



SE322 - INŽENJERSTVO ZAHTEVA

Potvrđivanje zahteva

Lekcija 13

PRIRUČNIK ZA STUDENTE

SE322 - INŽENJERSTVO ZAHTEVA

Lekcija 13

POTVRĐIVANJE ZAHTEVA

- ✓ Potvrđivanje zahteva
- ✓ Poglavlje 1: Proveravanje i potvrđivanje zahteva
- ✓ Poglavlje 2: Pregled zahteva
- ✓ Poglavlje 3: Izazovi pregleda zahteva
- ✓ Poglavlje 4: Primena prototipova sa zahtevima
- ✓ Poglavlje 5: Testiranje zahteva
- ✓ Poglavlje 6: Provera zahteva testovima prihvatanja
- ✓ Poglavlje 7: Vežba
- ✓ Poglavlje 8: Domaći zadatak
- ✓ Poglavlje 9: 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

Većina programera softvera doživela je frustraciju što su im predstavljeni zahtevi koji su bili dvosmisleni ili nepotpuni. Ako dobiju potrebne informacije, programeri moraju da naprave sopstvene interpretacije, koje su uvek tačne. Kao što ste videli u lekciji 1, „Suštinski softverski zahtev“, ispravljanja greške zahteva nakon primene košta mnogu više, nego ispravljanja pronađene tokom razvoja zahteva. Jedna studija otkrila je da je u proseku trebalo 30 minuta da se utvrdi kvar otkriven tokom faze zahteva. Suprotno tome, potrebno je 5 do 17 sati da se ispravi kvar otkriven tokom testiranja sistema. Jasno je da će sve mere koje preduzmete da otkrijete greške u specifikacijama zahteva uštedeti vreme i novac.

Na mnogim projektima testiranje je aktivnost u kasnoj fazi. Problemi u vezi sa zahtevima ostaju u proizvodu dok se konačno ne otkriju kroz dugotrajno testiranje sistema ili još gore od strane krajnjeg korisnika. Ako započnete planiranje testiranja i razvoj test slučajeva paralelno sa razvojnim zahtevima, naći ćete mnoge greške ubrzo nakon uvođenja. To ih sprečava da naprave dalju štetu i minimizira troškove vašeg razvoja i održavanja.

Učesnici projekta ponekad nerado troše vreme na preispitivanje i testiranje zahteva. Njihova intuicija im govori da bi ubacivanje vremena u raspored za poboljšanje zahteva u kvalitetu odložilo planirani datum isporuke za isto trajanje. Međutim, ovo očekivanje pretpostavlja nulti povraćaj ulaganja u validaciju (proveru) zahteva. U stvarnosti, ta investicija može zapravo skratiti raspored isporuke smanjenjem potrebnih prepravki i ubrzavanjem integracije i testiranja sistema. Bolji zahtevi dovode do većeg kvaliteta proizvoda i zadovoljstva kupaca, što smanjuje životni vek proizvoda za održavanje, unapređenje i korisničku podršku.

Ulaganje u zahteve za kvalitetom obično vam štedi mnogo više nego što trošite.

Različite tehnike mogu vam pomoći da procenite tačnost i kvalitet svojih zahteva. Jedan od pristupa je kvantifikovanje svakog zahteva, tako da možete smisliti način da izmerite koliko predloženo rešenje zadovoljava. Ova lekcija govori o tehnikama provere (validacije) formalnih i neformalnih pregleda zahteva, razvoju testova zahteva i klijentima koji definišu kriterijume prihvatanja proizvoda.

UVODNI VIDEO

Trajanje video snimka: 3min 48sek

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

▼ Poglavlje 1

Proveravanje i potvrđivanje zahteva

VIDEO PREDAVANJE ZA OBJEKAT "PROVERAVANJE I POTVRĐIVANJE ZAHTEVA"

Trajanje video snimka: 29min 4sek

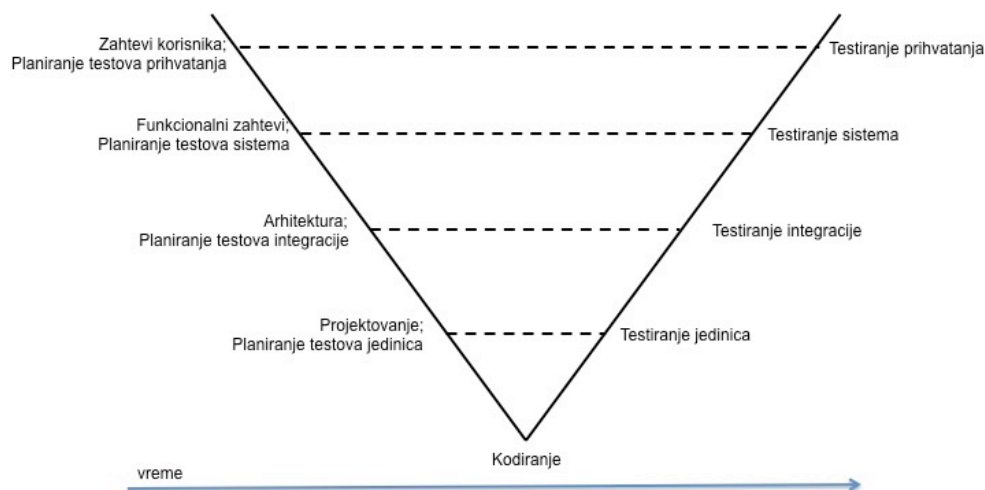
Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.

V MODEL RAZVOJA SOFTVERA

V model razvoja softvera pokazuje ispitne aktivnosti koje započinju paralelno sa odgovarajućim razvojnim aktivnostima

Slika 1 prikazuje V model razvoja softvera. Pokazuje ispitne aktivnosti koje započinju paralelno sa odgovarajućim razvojnim aktivnostima. Ovaj model ukazuje da su testovi prihvatanja izvedeni iz potreba korisnika, testovi sistema zasnovani su na funkcionalnim zahtevima, a testovi integracije zasnovani su na arhitekturi sistema. Ovaj model je primenljiv bez obzira da li su aktivnosti razvoja softvera koji se testiraju odnose na proizvod u celini, posebno izdanje, ili na pojedinačni priraštaj razvoja.

Kao što ćemo kasnije raspravljati u ovom poglavlju, testovi možete koristiti za proveru (validaciju) svake od ovih vrsta zahteva tokom razvoja zahteva. Verovatno zapravo izvršavate bilo kakve testove tokom razvoja zahteva jer još uvek imate bilo koji pokrenut softver.



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

Slika 1.1 V model razvoja softvera sadrži rana planiranja testova i projektovanje testova

VALIDACIJA (PROVERA) I VERIFIKACIJA (POTVRĐIVANJE) ZAHTEVA

Verifikacija (potvrđivanje) određuje da li ste pravilno napisali zahteve. Validacija (provera) zahteva procenjuje da li ste napisali ispravne zahteve.

Provera zahteva (requirenebts validation) je četvrta komponenta razvoja zahteva, zajedno sa zahtevima, analizama i specifikacijama. Termin "provera zahteva" se često i na srpskom naziva i "validacija zahteva", te ćemo koristiti oba termina koji se odnose na istu stvar.

Verifikovanje zahteva, s druge strane, je aktivnost kojom se želi da potvrdi (verifikuje) da su ostvarena sva željena svojstva zahteva visokog kvaliteta.

Precizno gledano, **validacija (provera)** i **verifikacija (potvrđivanje)** su dve različite aktivnosti u razvoju softvera. **Verifikacija određuje da li proizvod neke razvojne aktivnosti ispunjava njegove zahteve** (radi kako je propisano). **Validacijom se procenjuje da li proizvod zadovoljava potrebe kupaca** (radi pravu stvar). Proširivanje ovih definicija na zahteve, **verifikacija određuje da li ste pravilno napisali zahteve**: vaši zahtevi imaju poželjna svojstva opisana u Lekciji 7, poglavlju „Karakteristike odličnih zahteva.“ **Validacija zahteva procenjuje da li ste napisali ispravne zahteve**: da li su oni u saglasnosti sa poslovnim ciljevima. Ova dva koncepta su usko povezana. Radi jednostavnosti u ovom poglavlju govorimo o validaciji zahteva, ali tehnike koje opisujemo doprinose kako pravilnim zahtevima tako i visokim zahtevima kvaliteta.

Zahtevi provere (validacije) omogućavaju timovima da izgrade ispravno rešenje koje ispunjava navedene poslovne ciljeve.

.Aktivnosti **validacije** zahteva pokušavaju da osiguraju da:

- Softverski zahtevi precizno opisuju predviđene systemske mogućnosti i svojstva koja će udovoljiti različitim akterima.
- Softverski zahtevi tačno proizilaze iz poslovnih zahteva, systemskih zahteva, poslovnih pravila i drugih izvora.
- Zahtevi su potpuni, izvodljivi i proverljivi.
- Svi zahtevi su neophodni, a ceo set je dovoljan da ispuni poslovne ciljeve.
- Sva predstavljanja zahteva su u skladu jedno sa drugim.
- Zahtevi pružaju adekvatnu osnovu za nastavak projektovanja i izgradnje

Validacija je pojedinačna diskretna faza koju izvodite nakon dobijanja i dokumentovanja svih zahteva. Neke aktivnosti validacije, kao što su inkrementalni pregledi postavljenih rastućih zahteva, svrstane su u iterativne postupke izazivanja, analize i specifikacije. Ostale aktivnosti, kao što su formalne inspekcije, pružaju krajnji korak kvaliteta pre zasnivanja skupa zahteva. U svoj projektni plan uključite aktivnosti validacije zahteva. Naravno, možete potvrditi samo zahteve koji su dokumentovani, a ne implicitne zahteve koji postoje samo u nečijem umu.

▼ Poglavlje 2

Pregled zahteva

INSPEKCIJSKI PROCES

Recenzija daje izveštaj koji identifikuje ispitivani materijal, recenzente i ocenu tima za pregled da li su zahtevi prihvatljivi. Najbolje utvrđena vrsta formalne recenzije naziva se inspekcijom.

