UNIVERZITET METROPOLITAN BEOGRAD

SE322 - Analiza zahteva za softver

Lekcija 01

NAJČEŠĆE POSTAVLJANA PITANJA VEZANA ZA ZAHTEVE SISTEMA

Metropolitan

PRIRUČNIK ZA STUDENTE

SE322 - Analiza zahteva za softver

Lekcija 01

NAJČEŠĆE POSTAVLJANA PITANJA VEZANA ZA ZAHTEVE SISTEMA

Najčešće postavljana pitanja vezana za zahteve sistema

UVOD

Šta ćemo naučiti u ovoj lekciji?

Cilj ovog kursa je predstaviti procese koji su obuhvaćeni izborom, analizom, validacijom i upravljanjem zahteva za izgradnjom kompleksnih softverskih sistema. Prvih nekoliko predavanja je fokusirano na pitanje **„šta”** je obuhvaćeno analizom zahteva i specifikacijom sistema dok će u sledećim predavanjima biti reči o tome **„kako”** se u okviru svakog od ovih procesa mogu primeniti specifične tehnike.
U ovom predavanju će biti predstavljene notacije koje se koriste u predstavljanju sistemskih zahteva. Biće dati primeri nekih zahteva i odgovori na “pitanja koja se najčešće postavljaju ” (FAQ) o zahtevima. O specifikaciji zahteva se diskutuje u kontekstu šireg procesa inženjeringa zahteva koji se odnosi na razvoj sistema kao celine (softvera, hardvera, procesa). Na kraju će biti reči o organizaciji dokumentacije vezane za zahteve; biće jasno opisani standardi koji se koriste u ovoj oblasti i date smernice za pisanje jasnih i konciznih zahteva.

Napomena: Predavanje sastavljeno na osnovu sadržaja knjige: Gerald Kotonya and Ian Sommerville, Requiremets Engineering, processes and techniques, John Wiley & sons

Vezba napisana na osnovu Case study iz knjige: Dean Leffingwell, Don Widrig, Managing Software Requirements, Publisher: Addison Wesley, First Edition October 22, 1999

Poglavlje 1

Problemi vezani za zahteve sistema

ŠTA SU RAZLOZI ZA NASTANAK PROBLEMA U RAZVOJU SW SISTEMA?

Najčešći razlozi su poteškoće vezane za zahteve sistema

Razvoj softverskih sistema je vezan sa različitim problemima: oni se često isporučuju sa zakašnjenjem i prevazilaze planirani budžet. Ne rade ono što korisnici zaista žele i često se nikada ne iskoriste neke mogućnosti sistema za koje su korisnici platili.

Najčešći razlozi za takve probleme su poteškoće vezane za zahteve sistema.

- Kao što samo ime kaže, zahtevi sistema definišu šta se od sistema traži da radi i okolnosti pod kojima se traži da to radi. Drugim rečima, zahtevima se definišu servisi koje sistem treba da pruži i ograničenja koja se odnose na rad sistema.
- Obzirom da postoje različiti tipovi zahteva, nemoguće je opisati standardni način za opisivanje zahteva ili definisanje najboljeg načina da se oni specificiraju. To zavisi od toga ko zahteve specificira, ko će zahteve najverovatnije pročitati, prakse organizacije koja postavlja zahteve i aplikativnog domena sistema.
- Opis zahteva sistema može da uključi i druge informacije koje se moraju uzeti u obzir pri implementaciji sistema ali koje nisu iskazane u obliku servisa ili ograničenja sistema. Te informacije se mogu odnositi na opšte informacije o tipu sistema koji se specificira (informacioni domen), informacije o standardima koji se moraju pratiti prilikom razvoja sistema, informacije o drugim sistemima koji su u interakciji sa datim sistemom itd.

NAJČEŠĆI PROBLEMI U ZAHTEVIMA SISTEMA

Zahtevi ne odslikavaju realne potrebe, nekonzistentni su i nekompletni, nisu usaglašeni itd.

Najčešći problemi koji se javljaju sa zahtevima su:

- zahtevi ne odslikavaju realne potrebe korisnika
- zahtevi su nekonzistentni i/ili nekompletni
- zahteve je skupo menjati pošto je oko njih postignuto slaganje
- postoji nerazumevanje između korisnika, onih koji specificiraju zahteve i inženjera zaduženih za razvoj i održavanje sistema

Problem pisanja zahteva je univerzalni i nikada neće biti pronađeno kompletno rešenje problema vezanih za njega. Preporuke koje će biti date u okviru ovog kursa mogu da smanje broj problema i minimiziraju njihov uticaj na kompletan sistem.

Poglavlje 2

Najčešće postavljana pitanja vezana za zahteve sistema

ŠTA SE DOBIJANJEM ODGOVORA NA NAJČŠĆE POSTAVLJENJA PITANJE MOŽE POSTIĆI?

Dobijanje osnovnih informacija i definicija vezanih za razumevanje zahteva

Kroz najčešće postavljena pitanja vezana za zahteve sistema će biti predstavljene osnovne informacije i definicije vezane za razumevanje zahteva.

1. **Šta su zahtevi?**
2. **Šta je inženjering zahteva ? **
3. **Koliko košta inženjering zahteva ?**
4. **Šta se dešava kada su zahtevi loši ?**
5. **Šta je proces inženjeringa zahteva ?**
6. **Postoji li idealan proces inženjeringa zahteva ?**
7. **Šta je dokument zahteva ?**
8. **Šta su stakeholderi ?**
9. **Kakva je veza između zahteva i dizajna ?**
10. **Šta je upravljanje zahtevima ?**

ŠTA SU ZAHTEVI?

Može se reći da zahtevi sadrže mešavinu informacija o problemima, ponašanju sistema, njegovim svojstvima, dizajnu i ograničenjima u njegovoj izradi.

Zahtevi se definišu u ranim fazama razvoja sistema kao specifikacija onoga što treba implementirati. Oni opisuju kako sistem treba da se ponaša, informacioni domen aplikacije, ograničenja rada sistema,

specificiraju svojstva ili atribute sistema. Ponekad oni predstavljaju ograničenja u procesu razvoja sistema. Tako se zahtevi mogu opisati kao:

- Mogućnosti sistema na nivou korisnika(npr. “u word procesor mora da bude uključena i provera spellinga i komande za korekjciju”)
- Veoma uopštena svojstva sistema(npr. “sistem treba da omogući da personalne informacije nikada ne mogu postati dostupne bez prethodne autorizacije“)
- Specifična ograničenja sistema(npr. “senzor se mora puniti 10 puta u sec.“)
- Način na koji treba izvršiti neka izračunavanja(npr. “ukupna ocena se izračunava dodavanjem ocena sa ispita, projekata i zalaganja studenta na osnovu sledeće formule ukupna_ocena = ocena_sa_ispita + 2*ocena_sa_projekta + 2/3*ocena_sa_zalaganja “)
- Ograničenja koja se odnose na razvoj sistema(npr. “sistem mora biti razvijen korišćenjem Ade“)

Neki ljudi sugeriraju da zahteve treba tretirati kao naredbe onoga što sistem treba da radi a ne i kako to treba da radi. To je atraktivna ideja ali je u praksi nije lako sprovesti. Zbog toga se može reći da zahtevi sadrže mešavinu informacija o problemima, ponašanju sistema, njegovim svojstvima, dizajnu i ograničenjima u njegovoj izradi. To može proizvesti poteškoće, jer dizajn i ograničenja koja se odnose na njegovu izradu mogu biti u sukobu sa drugim zahtevima. To je realnost i procesom analize sistema i specifikacije zahteva treba uključiti aktivnosti za pronalaženje i rešavanje problema koji nastaju iz te činjenice.

ŠTA JE INŽENJERING ZAHTEVA ?

Relativno nov termin koji je smišljen sa namerom da pokrije sve aktivnosti obuhvaćene otkrivanjem, dokumentovanjem i održavanjem zahteva za sisteme bazirane na radu računara

Inženjering zahteva je relativno nov termin koji je smišljen sa namerom da pokrije sve aktivnosti obuhvaćene otkrivanjem, dokumentovanjem i održavanjem zahteva za sisteme bazirane na radu računara.Korišćenje termina inženjering podrazumeva sistematsko i višestruko korišćenje tehnika kojima se obezbeđuje da zahtevi sistema budu kompletni, konzistentni, relevantni itd. Inženjering zahteva ima mnogo zajedničkog sa sistem analizom – analizom i specifikacijom poslovnog sistema. Razlika je u tome što u praksi, sistem analizu prvenstveno treba fokusirati na poslovanje a ne na sistem, mada se kao i inženjering zahteva, ona često odnosi i na jedno i na drugo.

U ovom kursu se naglasak stavlja na proces inženjeringa zahteva za softverske sisteme. Procesi i metodi koji se koriste za razvoj i analizu softverskih zahteva su generalno primenljivi i na sistemske zahteve tj. zahteve koji se primenjuju na sistem kao celinu a ne samo na softverske komponente sistema.

KOLIKO KOŠTA INŽENJERING ZAHTEVA ?

Za velike softversko/hardverske sisteme, oko 15% ukupnog budžeta; Za manje sisteme koji su pretežno softverski iznose oko 10% ukupnog budžeta.

To zavisi od tipa i veličine sistema koji se razvija i aktivnosti koje su uključene u inženjering zahteva. U nekim slučajevima, sistemski zahtevi se ne razvijaju detaljno; u drugim se moraju proizvesti formalne specifikacije.

Na osnovu daljih istraživanja se može reći da za velike softversko/ hardverske sisteme, oko 15% ukupnog budžeta se troši na aktivnosti inženjeringa zahteva. To uključuje troškove detaljne specifikacije sistema. Za manje sisteme koji su pretežno softverski, troškovi zahteva su manji od ovoga i iznose oko 10% ukupnog budžeta.

ŠTA SE DEŠAVA KADA SU ZAHTEVI LOŠI ?

