



SE322 - INŽENJERSTVO ZAHTEVA

Nefunkcionalni zahtevi

Lekcija 11

PRIRUČNIK ZA STUDENTE

SE322 - INŽENJERSTVO ZAHTEVA

Lekcija 11

NEFUNKCIONALNI ZAHTEVI

- ✓ Nefunkcionalni zahtevi
- ✓ Poglavlje 1: Atributi kvaliteta softvera
- ✓ Poglavlje 2: Istraživanje atributa kvaliteta
- ✓ Poglavlje 3: Definisanje spoljnih zahteva kvaliteta
- ✓ Poglavlje 4: Unutrašnji zahtevi kvaliteta
- ✓ Poglavlje 5: Vežba
- ✓ Poglavlje 6: Domaći zadatak
- ✓ Poglavlje 7: 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

Uspeh softvera je više nego samo pružanje pravih funkcionalnosti. Korisnici takođe imaju očekivanja, često neusklađena, o tome kako će proizvod funkcionisati. Takva očekivanja uključuju svojstva, kao: koliko je lak za upotrebu, koliko brzo se izvršava, koliko retko ne uspeva, i kako se nosi sa neočekivanim uslovima. Takve karakteristike, zajednički poznate kao atributi kvaliteta, faktori kvaliteta, zahtevi kvaliteta ili zahtevi kvaliteta usluge predstavljaju glavni deo nefunkcionalnih zahteva sistema. U stvari, za mnoge, atributi kvaliteta su sinonim za nefunkcionalne zahteve, ali to je previše pojednostavljeno. Dve druge klase nefunkcionalnih zahteva su ograničenja i zahtevi za spoljašnji interfejs.

Ljudi se ponekad zakače na raspravu da li je određena potreba funkcionalni ili nefunkcionalni zahtev. Kategorizacija je važna manje nego osiguranje da identifikujete zahtev. Ova lekcija će vam pomoći da otkrijete i odredite nefunkcionalne zahteve koje možda drugačije niste pronašli. Atributi kvaliteta mogu razlikovati proizvod koji samo čini ono što treba, od proizvoda koji oduševljava svoje korisnike. Odlični proizvodi odražavaju optimalan balans konkurentskih kvaliteta. Ako ne istražite očekivanja kupaca u kvalitetu tokom emitovanja, imaćete samo sreće ako ih proizvod zadovoljava. Razočarani korisnici i frustrirani programeri su tipičniji ishod.

Atributi kvaliteta služe kao izvor mnogih funkcionalnih zahteva. Oni takođe upravljaju značajnim arhitektonskim i projektantskim odlukama. Mnogo je skuplje rekonstruisati kompletirani sistem za postizanje suštinskih ciljeva kvaliteta od projektovanje za njih na samom početku. Razmotrite brojne sigurnosne ispravke koje dobavljači operativnih sistema i često korištenih aplikacija periodično izdaju. Neki dodatni radovi na sigurnosti tokom razvojnog vremena mogu izbeći puno troškova i neugodnosti za korisnike

UVODNI VIDEO

Trajanje video snimka: 2min 13sek

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

▼ Poglavlje 1

Atributi kvaliteta softvera

VIDEO PREDAVANJE ZA OBJEKAT "ATRIBUTI KVALITETA SOFTVERA"

Trajanje video snimka: 20min 56sek

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

KLASIFIKACIJA KVALITETA

Jedan od načina klasifikacije kvaliteta razlikuje one karakteristike koje su vidljive izvršenjem softvera (spoljni kvalitet) od onih koji nisu (unutrašnji kva

Nekoliko desetina karakteristika proizvoda mogu se nazvati atributima kvaliteta, mada većina projektnih timova mora pažljivo razmotriti samo šačicu njih. Ako programeri znaju koje su od ovih karakteristika najvažnije za uspeh, mogu odabrati odgovarajuće projektantske pristupe za postizanje ciljeva kvaliteta. Atributi kvaliteta klasifikovani su prema širokom rasponu shema (DeGrace i Stahl 1993; IEEE 1998; ISO / IEC 2007; Miller 2009; ISO / IEC 2011). Neki autori su konstruisali opsežne hijerarhije koje grupiraju attribute u nekoliko glavnih kategorija.

Jedan od načina klasifikacije kvaliteta razlikuje one karakteristike koje su vidljive izvršenjem softvera (spoljni kvalitet) od onih koji nisu (unutrašnji kvalitet). Spoljni faktori kvaliteta su prvenstveno važni za korisnike, dok su unutrašnji kvaliteti važniji za osoblje za razvoj i održavanje. Interni atributi kvaliteta indirektno doprinose zadovoljstvu korisnika tako što je proizvod lakši za poboljšanje, ispravljanje, testiranje i prelazak na nove platforme.

Tabela na slici 1 ukratko opisuje nekoliko unutrašnjih i eksternih aspekata kvaliteta koje svaki projekat treba da uzme u obzir. Određeni atributi su od posebnog značaja za određene vrste projekata:

- Ugrađeni sistemi: performanse, efikasnost, pouzdanost, robusnost, sigurnost, sigurnost, upotrebljivost
- Internet i korporativne aplikacije: dostupnost, integritet, interoperabilnost, performanse, skalabilnost, sigurnost, upotrebljivost
- Desktop i mobilni sistemi: performanse, sigurnost, upotrebljivost

Pored toga, različiti delovi sistema možda će morati da naglase različite attribute kvaliteta. Performanse bi mogle da budu kritične za određene komponente, a upotrebljivost je od

presudne važnosti za druge. Vaše okruženje može imati druge jedinstvene attribute kvaliteta koji ovde nisu obuhvaćeni. Na primer, kompanije za proizvodnju igara možda žele da obuhvate emocionalne zahteve za svojim softverom.

Odeljak 6 obrazaca SRS opisan u lekciji 7. posvećen je atributima kvaliteta. Ako su neki zahtevi kvaliteta specifični za određene karakteristike, komponente, funkcionalne zahteve ili korisničke priče, povežite ih sa odgovarajućom stavkom u skladištu zahteva.

SPOLJNI I UNUTRAŠNJI ATRIBUTI KVALITETA

Spoljni faktori kvaliteta su prvenstveno važni za korisnike, dok su unutrašnji kvaliteti važniji za osoblje za razvoj i održavanje

Spoljni kvalitet	Kratak opis
Dostupnost	U kojoj su meri usluge sistema dostupne kada i gde su potrebne
Nestabilnost	Koliko je lako ispravno instalirati, deinstalirati i ponovo instalirati aplikaciju
Integritet	U kojoj meri sistem štiti od netačnosti podataka i gubitaka
Kompatibilnost	Koliko lako sistem može međusobno da poveže i razmenjuje podatke sa drugim sistemima ili komponentama
Performanse	Koliko brzo i predvidljivo sistem reaguje na korisničke unose ili druge događaje
Pouzdanost	Koliko dugo sistem radi prije nego što doživi kvar
Robustnost	Koliko dobro sistem reaguje na neočekivane radne uslove
Bezbednost	Koliko dobro sistem štiti od povreda i štete
Sigurnost	Koliko dobro sistem štiti od neovlašćenog pristupa aplikaciji i njenim podacima
Upotrebljivost	Koliko je lako ljudima da nauče, pamte i koriste sistem
Interni kvalitet	Kratak opis
Efikasnost	Koliko efikasno sistem koristi računarske resurse
Izmenljivost	Koliko je lako održavati, menjati, poboljšati i restrukturisati sistem
Prenosivost	Koliko lako sistem može da se koristi u drugim operativnim okruženjima
Ponovna upotreba	Do koje mere se komponente mogu koristiti u drugim sistemima
Prilagodljivost	Koliko lako sistem može da raste za više korisnika, transakcija, servera ili drugih dodataka
Proverljivost	Koliko lako programeri i testeri mogu potvrditi da je softver ispravno implementiran

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

Slika 1.1 Neki atributi kvaliteta softvera

▼ Poglavlje 2

Istraživanje atributa kvaliteta

SUŽAVANJE BROJA ATRIBUTA

U stvarnosti, postoje kompromisi i sukobi između određenih atributa koji onemogućavaju da ih istovremeno maksimalno iskoristite.

U idealnom univerzumu, svaki sistem bi pokazao maksimalnu moguću vrednost za sve svoje atribute. Sistem bi bio dostupan u svakom trenutku, nikada ne bi otkazao, pružio bi trenutne rezultate koji su uvek ispravni, blokirao bi sve pokušaje neovlašćenog pristupa i nikada ne bi zbunio korisnika. U stvarnosti, postoje kompromisi i sukobi između određenih atributa koji onemogućavaju da ih istovremeno maksimalno iskoristite. Kako je savršenstvo nedostižno, morate odrediti koji su atributi iz prethodno date tabele najvažniji za vaš uspeh u projektu. Tada možete kreirati određene ciljeve kvaliteta u pogledu ovih bitnih atributa, tako da projektanti softvera mogu donositi odgovarajuće izbore.

Različiti projekti zahtevaju različite skupove atributa kvaliteta za uspeh. Preporučuje se sledeći praktični pristup za identifikovanje i navođenje najvažnijih atributa za vaš projekat.

Korak 1: Počnite sa širokom taksonomijom.

Započnite s bogatim nizom atributa kvaliteta koji treba razmotriti, poput onih navedenih u prethodno datoj tabeli. Ovo široko polazište smanjuje verovatnoću previđenja važne dimenzije kvaliteta.

Korak 2: Smanjite listu

Uključite sve aktere da procenite koji će od atributa verovatno biti važni za projekat (za opsežnu listu mogućih aktera u projektu pogledajte sliku 2-2 u poglavlju 2, „Zahtevi iz perspektive kupca“, u referenci 1.). Na primer, šalter za prijavu odlazećih putnika na aerodromu mora da naglasi upotrebljivost (jer će se većina korisnika sa njim retko susresti) i bezbednost (jer mora da obrađuje plaćanje). Atributi koji se ne odnose na vaš projekat ne moraju se dalje razmatrati. Zapišite razloge za odlučivanje da se određeni atribut kvaliteta uzima u obzir ili ne uzima u obzir. Priznajte, međutim, da ako ne navedete ciljeve kvaliteta, niko ne bi trebao biti iznenađen ako proizvod ne pokazuje očekivane karakteristike. Zbog toga je važno dobiti doprinos više aktera. U praksi će neki atributi očigledno biti obuhvaćeni, neki će očigledno biti izvan okvira projekta, a samo će nekoliko zahtevati diskusiju o tome da li ih vredi razmotriti za projekat.