Formalne recenzije prate dobro definisan proces. Formalni pregled zahteva (formal peer review), tj. recenzija daje izveštaj koji identifikuje ispitivani materijal, recenzente i ocenu tima za pregled da li su zahtevi prihvatljivi. Glavni rezultat je rezime utvrđenih nedostataka i problema koji su se pojavili tokom pregleda. Članovi formalnog tima za pregled dele odgovornost za kvalitet recenzije, iako su autori na kraju odgovorni za kvalitet rezultata koji stvaraju.

Najbolje utvrđena vrsta formalne recenzije naziva se **inspekcijom**. Inspekcija dokumentacije o zahtevima jedna je od najkvalitetnijih tehnika kvaliteta softvera na raspolaganju. Nekoliko kompanija je izbeglo čak 10 sati rada za svaki sat koji su uložene u uvid u dokumentaciju o zahtevima i druge softverske isporuke (Gradi i Van Slack 1994). Na hiljadu procenata povrata investicije se ne uskraćuje.

Ako se ozbiljno bavite maksimiziranjem kvaliteta vašeg softvera, vaši timovi će pregledati većinu njihovih zahteva. Detaljna inspekcija velikog skupa zahteva je zamorna i dugotrajna. Ipak, praksa je pokazala da "je svaki minut inspekcije vredan".

Ako imate vremena da sve pregledate, koristite analizu rizika da biste razlikovali one zahteve koji zahtevaju inspekciju od manje kritičnog, manje složenog ili manje novog materijala za koji će biti dovoljan neformalni pregled. Inspekcije nisu jeftine. Nisu ni toliko zabavni. Ali oni su jeftiniji - i zabavniji - od alternative trošenju puno truda i problema u popravljanju dobre volje kupca koji su mnogo kasnije otkriveni

Svaki softverski proizvod može se pregledati, uključujući zahteve, projektna dokumentacija, izvorni kod, test dokumentacija i projektni planovi.

Inspekcija je dobro definisan višefazni proces. To uključuje mali tim učesnika koji pažljivo ispituju proizvod rada radi nedostataka i mogućnosti poboljšanja. Inspekcije služe kao kapija kvaliteta kroz koju moraju proći rezultati projekta prije nego što se primene. Postoji nekoliko oblika inspekcije, ali svaki od njih je moćna tehnika kvaliteta.

UČESNICI PROCESA INSPEKCIJE

Učesnici inspekcije trebalo bi da predstavljaju četiri perspektive: autora proizvoda, korisnike sistema, predstavnike razvojnog tima i ljude koji će pove

Uverite se da imate sve potrebne ljude na inspekcijskom sastanku pre nego što nastavite. U suprotnom, možete ispraviti probleme samo da biste kasnije saznali da se neko važan ne slaže sa promenom. Učesnici inspekcije trebalo bi da predstavljaju četiri perspektive:

- **Autor radnog proizvoda i možda vršnjaci autora** Poslovni analitičar koji je napisao dokument sa zahtevima pruža tu perspektivu. Ako možete, uključite još jednog iskusnog BA jer on zna koje vrste grešaka u pisanju zahteva da traži.
- **Ljudi koji su izvor informacija koji se ubacuju u predmet koji se pregledava** Ovi učesnici mogu biti stvarni predstavnici korisnika ili autori prethodne specifikacije. U nedostatku specifikacije višeg nivoa apstrakcije, inspekcija mora da uključi predstavnike kupaca, kao što su šampioni proizvoda, kako bi se osiguralo da zahtevi pravilno i potpuno opisuju njihove potrebe.
- **Ljudi koji će obavljati posao na osnovu materijala koji se pregledava** Za SRS možete uključiti programera, testera, menadžera projekata i pisca dokumentacije za korisnika, jer će otkriti različite vrste problema. Tester će najverovatnije ukazati na zahtev koji se ne može verifikovati; programer može uočiti zahteve koji su tehnički neizvodljivi.
- **Ljudi koji su odgovorni za povezivanje sistema na koje će uticati materijal koja se pregledava** Ovi inspektori će potražiti probleme sa zahtevima spoljnog interfejsa. Takođe mogu da primete efekte talasanja, pri čemu promena zahteva u SRS koji se pregledava utiče na druge sisteme.

Pokušajte da ograničite tim na sedam ili manje inspektora. To može značiti da su neke perspektive predstavljene u svakoj inspekciji. Veliki timovi se lako zamaraju u sporedne diskusije, rešavanje problema i rasprave oko toga da li je nešto zaista greška. To smanjuje brzinu kojom oni pokrivaju materijal tokom inspekcije i povećava troškove pronalaženja svakog oštećenja.

Rukovodilac autora obično ne treba da prisustvuje inspekcijskom sastanku, osim ako rukovodilac aktivno doprinosi projektu i ako je njegovo prisustvo prihvatljivo za autora. Efikasna inspekcija koja otkriva mnoge nedostatke može stvoriti loš utisak o autoru kod hiperkritičnog rukovodioca autora. Takođe, prisustvo rukovodioca može da uguši diskusiju drugih učesnika procesa inspekcije.

ULOGE U PROCESU INSPEKCIJE

U procesu inspekcije, učesniku obavljaju sledeće uloge: autor, moderator, čitalac i diktafon.

Svi učesnici inspekcije, uključujući autora, traže nedostatke i mogućnosti poboljšanja. Neki od članova inspeksijskog tima obavljaju sledeće posebne uloge tokom inspekcije:

Autor: Autor je kreirao ili održava proizvod pod inspekcijom. Autor dokumenta sa zahtevima je obično poslovni analitičar koji je iskazao potrebe klijenta i napisao zahteve. Tokom neformalnih pregleda kao što su uputstva za upotrebu, autor često vodi diskusiju. Međutim, autor preuzima više pasivnih uloga tokom inspekcije. Autor ne bi trebalo da preuzme nijednu drugu dodeljenu ulogu - moderatora, čitača ili diktafona. Kako nema aktivnu ulogu, autor može slušati komentare drugih inspektora, odgovarati - ali ne raspravljati - na njihova pitanja i razmišljati. Na ovaj način autor često može uočiti greške koje drugi inspektori često ne vide.

Moderator: Moderator planira inspekciju s autorom, koordinira aktivnosti i olakšava inspeksijski sastanak. Moderator deli materijale koje treba pregledati, zajedno sa svim relevantnim dokumentima prethodnika, učesnicima nekoliko dana pre sastanka inspekcije. Odgovornosti moderatora uključuju početak sastanka na vreme, podsticanje doprinosa svih učesnika i održavanje sastanka usredsređeno na pronalaženje glavnih nedostataka umesto na rešavanje problema ili ometanje sitnih stilskih pitanja i pogrešaka pri upisu. Moderator prati predložene izmene sa autorom kako bi se osiguralo da su pitanja koja su proizišla iz inspekcije pravilno rešena.

Čitalac (Reader): Jednom inspektoru dodeljena je uloga čitača. Tokom inspeksijskog sastanka, čitalac parafrazira zahteve i elemente modela koji se ispituju jedan po jedan. Ostali učesnici tada ističu potencijalne nedostatke i probleme koje vide. Izjašnjavanjem zahteva sopstvenim rečima, čitalac pruža tumačenje koje se može razlikovati od onog koji imaju drugi inspektori. Ovo je dobar način da se otkrije nejasnoća, mogući kvar ili pretpostavka. Takođe se naglašava vrednost postojanja čitaoca osim autora. U manje formalnim vrstama recenzija, uloga čitatelja je izostavljena, a moderator vodi ekipu kroz radni proizvod i prikuplja komentare na jednu sekciju.

Snimač (Recorder): Snimač, tj. zapisničar, koristi standardne obrasce za dokumentovanje postavljenih problema i nedostataka koji su pronađeni tokom sastanka. Snimač treba da pregleda ono što je napisao da bi potvrdio njegovu tačnost (dizajniranjem ili deljenjem u veb konferenciji). Ostali inspektori trebalo bi da pomognu snimaču da shvati suštinu svakog izdanja na način koji jasno saopštava autoru lokacije i prirodi izdanja kako bi mogao da ga efikasno i korektno adresira.

PROCES INSPEKCIJE I KRITERIJUMI ZA ULAZ U PROCES

Kriterijumi za ulazak postavljaju jasna očekivanja od autora koje treba da prate tokom priprema za inspekciju

Kriterijumi za ulazak u proces inspekcije:

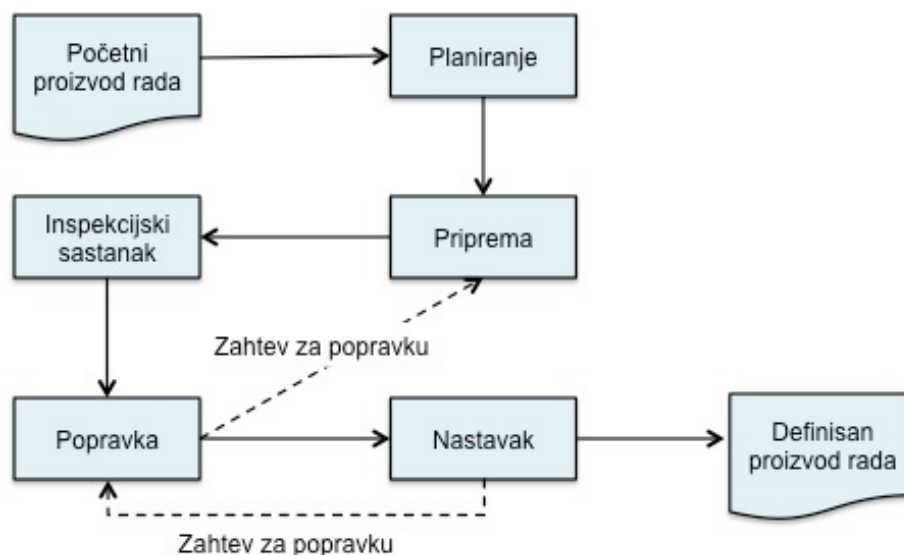
Spremni ste da pregledate dokument sa zahtevima kada ispunjava posebne preduslove. Ovi kriterijumi za ulazak postavljaju jasna očekivanja od autora koje treba da prate tokom priprema za inspekciju. Oni takođe drže inspeksijski tim da ne troši vreme na pitanja koja bi trebalo da budu rešena pre inspekcije. Moderator koristi kriterijume za unos kao kontrolnu

listu pre nego što odluči da nastavi sa inspekcijom. Slede neki predloženi kriterijumi za ulazak u inspekciju za dokumentaciju o zahtevima:

- Dokument je u skladu sa standardnim predloškom i ima očigledna pravopisna, gramatička ili formatiranja.
- Brojevi linija ili drugi jedinstveni identifikatori se štampaju na dokumentu u svrhu boljeg pozivanja na određene lokacije.
- Sva otvorena izdanja su u alatu za praćenje problema označena kao TBD (dostupan) ili dostupan.
- Moderator je pronašao više od tri velike nedostatke u desetominutnom ispitivanju reprezentativnog uzorka dokumenta.