Loši sistemski zahtevi izazivaju brojne konsekvence

Loši sistemski zahtevi izazivaju brojne konsekvence:

- Sistem se može isporučiti kasnije i koštati više nego što se prvobitno očekivalo.
- Kupci i krajnji korisnici nisu zadovoljni sistemom. Može se desiti da ne koriste sve mogućnosti sistema ili ga mogu čak i odbaciti kao neupotrebljiv.
- Sistem se može pokazati kao nepouzdan, i tokom njegovog rada se mogu pojaviti greške
- Ukoliko se nastavi sa korišćenjem sistema, njegovo održavanje je obično veoma skupo.

Troškovi pronalaženja grešaka zbog loših zahteva su obično mnogo veći što je proces razvoja kasniji. Problema nastali zbog loše definisanih

zahteva mogu prouzrokovati ponovni dizajn, implementaciju i testiranje sistema. Sami tim i troškovi su veći. Utvrđeno je da troškovi pronalaženja grešaka u zahtevima mogu biti i do 100 puta veći od troškova fiksiranja jednostavnih programskih grešaka.

ŠTA JE PROCES INŽENJERINGA ZAHTEVA ?

Struktuirani skup aktivnosti koje se izvršavaju s ciljem da se generiše, validira i održava dokument zahteva

Proces inženjeringa zahteva je struktuirani skup aktivnosti koje se izvršavaju s ciljem da se generiše, validira i održava dokument zahteva.Aktivnosti uključuju izbor zahteva, analizu zahteva, usaglašavanje i validaciju zahteva. Kompletan opis procesa uključuje nabrajanje aktivnosti koje se izvršavaju, struktuiranje ili raspoređivanje tih aktivnosti, definisanje odgovornosti za svaku aktivnost, opis ulaza i izlaza za svaku aktivnost i alata koji se koriste kao podrška.

Veoma mali broj organizacija ima eksplicitno definisan i standardizovan proces inženjeringa zahteva. Ljudi koji su uključeni u proces su odgovorni da donesu odluku šta treba raditi i kada to treba raditi, koje su informacije potrebne, koje alate treba koristiti.

POSTOJI LI IDEALAN PROCES INŽENJERINGA ZAHTEVA ?

Ne postoji jedinstveni proces koji odgovara svim organizacijama. Svaka organizacija mora da razvije svoj sopstveni proces;

Ne postoji jedinstveni proces koji odgovara svim organizacijama. Svaka organizacija mora da razvije svoj sopstveni proces koji odgovara tipu sistema koji se razvija, kulturi organizacije, nivou iskustva i mogućnosti ljudi koji su uključeni u inženjering zahteva. Postoji mnogo mogućih načina za organizovanje procesa zahteva sistema i oni se ne mogu transformisati iz jedne u drugu organizaciju. Da bi se definisao dobar proces inženjeringa zahteva, organizacije treba da uključe ljude koji su stvarno obuhvaćeni inženjeringom zahteva, a oni mogu zatražiti pomoć od konsultanata sa strane.

Većina standarda koji su razvijeni za potrebe inženjeringa zahteva se odnose na izlazne procese kao što je struktuiranje dokumenata zahteva.

ŠTA JE DOKUMENT ZAHTEVA ?

Oficijelni opis sistemskih zahteva za kupce, krajnje korisnike i one koji će razvijati softver

Dokument zahteva je oficijelni opis sistemskih zahteva za kupce, krajnje korisnike i one koji će razvijati softver. Zavisno od organizacije, dokument zahteva može imati različita imena kao što su „funkcionalna specifikacija”, „definicija zahteva”, „specifikacija softverskih zahteva” (SRS). U ovom kursu će se koristiti termin „dokument zahteva”.

ŠTA SU STAKEHOLDERI ?

Ljudi ili organizacije na koje sistem ima uticaja i koji s druge strane imaju posredan ili neposredan uticaj na sistemske zahteve.

Stakeholderi sistema su ljudi ili organizacije na koje sistem ima uticaja i koji s druge strane imaju posredan ili neposredan uticaj na sistemske zahteve. Oni uključuju krajnje korisnike sistema, menadžere i druge koji su uključeni u procese organizacije na koje sistem ima uticaja, inženjere koji su odgovorni za razvoj i održavanje sistema, kupce koji će koristiti sistem kako bi od njega dobili odgovarajuće usluge, eksterna tela kao što su vladine organizacije koje su zadužene za donošenje zakona ili sertifikata itd.

Na primeru razvoja jednog automatizovanog sistema za signalizaciju na železnici, mogući stakeholderi su:

- operatori u železničkim kompanijama koji su odgovorni za obavljanje signalizacije
- društvo železničara
- menadžeri na železnici
- putnici
- inženjeri koji su zaduženi za instalaciju i održavanje uređaja
- vlasti koje su odgovorne za izdavanje sertifikata koji se odnose na sigurnost

Potrebno je identifikovati značajne stakeholdere sistema i otkriti njihove zahteve. Ako se to ne uradi, može se desiti da oni za vreme razvoja sistema ili pošto je sistem isporučen, insistiraju na promenama.

KAKVA JE VEZA IZMEĐU ZAHTEVA I DIZAJNA ?

U ovom materijalu se ne podržava mišljenje da se inženjering zahteva odnosi na ono što treba uraditi a dizajn na ono kako to treba uraditi. Za to postoji mnogo razloga

Često između zahteva i dizajna postoji veoma složena veza. Neki autori sugeriraju da su to zasebne aktivnosti; zahtevi se uglavnom odnose na probleme koje treba rešiti; dizajn se odnosi na rešenje tih problema. Drugim rečima, inženjering zahteva se odnosi na ono što treba uraditi; dizajn se odnosi na ono kako to treba uraditi.

U ovom materijalu se ne podržava ovakvo mišljenje. Bilo bi lepo da je to zaista tako i posao ljudi koji rade na specifikaciji zahteva kao i dizajnera bi bio mnogo lakši. Međutim, u stvarnosti, inženjering zahteva i dizajn su međusobno zavisne aktivnosti. Za to postoji mnogo razloga:

- Sistemi se uvek instaliraju u nekom okruženju, i danas u tom okruženju gotovo uvek postoje i drugi sistemi. Ti drugi sistemi su obično predstavljaju ograničenja za dizajn sistema. Na primer, ograničenje za dizajn novog sistema može biti da sistem koji se razvija mora da svoje informacije dobija iz postojeće baze podataka. Ona je već dizajnirana i deo njegove specifikacije dizajna mora biti uključen u dokument zahteva.
- U okviru velikih sistema, neophodno je identifikovati podsisteme i njihove međusobne veze što se karakteriše kao arhitekturalni dizajn. U tom slučaju, zahtevi se definišu na nivou identifikovanih podsistema a ne čitavog sistema. Identifikacija podsistema znači da se procesi inženjeringa zahteva za svaki podsistem mogu paralelno odvijati.
- Zbog uštede budžeta, vremena ili kvaliteta, organizacija može da pri implementaciji novog sistema redizajnira deo ili čitav postojeći softverski sistem. Ovo ograničava kako sistemske zahteve tako i dizajn.
- Ako sistem treba da bude prihvaćen od strane nekog eksternog regulatora (npr. sistem u civilnom inženjerstvu), može biti neophodno koristiti standardni, „sertifikovani“ dizajn koji je već testiran u drugim sistemima.

ŠTA JE UPRAVLJANJE ZAHTEVIMA ?

Proces upravljanja promenama u zahtevima sistema

Upravljanje zahtevima je proces upravljanja promenama u zahtevima sistema. Zahtevi sistema se uvek menjaju kako bi odslikali promene potreba skateholdera, promene u okruženju u kojem je sistem instaliran, promene u preduzeću koje planira da instalira sistem, promene u zakonskoj regulativi itd. Tim promenama mora da se upravlja, kako bi one imale ekonomski smisao i doprinele poslovnim potrebama organizacije koja kupuje sistem. Mora se oceniti tehnička izvodljivost predloženih promena i omogućiti da se one odvijaju u okviru predviđenog budžeta i vremena.

Osnovne aktivnosti upravljanja zahtevima su kontrola promena i ocena uticaja promena. Kontrola promena se odnosi na utvrđivanje i izvršavanje formalnih procedura za sakupljanje, verifikovanje i ocenjivanje promena; ocena uticaja promena se odnosi na procenu kako će predložene promene uticati na sistem kao celinu. Kada se promene odnose na specifične zahteve, važno je proveriti na koje će druge zahteve verovatno te promene uticati. Upravljanje zahtevima zahteva beleženje informacija o npr. vezama između zahteva, izvoru zahteva i dizajnu sistema. O upravljanju zahtevima će se govoriti detaljnije u narednim predavanjima.

Poglavlje 3

Inženjering sistema

KATEGORIJE SISTEMA KOJI SE BAZIRAJU NA RAČUNARIMA

(1) sistemi koje konfigurišu sami korisnici i (2) korisnički ili naručeni sistemi

Ovaj kurs je fokusiran na inženjering softverskih zahteva, ali za mnoge tipove sistema je nemoguće zahteve za softverom izdvojiti od zahteva sistema kao celine. Pored softvera, sistem može da uključuje i računarski hardver, druge tipove hardverskih uređaja koji se koriste kao veza između računara i operativnih procesa prilikom instaliranja sistema u nekom radnom okruženju.