ODREĐIVANJE PRIORITETA I OČEKAVANJA KORISNIKA

Poređenja sa rangiranjem u parovima se mogu efikasno raditi sa malim spiskom atributa

Korak 3: Postavite prioritete atribute

Određivanje prioriteta relevantnim atributima postavlja fokus za buduće rasprave o izazivanju zahteva. Poređenja sa rangiranjem u parovima mogu efikasno raditi sa malim spiskom atributa. Slika 1 ilustruje kako se koristi Brosseauova tabela za procenu kvaliteta svojstva za slučaj prijemnog šaltera na aerodromu. Za svaku ćeliju na preseku dva atributa zapitajte se: „Ako bih mogao imati samo jedan od ovih atributa, koji bih uzeo?“ Unošenje znaka manjeg od ($<$) u ćeliju ukazuje da je atribut u redu važniji; simbol karata (\wedge) ukazuje da je atribut na vrhu stupca važniji. Na primer, upoređujući raspoloživost (tj. dostupnost) i integritet, vidimo da je integritet važniji. Putnik uvek može da se javi na info desku ako šalter ne radi (doduše, možda sa dugim nizom putnika). Ali ako šalter ne pokaže tačne podatke, putnik će biti vrlo nesrećan. Tako da smo stavili strelicu u ćeliju na preseku dostupnosti i integriteta, ukazujući na integritet kao najvažniji od dva.

Korak 4: Izdvojite specifična očekivanja za svaki atribut

Komentari koje korisnici tokom zahteva postavljaju neke indicije o kvalitetnim karakteristikama koje imaju na umu za proizvod. Trik je u tome da precizirate šta korisnici misle kada kažu da softver mora biti prilagođen korisnicima, brz, pouzdan ili robustan. Pitanja koja istražuju očekivanja korisnika mogu dovesti do specifičnih zahteva za kvalitetom koji pomažu programerima da stvore prijatan proizvod

Korisnici neće znati kako da odgovore na pitanja poput „Koji su vaši zahtevi za interoperabilnost?“ Ili, „Koliko pouzdan softver mora da bude?“ Biznis analitičar će morati da postavlja pitanja koja će voditi misaone procese korisnika kroz istraživanje interoperabilnosti, pouzdanosti i drugih atributa.

Zato ovaj proces zovemo "izazivanje", a ne "utvrđivanje" zahteva, jer moramo na posredni način da od korisnika saznamo (ili "iščupamo") stvarne zahteve. Najčešće oni nisu sposobni da vam ih direktno saopšte.

Atribut	Ocena	Raspoloživost	Integritet	Performanse	Pouzdanost	Robusnost	Bezbednost	Upotrebljivost	Poverljivost
Raspoloživost	2		>	>	>	<	>	>	<
Integritet	6			<	<	<	>	<	<
Performanse	4				<	<	>	>	<
Pouzdanost	2					<	>	>	>
Robusnost	1						>	>	<
Bezbednost	7							<	<
Upotrebljivost	5								<
Proverljivost	1								

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

Slika 2.1 Određivanje prioriteta atributa kvaliteta softvera

MOGUĆA PITANJA U TOKU IZAZIVANJA ZAHTEVA

Kada planira sesiju za izazivanje, BA treba da započne sa spisakom pitanja i da ga svede na ona pitanja koja su najvažnija za projekat.

Kada planira sesiju za izazivanje, BA treba da započne sa spisakom pitanja i da ga svede na ona pitanja koja su najvažnija za projekat. Kao ilustraciju, sledi nekoliko pitanja koja bi BA mogao da postavi kako bi razumeo očekivanja korisnika o performansama sistema koji upravlja prijavama za patente koje su izumitelji podneli:

1. Koje bi bilo razumno ili prihvatljivo vreme odziva za pronalaženje tipične prijave za patent kao odgovor na upit?
2. Šta bi korisnici smatrali neprihvatljivim vremenom odgovora za tipičan upit?
3. Koliko korisnika istovremeno očekujete u proseku?
4. Koji je maksimalni broj istodobnih korisnika koji biste predvideli
5. U koja doba dana, nedelje, meseca ili godine dolazi do težeg korišćenje od uobičajenih?

Slanje liste takvih pitanja unapred učesnicima razgovora pruža im priliku da razmisle ili istražuju svoje odgovore kako ne bi morali da odgovaraju na čitav niz pitanja. Dobro završno pitanje koje biste postavili tokom bilo koje rasprave o izazivanju zahteva je: „Postoji li nešto što nisam vas pitao o čemu bismo trebali da razgovaramo?“

Razmislite da pitate korisnike šta bi predstavljalo neprihvatljiv učinak, sigurnost ili pouzdanost. Odnosno, odredite svojstva sistema koja bi narušila korisnička očekivanja u pogledu kvaliteta, kao što je omogućavanje neovlašćenom korisniku da briše datoteke. Definisanje neprihvatljivih karakteristika omogućava vam da osmislite testove koji pokušavaju naterati sistem da demonstrira te neprihvatljive karakteristike. Ako ih on ne daje, verovatno ste postigli svoje ciljeve kvaliteta. Ovaj pristup je posebno vredan za bezbednosne primene u kojima sistem koji krši pouzdanost ili bezbednosne tolerancije predstavlja opasnost po život ili umiranje.

Druga moguća strategija izazivanja zahteva je da započnemo sa ciljevima kvaliteta koje zainteresovani akteri imaju sistem za neki sistem u razvoju. Kvalitetni cilj zainteresovanih strana može se razgraditi tako da otkriva i funkcionalne i nefunkcionalne podgrupe - a samim tim i zahteve - koji postaju i specifičniji i lakši za merenje dekompozicijom.

KAKO DO DOBRO STRUKTURISANIH ZAHTEVA ZA KVALITETOM?

Zahtevi za kvalitetom moraju biti merljivi kako bi se uspostavio precizan dogovor o očekivanjima između BA, kupaca i razvojnog tima

Korak 5: Navedite dobro struktuirane zahteve za kvalitetom

Jednostavni zahtevi kvaliteta kao što su „Sistem će biti prilagođen korisniku“ ili „Sistem će biti dostupan 24x7“ nisu korisni. Prva je previše subjektivna i nejasna; ovo je retko realno ili neophodno. Nijedna nije merljiva. Takvi zahtevi pružaju malo smernica programerima. Dakle, poslednji korak je stvaranje specifičnih i proverljivih zahteva iz informacija koje su dobijene u vezi svakog atributa kvaliteta. Kada pišete zahteve za kvalitetom, imajte na umu korisne SMART mnemonike - učinite ih specifičnim, merljivim, dostižnim, relevantnim i osetljivim na vreme.

Zahtevi za kvalitetom moraju biti merljivi kako bi se uspostavio precizan dogovor o očekivanjima između BA, kupaca i razvojnog tima. Ako to nije merljivo, malo je smisla u njegovom preciziranju, jer nikada nećete moći da utvrdite da li ste postigli željeni cilj. Ako ispitivač ne može da testira zahtev, to nije dovoljno dobro. Navedite u skali ili jedinice mere za svaki atribut i ciljnu, minimalnu i maksimalnu vrednost. Možda će biti potrebno nekoliko diskusija s korisnicima da bi se utvrdili jasni, merljivi kriterijumi za procenu zadovoljenja zahteva kvaliteta

Možete uključiti kriterijum podobnosti - „kvantifikacija zahteva koji pokazuje standard koji proizvod mora da ispuni“ - kao deo specifikacije svakog zahteva, i funkcionalnog i nefunkcionalnog. Ovo je odličan savet. Kriterijumi podobnosti opisuju merljiv način da se proceni da li je svaki zahtev pravilno realizovan. Oni pomažu projektantima softvera da odaberu rešenje za koje veruju da će ispuniti cilj, a testerima pomažu da procene rezultate.

Umesto da izmislite svoj način dokumentovanja nepoznatih zahteva, potražite postojeći obrazac zahteva koji treba slediti. Obrazac pruža smernice o tome kako napisati određenu vrstu zahteva, zajedno sa predloškom koji možete popuniti sa određenim detaljima za vašu situaciju. Sledeći obrasci pomoći će čak i BA početnicima da pišu odgovarajuće zahteve kvaliteta

▼ Poglavlje 3

Definisanje spoljnih zahteva kvaliteta

VIDEO PREDAVANJE ZA OBJEKAT "DEFINISANJE SPOLJNIH ZAHTEVA KVALITETA"

Trajanje video snimka: 1sat 10min 25sek

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

ATRIBUTI SPOLJNIH ZAHTEVA KVALITETA

Spoljni atributi kvaliteta opisuju karakteristike koje se primećuju prilikom izvršavanja softvera. Oni duboko utiču na korisničko iskustvo i korisnikovo opažanje kvaliteta sistema.

Ovaj odeljak opisuje svaki od atributa kvaliteta u Tabeli 14-1 i predstavlja neke zahteve za kvalitetom uzoraka iz različitih projekata. Postoji mnogo dodatnih primera dobro preciziranih zahteva za atributima kvaliteta. Kao i kod svih zahteva, dobra je ideja zabeležiti poreklo svakog zahteva za kvalitetom i obrazloženje navedenih ciljeva kvaliteta, ako oni nisu očigledni. Obrazloženje je važno u slučaju da se postave pitanja o potrebi za određenim ciljem ili da li je cena opravdana. Ta vrsta izvora informacija izostavljena je iz primera predstavljenih u ovom poglavlju.

Spoljni atributi kvaliteta opisuju karakteristike koje se primećuju prilikom izvršavanja softvera. Oni duboko utiču na korisničko iskustvo i korisnikovo opažanje kvaliteta sistema. Spoljni atributi kvaliteta opisani u ovom poglavlju su:

- dostupnost (ili raspoloživost)
- instaliranje,
- integritet,
- interoperabilnost,
- performanse,
- pouzdanost,
- robusnost,
- sigurnost,
- bezbednost i
- upotrebljivost.

