>					
Stavke akcija: <Napišite 3 do 10 stavki za svaki plan akcija>					
Stavka akcije	Vlasnik	Rok	Aktivnosti	Rezultati	Potrebni resursi

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

Slika 5.4 Uzorak plana akcija za poboljšanje procesa razvoja zahteva

KREIRANJE, PROBANJE ("PILOTIRANJE") I POSTAVLJANJE PROCESA

Ne postavljate previše novih očekivanja od projektnog tima odjednom, kada planirate poboljšanja procesa razvoja zahteva.

Za sada ste u mogućnosti da upravljate svojim poslom. Sada dolazi teški deo: sprovođenje plana. Implementacija akcionog plana znači razvoj procesa za koji verujete da će dati bolje rezultate od vaših trenutnih načina rada. Ne očekujte da će prvi proces biti savršen kod prvog pokušaja. Mnogi pristupi koji izgledaju kao dobra ideja u apstraktnom obliku. Stoga je većina postupaka za stvaranje malog obima. Koristite znanje pilota (probne primene) da biste prilagodili novi proces. Ovo povećava šansu da bude efikasno i efikasno. Imajte na umu sledeće predloge za svoje pilote:

- Odaberite pilot učesnike koji će novim pristupima pošteno pokušati i pružiti korisne povratne informacije. Ovi učesnici mogu biti saveznici ili skeptici, ali ne bi trebalo snažno protiviti naporima na poboljšanju.
- Kvantifikujte kriterijume koje će tim koristiti za procenu rezultata pilota.
- Identifikujte zainteresovane strane koje treba da budu obaveštene o pilotu i zašto se on izvodi.
- Razmotrimo pilotske delove novih procesa na različitim projektima. To uključuje više ljudi u isprobavanju novih pristupa, što povećava svest, povratne informacije i kupovinu.
- Kao deo evaluacije, kako pilotirati učesnike kako bi se osećali kada bi se vratili svojim načinima rada.

Ako je pilot uspeo, spremni ste da izvršite bilo kakva poslednja prilagođavanja procesa i da ga ustupite zajednici koja je pogođena. Čak i motivisani i prijemčivi timovi imaju

ograničen kapacitet da apsorbuju promene, tako da ne postavljate previše novih očekivanja od projektnog tima odjednom. Izrada plana implementacije koji definiše kako ćete distribuirati nove metode i materijale projektnim timovima i pružiti dovoljnu obuku, obuku i pomoć. Takođe razmislite kako će menadžment postaviti svoja očekivanja u pogledu novih procesa.

OCENA POSTIGNUTIH REZULTATA

Pri uvođenju novog poboljšanog procesa, u početku produktivnost pada dok se učesnici u procesu ne prilagode novom načinu rada.

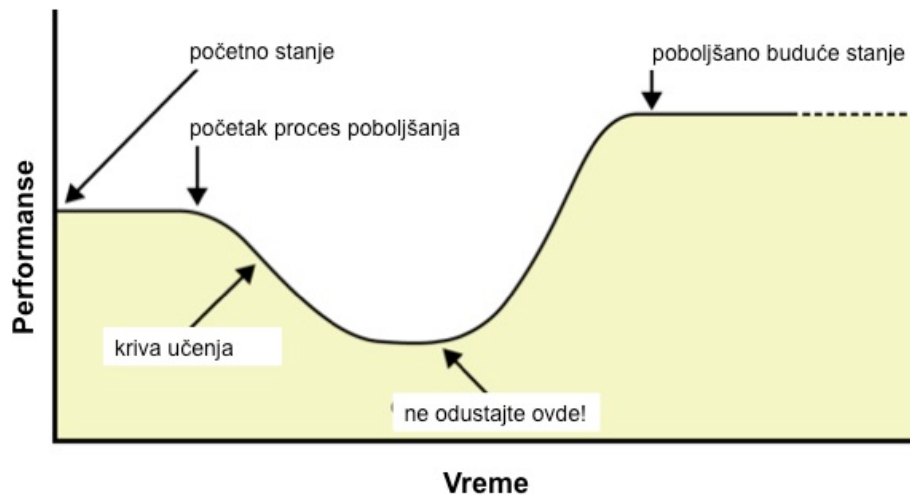
Izvršen je poslednji korak poboljšanja procesa i postignuti rezultati. Ovo ocenjivanje će pomoći timu da uradi još bolji posao na budućem poboljšanju. Procenite kako su glatko ostvarena probna rešenja (probni piloti). Koliko su bili efikasni u rešavanju nesigurnosti oko novih procesa? Da li biste nešto promenili sledeći put kada vodite pilot proces?