Sistemi koji se baziraju na računarima spadaju u dve široke kategorije:

- Sisteme koje konfigurišu sami korisnici tj. kod kojih kupac generiše sistem na osnovu postojećih softverskih proizvoda. Velika većina sistema koji se koriste na personalnim računarima je tog tipa (npr. word processor, paketi za crtanje itd.) U tom slučaju „sistemsko inženjerstvo" je u nadležnosti prodavca softvera koji se instalira na računaru opšte namene. Normalno, u tom slučaju nema eksplicitnih softverskih zahteva. Softverske zahteve kreiraju kompanije koje razvijaju različite softverske proizvode na osnovu zahteva korisnika i na osnovu svojih percepcija šta se može prodati na tržištu.
- Korisnički ili naručeni sistemi, gde korisnik definiše skup zahteva za softversko/hardverskim sistemom a ugovarač razvija i isporučuje taj sistem. Kupac i ugovarač mogu biti različita odeljenja u okviru iste organizacije ili mogu biti različite kompanije. Sistemski zahtevi opisuju usluge (servise) koje sistem kao celina treba da pruži, i kao deo procesa inženjeringa sistema se dobijaju specifični softverski zahtevi. Cela priča o softverskom inženjeringu u ovom kursu se odnosi na taj tip sistema.

Korisnički sistemi se protežu u opsegu od vrlo malih sistema koji se ugrađuju u uređaje kao što su mikrotalasna peć do gigantskih kontrolnih i komandnih sistema kao što su vojni sistemi za obaveštavanje. Ovim velikim sistemima može biti obuhvaćena mreža od hiljadu računara koji su locirani širom sveta.

KATEGORIJE SISTEMA KOJI SE RAZVIJAJU SPECIJALNO ZA KORISNIKE

(1) informacioni sistemi, (2) ugrađeni sistemi, (3) komandni i kontrolni sistemi

Postoje tri klase sistema koji se razvijaju specijalno za korisnike:

1. Informacioni sistemi: To su sistemi koji se prvenstveno odnose na obradu informacija koje se čuvaju u nekoj vrsti baza podataka. Oni se obično implementiraju korišćenjem standardnog računarskog hardvera (npr. mainframe računara, radnih stanica, PC-a) i komercijalnih operativnih sistema. Za ove sisteme, inženjering zahteva se prvenstveno odnosi na inženjering softverskih zahteva.
2. Ugrađeni sistemi: To su sistemi gde se softver koristi kao kontroler u okviru nekih većih hardverskih sistema. Oni se prostiru u opsegu od jednostavnih sistema (npr. u CD player-ima) do veoma složenih kontrolnih sistema (npr. u hemijskoj industriji). Oni se često oslanjaju na hardver i operativne sisteme specijalne namene. Inženjering zahteva za ove sisteme obuhvata inženjering zahteva kako za softver tako i za hardver.
3. Komandni i kontrolni sistemi: Onii su uglavnom kombinacija informacionih sistema i ugrađenih sistema u kojima računari specijalne namene odezbeđuju informacije koje se sakupljaju i pamte u bazama podataka a zatim koriste kao pomoć ljudima pri donošenju odluka. Ti sistemi obično obuhvataju mešavinu različitih tipova računara koji su na neki način povezani u mrežu, U okviru čitavog sistema, može postajati nekoliko ugrađenih i informacionih sistema. Najveći i najsloženiji sistemi kao što su sistemi za kontrolu vazdušnog saobraćaja, sistemi za signalizaciju na železnicama, vojni komunikacioni sistemi itd. su obično ovog tipa. Inženjeringom zahteva za ove sisteme su obuhvaćeni hardverske i softverske specifikacije i specifikacije operativnih procedura i procesa.

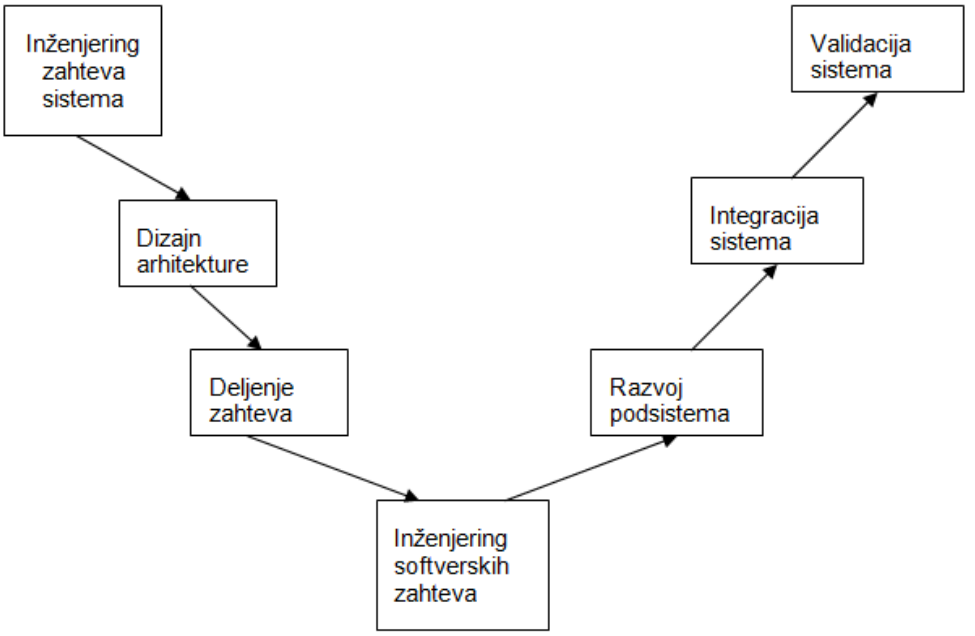
Karakteristika svih ovih sistema je da oni imaju skup izvedenih svojstava. Izvedena svojstva su svojstva sistema kao celine koja nastaju u trenutku kada se integrišu svi individualni podsistemi. Primer izvedenih svojstava za sisteme koji se baziraju na računarima su pouzdanost, mogućnost održavanja, performanse, zaštita, mogućnost korišćenja itd. Zahtevi sistema se često odnose upravo na ta izvedena svojstva a ne samo na funkcionalnost sistema.

GENERIČKI PROCES INŽENJERINGA SISTEMA

Obuhvata aktivnosti od inicijalnog opisa zahteva sistema do detaljnijeg razvoja specifičnih softverskih zahteva; uključuje i neke inicijalne aktivnosti dizajna

Mada se procesi koji se koristi za razvoj sistema razlikuju zavisno od tipa i veličine sistema koji se razvija i organizacije koje sistem razvijaju, postoje aktivnosti koje su zajedničke za sve tipove inženjeringa sistema.Slika 1. prikazuje jedan mogući generički proces inženjeringa sistema, odnosno proces posle donošenja odluke da se počne sa izgradnjom sistema. Pre toga, postoje aktivnosti koje se odnose na utvrđivanje poslovnih potreba za sistemom, ocenjivanja opravdanosti razvoja

Slika 3.1 Generički proces inženjeringa sistema



sistema, utvrđivanje budžeta za razvoj sistema. Aktivnosti prikazane na slici 1. su opisane u tabeli 1 (slika 2.).

Sa slike 1. i tabele 1. se može videti da se ukupan proces inženjeringa zahteva rasprostire na nekoliko aktivnosti, od inicijalnog opisa zahteva sistema do detaljnijeg razvoja specifičnih softverskih zahteva. On uključuje i neke inicijalne aktivnosti dizajna u kojima se definiše čitava struktura (arhitektura) sistema.

Ona se mora definisati na ovom nivou, kako bi zahtevi sistema mogli da se struktuiraju i omogući paralelni razvoj pojedinih podsistema.

Slika 3.2 tabelarni prikaz aktivnosti inženjeringa sistema

Aktivnost	Opis
Inženjering zahteva sistema	Utvrđuju se zahtevi sistema kao celine. Oni se obično iskazuju na vrlo visokom nivou i pišu u prirodnom jeziku. U njih mogu biti uključena neka detaljna ograničenja (kao što su ograničenja koja se odnose na kompatibilnost) ukoliko su ona kritična za uspeh sistema.
Dizajn arhitekture	Sistem se dekomponuje na skup nezavisnih podsistema
Deljenje zahteva	Zahtevi se dele prema utvrđenim podsistemima. Na ovom nivou se može doneti odluka da li se radi o softverskim ili hardverskim zahtevima
Inženjering softverskih zahteva	Softverski zahtevi visokog nivoa se dekomponuju na detaljnije
Razvoj podsistema	Paralelno se dizajniraju i implementiraju hardverski i softverski podsistemi
Integracija podsistema	Hardveski i softverski podsistemi se integrišu kako bi se dobio kompletan sistem
Validacija sistema	Sistem se validira u odnosu na svoje zahteve

RAZLIKA IZMEĐU SISTEMSKIH I SOFTVERSKIH ZAHTEVA

Sistemski zahtevi opisuju ponašanje sistema posmatrano spolja, sa strane korisnika; Softverski zahtevi služe za komunikaciju sa programerima

Sistemski i softverski zahtevi se često tretiraju zajedno, jer su sredstva i metode koje se koriste za njihovo izvođenje, kao i tehnike za dokumentovanje , veoma slični . (Može se primetiti daje većina alata i metoda nastala tokom razvoja softvera, i da se zatim utvrdilo da su oni pogodni i za korišćenje sistema) Međutim , trebalo bi istaći neke bitne razlike između sistemskih i softverskih zahteva.

Sistemski zahtevi opisuju ponašanje sistema posmatrano spolja , na primer, sa strane korisnik. Iako dokument i specifikacije zahteva korisnik ne može lako da pročita, zahtevi sistema služe kao sredstvo za delimičnu komunikaciju sa tehnički obrazovanijim korisnicima , ili nalazenje organizacije , analitičara i dizajnera koji se bave razvojem elemenata ili komponenata sistema.

Zahtevi za elemente ispod nivoa sistema (softverski zahtevi) , bez obzira da li su to elementi koji se odnose na hardver ili softver, ili i na hardver i softver, su obično su od minimalnog interesa za korisnike. Softverski zahtevi služe za komunikaciju sa programerima , koji treba da znaju šta se očekuje od elemenata za koje su odgovorni, a kojima su takođe potrebne informacije o elementima koji predstavljaju njihov interfejs.