DOSTUPNOST

Dostupnost je merilo planiranog vremena trajanja u kome su usluge sistema dostupne za upotrebu i u potpunosti rade.

Dostupnost je merilo planiranog vremena trajanja u kome su usluge sistema dostupne za upotrebu i u potpunosti rade. Formalno, dostupnost je jednaka odnosu vremena trajanja prema vremenu trajanja i vremena prekida. Još formalnije, dostupnost je jednaka srednjem vremenu između kvarova (MTBF) za sistem podeljenog sa zbirom broja MTBF i srednjim vremenom za oporavak (MTTR) sistema nakon što se dogodi kvar. Planirani periodi održavanja takođe utiču na dostupnost. Dostupnost je usko povezana sa pouzdanošću i na nju snažno utiče podkategorija održivosti koja se može modifikovati. Određeni zadaci su kritičniji od drugih. Pitajte korisnike koliki je procenat trajanja oporavka sistema zaista potreban ili koliko sati u određenom vremenskom periodu sistem mora da bude dostupan. Pitajte da li postoje vremenski periodi za koje je dostupnost neophodna za ispunjavanje poslovnih ili sigurnosnih ciljeva. Zahtevi za dostupnost su posebno složeni i važni za veb lokacije, aplikacije zasnovane na oblaku (cloud) i aplikacije čiji korisnici rade u više vremenskih zona. Zahtev za raspoloživost može biti naveden kao sledeće:

AVL-1. Sistem će biti dostupan najmanje 95 %u radnim danima između 6:00. i ponoć po istočnom vremenu, a najmanje 99 % dostupno je radnim danima između 3:00 po podne. i 5:00 P.M. ET. Ovaj zahtev je donekle pojednostavljen. Ne definiše nivo performansi koji čini dostupnost. Zahtevi za dostupnost se ponekad ugovorno određuju kao ugovor o nivou usluge.

Takvi zahtevi moraju tačno definisati šta predstavlja sistem koji je dostupan (ili ne) i mogu uključivati izjave kao što su sledeće: **AVL-2. Vreme prekida rada sistema koje je isključeno iz obračuna dostupnosti, sastoji se od vremena dodeljenog za održavanja zakazanog od 6:00 uveče, u nedelju, pacifičko vreme, do 3:00 ujutro u ponedeljak, pacifičko vreme.**

Kada postavljate zahteve za dostupnošću, postavite pitanja kako biste istražili sledeća pitanja:

- Koji su delovi sistema najkritičniji za dostupnost?
- Koje su poslovne posledice kvara sistema na strani korisnika?
- Ako se redovno održavanje mora vršiti periodično, kada to treba planirati? Kakav je uticaj na raspoloživost sistema? Koja su minimalna i maksimalna trajanja perioda održavanja? Šta se radi sa pokušajima pristupa korisnika za vreme perioda održavanja?
- Ako se aktivnosti održavanja moraju obavljati u vremenu kada sistema radi, kakav će uticaj imati na dostupnost i na koji način se taj uticaj može umanjiti?
- Koja su obaveštenja korisnicima potrebna ako sistem postane nedostupan?
- Koji delovi sistema imaju strožije zahteve za dostupnošću od ostalih?
- Koje zavisnosti o dostupnosti postoje između funkcionalnih grupa

LAKOĆA INSTALISANJA

Mera lakoće instaliranja sistema je srednje vreme za instaliranje sistema.

Softver nije koristan dok se ne instalira na odgovarajući uređaj ili platformu. Neki primeri instalacije softvera su: preuzimanje aplikacija na telefon ili tablet; premeštanje softvera sa računara na veb server; ažuriranje operativnog sistema; instaliranje ogromnog komercijalnog sistema, kao što je alat za planiranje resursa preduzeća; preuzimanje ažuriranja upravljačkog softvera u kablovsku televiziju; i instaliranje aplikacije krajnjeg korisnika na računar. Nestabilnost opisuje koliko je lako pravilno obavljati ove operacije. Povećavanje lakoće instaliranja sistema smanjuje vreme, troškove, ometanje korisnika, učestalost grešaka i nivo veština potrebnih za operaciju instalacije. Instalabilnost se bavi sledećim aktivnostima:

- Početna instalacija
- Oporavak od nepotpune, netačne ili korisničke prekidačke instalacije
- Ponovna instalacija iste verzije
- Instalacija nove verzije
- Vraćanje na prethodnu verziju
- Instalacija dodatnih komponenti ili ispravki
- Deinstalacija

Mera lakoće instaliranja sistema je srednje vreme za instaliranje sistema. To, međutim, zavisi od mnogo faktora: koliko je iskusen instalater brzina odredišnog računara, medij na kojem se softver instalira

(preuzimanje Interneta, lokalna mreža, CD / DVD), ručni koraci potrebni tokom instalacije , i tako dalje.

Detaljnu listu smernica i razmatranja za zahteve za instaliranje i testiranje za instalaciju možete naći na Sledi nekoliko zahteva vezana za lakoću instaliranja: http://www.testingstandards.co.uk/installability_guidelines.htm

INS-1. *Neobrazovani korisnik mora biti u mogućnosti da uspešno izvrši početnu instalaciju aplikacije u proseku 10 minuta.*

INS-2. *Kada instalirate novu verziju aplikacije, sva prilagođavanja u korisnikovom profilu će se zadržati i pretvoriti u format podataka nove verzije, ako je potrebno. (na primer, ne prihvatanje plaćanja kreditnom karticom za kupovinu ako funkcija autorizacije kreditne kartice nije dostupna)?*

INS-3. *Instalacioni program mora da proveri ispravnost preuzimanja pre nego što započne postupak instalacije.*

INS-4. *Instaliranje ovog softvera na server zahteva privilegiju administratora. INS-5.* *Nakon uspešne instalacije, instalacioni program će izbrisati sve privremene, rezervne kopije, zastarele i nepotrebne datoteke povezane sa aplikacijom.*

Primeri nekih pitanja koja treba istražiti kada postavljate zahteve za instalaciju: Koje instalacione operacije moraju biti izvedene bez ometanja sesije korisnika? Koji će postupci instalacije zahtevati ponovno pokretanje aplikacije?

INTEGRITET

Integritet se bavi sprečavanjem gubitka informacija i očuvanjem ispravnosti podataka.

Integritet (*integrity*) se bavi sprečavanjem gubitka informacija i očuvanjem ispravnosti podataka unesenih u sistem. Zahtevi za integritet nemaju toleranciju na greške: podaci su ili u dobrom stanju i zaštićeni, ili nisu. Podaci moraju biti zaštićeni od pretnji kao što su slučajni gubitak ili oštećenje, naizgled identični skupovi podataka koji se ne podudaraju, fizičko oštećenje medija za skladištenje, slučajno brisanje datoteka ili prepisivanje podataka od strane korisnika. Namerni napadi koji pokušavaju namerno pokvariti ili ukrasti podatke takođe su rizik. Sigurnost se ponekad smatra podskupom integriteta, jer su neki sigurnosni zahtevi namenjeni sprečavanju pristupa podacima od strane neovlašćenih korisnika. Zahtevi za integritet treba da osiguraju da se podaci primljeni od drugih sistema podudaraju sa poslatim i obrnuto. Izvršivi softveri sami su podložni napadu, tako da njihov integritet takođe mora biti zaštićen.

Integritet podataka takođe se odnosi na tačnost i pravilno oblikovanje podataka. To uključuje pažnju na formatiranje polja za datume, ograničavanje polja na ispravan tip podataka ili dužinu, osiguravanje da elementi podataka imaju validne vrednosti, proveru da li postoji odgovarajuće unošenje u jedno polje kada drugo polje ima određenu vrednost i tako dalje. Sledi nekoliko zahteva za integritet uzorka:

INT-1. *Nakon izvršenja sigurnosne kopije datoteke, sistem će potvrditi kopiju rezervne kopije u odnosu na original i prijaviti sve razlike.*

INT-2. *Sistem se štiti od neovlašćenog dodavanja, brisanja ili modifikacije podataka.*

INT-3. *Sistem za praćenje hemikalija potvrđuje da kodirana hemijska struktura uvezana od alata za crtanje drugih proizvođača predstavlja važeću hemijsku strukturu.*

INT-4 *Sistem svakodnevno potvrđuje da izvršni programi aplikacije nisu modifikovani dodavanjem neovlašćenog koda.*

Neki faktori koje treba uzeti u obzir prilikom rasprave o zahtevima integriteta uključuju sledeće:

- Osiguravanje da se promene podataka u potpunosti ili uopšte ne izvrše. To može značiti backup od promene podataka ako se tokom operacije nađe kvar u toku operacije.
- Osiguravanje trajnosti promena koje se unose u podatke.
- Koordinacija promena izvršenih u više skladišta podataka, posebno kada se promene moraju izvršiti istovremeno (recimo, na više servera) i u određeno vreme
- Osiguravanje fizičke sigurnosti računara i spoljnih uređaja za skladištenje podataka.
- Izvođenje sigurnosnih kopija podataka. (Na kojoj frekvenciji? Automatski i / ili na zahtev? Kojih datoteka ili baza podataka? Na koje medije? Sa ili bez kompresije i verifikacije?)
- Vraćanje podataka iz rezervne kopije.