Faze inspekcije

Inspekcija je postupak u više koraka, prikazan na slici 1. Možete istovremeno pregledati male skupove zahteva - možda one dodeljene određenoj razvojnoj iteraciji - koji ponekad mogu pokriti potpunu kolekciju zahteva.



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

Slika 2.1 Proces inspekcije sa više koraka ili aktivnosti.

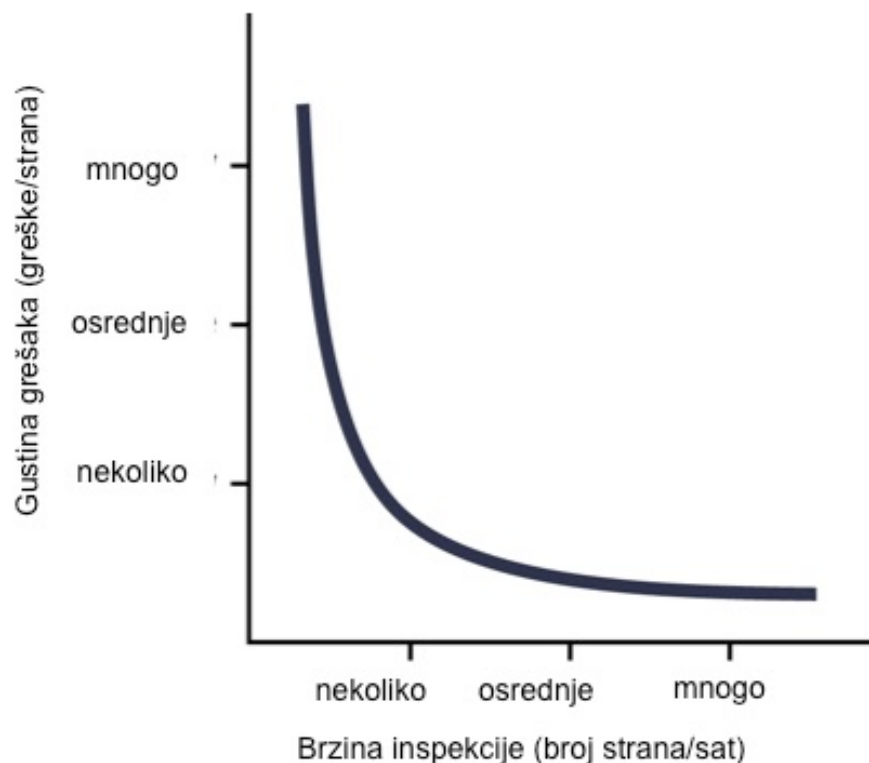
FAZA PLANIRANJA

Broj pregledanih stranica u satu ima veliki uticaj na to koliko nedostataka je pronađeno. Dve do četiri stranice na sat su praktične smernice.

Planiranje: Autor i moderator planiraju inspekciju zajedno. Oni određuju ko bi trebalo da učestvuje, koje materijale inspektori treba da dobiju pre inspeksijskog sastanka, ukupno vreme sastanka potrebno za pokrivanje materijala i kada treba da bude zakazana inspekcija. Broj pregledanih stranica u satu ima veliki uticaj na to koliko nedostataka je pronađeno Kao što prikazuje slika 2, postupanje kroz dokument sa zahtevima otkriva većinu nedostataka. Budući da nijedan tim nema na raspolaganju beskonačno vreme za inspeksijske zahteve,

odaberite odgovarajuću brzinu pregleda zasnovanu na riziku predviđanja glavnih oštećenja. Dve do četiri stranice na sat su praktične smernice, mada je optimalna brzina za maksimalno otkrivanje oštećenja otprilike polovina te brzine. Prilagodite ovu stopu na osnovu sledećih faktora:

- Prethodni podaci o inspekciji tima, koji pokazuju efikasnost inspekcije kao funkciju brzine
- Količina teksta na svakoj stranici
- Složenost zahteva
- Verovatnoća i uticaj grešaka ostaju neotkriveni
- Koliko je kritičan materijal koji treba da se pregleda je uspeh projekta
- Nivo iskustva osobe koja je napisala zahteve



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

Slika 2.2 Broj pronađenih defekata u zavisnosti od brzine inspekcije

SASTANAK INSPEKCIJE, PONOVDNA OBRADA I PRAĆENJE

Svrha sastanka inspekcije je identifikovati što više glavnih nedostataka.

Sastanak inspekcije Tokom inspekcijskog sastanka, čitač vodi druge inspektore kroz dokument, opisujući jedan uslov po jedan sopstvenim rečima. Dok inspektori postavljaju moguće nedostatke i druga pitanja, snimač ih zapisuje u spisak radnji za autora zahteve. Svrha sastanka je identifikovati što više glavnih nedostataka. Sastanak inspekcije ne bi

trebalo da traje više od dva sata; umorni ljudi su često efikasni inspektori. Ako vam treba više vremena da pokrijete sav materijal, zakažite dodatne sastanke.

Nakon što pregleda sav materijal, tim odlučuje da li će prihvatiti dokument sa zahtevima kakav je, prihvatiti ga uz manje revizije ili navesti da je potrebna velika revizija. Ishod „potrebne velike revizije“ mogao bi sugerisati da zahtevi razvojnog procesa imaju neke nedostatke ili da BA koji je napisao zahteve treba dodatnu obuku. Razmotrite održavanje retrospektive da biste istražili kako se proces može poboljšati pre sledeće specifikacije aktivnosti. Ako su neophodne velike revizije, tim će možda izabrati da ponovo pregleda delove proizvoda koji zahtevaju široku obradu, kao što pokazuje tačkasta linija između ponovne obrade i pripreme na slici 1.

Ponekad inspektori prijavljuju samo površna i kozmetička pitanja. Pored toga, inspektori se lako upuštaju u raspravu o tome da li je problem zaista nedostatak, raspravljaju o problemima u vezi sa projektom i razmišljanju o rešenju problema. Ove aktivnosti mogu biti korisne, ali odvlače pažnju od suštinskog cilja pronalaženja značajnih nedostataka i mogućnosti poboljšanja.

Ponovna obrada: Gotovo svaka aktivnost kontrole kvaliteta otkriva neke nedostatke. Autor treba da planira da provede neko vreme u preradi zahteva nakon sastanka inspekcije. Neispravljeni nedostaci zahteva biće skupi za popravku kasnije, tako da je ovo vreme da se reše nejasnoće, uklone nejasnoće i postave temelji uspešnom razvojnem projektu.

Praćenje: U ovom zadnjem koraku inspekcije, moderator ili određeni pojedinci sarađuju s autorom kako bi osigurali da su sva otvorena pitanja rešena i da su greške ispravno ispravljene. Nadalje sledi zatvaranje postupka inspekcije i omogućava moderatoru da utvrdi da li su ispunjeni kriteriji za izlaz inspeksijskog nadzora. Sledeći korak može otkriti da su neke od modifikovanih modifikacija bile nepotpune ili nisu pravilno izvedene, što je dovelo do dodatnih prepravki, kao što je prikazano isprekidanom linijom između Nadalje i Obrade na slici 1

Izlazni kriterijumi

Vaš inspeksijski postupak treba da definiše kriterijume za izlazak koji moraju biti zadovoljeni pre nego što moderator proglasi ceo postupak inspekcije završenim. Evo nekih mogućih kriterijuma za izlaz:

- Sva pitanja koja su postavljena tokom inspekcije su rešena
- Sve promene u zahtevima i srodnim proizvodima za rad izvršene su pravilno.
- Sva otvorena pitanja su rešena ili je svako otvoreno pitanje rešeno, ciljni datum i vlasnik su dokumentovani.

KONTROLNA LISTA NEISPRAVNOSTI

Kontrolne liste skreću pažnju recenzenata na istorijski česte probleme sa zahtevima.

Da biste pomogli recenzentima da potraže tipične tipove grešaka u proizvodima koje pregledavaju, razvijte kontrolni spisak neispravnosti za svaku vrstu dokumenta koji zahtevaju vaši projekti. Takve kontrolne liste skreću pažnju recenzenata na istorijski česte probleme sa zahtevima. Kontrolne liste služe kao podsetnik. Vremenom će ljudi internalizirati predmete i tražiti prava pitanja u svakom pregledu staništa. Ovde dajemo spisak zahteva koji je izložen

u ovom predmetu. Ako napravite posebne zahteve za predstavke ili modele, možete da proširite stavke na kontrolnom spisku kako biste ih detaljnije pregledali. Zahtevi poslovanja, poput dokumenta o viziji i opsegu, mogu da garantuju sopstvenu kontrolnu listu.

Niko se ne može setiti svih stavki na dugom spisku. Ako na listi postoji više od šest ili osam stavki, recenzent će verovatno imati više prolazaka kroz materijal da bi potražio sve na listi; većina recenzenata se nije mučila. Sastavite spiskove kako biste zadovoljili potrebe vaše organizacije i modifikujte stavke tako da odražavaju probleme sa kojima se ljudi najčešće susreću u skladu sa vašim potrebama. Neke studije su pokazale da je davanje recenzentima određene odgovornosti za otkrivanje nedostataka - pružanje strukturiranih misaonih procesa ili scenarija koji će im pomoći u potrazi za određenim vrstama grešaka - efikasnije od puštanja svih recenzenata na isti spisak i nadajući se najboljem.