Ova razlika je važna iz nekoliko razloga . Prvo , kao i svako sredstvo za komunikaciju, dokument zahteva bi trebalo da bude napisan imajući na

umu kome je namenjen. Stepen do kojeg tehnički neobrazovani korisnici mogu da čitaju i razumeju dokument zahteva je faktor koji određuje kako ga treba napisati. Drugo , i možda najviše važno je da se moraju uzeti u obzir veštine i iskustvo ljudi koji specificiraju zahteva. Vrlo često, sistem inženjeri sa ograničenim znanjem softvera su odgovorni za softver-intenzivne sisteme.Oni ne samo da pišu zahteve sistema na nivou koji zahteva znanje o tome koje funkcije i performanse softver-intenzivan sistem treba da zadovolji; oni takođe moraju funkcije i zahteve dodeliti odgovarajućem hardveru i softveru. Od onih koji razvijaju softver se traži da utvrde dizajne koji može da ispuni zahteve što može biti veoma nerealno ili neizvodljivo. Softver developeri oklevaju da se uključe u inženjering sistema; jedan od rezultata je da razvoj softverskih elemenata često počinje sa lošim zahtevima.

Poglavlje 4

Dokument zahteva

ČEMU SLUŽI DOKUMENT ZAHTEVA?

To je formalni dokument u kojem se definišu sistemski i softverski zahtevi i koji se koristi u komunikaciju između korisnika, inženjera koji rade na razvoju sistema i softvera i menadžera.

Sistemski i softverski zahtevi se obično dokumentuju u formalnom dokumentu koji se koristi u komunikaciju između korisnika, inženjera koji rade na razvoju sistema i softvera i menadžera procesa inženjeringa sistema. Kao što je rečeno, ne postoji standardno ime za ovaj dokument. U različitim organizacijama, ovaj dokument ima različito ime kao što je dokument zahteva, funkcionalna specifikacija, specifikacija sistemskih zahteva itd.

Dokument zahteva opisuje sledeće:

- Servise i funkcije koje sistem treba da obezbedi.
- Ograničenja pod kojima sistem treba da radi.
- Sva svojstva sistema npr. ograničenja vezana za izvedena svojstva sistema.
- Definicije drugih sistema sa kojima sistem mora da se integriše.
- Informacije o aplikativnom domenu sistema tj. kako izvršavati pojedine tipove izračunavanja.
- Ograničenja koja se odnose na proces razvoja sistema.

Za sisteme koji su prvenstveno softverski, dokument zahteva može da uključi i opis hadvera na kojem će se sistem izvršavati.

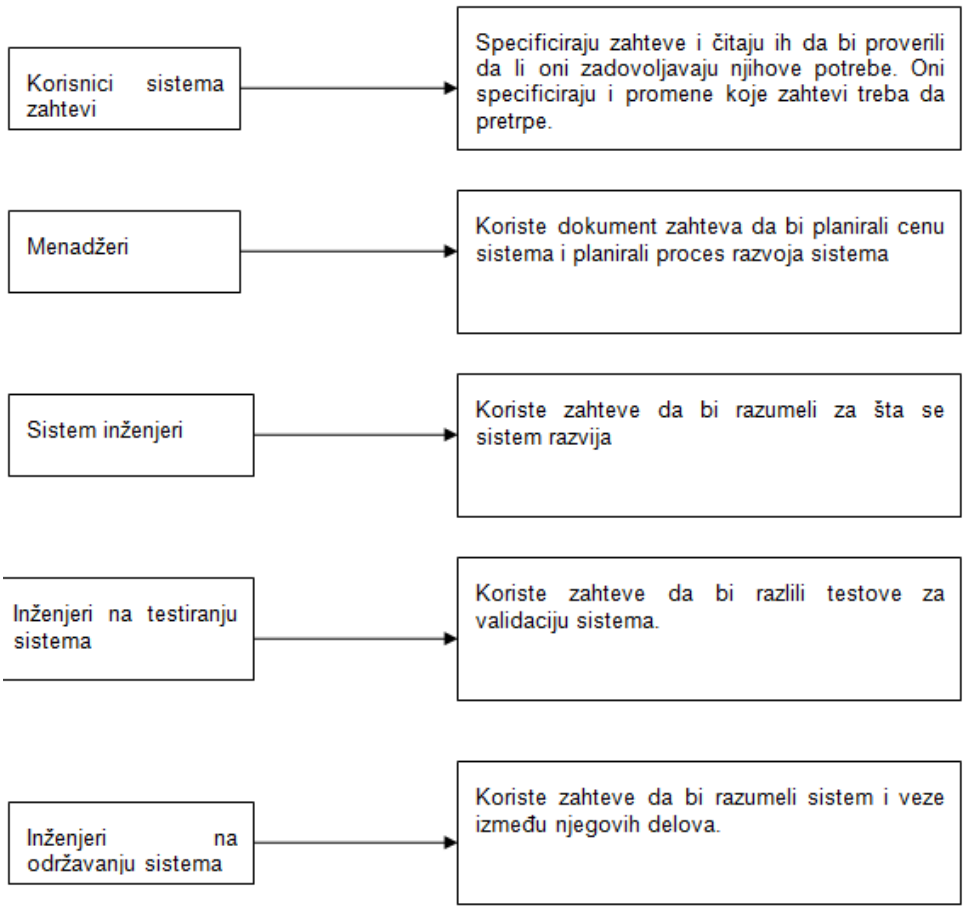
ŠTA SADRŽI DOKUMENT ZAHTEVA?

Mora da obavezno sadrži uvod u kojem se daje pregled celog sistema, poslovne potrebe koje sistemom treba podržati, rečnik koji definiše tehničke termine koji se koriste u dokumentu itd

Dokument zahteva mora da obavezno sadrži uvod u kojem se daje pregled celog sistema, poslovne potrebe koje sistemom treba podržati, rečnik koji definiše tehničke termine koji se koriste u dokumentu itd. Rečnik je naročito važan obzirom da dokument zahteva čitaju različiti stakeholderi sistema sa različitim obrazovanjem i koriste ga na različit način. (slika 1.) Osnovna namena rečnika je da obezbedi zajedničko razumevanje termina koji se koristeu dokumentu zahteva.

Postoji mnogo različitih načina za struktuiranje dokumenta zahteva, što zavisi od tipa sistema koji se razvija, nivoa detalja koji je uključen u zahteve, prakse organizacije, budžeta i vremena koji je definisan za proces inženjeringa zahteva. Da bi se obezbedilo da se u njega uključe sve važne informacije, organizacije moraju da definišu voj sopstveni standard za dokument zahtevakojim će biti definisane sekcije koje moraju biti uključene.

Slika 4.1.1 Različiti stakeholderi sistema koriste dokument zahteva na različit način



DOKUMENT ZAHTEVA PO STANDARDU IEEE/ ANSI 830-1993 (IEEE, 1993)

Sadrži uvod, generalni opis, specifične zahteve, priloge i indeks

Neke velike organizacije kao što su US Department of Defence i IEEE su definisale standarde za dokument zahteva.

Verovatno najčešće korišćen je standard IEEE/ ANSI 830-1993 (IEEE, 1993) kojim se sugerira sledeća struktura dokumenta zahteva:

Uvod

- Svrha dokumenta zahteva
- Područje rada
- Definicije, skraćenice
- Reference
- Pregled ostatka dokumenta

Generalni opis

- Perspektive proizvoda
- Funkcije proizvoda
- Karakteristike korisnika
- Generalna ograničenja
- Pretpostavke i zavisnosti

Specifični zahtevi

pokrivaju funkcionalne i nefunkcionalne zahteve kao i zahteve koji se odnose na interfejs. Treba da dokumentuju eksterne interfejse, funkcionalnost, zahteve za performansama, logičke zahteve baze podataka, ograničenja dizajna, atribute sistema i karakteristike kvaliteta.

Prilozi

Indeks

KOJI STANDARD ZA DOKUMENT ZAHTEVA TREBA USVOJITI?

IEEE standardi nisu idealni; kako on mora da podrži razlike koje postoje među sistemima, treba da sadrži

uvodnu stranicu na kojoj su objašnjene dozvoljene varijacije definisanog standarda.

Mada IEEE standardi nisu idealni, oni sadrže mnogo dobrih preporuka kako napisati zahteve i kako sprečiti probleme. U tom smislu mogu predstavljati dobru početnu tačku u definisanju strukture dokumenta zahteva. Prva dva dela u standardu su uvodna poglavlja kojim se definiše pozadina sistema i on uopšteno opisuje. Treća sekcija je glavna sekcija dokumenta koja sadrži detaljne specifikacije zahteva. Standard prepoznaje da specifikacije u velikoj meri variraju zavisno od tipa sistema koji se specificira i sugerira alternativne načine za organizovanje specifičnih zahteva.

Standard za dokument zahteva mora da podrži razlike koje postoje među sistemima. On mora da omogući izostavljanje nekih delova dokumenta i dodavanje novih sekcija ako je to neophodno. Dokument standarda treba da zbog toga sadrži i jednu uvodnu stranicu na kojoj su objašnjene dozvoljene varijacije definisanog standarda.

Da bi se prilagodio različitim tipovima dokumenata, standard za dokument zahteva mora da ima listu stalnih delova i nestalnih delova (koji variraju). Stalni delovi su ona poglavlja (kao što su uvod i rečnik) koji se moraju pojaviti u svim dokumentima zahteva. Nestalni delovi su ona poglavlja koja mogu ali ne moraju biti uključena i čiji sadržaj može varirati zavisno od specificiranog sistema.

PRIMER MOGUĆE ORGANIZACIJE DOKUMENTA ZAHTEVA KOJI JE BAZIRAN NA IEEE STANDARDU

Sistem se bavi računarski upravljanom proizvodnjom naučnih instrumenata.