INTEROPERABILNOST

Interoperabilnost ukazuje na to koliko lako sistem može da razmenjuje podatke i usluge sa drugim softverskim sistemima i koliko lako se može integrisati sa spoljnim hardverskim uređajima

Interoperabilnost (*interoperability*) ukazuje na to koliko lako sistem može da razmenjuje podatke i usluge sa drugim softverskim sistemima i koliko lako se može integrisati sa spoljnim hardverskim uređajima. Da biste procenili interoperabilnost, morate znati koje će druge

aplikacije korisnici koristiti zajedno sa vašim proizvodom i koje podatke očekuju da razmjene. Korisnici sistema za praćenje hemikalija bili su navikli da crtaju hemijske strukture s nekoliko komercijalnih alata, pa su izneli sledeće zahteve za interoperabilnost:

IOP-1. *Sistem za praćenje hemikalija mora biti u mogućnosti da uvozi bilo koju važeću hemijsku strukturu iz alata ChemDraw (verzija 13.0 ili starija) i MarvinSketch (verzija 5.0 ili starija).*

Možda biste radije to naveli kao zahtev za spoljnim interfejsom i definisali formate informacija koje sistem za praćenje hemikalija može da uvozi. Mogli biste definisati i nekoliko funkcionalnih zahteva koji se bave operacijom uvoza. Prepoznavanje i dokumentovanje takvih zahteva važnije je od načina na koji ih klasifikujete.

Zahtevi za interoperabilnost mogu nalagati da se koriste standardni standardni formati za razmenu podataka da bi se olakšala razmena informacija sa drugim softverskim sistemima. Takav zahtev za sistem praćenja hemikalija bio je:

IOP-2. *Sistem za praćenje hemikalija mora biti u mogućnosti da uvozi bilo koju hemijsku strukturu šifriranu pomoću SMILES (pojednostavljeni sistem za unos linija u molekularni unos).*

Razmišljanje o sistemu iz ugla atributa kvaliteta ponekad otkriva prethodno nestalne zahteve. Korisnici nisu izrazili potrebu za interoperabilnošću ove hemijske strukture kada smo razgovarali ili o spoljnim interfejsima ili o funkciji sistema. Čim je BA pitao o drugim sistemima na koje se sistem za praćenje hemikalija mora povezati, šampion proizvoda odmah je spomenuo dva paketa za crtanje hemijske strukture. Sledi nekoliko pitanja koja možete da koristite kada istražujete zahteve za interoperabilnošću:

- Kojim drugim sistemima mora biti ovaj interfejs?
- Koje usluge ili podatke moraju da razmenjuju?
- Koji su standardni formati podataka potrebni za podatke koje je potrebno razmeniti s drugim sistemima?
- Koje konkretne komponente hardvera moraju biti povezane sa sistemom?
- Koje poruke ili kodove sistem mora da prima i obrađuje iz drugih sistema ili uređaja? Koji su standardni protokoli za komunikaciju potrebni da bi se omogućila interoperabilnost? Koje spoljno mandatne zahteve za interoperabilnost mora sistem da ispunjava?

PERFORMANSE

Performanse predstavljaju odgovor sistema na različite korisničke upite i radnje.

Performanse su jedan od atributa kvaliteta koji će korisnici često spontano odrediti. Performanse predstavljaju odgovor sistema na različite korisničke upite i radnje, ali obuhvataju mnogo više od toga, kao što je prikazano u tabeli na slici 1.

Dimenzija performanse	Primer
Vreme odziva	Proj sekundi do prikaza veb strance
Protok	Broj transakcija sa kreditnim karticama u sekundi
Kapacitet podataka	Maksimalni broj istovremenih korisnika veb sajta
Predvidljivost u sistemima sa realnim vremenom	Strogi vremenski zahtevi za kontrolni sistem upravljanja avionom
Vreme kašnjenja	Vremenska kašnjenja u zapisu muzike i produkcijonog softvera
Ponašanje sistema u uslovima preopterećenja	U slučaju zemljotresa, dolazi do masovnog pozivanja telefonom

Slika 3.1 Neki aspekti performansi softvera

Problemi sa performansama mogu predstavljati ozbiljne rizike po bezbednost. Strogi zahtevi za performansama snažno utiču na strategije projektovanja softvera i izbor hardvera, pa definišite ciljeve performansi koji su pogodni za operativno okruženje. Zadovoljavanje svih zahteva za performansama može biti teško jer oni puno zavise od spoljnih faktora, kao što su brzina računara koji se koristi, mrežne veze i druge komponente hardvera.

Kada dokumentujete zahteve za performansama, takođe dokumentujte njihovo obrazloženje kako bi vodili projektante softvera u donošenju odgovarajućih izbora projektnog rešenja. Na primer, strogi zahtevi vremena reakcije baze podataka mogli bi dovesti projektante da kopiraju.

bazu podataka na više geografskih lokacija. Navedite broj transakcija u sekundi koje treba obaviti, vremena odziva i odnose zakazivanja zadataka za sisteme u realnom vremenu.

Možete takođe odrediti zahteve za memorijskim i diskovnim prostorom, istovremenskim opterećenjima korisnika ili maksimalnim brojem redova koji se čuvaju u tablicama baze podataka. Korisnici i BA ne znaju sve ove informacije, tako da planiraju da sarađuju sa različitim akterima kako bi istražili više tehničkih aspekata zahteva za kvalitetom. Sledi nekoliko zahteva za performansama uzoraka:

PER-1. *Odobrenje zahteva za podizanje bankomata traje ne više od toga 2,0 sekundi.*

PER-2. *Senzori brzine protiv blokiranja kočionog sistema moraju svaki izveštavati o brzinama točkova 2 milisekunde sa varijacijom koja ne prelazi 0,1 milisekundu.*

PER-3. *Veb stranice se u potpunosti preuzimaju u proseku 3 sekunde ili manje tokom 30 megabita / sekundi Internet veza.*

PER-4. *Najmanje 98 procenata vremena sistem trgovanja će ažurirati prikaz stanja transakcija u roku od 1 sekunde nakon završetka svake trgovine.*

Performanse su atribut spoljnog kvaliteta jer se može primetiti samo tokom izvođenja programa. One su usko povezane sa internim atributom kvaliteta efikasnosti, koji ima veliki uticaj na performanse koje posmatra korisnik.

POUZDANOST

Pouzdanost je verovatnoća da će softver raditi bez kvara tokom određenog vremenskog perioda.

Verovatnoća da se softver može izvršiti bez kvara tokom određenog vremenskog perioda poznata je kao pouzdanost. Problemi sa pouzdanošću mogu se pojaviti zbog nepravilnih unosa, grešaka u samom softverskom kodu, komponenti koje nisu dostupne po potrebi i kvara hardvera. Robusnost i dostupnost usko su povezani sa pouzdanošću. Načini za određivanje i merenje pouzdanosti softvera uključuju procenat operacija koje su pravilno

izvedene, prosečno vreme trajanja sistema pre neuspeha (srednje vreme između kvarova ili MTBF) i maksimalnu prihvatljivu verovatnoću kvara tokom određenog vremenskog perioda. Uspostavite zahteve za kvantitativnom pouzdanošću na osnovu toga koliko bi ozbiljan bio uticaj ako bi se desio kvar i da li je cena maksimiziranja pouzdanosti opravdana. Sistemi koji zahtevaju visoku pouzdanost takođe bi trebalo da budu projektovani za veliku proverljivost kako bi olakšali pronalaženje nedostataka koji bi mogli ugroziti pouzdanost. Evo jednog primera za zahtev za pouzdanost ovog sistema je bio

REL-1. *Zbog kvara softvera ne može se izgubiti više od 5 eksperimentalnih potrošnja od 1.000.*

Neki kvarovi sistema su ozbiljniji od drugih. Neuspeh može prisiliti korisnika da ponovo pokrene aplikaciju i obnovi sačuvane podatke. Ovo je neugodno, ali nije katastrofalno. Neuspesi koji rezultiraju izgubljenim ili oštećenim podacima, na primer, kada pokušaj transakcije baze podataka ne uspe, još su ozbiljniji. Sprečavanje grešaka je bolje nego otkrivanje i pokušaj oporavka od njih.

Kao i mnogi drugi atributi kvaliteta, pouzdanost je pokazatelj koji utvrdite sa kašnjenjem: ne možete da kažete da li ste to postigli sve dok sistem neko vreme ne radi. Razmotrite sledeći primer:

REL-2. *Srednje vreme između kvarova komponente čitača kartica biće najmanje 90 dana.*

Nema načina da se kaže da li je sistem ispunio ovaj zahtev sve dok nije prošlo najmanje 90 dana. Međutim, možete reći da li sistem nije uspeo da pokaže dovoljno pouzdanosti ako komponenta čitača kartica ne uspe više od jednom u roku od 90 dana. Sledi nekoliko pitanja koja treba da postavite predstavnicima korisnika kada postavljate zahteve za pouzdanošću: Kako biste presudili da li je ovaj sistem dovoljno pouzdan? Koje bi bile posledice doživljaja kvara prilikom obavljanja određenih operacija sa sistemom? Šta biste smatrali kritičnim promašajem, za razliku od smetnje? Pod kojim uslovima bi neuspeh mogao da ima ozbiljne posledice na vaše poslovanje? Niko ne voli da vidi pad sistema, ali postoje li određeni delovi sistema koji apsolutno moraju biti super pouzdani? Ako sistem propada, koliko dugo može da ostane van mreže pre nego što značajno utiče na vaše poslovanje? Razumevanje zahteva za pouzdanošću omogućava arhitektima, projektantima i programerima da preduzmu akcije za koje misle da će postići potrebnu pouzdanost. Iz perspektive zahteva, jedan od načina da se sistem učini pouzdanim i robusnim je odrediti uslove za izuzeće i način na koji se njima treba rukovati.

ROBUSNOST

Robusnost je stepen do koga sistem i dalje pravilno funkcioniše kada se suoči sa nevažecim ulazima i oštećenjima.