Razmotrimo koliko je napredovalo uvođenje novih procesa. Da li je dostupnost novih procesa ili obrazaca saopštena svima pogođenim? Da li su učesnici razumeli i uspešno primenili nove procese? Da li želite da promenite sledeći uvod?

Kritični korak je procena da li novi procesi daju željene rezultate. Neke nove prakse brzo postižu vidljiva poboljšanja, ali druge u potpunosti koriste svoju punu vrednost. Na primer, trebalo bi da budete u mogućnosti da brzo kažete da li je promena efikasna. Međutim, novom predlošku dokumenta može biti potrebno malo vremena da se iskoristi kada se poslovni analitičari i drugi akteri naviknu na njega. Dajte nove pristupe uspehu svake promene.

Ova kratkoročna produktivnost opada - koja se ponekad naziva i „dolina očaja“ - u delu ulaganja koju vaša organizacija ulaže u unapređenje procesa. Ljudi koji to ne razumeju, mogu se naći u iskušenju da odustanu od napora za poboljšanje pre nego što ono počne da se isplati, ostvarujući tako nula ili lošiji povrat svoje investicije. Educirajte svoje menadžere i vršnjake.

Prihvatajte realnost postojanja krive učenja. U početku, produktivnost pada dok se učesnici u procesu na prilagode novom načinu rada, kao što je pokazano na slici 5. Taj privremeni pad produktivnosti je investicija vaše organizacije u poboljšanje procesa. Neko može zbog ovoga da odustane od promena. Zato je vrlo važno da obrazujete menadžere i kolege o krivi učenja, da bi pridobili njihovo poverenje u promene.



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

Slika 5.5 Kriva učenja, neizbešan aspekt poboljšanja procesa

VIDEO 34 - REQUIREMENTS TRACEABILITY - WIEGERS (VIDEO)

Trajanje: 7:47 minuta

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

VIDEO 35 - RELEASE PLANNING WITH FEATURE LEVELS - WIEGERS (VIDEO)

Trajanje: 5:29 minuta

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

▼ Poglavlje 6

Vežba

PITANJA ZA DISKUSIJU

Tekst pitanja za diskusiju

PITANJE 1.

Navedite koje to doprinose praćenja zahteva mogu da osete drugi akteri u grupi za razvoj softvera. (5 min)

PITANJE 2.

Na osnovu informacija koje ste dobili na ovom predmetu, koje sve principe za unapređenje procesa razvoja softvera možete da identifikujete? (5 min)

PITANJE 3.

Izaberite jednu novu inženjersku praksu sa kojom ste se upoznali na ovom predmetu, a koju ćete se truditi da je primenite počevši od sledeće nedelje - bukvalno! Izaberite dve ili tri dodatne prakse o kojima biste želeli da saznate više. Razmislite o situacijama u kojima bi trebalo da koristite nove tehnike. Identifikujte barijere koje bi mogle da vam se nađu ne putu primene novih inženjerskih praksi i tehnika i razmislite ko bi vam mogao pomoći u razbijanju tih barijera. (10 min)

PITANJE 4.

Pogledajte matrice sledljivosti date u tekstu predavanja (poglavlje 2, slike 1 i 2). Koju matricu biste pre primenili na svojim projektima i zbog čega? (5 min)

ZADACI ZA VEŽBU

Tekst zadatka za vežbu

ZADATAK 1.

Na primeru sistema iz vežbe 13, postavite matricu praćenja zahteva (matricu sledljivosti). Možete primeniti bilo koju od predložene dve matrice u tekstu predavanja. (15 min)

ZADATAK 2.