Ovde je dat primer moguće organizacije dokument zahteva koji je baziran na IEEE standardu. Napomenimo da treći deo IEEE standarda (specifični zahtevi) treba predstaviti kroz nekoliko poglavlja i da takođe opciono treba uključiti priloge sa detaljnim informacijama. Pretpostavka je da se organizacija bavi računarski upravljanom proizvodnjom naučnih instrumenata.

Organizacija XYZ. Standardna struktura dokumenta zahteva

Uvod

U njemu treba definisati očekivane korisnike (čitaoce) dokumenta, opisati verzije dokumenta uključujući i obrazloženje za kreiranje nove verzije i dati kratak opis izmena koje su napravljene u svakoj verziji.

Predstavljanje projekta

Ovde treba definisati proizvod u koji se softver ugrađuje, očekivanja vezana za korišćenje softvera, prezentirati i dati pregled funkcionalnosti softvera.

Rečnik

On treba da definiše sve tehničke termine i skraćenice koje se koriste u dokumentu.

Opšti zahtevi korisnika

Ovde treba definisati sistemske zahteve iz perspektive korisnika sistema. Oni treba da budu predstavljeni korišćenjem prirodnog jezika i dijagrama.

Arhitektura sistema

U ovom poglavlju treba predstaviti očekivanu arhitekturu sistema na visokom nivou koja pokazuje distribuciju funkcija na module sistema. Komponente arhitekture koje se koriste u okviru modula treba posebno istaći.

Specifikacija hardvera

Ovo poglavlje je opciono i ono treba da
specificira hardver koji treba da upravlja projektovani softver. Može se izostaviti ukoliko se koristi platforma standardnih instrumenata.

Detaljne softverske specifikacije

Ovde se daje detaljan opis očekivane funkcionalnosti softvera sistema. Kada je neophodno, mogu se uključiti specifični algoritmi kojima se specificiraju potrebna sračunavanja. Ukoliko se za razvoj koristi pristup prototipa i platforma standardnih instrumenata, ovo poglavlje može biti izostavljeno.

Zahtevi koji se odnose na pouzdanost i performanse

U ovom poglavlju treba opisati zahteve koji se odnose na pouzdanost i performanse koje se očekuju od sistema. Oni treba da se odnose na opis korisničkih zahteva iz četvrtog poglavlja.

Prilozi

Index

4.1 PISANJE ZAHTEVA

PROBLEMI PRI PISANJU ZAHTEVA KORIŠĆENJEM PRIRODNOG JEZIKA

Zahtevi su pisani korišćenjem složenih kondicionalnih rečenica, terminologija se koristi na nekonzistentan način, u dokument zahteva se ne stavljaju sve značajne informacije

U mnogim organizacijama, zahtevi sistema se pišu kao parafgrafi korišćenjem prirodnog jezika (engleskog, francuskog, japanskog, srpskog itd.) kojima mogu biti dodati dijagrami i jednačine. Prirodni jezik je jedina notacija koju generalno razumeju svi potencijalni čitaoci sistemskih zahteva. Međutim, najčešći problemi koji se javljaju pri pisanju zahteva korišćenjem prirodnog jezika su:

- Zahtevi su pisani korišćenjem složenih kondicionalnih rečenica (ako A onda ako B onda ako C.....) što može delovati zbunjujuće.
- Terminologija se koristi na neuredan i nekonzistentan način
- Pisac zahteva pretpostavlja da onaj ko čita dobro poznaje domen sistema i u dokument zahteva ne stavlja sve značajne informacije

Ovi problemi otežavaju proveru zahteva na greške i nedostatke. Različita interpretacija zahteva može dovesti do neslaganja između korisnika i inženjera sistema sa kojima oni sklapaju ugovor.

Da bi se ove poteškoće uklonile, neke organizacije su razvile specijalnu notaciju za pisanje zahteva pa su se zahtevi pisani na prirodnom jeziku mogli zameniti modelom sistema koji može biti grafički ili matematički model. Ali notacije specijalnih namena nikada nisu postale široko prihvaćene a modeli sistema su često na suviše niskom nivou za ostvarivanje komunikacije među ljudima koji nemaju potrebu da zahteve detaljno analiziraju. Zbog toga postoji potreba za dobro napisanim zahtevima korišćenjem prirodnog jezika.

ŠTA TREBA IMATI NA UMU PRI PISANJU ZAHTEVA?

Tri osnovne stvari

Bez obzira na nivo detalja u opisu zahteva, postoje tri osnovne stvari koje treba imati na umu pri pisanju zahteva:

1. Zahtevi se mnogo češće čitaju nego što se pišu. Uložiti napor u pisanje zahteva koji se lako čitaju i razumeju je skoro uvek isplativo.

2. Čitaoci zahteva dolaze iz različitih okruženja. Ako ste vi onaj koji piše zahteve, ne treba da pretpostavite da čitalac ima ista predznanja kao i vi.
3. Jasno i koncizno pisanje nije lako. Ako nemate dovoljno vremena da napišete zahteve koji su potpuno određeni, pregledani i potvrđeni neminovno ćete dobiti loše napisane specifikacije.

Različite organizacije pišu zahteve na različitom nivou apstrakcije od umereno nejasnih specifikacija proizvoda do detaljnih i preciznih opisa svih aspekata sistema. Vi sami morate odlučiti o nivou detaljnosti koji vam je potreban. To zavisi od tipa zahteva (zahtevi za stakeholdere, sistem ili procese) očekivanja korisnika, vaših organizacionih procedura i eksternih standarda ili regulativa kojih se treba pridržavati.

Poglavlje 5

Vežba: Analiza problema

KAKO DEFINISATI ANALIZU PROBLEMA?

Kao proces razumevanja problema iz realnog sveta, potrebe korisnika i predlaganje rešenja koja može da zadovolje te potrebe

Ova vežba se fokusira na načine na koji razvojni tim može da razume realne potrebe korisnika i zainteresovanih strana (stejkholdera) novog sistema ili aplikacije. Da bismo bili sigurni da razumemo šta je problem koji treba rešiti sistemom, koristićemo *tehnike analize problema*.

Ali isto tako treba shvatiti da nije svaka aplikacija razvijena da reši problem; neke su napravljene da bi se iskoristile mogućnosti koje pruža tržište, čak i kada ne postoje jasni problemi.Na primer ,softverske aplikacije , kao šo su SimCiti i Mist , su pokazale svoju vrednost onima koji vole kompjuterske igrice i uživaju u modeliranju i simulaciji ili igranju igara na svojim računarima. Dakle, teško je reći kakav problem SimCiti ili Mist rešavaju - možda problem " nemanja dovoljno zabavnih stvari za rad sa računarom " ili problem " previše slobodnog vremena za nečije ruke "- jasno je da proizvodi imaju realnu vrednost za veliki broj korisnika .

U izvesnom smislu, problemi i mogućnosti su samo različite strane istog novčića; tvoj problem je moja prilika. Ali pošto se većina sistema bavi nekim identifikovanim problemima, možemo pojednostaviti diskusiju i izbeći probleme fokusiranjem na samo jednu stranu medalje. Na kraju krajeva, mi volimo da mislimo o sebi kao da rešavamo problem. Analizu problema ćemo definisati kao proces razumevanja problema iz realnog sveta, potreba korisnika i predlaganje rešenja koja može da zadovolje te potrebe. Na taj način,domeni problema se moraju analizirati i razumeti i istražiti niz rešenja problema. Obično, su moguća različiti rešenja, a naš posao je da se pronađe rešenje koje je optimalno za problem koji se rešava.

ŠTA JE PROBLEM?

Može se definisati kao razlika između percepcije i realnosti (želja i stanovišta); Problem se rešava u pet koraka;

Da bi bili u mogućnosti da uradimo analizu problema, bilo bi korisno definisati šta je problem. Prema Gause i Veinberg (1989), problem se može definisati kao razlika između stanovišta kako stvari treba izvesti i načina na koji su one izvedene.

Prema definiciji, ako korisnik vidi nešto kao problem, onda je to pravi problem, i njega treba rešavati. Ipak, na osnovu ove definicije, naš kolega Elemer Magaziner napominje da postoji više načina za bavljenje problemima. Na primer, jedan od najefikasnijeg pristup je promena želje ili percepcije korisnika. Praktično iskustvo pokazuje mnoge primere gde je promena percepcija u sagledavanju razlika dovelo do rešenja najvišeg kvaliteta, koja su najbrža, i najjeftinija. Dužnost onih koji rešavaju problem je da se istraže sva alternativna rešenja pre nego što se pređe na novo rešenje sistema.

Međutim , ove alternativne aktivnosti ne mogu dovoljno smanjiti jaz između percepcije i želje, pa tako ostaje najveći i najskuplji izazov: da se smanji udaljenost između percepcije i realnosti . Ovo moramo postići definisanjem i primenom novih sistema koji sužavaju razliku između želja i stanovišta .

Kao i u svakoj vežbi rešavanja kompleksnog problema,na početku moramo imati cilj. Cilj analize problema je da se, pre nego što počne razvoj , stekne bolji uvid u problem koji treba da bude rešen .Specifični koraci koji se moraju preduzeti u cilju postizanja cilja su:

1. Postizanje dogovora o definiciji problema .
2. Razumevanje potreba korisnika
3. Utvrđivanje aktera i korisnika .
4. Definisanje granica sistemskog rešenja .
5. Utvrđivanje ograničenja kojima je izloženo rešenje.

KORAK 1 I 2: DEFINISANJE PROBLEMA I RAZUMEVANJE POTREBA KORISNIKA

Problem treba opisati u standardnom formatu

Problem treba opisati u standardnom formatu . Popunjavanje same tabele kojom se opisuje problem za aplikaciju je jednostavno, mada se treba pridržavati određenih tehnika kako bi se osiguralo da svi učesnici na projektu rade ka istom cilju .