Robusnost je stepen do koga sistem i dalje pravilno funkcioniše kada se suoči sa nevažecim ulazima, oštećenjima na povezanim softverskim ili hardverskim komponentama, spoljnim napadima ili neočekivanim radnim uslovima. Robustan softver se graciozno oporavlja od problematičnih situacija i oprašta korisničke greške. Vraća se od unutrašnjih kvarova bez negativnog uticaja na iskustvo krajnjeg korisnika. Softverskim greškama upravlja se na način koji korisnik doživljava kao razumne, a ne dosadne. Ostali termini atributa povezani sa

robusnošću su tolerancija na greške (da li se greške unose i ispravljaju greške kod korisnika?), održivost (može li kamera doživeti pad s određene visine bez oštećenja?) i obnovljivost (može li računar nastaviti pravilan rad ako izgubi napajanje usred ažuriranja operativnog sistema?). Kada postavljate zahteve za robusnošću, pitajte korisnike o uslovima greške na koje sistem može da naiđe i kako sistem treba da reaguje. Razmislite o načinima kako da otkrijete moguće kvarove koji bi mogli dovesti do kvara sistema, prijavite ih korisniku i oporavite ih od njih ako dođe do kvara. Obavezno shvatite kada jedna operacija (kao što je priprema podataka za prenos) mora biti ispravno završena pre nego što druga započne (slanje podataka u drugi računarski sistem). Jedan primer zahteva za robusnošću je

ROB-1. *Ako uređivač teksta padne pre nego što korisnik sačuva datoteku, on će oporaviti sadržaj datoteke koja se uređuje najviše od jedne minute pre pada, a drugi put kada isti korisnik pokrene aplikaciju.*

Zahtev poput ovog mogao bi dovesti programera da primenjuje kontrolnu tačku ili periodično automatsko skladištenje podataka radi smanjivanja gubitka podataka, zajedno sa funkcionalnošću za traženje sačuvanih podataka po pokretanju i vraćanje sadržaja datoteke. Jedan od naših zahteva za robusnošću u jednom projektu je bio

ROB-2. *Svi parametri opisa crteža moraju imati zadane vrednosti koje će grafički motor koristiti ako ulazni podaci parametra nedostaju ili su nevažeći.*

Uz ovaj zahtev, program se ne bi srušio ako je, na primer, aplikacija zatražila nepodržani stil linije. Grafički motor isporučio bi zadani stil pune linije i nastavio s izvršavanjem. To bi i dalje predstavljalo kvar proizvoda, jer krajnji korisnik nije dobio željeni izlaz. Ali projektovanje za robusnost smanjilo je ozbiljnost otkaza od pada programa do generisanja pogrešnog stila linije, primera tolerancije greške.

BEZBEDNOST

Bezbednosni zahtevi bave se potrebom da se spreči sistem da nanosi bilo kakvu povredu ljudi ili materijalnu štetu.

Bezbednosni zahtevi bave se potrebom da se spreči sistem da nanosi bilo kakvu povredu ljudi ili materijalnu štetu. Zahtevi za bezbednost(safety) mogu biti diktirani državnim propisima ili drugim poslovnim pravilima, a pravna ili sertifikaciona pitanja mogu se povezati sa zadovoljavanjem takvih zahteva. Zahtevi za bezbednost često se pišu u obliku uslova ili radnji koje sistem ne sme da dozvoli. Ljudi se retko povrede eksplozivnim proračunskih tablica. Međutim, hardverski uređaji koji upravljaju softverom sigurno mogu predstavljati rizik za život. Čak i neke aplikacije samo za softver mogu imati nevidljive sigurnosne zahteve. Aplikacija kojom se ljudima dozvoljava da naručuju obroke iz kafeterije možda uključuje bezbednosne zahteve poput sledećeg:

SAF-1. *Korisnik će moći da vidi spisak svih sastojaka u bilo kojim stavkama menija, sa istaknutim sastojcima za koje se zna da izazivaju alergijske reakcije u više od 0,5 procenata stanovništva Severne Amerike.*

Mogućnosti veb pregledača poput roditeljskog nadzora koji onemogućavaju pristup određenim funkcijama ili URL-ovima mogu se smatrati rešenjima bilo za bezbednosne, bilo za

sigurnosnih zahteva. Češće je videti sigurnosne zahteve napisane za sisteme koji uključuju hardver, kao što su sledeći primeri:

SAF-2. *Ako temperatura posude reaktora raste brže od 5C u minutu, sistem za kontrolu hemijskog reaktora isključuje izvor toplote i uputiti upozorenje rukovaocu.*

SAF-3. *Terapeutska mašina za zračenje mora dozvoliti ozračivanje samo ako je na raspolaganju odgovarajući filter.*

SAF-4. *Sistem će prekinuti bilo koju operaciju u roku od jedne sekunde ako izmereni pritisak u rezervoaru prelazi 90 procenata specificiranog maksimalnog pritiska.*

Kada postavljate bezbednosne zahteve, možda će biti potrebno da intervjuirate stručnjake za stvari koje su veoma dobro upoznate sa operativnim okruženjem ili ljude koji su puno razmišljali o projektnim rizicima. Razmislite o postavljanju pitanja kao što je sledeće:

Pod kojim uslovima se čoveku može naneti šteta upotrebom ovog proizvoda? Kako sistem može otkriti te uslove? Kako treba da reaguje? Koja je maksimalna dozvoljena učestalost grešaka koji mogu izazvati povrede? Koji načini kvara mogu uzrokovati štetu ili štetu na imovini? Koje radnje operatera mogu da slučajno nanesu štetu ili oštete imovinu? Postoje li posebni načini rada koji predstavljaju rizik za ljude ili imovinu?

SIGURNOST

Sigurnost se bavi blokiranjem neovlašćenog pristupa sistemskim funkcijama ili podacima osiguravajući da je softver zaštićen od napada zlonamernog softvera.

Sigurnost se bavi blokiranjem neovlašćenog pristupa sistemskim funkcijama ili podacima, osiguravajući da je softver zaštićen od napada zlonamernog softvera i tako dalje. Sigurnost je glavni problem Internet softvera. Korisnici sistema e-trgovine žele da podaci o njihovim kreditnim karticama budu sigurni. Veb surferi ne žele da se lični podaci ili snimci lokacija koje posećuju koriste neprimereno. Kompanije žele da zaštite svoje veb stranice od napada uskraćivanja usluge ili hakerskih napada. Kao i kod zahteva integriteta, bezbednosni zahtevi nemaju toleranciju na greške. Sledi nekoliko razloga za razmatranje prilikom postavljanja bezbednosnih zahteva:

- Korisnička ovlašćenja ili nivoi privilegija (obični korisnik, gost korisnik, administrator) i korisničke kontrole pristupa (matrica uloga i dozvola koja je prikazana na slici 9-2 može biti korisno sredstvo)
- Identifikacija i autentifikacija korisnika (pravila konstrukcije lozinke, frekvencija promene lozinke, bezbednosna pitanja, zaboravljeno ime za prijavu ili lozinka, biometrijska identifikacija, zaključavanje računara posle neuspešnih pokušaja pristupa, neprepoznati računar)
- Privatnost podataka (ko može kreirati, videti, menjati, kopirati, štampati i brisati podatke)
- Namerno uništavanje podataka, korupcija ili krađa
- Zaštita od virusa, crva, trojanskih konja, špijunskog softvera, rootkita i drugog zlonamernog softvera
- Zaštitni zid i druga pitanja vezana za bezbednost mreže
- Šifriranje sigurnih podataka

- Izvršenje revizije izvršenih operacija i pokušaja pristupa

Sledi nekoliko primera bezbednosnih zahteva.. Lako je videti kako biste mogli da projektujete testove da biste proverili da li su ovi zahtevi pravilno implementirani.

SEC-1. *Sistem će zaključati korisnički račun nakon četiri uzastopna neuspešna pokušaja prijave u roku od pet minuta.*

SEC-2. *Sistem beleži sve pokušaje pristupa bezbednim podacima od strane korisnika koji nemaju dovoljan nivo privilegija.*

SEC-3. *Korisnik mora da promeni privremenu lozinku koju mu je dodelio službenik za bezbednost u prethodno nekorišćenu lozinku odmah nakon prve uspešne prijave sa privremenom lozinkom.*

SEC-4. *Otključavanje vrata koje je rezultat uspešnog očitavanja sigurnosne značke moraće da vrata budu otključana 8,0 sekundi, sa tolerancijom od 0,5 sekunde.*

SEC-5. *Rezidentni antimalware softver mora garantovati svaki dolazni Internet saobraćaj koji pokazuje karakteristike poznatih ili sumnjivih potpisa virusa.*

SIGURNOST (NASTAVAK)

Sigurnosni zahtevi često potiču iz poslovnih pravila.

SEC-6. *Magnetometar mora otkriti najmanje 99,9 posto zabranjenih predmeta, s lažno pozitivnom stopom koja ne prelazi 1 posto.*

Sigurnosni zahtevi često potiču iz poslovnih pravila, kao što su politike bezbednosti preduzeća, kao što ilustruje sledeći primer:

SEC-7. *Samo korisnici koji imaju privilegovane pristupe revizoru mogu da pregledaju istoriju transakcija.*

Pokušajte da izbegavate pisanje bezbednosnih zahteva sa ugrađenim ograničenjima projekta. Navođenje lozinki za kontrolu pristupa je primer. Pravi zahtev je da se ograniči pristup sistemu ovlašćenim korisnicima; lozinke su samo jedan od načina (iako najčešći način) da se taj cilj postigne. U zavisnosti od toga koji je način provere identiteta korisnika izabran, ovaj bezbednosni zahtev će dovesti do specifičnih funkcionalnih zahteva koji implementiraju metodu autentifikacije.

Sledi nekoliko pitanja koja ćete istražiti kada postavljate bezbednosne zahteve:

- Koji osetljivi podaci moraju biti zaštićeni od neovlašćenog pristupa?
- Ko je ovlašćen za pregled osetljivih podataka? Ko konkretno nije ovlašćen?
- Pod kojim uslovima poslovanja ili vremenskim okvirom rada ovlašćeni korisnici imaju pristup funkcijama?
- Koje provere se moraju izvršiti da bi se potvrdilo da korisnik upravlja aplikacijom u sigurnom okruženju?
- Koliko često virusni softver treba da pretražuje viruse?
- Postoji li određeni način provere autentičnosti korisnika koji se mora koristiti?

UPOTREBLJIVOST

Upotrebljivost je prilagođenost korisnicima, jednostavnost upotrebe i ljudski inženjering

Upotrebljivost se odnosi na bezbroj faktora koji čine ono što ljudi kolokvijalno opisuju kao prilagođenost korisnicima, jednostavnost upotrebe i ljudski inženjering. Analitičari i programeri ne bi trebalo da pričaju o "prijateljskom" softveru, već o softveru koji je osmišljen za efikasno i nenametljivo korišćenje. Upotrebljivost meri napor potreban da se pripremi ulaz za sistem, da se njime upravlja i da se interpretiraju njegovi rezultati.