Kompletnost:

- Da li zahtevi obuhvataju sve poznate potrebe kupca ili sistema?
- Da li nedostaje neka potrebna informacija? Ako da, da li označena kao TBD ("to be defined")?
- Da li su potrebni algoritmi za funkcionalne zahteve definisani?
- Da li su interfejsi ka spoljnjem hardveru, softveru i komunikacijskim interfejsima definisani?
- Da li je očekivano ponašanje dokumentovano za sve predviđene greške?
- Da li zahtevi predstavljaju odgovarajuću bazu za projektovanje i testiranje softvera?
- Da li su prioriteti implementacije za svaki zahtev uključeni?
- Da li svaki zahtev u predviđenom okviru projekta, izdanja ili iteracije?

Ispravnost:

- Da li postoji sukob zahteva ili duplira druge zahteve?
- Da li je svaki zahtev jasno, sažeto, i nedvosmisleno napisan?
- Da li je svaki zahtev proverljiv testiranjem, pokazivanjem, pregledom, ili analizom?
- Da li svaka poruka o grešci jasna i razumljiva?
- Da li su svi zahtevi stvarni zahtevi, a ne rešenja ili ograničenja?
- Da li su zahtevi tehnički ostvarljivi, i primenljivi?

KONTROLNA LISTA NEISPRAVNOST (NASTAVAK)

Kontrolna lista sadrži proveru zahteva grupisanih u sledeće grupe: kompletnost, ispravnost, atributi kvaliteta, organizacija i sledljivost, i ostala itanja.

Atributi kvaliteta:

- Da li su svi ciljevi upotrebljivosti, performansi, sigurnosti, i bezbednosti dobro specificirani?
- Da li su drugi atributi kvaliteta dokumentovani i kvantifikovani, sa prihvatljivim kompromisima?

- Da li utvrđene vremenski-kritične funkcije i da li su za njih specificirani vremenski kriterijumi?
- Da li su pitanja internacionalizacije i lokalizacije adekvatno rešena?
- Da li su svi atributi kvaliteta merljivi?

Organizacija i sledljivost:

- Da li su zahtevi organizovani na logičan i dostupan način?
- Da li su reference ka ostalim zahtevima i dokumentima ispravne?
- Da li su svi zahtevi napisani na jedan konzistentan način i na odgovarajućem nivou detaljnosti?
- Da li svako zahtev jedinstveno i ispravno obeležen?
- Da li svaki funkcionalni zahtev ima trag ka svom izvoru (npr, ka zahtevima sistema ili poslovnim pravilima)?

Ostala pitanja:

- Da li nedostaje neki slučaj korišćenja ili tok procesa?
- Da li u slučajevima korišćenja nedostaju neki alternativni tokovi, izuzeci, ili informacije?
- Da li su sva poslovna pravila određena?
- Da li nedostaju neki vizualni modeli koji bi trebalo da obezbede jasnoću ili kompletnost?
- Da li su sve potrebne specifikacije izveštaja obuhvaćene i kompletne?

SAVETI ZA PREGLED ZAHTEVA

Saveti se odnose i na sprovođenje formalnih i neformalnih pregleda zahteva.

Sljedeći saveti odnose sprovođenje li neformalne ili formalne preglede na svojim projektima.

Planirajte ispitivanje: Kad od vas neko zatraži da pregledate dokument, iskušenje je započeti na vrhu stranice i čitati ga po redu. Ali to ne trebate da radite. Korisnici specifikacija zahteva neće ih čitati unapred poput knjige; ni recenzenti ne moraju. Pozovite određene recenzente da se usredsrede na određene delove dokumenata.

Započnite rano: Počnite pregledavati skupove zahteva kada su ispunjeni možda tek 10 posto, a ne kada mislite da su "gotovi". Otkrivanje velikih kvarova rano i uočavanje sistemskih problema na način na koji su uneseni zahtevi moćan je način za sprečavanje - ne samo pronaći - nedostatke.

Odredite dovoljno vremena Dajte recenzentima dovoljno vremena za obavljanje recenzija, u smislu stvarnih sati pregleda (napor) i kalendarskog vremena. Imaju i druge važne zadatke u koje se recenzija mora uklopiti.

Navedite kontekst Dajte recenzentima kontekst za dokument i možda za projekt ako ne rade svi na istom projektu. Potražite recenzente koji na temelju svog znanja mogu pružiti korisnu perspektivu. Na primer, možete znati saradnika iz drugog projekta koji ima dobro oko za pronalaženje velikih propusta u zahtevima, čak i ako nije usko upoznat s projektom.

Postavite opseg ispitivanja: Recenzentima recite koji materijal treba pregledati, gde usmeriti njihovu pažnju i koja pitanja trebaju da potraže. Predložite da koriste kontrolne liste (moguće) neispravnosti kao što je onaj opisan u prethodnom odeljku. Možda želite povećati raspoloživost i veštine tražeći od različitih recenzenata da pregledaju različite odeljke ili upotrebe različitih delova popisa.

Ograniči ponovno preispitivanje: Nemojte nikoga tražiti da pregleda isti materijal više od tri puta. Umoriće se od gledanja i neće primetiti velike probleme posle trećeg ciklusa zbog „umora recenzenata.“ Ako vam treba neko da ga pregleda više puta, označite promene kako bi se mogao usredrediti na njih.

Postavljanje prioriteta područja pregleda: Prioritet za pregled delova zahteva visokog rizika ili funkcionalnosti koji će se često koristiti. Potražite i područje zahteva za koje je već zabeleženo nekoliko problema. Može se dogoditi da ti odeljci još nisu pregledani, a ne da ne postoje problemi.

▼ Poglavlje 3

Izazovi pregleda zahteva

VIDEO PREDAVANJE ZA OBJEKAT "IZAZOVI PREGLEDA ZAHTEVA"

Trajanje video snimka: 23min 2sek

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

PREDLOZI ZA REŠAVANJE IZAZOVA U VEZI RECENZIJA ZAHTEVA

Daju zajedničke izazove sa kojima se organizacije suočavaju u pogledu preispitivanja zahteva, od kojih se neki odnose na formalne inspekcije, s predlozima kako da se reše svake od njih.

Recenzija je i tehnička i društvena aktivnost. Zamoliti neke kolege da vam kažu što nije u redu s vašim radom je naučeno - ne instinktivno - ponašanje. Potrebno je vreme da softverska organizacija ugradi recenziju u svoju kulturu. Slede neki zajednički izazovi s kojima se organizacije suočavaju u pogledu preispitivanja zahteva, od kojih se neke odnose posebno na formalne inspekcije, s predlozima kako da se reše svake od njih.

Dokumenti s velikim zahtevima: Izgledi da se temeljito ispita dokument sa zahtevima od nekoliko stotina stranica su zastrašujući. Možda ćete biti u iskušenju da preskočite pregled u potpunosti i jednostavno nastavite s izgradnjom softvera - nije mudar izbor. Čak i ako imaju dokument umerene veličine, svi bi recenzenti mogli pažljivo ispitati prvi deo, a nekoliko njih i će proučiti sredinu, ali malo je verovatno da će neko pogledati i zadnji deo.

Da biste izbegli preopterećenje tima za pregled, tokom izrade zahteva izvršite inkrementalne preglede. Utvrdite područja visokog rizika koja trebaju pažljivo pregledati pregledom i upotrebite neformalne preglede za manje rizične materijale. Zatražite od određenih recenzenata da počnu s različitih mesta u dokumentu kako bi bili sigurni da su sveže oči pregledavale svaku stranicu.

Da biste procenili trebate li zaista pregledati celu specifikaciju, ispitajte reprezentativni uzorak . Broj i vrste grešaka koje ste pronašli pomoći će vam da odredite hoće li se ulaganje u potpunu inspekciju isplatiti.

Veliki inspeksijski timovi: Mnogi učesnici projekta i kupci imaju udela u zahtevima, tako da ćete možda imati dugu listu potencijalnih učesnika za inspeksijske zahteve. Međutim,

veliki timovi za pregled povećavaju troškove pregleda, otežavaju planiranje sastanaka i imaju poteškoća u postizanju dogovora o pitanjima. Pokušajte sa sledećim pristupima da biste se suočili sa potencijalno velikim inspekcijskim timom:

- Obavezno pobrinite se da svaki učesnik pronađe nedostatke, da se ne edukuje ili da zaštiti položaj.
- Shvatite koju perspektivu (kao što je kupac, programer ili ispitivač) predstavlja svaki inspektor. Nekoliko ljudi koji predstavljaju istu zajednicu mogu objediniti svoj doprinos i poslati samo jednog predstavnika na inspekcijski sastanak.
- Uspostavite nekoliko malih timova za paralelni uvid u zahteve i kombinujte njihove liste nedostataka uklanjajući duplikate. Istraživanje je pokazalo da više inspekcijskih timova pronalazi više nedostataka u odnosu na jednu veliku grupu. Rezultati paralelnih inspekcija su prevashodno aditivni, a ne suvišni.

PREDLOZI ZA REŠAVANJE IZAZOVA U VEZI RECENZIIJA ZAHTEVA (NASTAVAK)

Jedan od preduslova formalnog sastanka za recenziju je da su učesnici pregledali materijal koji se preispituje unapred, pojedinačno identifikujući svoje početne skupove pitanja.

Geografski razdvojeni recenzenti Organizacije često grade proizvode kroz saradnju geografski raspršenih timova. To čini kritike izazovnijim. Telekonferencija ne otkriva govor tela i izraze drugih recenzenata kao što je to susret licem u lice, ali video konferencije mogu biti efikasno rešenje. Alat za veb konferencije omogućava recenzentima da osiguraju da svi gledaju isti materijal tokom diskusije.

Pregledi elektronskog dokumenta smeštenog u deljeno skladište mreže pružaju alternativu tradicionalnom sastanku za pregled. U ovom pristupu, recenzenti koriste funkcije programa za obradu teksta da bi umetnuli svoje komentare u tekst. Svaki komentar označen je inicijalima recenzenata, a svaki recenzent može videti šta su prethodni recenzenti morali da kažu. Interni alati za saradnju takođe mogu pomoći. Neki alati za upravljanje zahtevima uključuju komponente koje olakšavaju distribuirane asinhrono preglede koji ne uključuju sastanke uživo. Ako odlučite da ne održavate sastanak, priznajte da to može smanjiti efikasnost pregleda, ali svakako je bolje nego da reviziju uopšte ne izvršite.