Potrebno vreme da se dobije ugovor o problemu koji se rešava može izgledati kao mali i beznačajan korak, a u većini slučajeva, to je to. Međutim, ponekad, nije. Na primer , jedan od naših klijenata, proizvođač opreme je bio angažovan na nadogradnji IS / IT sistema , koji je trebalo da obezbedi fakturisanje i finansijsko izveštavanje između kompanije i njenih dilera. Cilj ovog projekta je bio da se "poboljša komunikacija sa dilerima ". Tim je krenuo u razvoj novog sistem koji predviđa bolje finansijsko izveštavanje , poboljšanje formata faktura , onlajn naručivanje delova i elektronsku poštu. I oh, uzgred , tim se nada da će imati mogućnost elektronskog prenosa sredstava između kompanije i distributera.

Tokom definisanja problema, menadžment preduzeća ima priliku da da svoj doprinos . Vizija rukovodstva je bila bio bitno drugačija : Primarni cilj novog sistema je bio da se obezbedi elektronski transfer sredstava koji će poboljšati protok gotovine kompanije. Nakon žestoke rasprave , postalo je jasno da je problem prvog reda kojim će se baviti novi sistem je elektronski transfer sredstava ; e-mail i drugi način komunikacije sa delireima su osobine za koje se smatra da je samo " lepo imati . "

KORAK 3: IDENTIFIKOVANJE KORISNIKA I STEJKHOLDERA

Stejkholder je bilo ko ko je materijalno pogođen implementacijom novog sistema; neki stejkholderi su direktni a neki indirektni korisnici sistema

Efikasnim rešavanjem bilo kog kompleksnog problema je obično obuhvaćeno zadovoljavanje potreba različitih grupa stejkholdera. Stejkholderi obično imaju različite poglede na problem i različite potrebe koje moraju biti rešene. Stejkholder se može definisati kao bilo ko ko bi mogao biti materijalno pogođen implementacijom novog sistema ili aplikacije .

Mnogi stejkholderi su i korisnici sistema i njihove potrebe se mogu lako fokusirati, jer su oni direktno obuhvaćeni definicijom i korišćenjem sistema . Međutim, neki stejkholderi su samo indirektni korisnici sistema ili na njih utiču samo poslovni rezultati koje sistem generiše . Ovi stejkholderi se mogu naći bilo gde u biznisu ili u " okruženju ". U drugim slučajevima , stejkholderi su još dalje uklonjene iz okruženja aplikacije

. Na primer, oni uključuju ljude i organizacije koje se bave razvojem sistema, kupčeve kupce, spoljne agencije koje su u interakciji sa sistemom ili razvojnim procesom.

NAČIN ZA PRONALAŽENJE STEJKHOLDERA SISTEMA

Pronalaženje stejkholdera sistema i njihovih posebnih potreba je važan faktor u razvoju efikasnog rešenja

Svaka od ovih klasa stejkholdera može uticati na zahteve sistema ili će na neki način biti uključena u ishode sistema. Pronalaženje stejkholdera sistema i njihovih posebnih potreba je važan faktor u razvoju efikasnog rešenja . U zavisnosti od poznavanja domena problema razvojnog tima, identifikovanje stejkholdera može biti trivijalan ili netrivijalan korak u analizi problema . Često, to podrazumeva razgovor sa donosiocima odluka , potencijalnim korisnicima, i drugim zainteresovanim stranama .Pitanja koja mogu biti od pomoći u tom procesu su:

1. Ko su korisnici sistema ?
2. Ko je kupac sistema ?
3. Koga pogođaju izlazi koji sistem proizvodi ?Ko procenjuje sistem kada se on isporuči i razmesti ?
4. Da li postoje neki drugi interni ili eksterni korisnici sistema čije se potrebe moraju se rešavati ?
5. Ko će održati novi sistem ?
6. Da li postoji neko drugi ?

U našem primeru sistema za evidentiranje narudžbenica, osnovni i najočigledniji korisnici su službenici za unos narudžbenica. Ovi korisnici su očigledno stejkholderi i njihova produktivnost , udobnost, performanse i zadovoljstvo poslom utiču na sistem. Koje druge stekholdere možemo identifikovati ?

Kada se identifikuju korisnici i stejkholderi sistema, možemo da obratimo pažnju na definisanje sistema koji može da reši naš problem . Na taj način , ulazimo u stanje tranzicije u kome moramo imati na umu dve stvari: razumevanje problema i razmatranja potencijalnog rešenja .

KORAK 4 : DEFINISANJE GRANICA SISTEMA

Granice sistema definiše ivicu između rešenja i stvarnog sveta koji okružuje naše rešenja

Sledeći važan korak je utvrđivanje granica sistemskog rešenja.

Granice sistema definiše ivicu između rešenja i stvarnog sveta koji okružuje naše rešenja. Drugim rečima, granica sistem opisuje omotač u kojoj je sadržano rešenje sistema . Informacije , u obliku ulaza i izlaza , idu napred i nazad od sistema do korisnika koji se nalaze izvan njega. Sva interakcija sa sistemom odvija preko interfejsa između sistema i spoljašnjeg sveta .

KORAK 5: UTVRDJIVANJE OGRANIČENJA KOJIMA SISTEM MOŽE BITI IZLOŽEN

Ograničenje se može definisati kao zabrana izvesnog stepena slobode prilikom isporuke sistema; Izvori ograničenja mogu biti raznovrsni;

Pre bilo kakvih početnih ulaganja u sistem za evidentiranje narudžbenica se moramo zaustaviti i uzeti u obzir ograničenja koja će rešenje biti izloženo. Ograničenje se može definisati kao zabrana izvesnog stepena slobode prilikom isporuke sistema. Svako ograničenje ima potencijal da ozbiljno ograniči našu mogućnost da isporučmo rešenje kao smo zamislili. Dakle, svako ograničenje se mora pažljivo razmotriti kao deo procesa planiranja, a mnoga mogu prouzrokovati potrebu da se preispita tehnološki pristup koji je na početku predviđen.

Izvori ograničenja mogu biti raznovrsni: raspored, povratak investicija, budžet za rad i opremu, zaštita životne sredine, operativni sistemi, baze podataka, host i sistema klijenata, tehnička pitanja, politička pitanja unutar organizacije, kupovina softvera, politika i procedure kompanije, izbor alata i jezika, osoblje ili drugi izvori.

TEHNIKE ZA ANALIZU PROBLEMA

Poslovno modeliranje, specifične tehnike za analizu problema itd.

Treba takođe napomenuti da se prilikom analize problema mogu koristiti različite tehnike: modeliranje poslovanja, specifične tehnike za analizu problema koje se mogu uspešno primeniti u složenim informacionim sistemima koji podržavaju ključnu poslovnu infrastrukturu. Poslovno modeliranje se može koristiti i za razumevanje načina na koji se poslovanje odvija i identifikovanje mesta na kojima je najproduktivnije smestiti neku aplikaciju. Poslovni slučajevi korišćenja se takođe mogu koristiti za definisanje zahteva za samu aplikaciju

Za softverske aplikacije u ugrađenim sistemima, inženjerstvo softvera se koristi tehnika za analizu problema koja pomaže u dekompoziciji složenih sistema na podsisteme. Taj proces pomaže da se razume gde treba softver smestiti i šta je njegova svrha.

5.1 ANALIZA PROBLEMA ZA LUMENATIONS LCD.

LUMENATIONS, LTD.

Opis sistema

Lumenations , Ltd , je svetski snabdevač sistema za reklamno osvetljenje koje se koristi u profesionalnim pozorištima i amaterskoj produkciji više od 40 godina . 1999. godine, njegov godišnji prihod je dostigao vrhunac od oko 120 miliona dolara a prodaja se ujednačila. Lumenations je javno preduzeće i izostanak rasta prodaje- još gore , nedostatak bilo kakvih razumnih izglede za poboljšanje rasta prodaje - uzima svoj danak kompaniji i njenim akcionarima. Poslednji godišnji sastaanka je bio prilično neprijatan, jer je bilo malo toga novog u izveštajima o izgledima da kompanija postigne nekakav za rast .

Industrija pozorišne opreme u celini je u stagnaciji sa malo novog razvoja. Kako su Lumenations-ove zalihe u magacinima a njegova kapitalizacija bi bila vrlo skromna, akvizicija nije opcija za kompaniju .

Ono što je potrebno jesu nova tržišta, ne previse udaljena od onoga što kompanija najbolje radi, ili u kojima postoji značajna mogućnost za rast prihoda i profita . Nakon temeljnog projekta istraživanja tržišta i mnogo

potrošenih dolara za marketing konsultante, kompanija je odlučila da uđe na novo tržište: *automatizacija osvetljenje za stambene sisteme.* Ovo tržište očigledno raste 25% -35% godišnje. Tržište je nezrelo jer niko od utvrđenih igrača nema dominantne tržišne pozicije. Lumenations-ov jak, širom sveta rasprostanjen distributivni kanal predstavlja preimućstvo na tržištu , a distributeri su gladni za novim proizvodima . Izgleda kao odlična prilika.

ANALIZA PROBLEMA ZA LUMENATIONS LCD.

Identifikovano tri problema: sa aspekta fireme, vlasnika kuća i distributera

Tim je definisao tri problema, od kojih je jedan bio očigledan iz perspektive same kompanije (tabela 1).