Upotrebljivost obuhvata nekoliko poddomena izvan očigledne lakoće upotrebe, uključujući lakoću učenja; pamćenje; izbegavanje grešaka, rukovanje i oporavak; efikasnost interakcija; pristupačnost; i ergonomija. Sukobi između tih kategorija mogu nastati. Na primer, lakoća učenja može biti u kontradikciji s lakoćom korišćenja. Postupci koje bi projektant softvera mogao da preduzme da olakša novom ili retkom korisniku da koristi sistem mogu biti iritantne prepreke korisniku energije koji tačno zna šta želi da radi i žudi za efikasnošću. Različite funkcije unutar iste aplikacije takođe mogu imati različite ciljeve upotrebljivosti. Možda je važno da budete u mogućnosti da unosite podatke veoma efikasno, ali i da možete lako da shvatite kako generisati prilagođeni izveštaj.

Kao i kod ostalih atributa kvaliteta, moguće je izmeriti mnoge aspekte „korisnosti“. Pokazatelji upotrebljivosti uključuju:

- Prosečno vreme potrebno određenoj vrsti korisnika da pravilno izvrši određeni zadatak.
- Koliko transakcija korisnik može pravilno izvršiti u određenom vremenskom periodu.
- Koliki procenat skupa zadataka korisnik može pravilno da izvrši bez da mu je potrebna pomoć.
- Koliko grešaka korisnik pravi prilikom dovršavanja zadatka.
- Koliko pokušaja treba korisniku da ispuni određeni zadatak, poput pronalaska određene funkcije zakopane negde u menijima.
- Kašnjenje ili vreme čekanja prilikom obavljanja zadatka.
- Broj interakcija (klikovi miKaša, pritisci tastera, gestovi dodirnog ekrana) potreban da biste došli do informacija ili radi izvršavanja zadatka.

Da bi istražili njihova očekivanja u pogledu upotrebljivosti, poslovni analitičari sistema za praćenje hemikalija postavljali su šampionima proizvoda pitanja poput „*Koliko koraka biste bili spremni da prođete da biste zahtevali hemikaliju?*“ I „*Koliko vremena vam treba da dovršite upotrebu hemikalije*“. Ovo je jednostavna početna tačka za definisanje mnogih karakteristika koje će softver učiniti jednostavnim za upotrebu. Rasprave o upotrebljivosti mogu dovesti do merljivih ciljeva kao što su sledeći:

UP1. *Obučeni korisnik mora biti u mogućnosti da podnese zahtev za hemikalije iz kataloga dobavljača u proseku tri minuta, a u roku od najviše pet minuta, 95 odsto vremena.*

UPOTREBLJIVOST (NASTAVAK)

Neophodno je pažljivo određivanje zahteva za različite dimenzije upotrebljivosti

Upitajte da li novi sistem mora biti u skladu sa bilo kojim standardima ili konvencijama korisničkog interfejsa ili da li njegov korisnički interfejs mora biti u skladu sa standardima drugih često korišćenih sistema. Takav zahtev za upotrebljivost možete navesti na sledeći način:

UP-2. *Sve funkcije u meniju Datoteka treba da imaju definisane tastere za prečice koji koriste kontrolni taster pritisnut istovremeno sa jednom drugom tipkom.*

Komande menija koje se takođe pojavljuju u programu Microsoft Word koristiće iste podrazumevane tipke za prečice koji koristi Word. Takva doslednost upotrebe može vam pomoći da izbegnete one frustrirajuće greške koje nastaju kada prsti navikavaju neku radnju koja ima nešto drugačije značenje u aplikaciji koju ne koristite često. Ciljevi jednostavnosti učenja takođe se mogu kvantifikovati i meriti, kao što sledeći primer pokazuje:

UP-3. *95 procenata hemičara koji nikada ranije nisu koristili sistem za praćenje hemikalija moći će da podnesu zahtev za hemijsko ispravno ne više od 15 minuta orijentacije.*

Pažljivo određivanje zahteva za različite dimenzije upotrebljivosti može pomoći projektantima softvera da donesu izbor koji razlikuje oduševljene korisnike od onih koji koriste aplikaciju sa mrštenjem na licu ili, još gore, od onih koji odbiju uopšte da ga koriste.

▼ Poglavlje 4

Unutrašnji zahtevi kvaliteta

EFIKASNOST

Efikasnost je merilo koliko sistem koristi kapacitet procesora, prostor na disku, memoriju ili opseg komunikacije.

Interni atributi kvaliteta se direktno ne mogu primetiti tokom izvođenja softvera. To su svojstva koja programer ili održavač opaža dok gleda projektno rešenje softvera ili programski kod da bi ga modifikovao, ponovo iskoristio ili premeštao na drugu platformu. Unutrašnji atributi mogu posredno uticati na percepciju kupca o kvalitetu proizvoda ako mu je kasnije teško da dodali novu funkcionalnost ili ako unutrašnja neefikasnost rezultira degradacijom performansi.

Efikasnost (*efficiency*) je usko povezana sa spoljnim atributom kvaliteta performansi. Efikasnost je merilo koliko sistem koristi kapacitet procesora, prostor na disku, memoriju ili opseg komunikacije. Ako sistem troši previše raspoloživih resursa, korisnici će naići na smanjene performanse.

Efikasnost - a samim tim i performanse - je pokretački faktor u arhitekturi sistema, koji utiče na to kako dizajner odabere da distribuira račune i funkcije po komponentama sistema. Zahtevi za efikasnost mogu ugroziti postizanje drugih svojstava kvaliteta. Uzmite u obzir minimalne hardverske konfiguracije prilikom definisanja ciljeva efikasnosti, kapaciteta i performansi. Da biste omogućili inženjerske marže za neočekivane uslove i budući rast (čime utiče na skalabilnost), možete navesti nešto poput sledećeg:

EFF-1. *Najmanje 30 posto kapaciteta procesora i memorije dostupne aplikaciji ne sme se koristiti pri planiranim uslovima najvećeg opterećenja.*

EFF-2. *Sistem će operateru pružiti poruku upozorenja kada opterećenje upotrebe prelazi 80 posto maksimalno planiranog kapaciteta.*

Korisnici neće postaviti zahteve efikasnosti u takvim tehničkim uslovima; umesto toga, oni će razmišljati u smislu vremena odziva ili drugih zapažanja. BA mora postaviti pitanja koja će nadvladati očekivanja korisnika u vezi sa pitanjima poput prihvatljive degradacije performansi, povećanja potražnje i očekivanog rasta. Primeri takvih pitanja su:

- Koliki je maksimalni broj istodobnih korisnika sada i očekuje se u budućnosti?
- Za koliko bi se vremena odziva ili drugi pokazatelji performansi mogli smanjiti pre nego što korisnici ili preduzeće dožive negativne posledice?
- Koliko operacija mora sistem biti sposoban da izvrši istovremeno i u normalnim i u ekstremnim radnim uslovima?

PROMENLJIVOST

Promenljivost se odnosi na to kako se jednostavno projektovanje i kod softvera može razumeti, promeniti i proširiti

Promenljivost (*modifiability*) govori o tome kako se jednostavno projektovanje i kod softvera može razumeti, promeniti i proširiti. Izmenjivanje obuhvata nekoliko drugih termina atributa kvaliteta koji se odnose na različite oblike održavanja softvera, kao što je prikazano u tabeli na slici 1. Promenljivost je usko povezana sa verifikovanjem. Ako programeri predviđaju mnogo poboljšanja, mogu odabrati projektne pristupe koji će povećati izmenljivost softvera. Velika izmenljivost je kritična za sisteme koji će biti podvrgnuti čestoj reviziji, poput onih koji se razvijaju korišćenjem inkrementalnog ili iterativnog životnog ciklusa.

Načini za merenje promene izmena uključuju prosečno vreme potrebno za dodavanje mogućnosti ili rešavanje problema i procenat ispravki koji su pravilno napravljeni. Sistem za praćenje hemikalija uključio je sledeće zahteve za promenljivost:

MOD-1. Programer za održavanje koji ima iskustva sa sistemom biće u mogućnosti da modifikuje postojeće izveštaje u skladu sa revidiranim propisima o hemijskom izveštavanju od savezne vlade sa 10 ili manje sati razvojnog napora.

Na osnovu iskustva usvojili smo smernice za projektovanje kao što je sledeća da bismo programere usmerili u pisanju koda na način koji bi poboljšao razumljivost programa i samim tim održivost:

MOD-2. Funkcijski pozivi ne smeju se ugnijezditi u više od dva nivoa.

BA treba da sarađuje sa programerima za održavanje kako bi razumeli koja svojstva koda bi im olakšala modifikaciju ili ispravljanje oštećenja.

Hardverski uređaji koji sadrže ugrađeni softver često imaju zahteve za podržavanje na terenu. Neki od njih vode do izbora projektnog rešenja softvera, dok drugi utiču na dizajn hardvera. Evo primera takvog zahteva:

SUP-1. Sertifikovani tehničar za popravku mora biti u mogućnosti da zameni modul skenera za najviše 10 minuta.

Zahtevi za podršku mogu takođe da olakšaju život korisnika, kao što ovaj primer ilustruje:

SUP-2. Štampač će prikazati poruku o grešci ako rezervni kertridži nisu uneti u odgovarajuće proreze.

Tip održavanja	Dimenzije promenljivosti	Opis
Korektivno	održljivost, razumljivost	ispravka grešaka
Svršeno	fleksibilnost, proširivost	proširenje i promena funkcionalnosti
Adaptivno	održivost	promena sistema da radi u promenjenom radnom okruženju bey novih sposobnosti
Podrška na terenu	podrška	Korekcija grešaka, servisiranje uređaja

Slika 4.1 Neki aspekti promenljivosti zahteva (Izvor: Wiegers, Software Requirements, 3rd ed., Microsoft, 2013)

PRENOSIVOST

Napor potreban za migraciju softvera iz jednog operativnog okruženja u drugi je mera prenosivosti.