Nepripremljeni recenzenti Jedan od preduslova formalnog sastanka za recenziju je da su učesnici pregledali materijal koji se preispituje unapred, pojedinačno identifikujući svoje početne skupove pitanja. Bez ove pripreme rizikujete da ljudi provode vreme sastanka radeći sve svoje mišljenje na licu mesta i verovatno će vam nedostajati mnoga važna pitanja.

Jedan projekat je imao SRS na 50 stranica koje je trebalo pregledati 15 ljudi, što je previše da bi bilo efikasno i efikasno. Svi su imali nedelju dana da sami pregledaju dokument i pošalju nazad autoru. Nije iznenađujuće što većina ljudi to uopšte nije gledala. Tako je vodeći BA zakazao obavezan sastanak za recenzente kako bi zajedno pregledali dokument. Projektovao je SRS na ekran, prigušio svetla i počeo da čita jedan po jedan zahtev. (U sobi se nalazilo jedno jako jarko svetlo u sredini, direktno na glavnom BA-u - razgovarajte o tome da budete u

centru pažnje!) Nekoliko sati nakon sastanka za pregled, učesnici su zevali, a njihova pažnja je bledila. Nije iznenađujuće što se stopa otkrivanja problema smanjila. Svi su čeznuli da se sastanak završi. Ovaj BA je pustio učesnike da ga napuste, sugerirajući im da dokument pregledaju sami kada bi ubrzali naredni sastanak. Sigurno da ih je dosada tokom sastanka pokrenula da rade svoj pripremni posao.

▼ Poglavlje 4

Primena prototipova sa zahtevima

SVRHA PROTOTIPOVA

Prototipovi su alati za validaciju koji prave zahteve. Sve vrste prototipova omogućavaju vam da pronađete nedostajuće zahteve pre skupljih aktivnosti kao što su razvoj i testiranje.

Teško je predočiti kako će sistem funkcionisati u određenim okolnostima samo čitanjem zahteva. Prototipovi su alati za validaciju koji prave zahteve. Omogućuju korisniku da iskusi neke aspekte o tome kakav bi bio sistem na osnovu zahteva. U lekciji 11, u delu „Smanjenje rizika prototipovanjem“ dato je više informacija o različitim vrstama prototipova i načinu na koji poboljšavaju zahteve. Ovde smo opisali kako prototipovi mogu da pomognu akterima da procene da li će proizvod izgrađen u skladu sa zahtevima ispuniti njihove potrebe i da li su zahtevi potpuni, izvodljivi i jasno saopšteni.

Sve vrste prototipova omogućavaju vam da pronađete nedostajuće zahteve pre skupljih aktivnosti kao što su razvoj i testiranje. Nešto tako jednostavno kao maketa na papiru može se koristiti za hodanje kroz slučajeve upotrebe, procese ili funkcije za otkrivanje izostavljenih ili pogrešnih zahteva. Prototipovi takođe pomažu u potvrđivanju da zainteresovane strane imaju zajedničko razumevanje zahteva. Neko bi mogao implementirati prototip zasnovan na njegovom razumevanju zahteva, samo kako bi naučio da zahtev nije bio jasan kada se procenjivači prototipa ne slažu sa njegovom interpretacijom.

Dokazani prototipovi mogu pokazati da su zahtevi izvodljivi. **Evolucijski prototipovi omogućavaju korisnicima da vide kako bi zahtevi funkcionisali kada su implementirani, da potvrde da je rezultat ono što očekuju.** Dodatni nivoi sofisticiranosti prototipa, poput simulacija, omogućavaju precizniju validaciju zahteva; međutim, izgradnja sofisticiranijih prototipa će takođe trebati više vremena.

▼ Poglavlje 5

Testiranje zahteva

PISANJE KONCEPTUALNIH TESTOVA I NJIHOVO TESTIRANJE

Testovi bi trebalo da obuhvate i normalan protok svakog slučaja upotrebe, alternativne tokove i izuzetke, koje ste prepoznali tokom ispitivanja i analize.

Testovi koji se zasnivaju na funkcionalnim zahtevima ili proizilaze iz potreba korisnika pomažu da očekivano ponašanje sistema bude opipljivo za učesnike u projektu. Jednostavan čin projektovanja testova otkriće mnoge probleme sa zahtevima mnogo pre nego što možete da izvršavate te testove na pokretanom softveru. Pisanje funkcionalnih testova kristalizira vašu viziju kako bi sistem trebalo da se ponaša u određenim uslovima. Nejasni i dvosmisleni zahtevi izbaciće vas, jer nećete moći da opišete očekivani odgovor sistema. Kada BA, programeri i kupci zajedno prođu testove, postižu zajedničku viziju kako će proizvod raditi i povećavaju uverenje da su zahtevi tačni. Testiranje je moćan alat za potvrđivanje i proveru zahteva.

Možete započeti izvođenje konceptualnih testova iz korisničkih potreba već u procesu razvoja. Koristite testove za procenu funkcionalnih zahteva, analiza modela i prototipa. Testovi bi trebalo da obuhvate i normalan protok svakog slučaja upotrebe, alternativne tokove i izuzetke koje ste prepoznali tokom ispitivanja i analize. Slično tome, ako ste identifikovali tokove poslovnih procesa, testovi bi trebalo da obuhvate korake poslovnog procesa i sve moguće puteve donošenja odluka.

Ovi konceptualni testovi ne zavise od primene. Na primer, razmotrite slučaj upotrebe koji se zove „Pregled uskladištenog naloga“ za sistem praćenja hemikalija (Chemical Tracking System). Neki konceptualni testovi su:

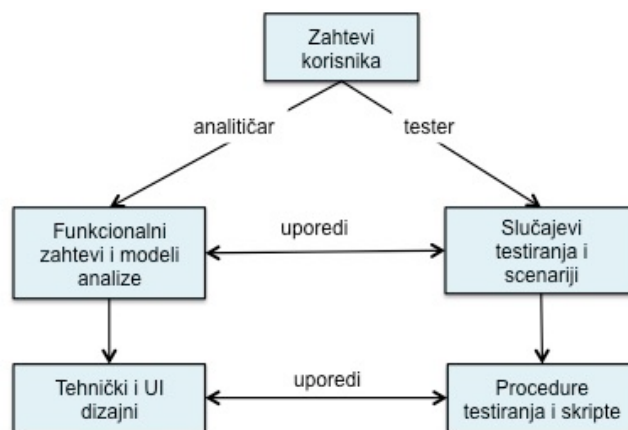
- "Korisnik unosi broj naloga da bi ga pregledao, nalog postoji, korisnik je dao porudžbinu. Očekivani rezultat: pokažite detalje narudžbine.
- Korisnik unosi broj naloga da bi ga pregledao, a nalog ne postoji. Očekivani rezultat: Prikažite poruku „Izvini, ne mogu da nađem taj nalog“.
- Korisnik unosi broj naloga da bi ga pregledao, nalog postoji, korisnik nije stavio porudžbinu. Očekivani rezultat: Prikažite poruku „Izvinite, to nije vaša narudžba“.

PARALELIZAM RAZVOJA I TESTIRANJA ZAHTEVA

BA piše funkcionalne zahteve, a tester - testove sa zajedničke početne tačke: sa korisničkih zahteva.

U idealnom slučaju, BA će napisati funkcionalne zahteve, a tester će napisati testove sa zajedničke početne tačke: korisničkih zahteva, kao što je prikazano na slici 1. Nejasnoće u potrebama korisnika i razlike u tumačenju dovešće do nedoslednosti između stavova predstavljenih funkcionalnim zahtevima, modelima i testovima. Dok programeri prevode zahteve u korisničke interfejsa i tehničke dizajne, tester (ispitivači) mogu razraditi konceptualne testove u detaljne procedure ispitivanja

Da vidimo kako je tim za sistem za praćenje hemikalija povezoao zahteve i vizuelne modele sa ranim testiranjem. Sledi nekoliko delova informacija u vezi sa zahtevima, a svi se odnose na zadatak da se zatraži hemikalija.



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

Slika 5.1 Razvoj i testiranje proizvoda rada dolaze iz istog izvora

.Zahtevi poslovanja Kao što je opisano u lekciji 3, „Uspostavljanje poslovnih zahteva“, jedan od glavnih poslovnih ciljeva za Sistem za praćenje hemikalija bio je:

"Smanjite troškove kupovine hemikalija za 25% u prvoj godini."

Slučaj upotrebe: Slučaj upotrebe koji se usklađuje sa ovim poslovnim zahtevom je „Zahtevanje hemikalije.“ Ovaj slučaj upotrebe uključuje putanju koja korisniku omogućava da zatraži hemijsku posudu koja je već dostupna u skladištu hemikalija. Evo opisa slučaja upotrebe "Zahtev hemikalije" iz lekcije 5:

"Podnosilac zahteva specificira željenu hemikaliju koju treba da unese svojim imenom ili hemijskim ID brojem ili uvoživši njenu strukturu iz hemijskog alata za crtanje. Sistem nudi Zahtevaču ili kontejner hemikalije iz skladišta hemikalija ili daje Zahtevaču da naruči jedan od prodavca."

Funkcionalni zahtev: Evo nekoliko funkcionalnosti koje proizilaze iz ovog slučaja upotrebe:

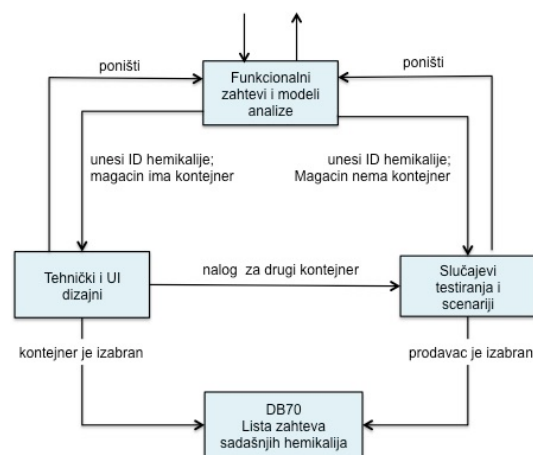
1. Ako u skladištu ima kontejnera hemikalije koja se traži, sistem će prikazati listu raspoloživih kontejnera.