Slika 5.1.1 Problemi sa aspekta kompanije

Za Lumenations kompaniju	
Problemi	usporavanje napretka u ključnom poslu kompanije
Utiče na	kompaniju, zaposlene i aktere
Rezultat koji	je neprihvatljiv za biznis performanske kompanije uz nedostatak prilika za rast prihoda i profitabilnosti
Prednosti	novi proizvodi su potencijalno nova tržišta za kompanijske proizvode i servise
	• Oporavak kompanije i zaposlenih
	• Povećanje lojalnosti i zadržavanje distributera u kompaniji
	• Veći rast prihoda i profitabilnosti
	• Rast cena akcija kompanije

Tim je takođe odlučio da vidi da li problem može biti rešen postojećim rešenjima koja se nude na tržištu iz perspektive budućih kupaca (krajnjih korisnika-vlasnika kuća tabela 2.) i potencijalnih distributera (kupaca, tabela 3):

Za vlasnika stambenog prostora	
Problemi	nedostatak izbora proizvoda, ograničene funkcionalnosti i visoki troškovi sadašnjeg sistema za automatsko osvetljenje prostora
Utiče na	vlasnike kuća i stambenih sistema
Rezultat koji	daje neprihvatljive performance kupljenog sistema gde se često dešava da je potrebno samostalno odlučiti a ne koristiti automatizovan sistem
Prednosti	pravilan sistem za automatsko regulisanje svetla podrazumeva
	• Veće zadovoljstvo kod korisnika sistema, ponos uzrokovan posedovanjem ovakvog sistema
	• Povećanje fleksibilnosti i korišćenja stambenog prostora

Tabela 2. Problemi sa aspekta vlasnika kuća

Slika 5.1.2 Problem sa aspekta distributera

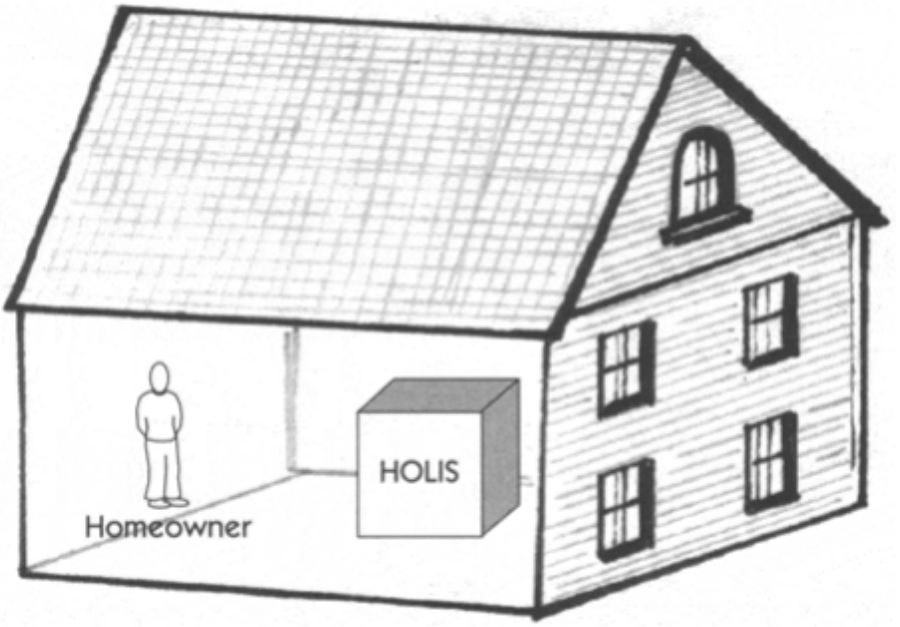
Za distributere	
Problemi	nedostatak izbora proizvoda, ograničena funkcionalnost i veliki troškovi sistema za kontrolu osvetljenja u stambenim prostorima
Utiče na	distributere i građevince stambenih sistema
Rezultat koji	predstavlja nekoliko prilika za markete u kreiranju konkurentog proizvoda bez prilika za kompleksnije proizvode
Prednosti	pravi sistem automatskog sistema može uključivati:
	• Diferencijaciju proizvoda
	• Povećanje prihoda i veću profitabilnost
	• Povećanje tržišnog udela

HOLIS: SISTEM, AKTERI, STEJKHOLDERI

Definisanje konteksta i aktera HOLIS sistema

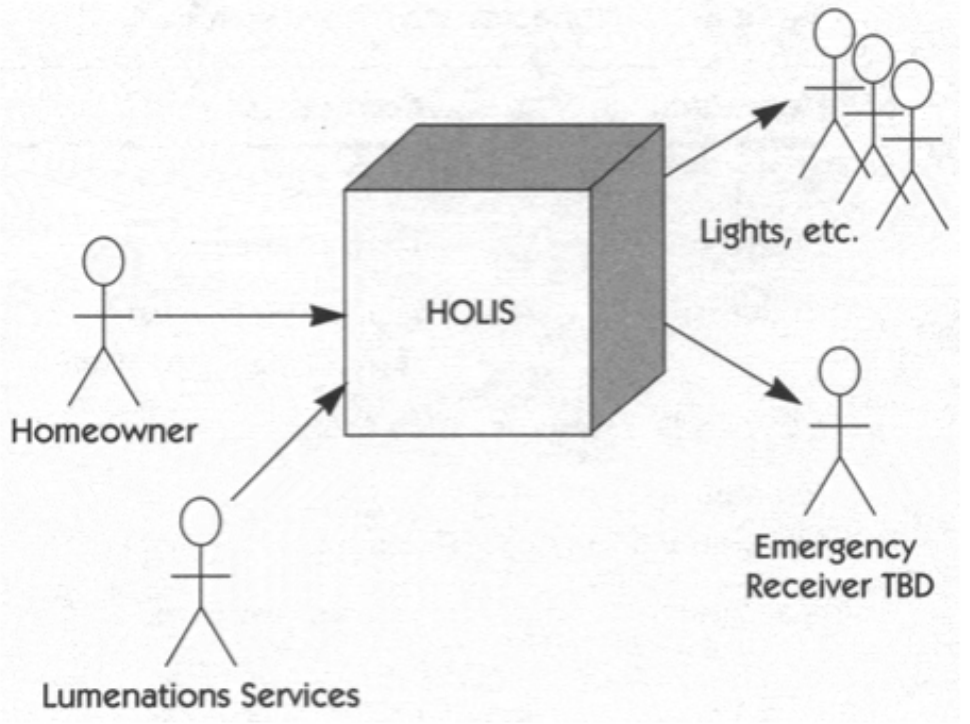
Iz perspektive sistema, prva impresija je da je HOLIS sistem namenjen vlasnicima kuća. Slika 4. predstavlja jednostavni dijagram sistema koji pokazuje HOLIS u kontekstu vlasnika kuće.

Slika 5.1.3 HOLIS u svom okruženju



Da bi unapredili znanje o kontekstu HOLIS sistema, identifikovani su akteri koji su u interakciji sa HOLIS. Slika 5. prikazuje četiri aktera.

Slika 5.1.4 Akteri sistema HOLIS



1. Vlasnik kuće koji koristi HOLIS da bi kontrolisao osvetljenje
2. Različita osvetljenja koje HOLIS kontroliše
3. Lumenations Services, proizvođač koji ima mogućnost da udaljeno pozove HOLIS I izvrši udaljeno programiranje.
4. Primalac signala za uzbunu (eng. Emergency Receiver), nedefinisani akter koji će primate poruke za uzbunu.

BLOK DIJAGRAM SISTEMA SA IDENTIFIKOVANIM KORISNICIMA

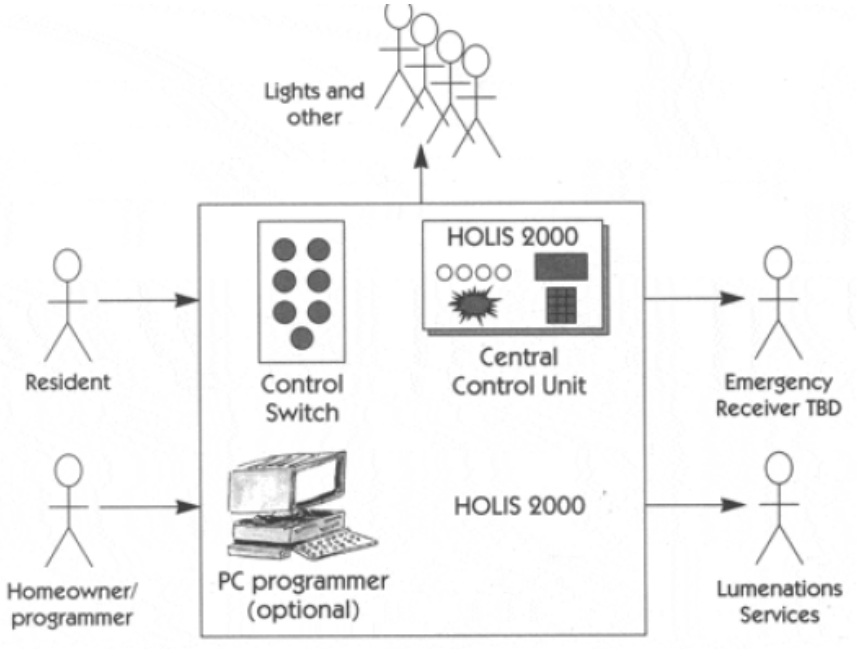
Identifikuje podsisteme i aktere ovog sistema

HOLIS se satoji od tri podsistema:

- **Control Switch,** uređaj za daljinsku kontrolu isključivanje koji može da se programira
- **Central Control Unit,** centralni računarski kontrolni sistem
- **PC Programmer**, opcioni PC system koji zahtevaju pojedini vlasnici kuća

Blok diagram je prikazan na slici 4.

Slika 5.1.5 Podsistemi i akteri sistema



Napomenimo da je sada identifikovano pet korisnika-ponovo vlasnik kuće ali koji ovoga puta koristi PC za programiranje HOLIS-a umesto da pali I gasi osvetljenje. Ovaj akter vlasnik/programmer ima različite potrebe u pogledu sistema I zbog toga je prikazan kao poseban actor sistema.