Napor potreban za migraciju softvera iz jednog operativnog okruženja u drugi je mera prenosivosti. Neki stručnjaci uključuju mogućnost internacionalizacije i lokalizacije proizvoda pod naslovom prenosivost. Pristupi projektovanju koji softver čini prenosivim slični su onima koji ga čine ponovo upotrebljivim. Prenosivost (*portability*) postaje sve važnija jer se aplikacije moraju pokrenuti u više okruženja, kao što su Windows, Mac i Linux; iOS i Android; i računari, tableti i telefoni.

Zahtevi za prenosivost podataka su takođe važni. Ciljevi prenosivosti treba da identifikuju one delove proizvoda koji moraju da se premeštaju u druga okruženja i opisuju ta ciljna okruženja. Jedan proizvod za analizu hemikalija bio je u dva veoma različita okruženja. Jedna verzija naletela je u laboratoriju gde je doktorski hemičar koristio softver za kontrolu nekoliko analitičkih instrumenata. Druga verzija radila je u ručnom uređaju koji će na terenu, na primer na naftovodu, koristiti neko ko je imao mnogo manje tehničkog obrazovanja. Osnovne mogućnosti dve verzije bile su uglavnom iste. Takav proizvod mora biti projektovan od samog početka da deluje u obe vrste okruženja sa minimalnom količinom razvojnog rada. Ako programeri znaju za očekivanja kupaca od prenosivosti, mogu odabrati razvojne pristupe koji će na odgovarajući način poboljšati prenosivost proizvoda. Evo nekih zahteva za prenosivost uzoraka:

POR-1. Izmena iOS verzije aplikacije koja će se izvoditi na Android uređajima zahteva promenu ne više od 10 procenata izvornog koda.

POR-2. Korisnik će moći da prenosi obeleživače pregledača u Firefox, Internet Explorer, Opera, Chrome i Safari.

POR-3. Alat za migraciju platforme prenosi prilagođene korisničke profile u novu instalaciju bez potrebe za korisničkim akcijama.

Kada istražujete prenosivost, pitanja poput sledećeg mogu biti korisna:

- Na kojim različitim platformama će ovaj softver trebati da se pokreće, i sada i u budućnosti?
- Koje delove proizvoda treba dizajnirati za veću prenosivost od ostalih delova?
- Koje datoteke podataka, programske komponente ili drugi elementi sistema moraju da budu prenosivi?
- Ako softver postane mnogo prenosivi, koji drugi atributi kvaliteta mogu biti ugroženi?

PONOVNA UPOTREBLJIVOST

Ponovna upotreba ukazuje na relativni napor potreban za preradu softverske komponente za upotrebu u drugim aplikacijama

Ponovna upotreba ukazuje na relativni napor potreban za pretvaranje softverske komponente za upotrebu u drugim aplikacijama. Softver za višekratnu upotrebu mora biti modularan, dobro dokumentovan, nezavisno od posebne aplikacije i operativnog okruženja i donekle

generičan po mogućnostima. Brojni artefakti projekta nude potencijal ponovne upotrebe, uključujući zahteve, arhitekture, nacрте, kod, testove, poslovna pravila, modele podataka, opise korisničkih klasa, profile zainteresovanih strana i pojmovnike (vidi Poglavlje 18, „Ponovna upotreba uslova“). Popravka softvera za višekratnu upotrebu olakšava se temeljnim određivanjem zahteva i dizajna, rigoroznim pridržavanjem standarda kodiranja, održavanjem regresijskog paketa test slučajeva i održavanjem standardne biblioteke komponenata za višekratnu upotrebu. Ciljeve ponovne upotrebe teško je kvantifikovati. Navedite koji elementi novog sistema moraju biti konstruisani na način koji olakšava njihovu ponovnu upotrebu, ili odredite komponente za višekratnu upotrebu koje bi trebalo kreirati kao odvajanje od projekta. Evo nekoliko primera:

REU-1. *Funkcije unosa hemijske strukture ponovo će se koristiti na nivou objektnog koda u drugim aplikacijama.*

REU-2. *Najmanje 30 posto arhitekture aplikacija ponovo će se koristiti iz odobrenih referentnih arhitektura.*

REU-3. *Algoritmi za cene ponovo će se koristiti u budućim aplikacijama za upravljanje prodavnicama.*

Razmislite o sledećim pitanjima kada pokušavate da saznate o zahtevima za ponovnu upotrebu za svoj projekat:

- Koji postojeći zahtevi, modeli, dizajnerske komponente, podaci ili testovi mogu se ponovo koristiti u ovoj aplikaciji?
- Koja funkcionalnost dostupna u povezanim aplikacijama može ispuniti određene zahteve za ovu aplikaciju?
- Koji delovi ove aplikacije nude dobar potencijal za ponovnu upotrebu na drugom mestu?
- Koje posebne akcije treba preduzeti kako bi se olakšalo korišćenje delova ove aplikacije?

SKALABILNOST

Zahtevi za skalabilnost odnose se na mogućnost aplikacije da raste kako bi se prilagodio većem broju korisnika.

Zahtevi za skalabilnost odnose se na mogućnost aplikacije da raste kako bi se prilagodio većem broju korisnika, podataka, servera, geografskih lokacija, transakcija, mrežnog saobraćaja, pretraga i drugih usluga bez ugrožavanja performansi ili tačnosti. Skalabilnost (*scalability*) ima i hardverske i softverske implikacije. Smanjivanje sistema može značiti nabavku bržih računara, dodavanje memorije ili prostora na disku, dodavanje servera, paralelne baza podataka ili povećanje mrežnog kapaciteta. Softverski pristupi mogu uključivati distribuciju računanja na više procesora, komprimovanje podataka, optimizaciju algoritama i druge tehnike podešavanja performansi. Skalabilnost je povezana sa izmenljivošću i robusnošću, jer jedna kategorija robusnosti ima veze sa ponašanjem sistema kada se približe ili premaše ograničenja kapaciteta. Evo nekoliko primera zahteva za skalabilnost:

SCA-1. *Kapacitet telefonskog sistema za hitne slučajeve mora biti u mogućnosti da se poveća sa 500 poziva dnevno na 2500 poziva dnevno u roku od 12 sati.*

SCA-2. *Veb lokacija će moći da obrađuje stopu rasta stranice 30 procenata po kvartalu tokom*

najmanje dve godine bez degradacije performansi koje će opaziti korisnik.

SCA-3. *Sistem distribucije mora biti u stanju da primi do 20 novih skladišnih centara.*

Poslovni analitičar možda nema dobar osećaj za buduće planove za širenje za određenu aplikaciju. Možda će joj trebati da sarađuje sa sponzorom projekta ili ekspertima za teme kako bi stekla osećaj koliko bi baza korisnika, količina podataka ili drugi parametri mogli da narastu tokom vremena. Sledeća pitanja bi mogla biti od pomoći tokom tih diskusija:

- Koje su vaše procene o broju ukupnih i istodobnih korisnika sa kojima sistem mora biti u mogućnosti da se bavi u narednih nekoliko meseci, kvartala ili godina?
- Možete li opisati kako i zašto bi zahtevi za sistemom podataka u budućnosti mogli da porastu?
- Koji su minimalni prihvatljivi kriterijumi performansi koji moraju biti zadovoljeni bez obzira na broj korisnika?
- Koji su planovi rasta dostupni u pogledu koliko servera, podaktovnih centara ili pojedinačnih instalacija za koje se očekuje da će sistem raditi?

PROVERLJIVOST

Proverljivost se odnosi na to koliko dobro softverske komponente ili integrisani proizvod mogu da se procene kako bi se pokazalo da li sistem funkcioniše

Uže nazvano proveravanjem, proverljivost (*verifiability*) se odnosi na to koliko dobro softverske komponente ili integrisani proizvod mogu da se procene kako bi se pokazalo da li sistem funkcioniše onako kako se očekuje. Projektovanje za proverljivost je kritičan ako proizvod ima složene algoritme i logiku ili ako sadrži suptilne međusobne veze u funkcionalnosti. Proverljivost je takođe važna ako će se proizvod često menjati, jer će biti podvrgnut čestim regresionim testiranjima kako bi se utvrdilo da li su promene oštetile bilo koju postojeću funkcionalnost. Sistemi sa visokom proverljivošću mogu se testirati i efikasno. Dizajniranje softvera za verifikovanje podrazumeva olakšavanje postavljanja softvera u željeno predtestovno stanje, pružanje potrebnih test podataka i posmatranje rezultata testa. Evo nekoliko primera zahteva za verifikovanje:

VER-1. *Konfiguracija razvojnog okruženja mora biti identična okruženju test konfiguracije kako bi se izbegli nepodnošljivi kvarovi ispitivanja.*

VER-2. *Tester mora biti u mogućnosti da konfiguriše koji se rezultati izvođenja beleže tokom testiranja.*

VER-3. *Programer će biti u mogućnosti da postavi računski modul da prikazuje privremene rezultate bilo koje određene grupe algoritama za potrebe uklanjanja pogrešaka.*

VER-4. *Maksimalna ciklomatska složenost modula ne smije biti veća od 20*

Ciklomatička složenost (*cyclomatic complexity*) je merilo broja logičkih grana u modulu izvornog koda. Dodavanje više grana i petlji u modul otežava razumevanje, testiranje i održavanje. Projekat neće biti neuspeh ako je neki modul imao ciklomatsku složenost od 24, ali dokumentovanje takvih smernica pomoglo je programerima da postignu željeni cilj

kvaliteta. Definisanje zahteva za proverljivost može biti teško. Istražite pitanja poput sledećeg:

- Kako možemo potvrditi da konkretni proračuni daju očekivane rezultate?
- Postoje li delovi sistema koji ne daju determinirane izlaze, tako da je teško utvrditi da li rade ispravno?
- Da li je moguće smisliti skupove podataka s testovima koji imaju veliku verovatnoću da otkriju greške u zahtevima ili njihovoj primeni?
- Koje referentne izveštaje ili druge izlaze možemo da koristimo da potvrdimo da sistem pravilno proizvodi svoje izlaze?