2. Korisnik mora odabrati jedan od prikazanih kontejnera ili tražiti da od prodavca naruči novi kontejner.

MAPA DIJALOGA SLUČAJA KORIŠĆENJA "ZAHTEV HEMIKA LIJE"

Možete predvideti više testova kako biste se obratili normalnom toku, alternativnim tokovima i izuzecima i proveriti da li su podržani mapom dijaloga.

Mapa dijaloga: Slika 2 ilustruje deo mapa dijaloga slučaja korišćenja „Zahtevaj hemikaliju“ koji se odnosi na ovu funkciju. Okviri u ovoj dijaloškoj mapi predstavljaju prikaze korisničkog interfejsa, a strelice predstavljaju moguće navigacione puteve sa jednog ekrana na drugi. Ta dijaloška mapa je kreirana dovoljno rano u toku razvoja zahteva da su učesnici projekta počeli da identifikuju određene ekrane, menije, dijaloške okvire i druge elemente dijaloga kako bi im mogli dati imena i razmotriti moguću arhitekturu korisničkog interfejsa.



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

Slika 5.2 Deo mape dijaloga slučaja upotrebe "Zahtev hemikalije" projekta CTS

Test: Budući da ovaj slučaj upotrebe ima nekoliko mogućih puteva izvršenja, možete predvideti više testova kako biste se obratili normalnom toku, alternativnim tokovima i izuzecima. Sledi samo jedan test, zasnovan na protoku koji korisniku pokazuje raspoložive kontejnere u skladištu hemijskih proizvoda

"U dijaloškom okviru DB40 unesite važeći hemijski ID; hemijska magacin ima dve posude ove hemikalije. Pojavi se dijaloški okvir DB50, koji prikazuje dva kontejnera. Izaberite drugi kontejner. DB50 se zatvara i kontejner 2 je dodan na dnu trenutne liste zahteva za hemijskim proizvodima u dijaloškom okviru DB70."

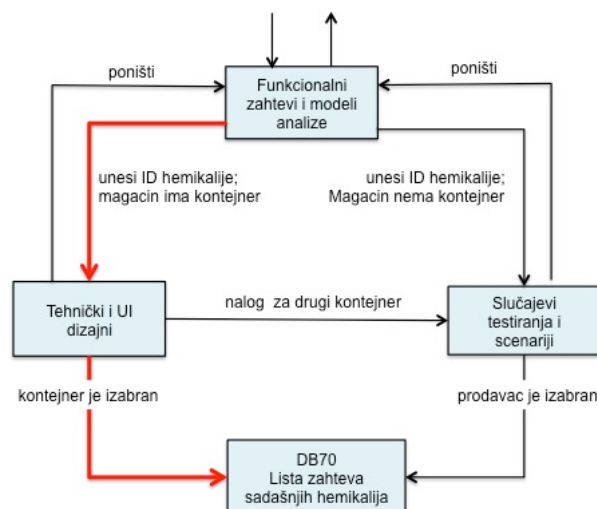
Ramesh, vodeći tester za sistem za praćenje hemikalija (Chemical Tracking System), napisao je nekoliko testova poput ovog na osnovu svog razumevanja slučaja upotrebe. **Takvi apstraktni testovi ne zavise od detalja implementacije.** Oni ne razgovaraju o unošenju podataka u određena polja, klikom na dugmad ili drugim specifičnim tehnikama interakcije.

Kako razvoj napreduje, ispitivač može doraditi takve konceptualne testove u posebne postupke ispitivanja.

PRAĆENJE PUTANJE IZVOĐENJA TESTA U MAPI DIJALOGA

Praćenjem putanje izvođenja za svaki test možete pronaći pogrešne ili nedostajuće zahteve, poboljšati korisničke navigacijske opcije i precizirati testove.

Sada dolazi zabavni deo - testiranje zahteva. Ramesh je prvo testirao svaki test prema funkcionalnim zahtevima. Proverio je da se uveri da se svaki test može „izvršiti“ prolazeći kroz set postojećih zahteva. Takođe se pobrinuo da bar jedan test pokrije svaki funkcionalni zahtev. Zatim je Ramesh, sledeći tragove, izvršio svaki test na dijaloškoj mapi markerima za označavanje. Crvena i zadebljana linija na slici 3 pokazuje kako se prethodni test prati na mapi dijaloga.



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

Slika 5.3 Prikaz traga za test u mapi dijaloga slučaja korišćenja "Zahtev hemikalije"

Praćenjem putanje izvođenja za svaki test možete pronaći pogrešne ili nedostajuće zahteve, poboljšati korisničke navigacijske opcije i precizirati testove. Pretpostavimo da nakon „izvršavanja“ svih testova na ovaj način, navigaciona linija dijaloške mape sa oznakom „naručite novi kontejner“ koja ide od DB50 do DB60 na slici 2 nije istaknuta. Postoje dve moguće interpretacije:

- Ta navigacija nije dozvoljeno ponašanje sistema. BA mora ukloniti tu liniju iz dijaloške mape. Ako SRS sadrži zahtev koji određuje prelaz, taj zahtev takođe mora biti uklonjen.
- Navigacija je legitimna, ali nedostaje test koji pokazuje ponašanje.

PROVERA POSTOJANJA NAVIGACIONE LINIJE ZA DRUGI SCENARIO

U ovim primerima BA i ispitivač kombinuju zahteve, analize modela i testove za otkrivanje nedostajućih, pogrešnih ili nepotrebnih zahteva mnogo pre nego što je bilo koji zapis napisan.

Pretpostavimo da je u drugom scenariju ispitivač napisao test na osnovu svoje interpretacije slučaja upotrebe koji kaže da *korisnik može poduzeti neku radnju kako bi se direktno prebacio iz dijaloškog okvira DB40 u DB70*. Međutim, mapa dijaloga na slici 2 ne sadrži takvu navigacionu liniju, tako da se test ne može „izvršiti“ sa postojećim postavljenim zahtevima. Opet postoje dve moguće interpretacije. Morate da utvrdite šta je od sledećeg tačno:

- Navigacija od DB40 do DB70 nije dozvoljeno ponašanje sistema, pa je test pogrešan
- Navigacija od DB40 do DB70 je legitimna, ali u dijaloškoj mapi i možda u SRS nedostaje zahtev koji se koristi testom

U ovim primerima BA i ispitivač kombinuju zahteve, analize modela i testove za otkrivanje nedostajućih, pogrešnih ili nepotrebnih zahteva mnogo pre nego što je bilo koji zapis napisan. Konceptualno testiranje softverskih zahteva snažna je tehnika za kontrolu troškova i rasporeda projekta pronalaženjem nejasnoća i grešaka u ranoj fazi igre.

▼ Poglavlje 6

Provera zahteva testovima prihvatanja

KRITERIJUMI PRIHVATANJA

Kriterijumi prihvatanja trebalo bi da procene da li proizvod ispunjava svoje dokumentovane zahteve i da li je pogodan za upotrebu u predviđenom radnom okruženju.

Programeri softvera mogu verovati da su izgradili savršen proizvod, ali kupac je konačni arbitar. Kupci moraju da procene da li sistem ispunjava svoje unapred definisane kriterijume prihvatanja. Kriterijumi prihvatanja (**acceptance criteria**) - a samim tim i testiranje prihvatanja - trebalo bi da procene da li proizvod ispunjava svoje dokumentovane zahteve i da li je pogodan za upotrebu u predviđenom radnom okruženju . Postavljanje testova prihvatanja od strane korisnika testiranja, važan je doprinos razvoju efektivnih zahteva. Što se ranije napišu testovi prihvatanja, tim pre će oni moći da pomognu timu da otkloni nedostatke u zahtevima i, na kraju, u implementiranom softveru.

Rad sa kupcima na razvoju kriterijuma prihvatanja pruža način za potvrđivanju (validaciju) i zahteva i samog rešenja. Ako kupac ne može da izrazi kako bi ocenio zadovoljstvo određenim zahtevom, taj zahtev nije dovoljno jasan. **Kriterijumi prihvatanja definišu minimalne uslove da se aplikacija smatra poslovnom spremnom.**

Razmišljanje o kriterijumima prihvatanja nudi pomak u perspektivi sa pitanja postavljanja „**Šta treba da uradite sa sistemom?**“ Na pitanje „**Kako biste procenili da li rešenje zadovoljava vaše potrebe?**“ , podstaknite korisnike da koriste SMART mnemonic - Specific, Measurable, Attainable, Relevant, and Time-sensitive- kada se definišu kriterijumi prihvatanja.

Trebalo bi navesti kriterijume tako da bi više objektivnih posmatrača došlo do istog zaključka o tome da li su zadovoljni. Kriterijumi prihvatanja zadržavaju fokus na poslovnim ciljevima zainteresovanih strana i uslovima koji bi omogućili sponzoru projekta da proglasi pobedu. Ovo je važnije od prostog pružanja specifikacije za zahteve koji zaista ne bi mogli da reše poslovne probleme zainteresovanih strana.

DIMENZIJE KRITERIJUMA PRIHVATANJA

Pri agilnom razvoju softvera, ako su ispunjeni svi kriterijumi prihvatanja za korisničku priču, vlasnik proizvoda će prihvatiti korisničku priču kao dovršenu.

Definisanje kriterijuma prihvatanja više je od puke reči da su svi zahtevi implementirani ili da su prošli svi testovi. Testovi prihvatanja predstavljaju samo niz kriterijuma prihvatanja. Kriterijumi za prihvatanje mogu takođe obuhvatiti dimenzije kao što su sledeće:

- Specifične funkcionalnosti visokog prioriteta koje moraju biti prisutne i raditi ispravno pre nego što se proizvod može prihvatiti i koristiti. (Druga planirana funkcionalnost možda će se isporučiti kasnije ili bi se mogućnosti koje ne rade sasvim ispravno mogle popraviti bez odlaganja početnog izdanja.)
- Bitni nefunkcionalni kriterijumi ili metrike kvaliteta koji moraju biti zadovoljeni. (Određeni atributi kvaliteta moraju biti barem minimalno zadovoljeni, mada bi moglo biti odloženo poboljšanje upotrebljivosti, kozmetika i podešavanje performansi. Proizvod će možda morati da zadovolji metrike kvaliteta, kao što je određeno minimalno trajanje operativne upotrebe, a da ne doživi kvar.)
- Preostala otvorena pitanja i nedostaci. (Možete odrediti da nijedna oštećenja koja prelaze određeni stepen ozbiljnosti ne ostaju otvorena u odnosu na zahteve visokog prioriteta, mada i dalje mogu biti prisutni mali greški.)
- Specifični pravni, regulatorni ili ugovorni uslovi. (Oni moraju biti u potpunosti zadovoljni pre nego što se proizvod smatra prihvatljivim.)
- Podržavanje potreba za tranzicijom, infrastrukturom ili drugim projektima (nije proizvod). (Možda će biti dostupni materijali za obuku, a konverzije podataka završene pre nego što se rešenje objavi.)