Slika 5.1.6 Akteri sistema

Naziv proizvoda	Komentar
Svetla i drugo	Izlazni uređaji, svetla i prekidači, ostalo
Vlasnik kuće/programmer	Programi za vlasnike direktno sa CC kroz programerski računar
Primalac hitnih poruka	Nepoznato, pod istragom
Stanovnik	Vlasnik koristi kontrolni prekidač za promenu osvetljenja
Lumenations servisi	Lumenations zaposleni podržavaju daljinsko programiranje i aktivnosti održavanja

BLOK DIJAGRAMI PODSISTEMA

Blok dijagrami za podsisteme: Control Switch blok, Central Control Unit i Homeowner PC

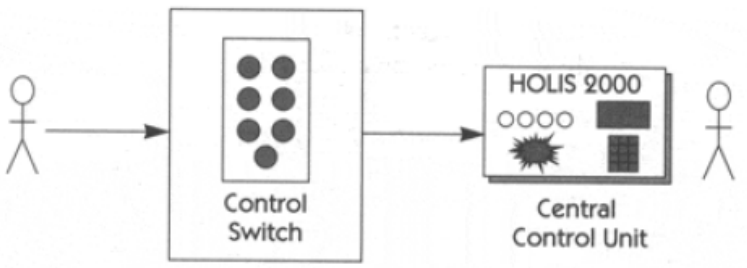
Slike 6, 7 i 8 prikazuju blok dijagrame.

Na slici 6. na kojoj se sistem gleda iz perspektive Control Switch, se nalazi još jedan akter: podsistem Central Control Unit (CCU). Drugim rečima, CCU je akter- Control Switch-a, pa ćemo kasnije morati da identifikujemo sve vrste zahteva i use case-ove koje CCU zahteva od Control Switch-a.

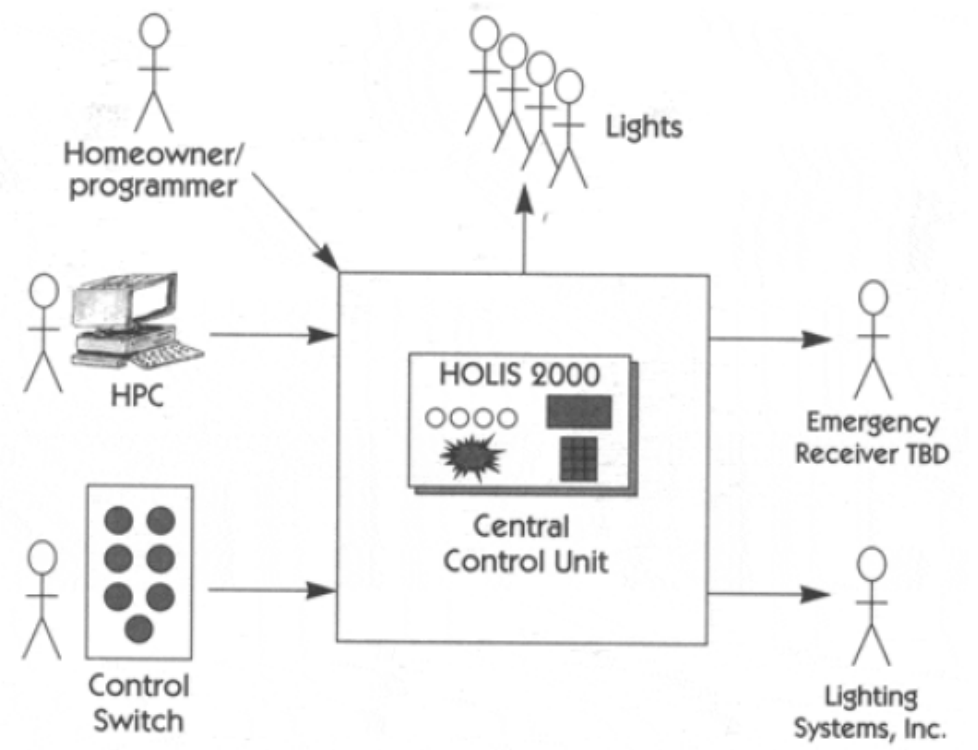
Na slici 8. prikazana je perspektiva sistema s tačke gledišta PC-a vlasnika kuće i tu se ne vidi ništa novo bar što se tiče aktera i podsistema u odnosu na ono što je prethodno identifikovano.

Slika 7. međutim predstavlja nešto bogatiji pogledIl vidimo da CCU ima više aktera od ostalih. To se čini logičnim, obzirom da se CCU može smatrati mozgom HOLIS-a, tako da njega treba da opslužuje više korisnika.

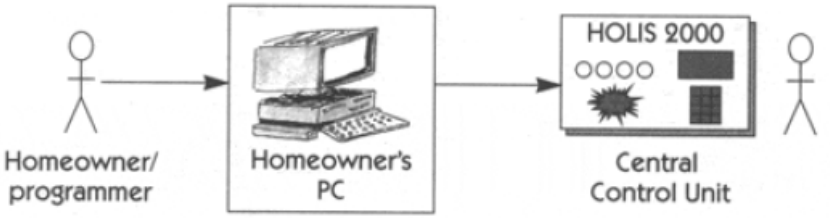
Slika 5.1.7 Blok dijagram podsistema Control Switch sa akterima, 6-8



Slika 5.1.8 Blok dijagrame podsistema Central Control Unit sa akterima, 6-10



Slika 5.1.9 Blok dijagrame podsistema PC Programmer sa akterima, 6-9



STEJKHOLDERI

HOLIS ima izvestan broj stejkoldera koji ne koriste direktno sistem

HOLIS ima izvestan broj stejkoldera koji ne koriste direktno sistem, kako internih tako i eksternih. Lista stejkholdera sa komentarima je data u tabeli 9.

Slika 5.1.10 Lista stejkholdera sistema

Naziv proizvoda	Komentari
Eksterni	
Distributeri	Direktni kupci kompanije Lumenations
Građevinci	Klijenti kompanije Lumenations, generalni izvođač radova, odgovoran vlasniku za kranji rezultat
Interni	
Razvojni tim	Lumenations tim
Marketing/ menadžment proizvodnje	Prezentovan od strane Keti, menadžera proizvoda
Generalni menadžment kompanije	Finansiranje i odgovornost za krajnji ishod

OGRANIČENJA KOJIMA JE REŠENJE IZLOŽENO

Opis ograničenja

U period od 45 dana od početka razvoja proizvoda, HILIS razvojni tim i menadžment Lumenations-a su identifikolvali, prodiskutovali i složili se oko sledećih ograničenja prikazanih u tabeli 10.

Slika 5.1.11 Tabela sa ograničenjima sistema

ID	Opis	Obrazloženje
1	Verzija 1.0 proizvoda treba da bude puštena u rad 5og Januara, 2000. Godine.	Jedino proizvod lansira priliku ove godine
2	Tim treba da prihvati UML modelovanje, OO metodologije i "Unified Software Development Process".	Mi verujemo da će ove tehnologije omogućiti poboljšanu produktivnost i robusnost sistema
3	Softver za Centralnu Kontrolnu Jedinicu programer treba da napiše u asemblerskom C++ jeziku koji će se koristiti i za prekidač (Control Switch).	Za konzistentnost i sposobnost snadbevanja; takođe, ima potrebno iskustvo u radu sa ovim programskim jezicima
4	Prototip sistema mora biti prikazan na decembarskom sajmu kućne automatizovane opreme	Uzimanje narudžbina od distributera za Q1 FY2000
5	Podsistem mikroprocesora za centralnu kontrolnu jedinicu treba da bude kopiran sa projekta "Proffesional Divisons advanced lighting system project (ALSP"	Uzbudljiv dizajn i deo popisa
6	Konfiguracija sistema biće kompatibilna sa Windows 7 operativnim sistemom.	Obim menadžmenta za puštanje 1.0
7	Timu treba da bude dozvoljeno angažovanje dva nova zaposlena (full time angažovanje) posle početne faze sa znanjem za koje se smatra da je neophodno u projektu	Maksimalano dozvoljeno proširenje budžeta
8	KCH5444 mikroprocesor će biti korišćen na prekidaču (Control Switch)	Već u upotrebi u kompaniji
9	Kupljene softverske komponente su dozvoljene do trenutka dok se ne zloupotrebe od strane kompanije	Bez dugotrajnih troškova i uticaja prodaje na softver

VIDEO

An introduction to Requirements Engineering

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

Poglavlje 6

Domaći zadatak

ZADATAK BR. 1.

Opis domaćeg zadatka

1. Integracija sistema je važna aktivnost inženjeringa zahteva. Navedite probleme koji se mogu pojaviti kada je potrebno integrisati sistemi različitih proizvođača. Kako se ti problemi mogu rešiti?
2. Sugerišite kako bi sledeće zahteve trebalo preformulisati kako bi oni bili izraženi kvantitativno. Možete koristiti bilo koju metriku za iskazivanje zahteva.
a. Bibliotečki sistem treba da bude lak za korišćenje
b. Bibliotečki sistem treba da obezbedi pouzdane servise za sve vrste korisnika
c. Bibliotečki sistem treba da obezbedi brz odgovor za sve korisnike koji žele da dobiju informaciju o knjigama
Rešenje zadatka pošaljite na mail adresu nikola.gavrilovic@metropolitan.ac.rs ili na jovana.jovic@metropolitan.ac.rs

Zaključak

ZAKLJUČAK

Šta smo naučili u ovoj lekciji?

U predavanju je naglašeno da se problemi u razvoju sw proizvoda najčešće odnose na zahteve sistema jer zahtevi ne odslikavaju realne potrebe korisnika, često su nekonzistentni ili nekompletni, postoji nerazumevanje između onih koji specificiraju zahteve i inženjera zaduženih za razvoj i održavanje sistema.

Kroz najčešće postavljena pitanja vezana za zahteve sistema je učinjen pokušaj da se daju osnovne informacije i definicije vezane za razumevanje zahteva. Tako je objašnjeno šta su zahtevi, šta podrazumeva proces inženjeringa zahteva, šta se dešava kada su zahtevi loši, šta je dokument zahteva, šta su stejkholderi sistema itd.
U predavanju je napravljena razlika između sistemski i softverskih zahteva i detaljno objašnjena struktura dokumenta zahteva koja se može prilagođavati premu tipu sistema za koji se pravi.