VIDEO 14 - NONFUNCTIONAL REQUIREMENTS - WIEGERS (VIDEO)

Trajanje: 4:28 minuta

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

VIDEO 18 - QUALITY ATTRIBUTES - WIEGERS (VIDEO)

Trajanje: 6:36 minuta

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

▼ Poglavlje 5

Vežba

ZADATAK ZA VEŽBU 1

Tekst zadatka za vežbu 1

ZADATAK 1.

U nastavku su dati korisnički zahtevi za različite sisteme:

- Korisnik može pretraživati letove na osnovu destinacije i prevoznika
- Sistem treba da svakog dana generiše, za svaku kliniku, spisak pacijenata koji će tog dana biti pregledani
- Korisnik treba da bude obavešten o novim porukama u aplikaciji čim ponovo uključi internet
- Dva dana nakon održanog ispitnog roka, svi profesori univerziteta moraju da upišu ocene
- U sistemu treba obezbedi da se podaci štite od neautorizovanog pristupa
- Svaki član osoblja koji koristi sistem mora biti jedinstveno identifikovan svojim matičnim brojem
- Bibliotečki sistem treba da bude lak za korišćenje
- Svim korisnicima banke treba svakog prvog u mesecu da stigne automatski mejl obaveštenja sa mesečnim izvodom za prethodni mesec
- Samo direktor firme sme da generiše finansijske izveštaje
- Veb sajt treba da bude namenjen grafičkim dizajnerima koji traže posao
- Korisnik ima pravo da pogreši šifru tri puta, a nakon toga mu se zaključava nalog
- Profesor treba da vidi na kojim predmetima drži nastavu
- Pored veb sajta, treba da postoji i mobilna aplikacija za kupovinu koja će da radi na svim mobilnim uređajima
- Na stranici za pregled preparata, korisnik treba da vidi pun naziv, cenu i sastav preparata
- U editoru za kreiranje vesti, korisniku treba da bude omogućeno da koristi sve standardne prečice na tastaturi

a) Probajte da izdvojite one koji se odnose na attribute kvaliteta. (10 min)

b) One koji se odnose na attribute kvaliteta, preformulišite tako da zaista budu izraženi kao nefunkcionalni zahtevi. (15 min)

ZADACI ZA VEŽBU 2-4

Tekst zadataka za vežbu 2-4

ZADATAK 2.

Sugerišite kako bi sledeće korisničke zahteve trebalo preformulisati kako bi oni bili izraženi kvantitativno, odnosno da budu proverljivi. Možete koristiti bilo koju metriku za iskazivanje zahteva. (15 min)

- Bibliotečki sistem treba da bude lak za korišćenje
- Bibliotečki sistem treba da obezbedi pouzdane servise za sve vrste korisnika
- Bibliotečki sistem treba da obezbedi brz odgovor za sve korisnike koji žele da dobiju informaciju o knjigama
- Bibliotečki sistem treba da bude uvek dostupan, pogotovo osoblju koje radi u prepodnevnoj smeni
- Bibliotečki sistem ne sme da pravi probleme kada bibliotekar treba da registruje nove korisnike ili zaduži stare korisnike
- Bibliotečki sistem ne sme da pogreši kada zadužuje korisnike za prekoračenje u zadržavanju knjiga
- Bibliotečki sistem treba da bude dopadljiv korisnicima koji žele da dobiju informacije o knjigama pre nego što dođu da se registruju
- Bibliotečki sistem treba da bude tako jednostavan da svaki nov radnik može da se snađe

ZADATAK 3.

Pogledajte attribute kvaliteta iz tabele na slici 1 u poglavlju 2. Formirajte nekoliko pitanja o svakom atributu koji bi pomogli korisnicima da jasno iznesu svoja očekivanja. (10 min)

ZADATAK 4.

Na osnovu odgovora iz prethodnog pitanja, a na primeru onlajn prodavnice sportske opreme i suplemenata, napišite po jedan ili dva nefunkcionalna zahteva za attribute kvaliteta iz prethodne tabele. Uradite isto i za sistem za upravljanje radom taksi udruženja. (20 min)

ZADATAK ZA VEŽBU 5

Tekst zadatka za vežbu 5

ZADATAK 5.

Pogledajte na sledećoj slici, koja ukazuje na pozitivne i negativne odnose između atributa kvaliteta. Možete li da objasnite zbog čega su neki od atributa u pozitivnom, a drugi u negativnom odnosu? (5 min)

	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability								+	+							
Efficiency	+			-	-	+	-			-		+		-		
Installability	+							+					+			
Integrity			-		-				-		+		+	-	-	
Interoperability	+		-	-		-	+	+		+	-		-			
Modifiability	+		-			-	+	+			+				+	
Performance		+		-	-		-			-		-		-		
Portability		-		+	-	-			+				-	-	+	
Reliability	+	-	+	+	+	-			+	+	+	+	+	+	+	
Reusability		-	-	+	+	-	+							-	+	
Robustness	+	-	+	+	+	-	+	+		+	+	+	+	+	+	
Safety		-	+	+		-				+		+		+	-	-
Scalability	+	+	+			+	+	+		+						
Security	+		+	+		-	-	+		+	+			-	-	
Usability		-	+			-	-	+		+	+					-
Verifiability	+		+	+	+			+	+	+	+	+	+	+	+	

Slika 5.1 Pozitivni i negativni odnose između atributa kvaliteta

Kada imate više stejkholdera, čiji su zahtevi sukobljeni, kako ćete odlučiti čije ćete zahteve uzeti u obzir? Na koji način ćete sprovesti usaglašavanje interesa i zahteva? (5 min)

▼ Poglavlje 6

Domaći zadatak

DOMAĆI ZADATAK 11

Tekst domaćeg zadatka

Za sistem na kome radite od DZ01, odredite nefunkcionalne zahteve zasnovane na atributima kvaliteta koji su analizirani u ovoj lekciji. Opredelite se za dva atributa kvaliteta i navedite odgovarajuće zahteve za koje ste procenili da je neophodno zadovoljiti ih.

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 7

Projektni zadatak

ZADATAK ZA RAD NA PROJEKTU

Tekst zadatka za rad na projektu

Proučite **Poglavlje 6 - Atributi kvaliteta** u uzorku SRS dokumenta.

U svom SRS dokumentu, na kome radite već tri nedelje, popunite poglavlje 6. Pažljivo razmotrite koje attribute kvaliteta treba da poseduje sistem na kome radite. Vodite računa o tome da te nefunkcionalne zahteve izrazite kvantitativno, odnosno tako da budu merljivi i proverljivi. Odredite svoje zahteve za atributima kvaliteta prema funkcionalnim zahtevima, različitim ograničenjima i pretpostavkama koje ste već definisali. Niste ograničeni predlogom i brojem nefunkcionalnih zahteva koji su dati u uzorku – navedite samo one koji su relevantni za vaš projekat.

▼ Poglavlje 8

Zaključak

ZAKLJUČAK

1. Jedan od načina klasifikacije kvaliteta razlikuje one karakteristike koje su vidljive izvršenjem softvera (spoljni kvalitet) od onih koji nisu (unutrašnji kvalitet).
2. Spoljni faktori kvaliteta su prvenstveno važni za korisnike, dok su unutrašnji kvaliteti važniji za osoblje za razvoj i održavanje
3. U stvarnosti, postoje kompromisi i sukobi između određenih atributa koji onemogućavaju da ih istovremeno maksimalno iskoristite.
4. Poređenja sa rangiranjem u parovima se mogu efikasno raditi sa malom spiskom atributa
5. Kada planira sesiju za izazivanje, BA treba da započne sa spiskom pitanja i da ga svede na ona pitanja koja su najvažnija za projekat.
6. Zahtevi za kvalitetom moraju biti merljivi kako bi se uspostavio precizan dogovor o očekivanjima između BA, kupaca i razvojnog tima
7. Spoljni atributi kvaliteta opisuju karakteristike koje se primećuju prilikom izvršavanja softvera. Oni duboko utiču na korisničko iskustvo i korisnikovo opažanje kvaliteta sistema.
8. Dostupnost je merilo planiranog vremena trajanja u kome su usluge sistema dostupne za upotrebu i u potpunosti rade
9. Lakoća instaliranja
10. Integritet se bavi sprečavanjem gubitka informacija i očuvanjem ispravnosti podataka.
11. Integritet se bavi sprečavanjem gubitka informacija i očuvanjem ispravnosti podataka.
12. Interoperabilnost ukazuje na to koliko lako sistem može da razmenjuje podatke i usluge sa drugim softverskim sistemima i koliko lako se može integrisati sa spoljnim hardverskim uređajima
13. Performanse predstavljaju odgovor sistema na različite korisničke upite i radnje.
14. Pouzdanost je verovatnoća da se softver radi bez kvara tokom određenog vremenskog perioda.
15. Robusnost je stepen do koga sistem i dalje pravilno funkcioniše kada se suoči sa nevažecim ulazima, i oštećenjima.
16. Bezbednosni zahtevi bave se potrebom da se spreči sistem da nanosi bilo kakvu povredu ljudi ili materijalnu štetu.
17. Sigurnost se bavi blokiranjem neovlaštenog pristupa sistemskim funkcijama ili podacima osiguravajući da je softver zaštićen od napada zlonamernog softvera i tako dalje
18. Upotrebljivost je prilagođenost korisnicima, jednostavnost upotrebe i ljudski inženjering

19. Efikasnost je merilo koliko sistem koristi kapacitet procesora, prostor na disku, memoriju ili opseg komunikacije.
20. Promenljivost se odnosi na to kako se jednostavno projektovanje i kod softvera može razumeti, promeniti i proširiti

ZAKLJUČAK (NASTAVAK)

21. Napor potreban za migraciju softvera iz jednog operativnog okruženja u drugi je mera prenosivosti.
22. Ponovna upotreba ukazuje na relativni napor potreban za preradu softverske komponente za upotrebu u drugim aplikacijama
23. Zahtevi za skalabilnost odnose se na mogućnost aplikacije da raste kako bi se prilagodio većem broju korisnika.
24. Proverljivost se odnosi na to koliko dobro softverske komponente ili integrisani proizvod mogu da se procene kako bi se pokazalo da li sistem funkcioniše onako kako se očekuje

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 14 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,