Takođe može biti dragoceno razmišljanje o „kriterijumima odbacivanja“, uslovima ili rezultatima procene koji bi zainteresovali zainteresovanog da proceni da sistem još nije spreman za isporuku. Pazite na sukobljene kriterijume prihvatanja, tako da zadovoljavanje jednog može blokirati zadovoljstvo drugog. U stvari, rano traženje sukobljenih kriterijuma prihvatanja način je da se otkriju sukobljeni zahtevi.

Projekti agilnog razvoja softvera stvaraju kriterijume prihvatanja na osnovu korisničkih priča.

"Kriterijumi prihvatanja nisu funkcionalni ili jedinični testovi; pre su oni uslovi zadovoljstva koji se postavljaju u sistem. Funkcionalni i jedinični testovi idu mnogo dublje u testiranju svih funkcionalnih tokova, izuzetaka, graničnih uslova i srodne funkcionalnosti povezane s pričom."

U principu, ako su ispunjeni svi kriterijumi prihvatanja za korisničku priču, vlasnik proizvoda će prihvatiti korisničku priču kao dovršenu. Stoga bi kupci trebali biti veoma specifični u pisanju kriterija prihvatanja koji su im važni.

TESTOVI PRIHVATANJA

Kreatori testova prihvatanja treba da razmotre najčešće izvedene i najvažnije scenarije upotrebe prilikom odlučivanja o tome kako oceniti prihvatljivost softvera.

Testovi prihvatanja (acceptance tests) predstavljaju najveći deo kriterijuma prihvatanja. Kreatori testova prihvatanja treba da razmotre najčešće izvedene i najvažnije scenarije upotrebe prilikom odlučivanja o tome kako oceniti prihvatljivost softvera. Usredsredite se na testiranje normalnih tokova slučajeva upotrebe i njihovih odgovarajućih izuzetaka, posvećujući manje pažnje manje često korišćenim alternativnim tokovima.

Agilni razvojni pristupi često stvaraju testove prihvatanja umesto pisanja preciznih funkcionalnih zahteva. Svaki test opisuje kako treba da funkcioniše korisnička priča u izvršnom softveru. Budući da oni uveliko zamenjuju detaljne zahteve, testovi prihvatanja agilnog projekta treba da obuhvate sve scenarije uspeha i neuspeha. Vrednost pisanja testova prihvatanja je ono što usmerava korisnike da razmišljaju o tome kako će se sistem ponašati nakon implementacije. Problem sa pisanjem samo testova prihvatanja je što zahtevi postoje samo u glavama ljudi. Ako ne dokumentujete i ne upoređujete alternativne prikaze zahteva - zahteve korisnika, funkcionalni zahtevi, modeli analize i testovi - možete propustiti priliku za prepoznavanje grešaka, nedoslednosti i nedostataka.

Automatizirajte izvršavanje testova prihvatanja kad god je to moguće. To olakšava ponavljanje testova kada su uvedene promene i dodana funkcionalnost u budućim iteracijama ili izdanjima. Testovi prihvatanja takođe moraju da se odnose na nefunkcionalne zahteve

Oni bi trebali osigurati da su postignuti ciljevi performansi, da je sistem u skladu sa standardima upotrebljivosti i da su ispunjena sigurnosna očekivanja.

Neka testiranje prihvatanja mogu korisnici da obavljaju i ručno. Testovi koji se koriste u testiranju prihvatanja od strane korisnika (UAT) trebalo bi da se izvrše nakon što se veruje da je skup funkcionalnosti spreman za puštanje. To omogućava korisnicima da se upoznaju sa radnim softverom pre nego što se zvanično isporuči i omogućava korisnicima da se upoznaju sa novim softverom. Kupac ili šampion proizvoda treba odabrati testove za UAT koji predstavljaju područja sa najvišim rizikom u sistemu. Testovi prihvatanja potvrdiće da rešenje čini ono što bi trebalo. Obavezno podesite ove testove koristeći verodostojne podatke testa. Pretpostavimo da podaci ispitivanja koji se koriste za generisanje izveštaja o prodaji nisu realni za aplikaciju. Korisnik koji obavlja UAT može pogrešno prijaviti kvar samo zato što izveštaj ne izgleda ispravno prema njemu ili može propustiti pogrešan proračun jer su podaci neverovatni.

Zahtevi za pisanje nisu dovoljni. Morate biti sigurni da su oni pravi zahtevi i da li su dovoljno dobri da služe kao osnova za dizajn, izgradnju, testiranje i upravljanje projektima. Planiranje testova prihvatanja, neformalne recenzije, inspekcije i tehnike testiranja zahteva pomoći će vam da brže i jeftinije izgradite sisteme visokog kvaliteta nego ikad pre.

VIDEO 22 - REVIEWING REQUIREMENTS - WIEGERS (VIDEO)

Trajanje: 7:32 minuta

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

▼ Poglavlje 7

Vežba

KONTROLNA LISTA NEISPRAVNOSTI

U nastavku vežbe je dat opis sistema za naručivanje hrane u kafeteriji, radi provere ispravnosti specificiranih zahteva. Zatim su navedene očekivane funkcije

Sistem za naručivanje hrane u kafeteriji je nov softver koji zamenjuje trenutni ručni i telefonski proces naručivanja i preuzimanja hrane/pića iz jedne kafeterije. Akteri sistema su:

1. Klijent
2. Osoblje kafeterije
3. Menadžer menija
4. Isporučilac hrane

Očekuje se da se sistem razvija u nekoliko faza, a na kraju i poveže sa servisima za plaćanje putem kreditnih i debitnih kartica i sistemom za autorizaciju.

Korisnici sistema:

Klijent je osoba koja želi da naruči hranu iz kafeterije. Postoji oko 600 potencijalnih klijenata i očekuje se da 300 njih koriste sistem, u proseku pet puta u toku svake nedelje. Klijenti će ponekad naručivati više jela za grupne događaje ili goste. Oko 60% porudžbina biće plasirano korišćenjem intraneta na nivou same kafeterije dok će 40% narudžbina biti plasirano od kuće ili korišćenjem aplikacija za pametne telefone i tablete.

Osoblje kafeterije su zaposleni kojih ima oko 20, treba da primaju narudžbine preko sistem, pakuju ih za dostavu i iniciraju dostavu. Većina osoblja treba da prođe trening za korišćenje hardvera i softvera za sistem.

Menadžer menija je zaposleni u kafeteriji i on postavlja dnevni meni hrane koji je raspoloživ tog dana u kafeteriji. Neke stavke menije mogu biti nedostupne za dostavu. Menadžer menija takođe definiše dnevne specijalitete. Menadžer menija treba periodično da vrši izmene menija.

Isporučilac hrane je onaj koji preuzima hranu i dostavlja je klijentu. Kada osoblje kafeterije primi narudžbinu, zahtev se šalje dostavljaču hrane putem pametnog telefona. Isporučilac hrane u interakciji sa sistemom izveštava da li je jelo dostavljeno ili ne.

PRETPOSTAVKE, ZAVISNOSTI I FUNKCIONALNOSTI

Date su pretpostavke, zavisnosti i funkcionalnosti sistema za naručivanje hrane u kafeteriji radi provere ispravnosti specificiranih zahteva.

Pretpostavke i zavisnosti

AS-1: Kafeterija je otvorena za doručak, ručak i svaki radni dan kada se očekuje da svaki zaposleni bude na svom radnom mestu.

DE-1: Operacije u sistemu zavise od izmena napravljenih u platnom sistemu uz prihvatanje zahteva za plaćanje za jela naručena korišćenjem sistema.

DE-2: Operacije u sistemu zavise od izmena napravljenih u inventaru kafeterije i raspoložive hrane.

Neke od izdvojenih i dogovorenih **funkcionalnosti** u prethodnim aktivnostima inženjeringa zahteva su:

FE-1. Naručivanje hrane iz kafeterije (naručivanje više jela i proizvoda)

FE-2. Dostavljanje hrane

FE-3. Pregled menija

FE-4. Potvrda narudžbine i plaćanje narudžbine

FE-5. Završavanje narudžbine

FUNKCIONALNI ZAHTEVI SISTEMA ZA NARUČIVANJE HRANE U KAFETERIJI

Dati su funkcionalni zahtevi sistema za naručivanje hrane u kafeteriji radi provere ispravnosti specificiranih zahteva.

FE-1. Naručivanje hrane iz kafeterije

REQ1. Klijent nakon pregledavanja menija bira hranu i piće koje želi da naruči

REQ2. Ako klijent nije registrovan, sistem treba da ponudi opciju za registrovanje i omogući nastavak naručivanja, uz obavezno preuzimanje hrane u kafeteriji (bez dostave) ili izlazak iz aplikacije

REQ3. Sistem treba da dozvoli korisniku da naruči više identičnih jela ukoliko su na meniju i ukoliko ih ima u inventaru kafeterije. Ukoliko narudžbine klijenta vezane za neku hranu ili proizvod prelaze raspoložive zalihe u inventaru kafeterije, sistem treba da obavesti korisnika da nije moguće naručiti hranu u tolikom broju.

REQ4. Ako je služenje i dostava određenog jela istekla ili kafeterija više ne radi sistem treba da informiše klijenta da je kasno da pošalje narudžbinu za taj dan. Klijent može promeniti datum dostave ili otkazati narudžbinu.

FE-2. Pregled menija

REQ1. Sistem treba da prikaže meni za datum koji klijent želi da odabere

REQ2. Meni za specifičan datum treba da bude prikazan jedino za onu hranu koja je na raspolaganju u kafeterijom inventaru i koja može biti dostavljena.

FE-3. Dostavljanje hrane. Lično preuzimanje hrane

REQ1. Klijent treba da odredi da li želi dostavu hrane ili preuzimanje u restoranu.