Rekapitulacija

Na osnovu opisa sistema za naručivanje hrane iz kafeterije, datog u vežbi 13, uraditi sledeće (40 min):

1. Kreirati katalog poslovnih pravila
2. Napraviti matricu povezivanja poslovnih pravila i navedenih slučajeva korišćenja
3. Razmotriti koji atributi kvaliteta još moraju biti ispunjeni, osim navedenih, i dopisati nefunkcionalne zahteve na bazi njih
4. Nacrtati dijagram slučajeva korišćenja
5. Za svaki slučaj korišćenja nacrtati sekvencijalni dijagram
6. Nacrtati dijagram stanja za jednu porudžbinu
7. Nacrtati model podataka

▼ Poglavlje 7

Domaći zadatak

DOMAĆI ZADATAK 15

Tekst domaćeg zadatka

Prema uzorku datom u lekciji, kreirati **Akcioni plan za poboljšanje procesa razvoja zahteva** na vašem projektu koji ste radili za potrebe ovog predmeta. Uzmite iz templejta samo elemente koji su vam neophodni da dobro koncipirate i obrazložite svoj akcioni plan za poboljšanje. Uzmite u obzir tehnike i prakse koje ste naučili na ovom predmetu i kreirajte plan tako da nagovestite šta biste mogli da uradite bolje kada biste imali još jednu priliku.

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.

Pripremite se za odbranu projektnog zadatka. Osim priložene dokumentacije i modela, za potrebe odbrane možete iskoristiti i plan koji ste osmislili za DZ15.

▼ Poglavlje 9

Zaključak

ZAKLJUČAK

1. Sledljivost zahteva omogućava utvrđivanje veza sa potrebama korisnika i sa funkcijama proizvoda.
2. Sledenjem tragova, tj. veza, saznajemo sa kojim potrebama, s jedne strane, i sa kojim elementima softvera, s druge strane, je neki zahtev povezan
3. Sledenje zahteva omogućava način da se pokaže usaglašenost sa specifikacijom, ugovorom ili regulativom
4. Uspostavljanje tragova nije mnogo posla ako informacije prikupljate u toku razvoja, ali naporno je i skupo uraditi na dovršenom sistemu.
5. Matrica sledljivosti prikazuje kako je svaki funkcionalni zahtev povezan unazad za određeni slučaj upotrebe i prosleđuje se jednom ili više elemenata projektnog rešenja, koda i ispitivanja. Svaka matrice ćelija na preseku dve povezane komponente sadrži simbol koji označava vezu koje mogu da definišu odnose jedan na jedan, jedan na mnoge ili između mnogih i drugih sistema.
6. Nefunkcionalni zahtevi ne prate direktno kod. Odgovarajuće funkcionalne zahteve možete pratiti unazad prema njihovim roditeljskim nefunkcionalnim zahtevima i prosleđuju se dole proizvodima. Odredite uloge i pojedince koji bi trebalo da dostave svaku vrstu informacija u tragovima za vaš projekat.
7. Neki alati automatski označavaju vezu traga kao sumnjivu kada se objekt na bilo kojem kraju veze menja. Sumnjiva veza prikazuje vizuelni indikator u odgovarajućoj ćeliji matrice sledljivosti.
8. Promene koje unesete u svoje zahteve odraziće se na ostale procese u projektu i obrnuto. Ako ste poslovni analitičar ili rukovodilac projekata, morate naučiti aktere šta od njih očekujete da bi razvoj proizvoda uspeo.
9. Da biste umanjili strah od nepoznatog, obavestite o poboljšanju svog procesa. Objasnite prednosti koje će druge grupe dobiti od novog procesa.
10. Veliki napor za unapređenje procesa može se postići samo ako je osoba motivisana da angažuje resurse, postavi očekivanja i za doprinos inicijativi za promenama.
11. Ciklus poboljšanja softverskog procesa odražava važnost saznanja gde ste sada, pre nego što krenete dalje i vodi vas ka promenama koje treba da poboljšaju dosadašnje procese. Grupišite više akcija poboljšanja procesa razvoja zahteva u celokupni plan poboljšanja softverskog procesa.
12. Ne postavljajte previše novih očekivanja od projektnog tima odjednom, kada planirate poboljšanja procesa razvoja zahteva. Pri uvođenju novog poboljšanog procesa, u početku, produktivnost pada dok se učesnici u procesu ne prilagode novom načinu rada.

REFERENCE

Nastavni 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 29 i 30** 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,