REQ2. Ako porudžbina treba da bude dostavljena i postoje slobodni termini za dostavu za taj dan, klijent treba da da validnu lokaciju za dostavu.

REQ3. Sistem treba da obavesti klijenta ukoliko ne postoje slobodni termini za dostavu hrane. Klijent može da otkaže narudžbinu ili da naglasi da će hranu preuzeti u kafeteriji.

REQ4. Sistem treba da prikaže preostale termine za dostavu za taj dan.

REQ5. Takođe, treba da omogući klijentu da odabere jedan od termina ili da promeni detalje dostave i preuzme hranu u kafeteriji ili da otkaže narudžbinu.

FUNKCIONALNI ZAHTEVI SISTEMA ZA NARUČIVANJE HRANE U KAFETERIJI - NASTAVAK

Dati su funkcionalni zahtevi sistema za naručivanje hrane u kafeteriji radi provere ispravnosti specificiranih zahteva - nastavak

FE-4. Potvrda narudžbine

REQ1. Kada klijent završio naručivanje jela i proizvoda, sistem treba da prikaže koja je hrana naručena i njihove individualne cene kao i koliko je potrebno platiti za kompletnu narudžbinu.

REQ2. Sistem treba da omogući klijentu da potvrdi narudžbinu.

REQ3. Klijent treba da potvrdi, izmeni ili otkaže narudžbinu.

REQ4. Sistem treba da ponudi meni za naredni dan.

FE-5. Završavanje narudžbine

Kada klijent potvrdi narudžbinu sistem treba da uradi sledeću transakciju

REQ1. Dodeljivanje sledećeg slobodnog broja porudžbini i promenu statusa porudžbine u "Prihvaćeno"

REQ2. Slanje poruke sistemu inventara kafeterije sa količinom potrebne hrane za porudžbinu.

REQ3. Izmena menija u skladu sa porudžbinom koja se odnosi na raspoloživost hrane za naručivanje u inventaru kafeterije.

REQ4. Izmena u preostalim slobodnim terminima dostave za taj dan.

REQ5. Slanje poruke ili mejla klijentu sa porudžbinom hrane i odabranim metodama plaćanja.

REQ6. Slanje poruke ili mejla zaposlenima u kafeteriji sa informacijama o narudžbini.

REQ7. Ako neki korak u procesu naručivanja otkaže, sistem treba da vrati unazad transakciju i obavesti korisnika da je narudžbinu nemoguće obaviti uz objašnjenje o uzroku otkazivanja.

ATRIBUTI KVALITETA

Dati su atributi kvaliteta sistema za naručivanje hrane u kafeteriji radi provere ispravnosti specificiranih zahteva.

Zahtevi korišćenja

USE-1: Sistem treba da dozvoli klijentu da dobije prethodne narudžbine korišćenjem jednostavne interakcije.

USE-2: 95% novih korisnika treba da bude sposobno da uspešno naruči hranu bez grešaka prilikom prvog pokušaja.

Zahtevi za performansama

PER-1: Sistem treba da opslužuje 400 korisnika sa maksimalno 100 istovremenih korisnika u vreme najvećeg korišćenja sistema od 9:00 A.M do 10:00 A.M

PER-2: 95% veb strana generisanih sistemom treba da budu preuzimane u roku od 4 sekunde od trenutka kada se zahtevaju uz brzinu od 20Mbps ili bržom internet konekcijom.

PER-3: Sistem treba da prikaže korisnicima poruku o poslatoj porudžbini u prosečnom vremenu od 3 sekunde i maksimalnim vremenom od 6 sekundi u periodu od trenutka slanja informacije ka sistemu.

Bezbednosni zahtevi

SEC-1: Mrežne transakcije koje sadrže finansijske informacije ili personalne informacije klijenata treba da budu enkriptovane.

SEC-2: Jedino autorizovanom menadžeru menija treba da se omogući rad sa menijem.

SEC-4: Sistem treba da omogući klijentima da vide samo svoje narudžbine.

Zahtevi za raspoloživost

AVL-1: Sistem treba da bude dostupan u bar 98% vremena između 5.00 A.M. i ponoći po lokalnom vremenu bar 90% vremena između ponoći i 5:00 A.M. lokalnog vremena ne računajući predviđeno vreme za održavanje.

ZADATAK ZA VEŽBU

Tekst zadatka za vežbu

Zadatak asistenta je da organizuje proces recenzije prethodno izloženih zahteva.

Grupu studenata treba podeliti na sledeći način:

- Jedan student dobija ulogu autora. Autor ima zadatak da brani svoj rad, odnosno urađenu specifikaciju zahteva, i da adekvatno argumentuje svoj rad.
- Jedan student će preuzeti ulogu moderatora i čitaoca zahteva. Uloga moderatora je da vodi tok inspekcije. Uloga čitaoca je da, prema planiranom toku, čita deo po deo specifikacije zahteva, koje će recenzenti komentarisati.
- Jedan student će preuzeti ulogu zapisničara. Uloga zapisničara je da zapisuje uočene probleme od strane recenzenata, smernice za prepravku dokumentacije i pojašnjenja autora.
- Ostali studenti preuzimaju ulogu recenzenata. Svako treba da dobije jedan deo specifikacije koji će pročitati, samostalno analizirati, zabeležiti pitanja i probleme, a zatim ih iznositi kada dobije reč od strane moderatora. Ukoliko ima mnogo studenata u grupi, mogu raditi u parovima.

Recenzenti u svom radu treba da se služe kontrolnom listom ispravnosti koja je data u lekciji. Treba da prođu kroz sva pitanja koja se odnose na Kompletnost, Ispravnost, Atribut kvaliteta, Organizaciju i doslednost i Ostala pitanja (15 min). Treba da izvrše proveru onog dela specifikacije koji su dobili (30 min), po svim navedenim grupama zahteva, postavljajući

jedno po jedno pitanje i dajući odgovore na njih. Na kraju, kada su svi učesnici spremni, moderator (čitalac) prolazi redom kroz specifikaciju zahteva i diskutuje sa recenzentima (45 min). Za to vreme, zapisničar vodi evidenciju sa sastanka.

Budite savestan recenzent.

▼ Poglavlje 8

Domaći zadatak

DOMAĆI ZADATAK 13

Tekst domaćeg zadatka

Iz kontrolne liste zahteva date u lekciji uzmite pitanja koja se odnose na Ispravnost zahteva. Od asistenta ćete dobiti domaći zadatak jednog vašeg kolege u kome je pisao funkcionalne zahteve. Priminite predložena pitanja nad zahtevima iznetim u njegovom dokumentu. Potrudite se da budete savestan i objektivan recenzent i odgovorite na pitanja.

Ukoliko ste internet student, uradite recenziju zahteva iznetih u vežbi. Odgovorite na pitanja i pritom se potrudite da što objektivnije ocenite. Pošaljite domaći zadatak.

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 9

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 10

Zaključak

ZAKLJUČAK

1. V model razvoja softvera pokazuje ispitne aktivnosti koje započinju paralelno sa odgovarajućim razvojnim aktivnostima.
2. Verifikacija (potvrđivanje) određuje da li ste pravilno napisali zahteve. Validacija (provera) zahteva procenjuje da li ste napisali ispravne zahteve.
3. Recenzija daje izveštaj koji identifikuje ispitivani materijal, recenzente i ocenu tima za pregled da li su zahtevi prihvatljivi. Najbolje utvrđena vrsta formalne recenzije naziva se inspekcijom.
4. Učesnici inspekcije trebalo bi da predstavljaju četiri perspektive: autora proizvoda, korisnike sistema, predstavnike razvojnog tima i ljude koji će povezivati sistem sa spoljnim sistemima. U procesu inspekcije, učesniku obavljaju sledeće uloge: autor, moderator, čitalac i diktafon.
5. Broj pregledanih stranica u satu ima veliki uticaj na to koliko nedostataka je pronađeno. Dve do četiri stranice na sat su praktične smernice.
6. Svrha sastanka inspekcije je identifikovati što više glavnih nedostataka. Kontrolne liste skreću pažnju recenzenata na istorijski česte probleme sa zahtevima. Kontrolna lista sadrži proveru zahteva grupisanih u sledeće grupe: Kompletnost, Ispravnost, Atributi kvaliteta, Organizacija i sledljivost, i Ostala pitanja.
7. Predlozi za rešavanje izazova u vezi recenzija zahteva daju zajednički izazovi s kojima se organizacije suočavaju u pogledu preispitivanja zahteva, od kojih se neke odnose na formalne inspekcije, s predlozima kako da se reše svake od njih. Jedan od preduslova formalnog sastanka za recenziju je da su učesnici pregledali materijal koji se preispituje unapred, pojedinačno identifikujući svoje početne skupove pitanja.
8. Prototipovi su alati za validaciju koji prave zahteve. Sve vrste prototipova omogućavaju vam da pronađete nedostajuće zahteve pre skupljih aktivnosti kao što su razvoj i testiranje.
9. Testovi bi trebalo da obuhvate i normalan protok svakog slučaja upotrebe, alternativne tokove i izuzetke koje ste prepoznali tokom ispitivanja i analize. BA piše funkcionalne zahteve, a tester - testove sa zajedničke početne tačke: sa korisničkih zahteva. Možete predvideti više testova kako biste se obratili normalnom toku, alternativnim tokovima i izuzecima i proveriti da li su podržani mapom dijaloga.
10. Praćenjem putanje izvođenja za svaki test možete pronaći pogrešne ili nedostajuće zahteve, poboljšati korisničke navigacijske opcije i precizirati testove. U datim primerima BA i ispitivač kombinuju zahteve, analize modela i testove za otkrivanje nedostajućih, pogrešnih ili nepotrebnih zahteva mnogo pre nego što je bilo koji zapis napisan.
11. Kriterijumi prihvatanja trebalo bi da procene da li proizvod ispunjava svoje dokumentovane zahteve i da li je pogodan za upotrebu u predviđenom radnom

okruženju. Pri agilnom razvoju softvera, ako su ispunjeni svi kriterijumi prihvatanja za korisničku priču, vlasnik proizvoda će prihvatiti korisničku priču kao dovršenu.

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 poglavlju 17c 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,