**1, 2, 3. NEDELJA**

**==============**

**1. Na šta se odnose zahtevi projekta, a na šta zahtevi proizvoda?**

* Zahtevi proizvoda utiču na svojstva softvera u razvoju.
* Zahtevi projekta obuhvataju elemente koji učestvuju u projektovanju i u izradi softvera.

Zahtevi proizvoda (product requirements) su svi zahtevi koji utiču na svojstva softverskog sistema koji treba razviti. Kako projekti imaju i druge rezultate, sem dobijanje proizvoda, to se zahtevi projekta razlikuju od zahteva proizvoda.

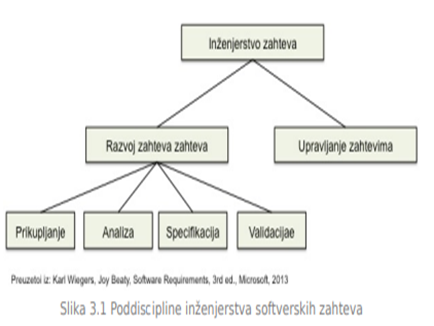
Dokument Specifikacija softverskih zahteva (SRS) sadrži zahteve proizvoda, ali ne i zahteve projekta, te ne sadrži detalje vezane za projektovanje i izradu softvera, plan projekta, planove testiranja, i slične informacije.

**2. Objasnite isciplinu inženjerstva zahteva koja se naziva razvoj zahteva.**

**3. Opišite osnovne aktivnosti procesa razvoja zahteva.**

Isti odgovor:

Inženjerstvo zahteva se deli na razvoj zahteva i upravljanje zahtevima. Razvoj zahteva se realizuje izvršenjem sledećih aktivnosti:

• Pikupljanje zahteva (elicitation)

• Analiza zahteva (analysis)

• Specifikacija zahteva (specification), i

• Potvrđivanje (validation)

Dole opisujem sve aktivnosti i postoji pitanje za potvrdjivanje (jedan odgovor za 3 pitanja)

PRIKUPLJANJE ZAHTEVA Prilikom prikupljanja zahteva koristite različite tehnike.

Definišite viziju i okvir projekta, što obuhvata poslovne zahteve i granice sistema. Identifikujte klase korisnika, opišite njihove karakteristike i odaberite "šampiona proizvoda" za svaku klasu. Vodite fokus grupe s tipičnim korisnicima i radite s predstavnicima korisnika kako biste utvrdili zahteve. Definišite interakciju korisnik-sistem, utvrdite događaje i odgovore sistema. Organizujte intervjue, radionice i posmatrajte rad ljudi na poslu za efikasno prikupljanje zahteva. Proučavanje dokumentacije radi dodatnih informacija.

ANALIZA ZAHTEVA

Analiza zahteva poboljšava zahteve, otklanja greške i nedostatke u njima, a omogućava bolje razumevanje zahteva od strane svih aktera

Analiza zahteva (requirements analysis) poboljšava utvrđene zahteve i obezbeđuje da ih svi akteri razumeju i ukažu na eventualne greške, ispuštene zahteve i druge nedostatke.

Analiza uključuje dekompoziciju uopštenih zahteva na odgovarajući nivo detalja**, izradu prototipova, ocenjuje ostvarljivost, i usklađuje prioritete zahteva.**

§ Cilj je dobijanje zahteva dovoljnog kvaliteta i sa dovoljnom preciznošću koje omogućavaju menadžerima da urade realističnu procenu projekta, a tehničkim članovima razvojnog tima mogućnost projektovanja, konstrukcije i testiranja softvera.

**Analiza obuhvata sledeće aktivnosti:**

• Određivanje prioriteta zahtevima - daje se prednost onim zahtevima koji omogućavaju najveću vrednost ili hitnu funkcionalnost.

• Kreiranje rečnika podataka - definicija podataka, olakšava komunikaciju

• Modeliranje zahteva - daje grafičko predstavljanje zahteva (model toka podataka, ER dijagrami, dijagrami stanja, mape dijaloga i dr.)

• Analiza interfejsa između sistema i spoljnjeg okruženja - to omogućava dobru usaglašenost sistema sa okruženjem.

• Dodeljivanje zahteva podsistemima - Složeni sistemi imaju podsisteme i preporučljivo je da im se dodele odgovarajući zahtevi.

SPECIFIKACIJA ZAHTEVA

Specifikacija zahteva ima **svrhu dokumentovanja različitih vrsta zahteva** na konzistentan, pristupačan i proverljiv način, razumljiv svim akterima sistema.

**Poslovni zahtevi** se često beleže u dokumentu o viziji i okviru projekta,

**Zahtevi korisnika** izražavaju u formi slučajeva korišćenja ili priča korisnika.

Detaljniji **funkcionalni i nefunkcionalni** softverski zahtevi se dokumentuju u specifikaciji softverskih zahteva (SRS).

4. Koje akcije podrazumeva potvrđivanje zahteva?

POTVRĐIVANJE ZAHTEVA (validation) obezbeđuje kontrolu da li su zahtevi korektni, i pokazuju karakteristike kvaliteta i zadovoljenje potreba korisnika.

Ova aktivnost podrazumeva sledeće akcije:

• Recenzija zahteva: Recenzenti koji predstavljaju različite poglede na softver (analitičar, kupac, programer, tester) pažljivo analiziraju zapisane zahteve, modele analize, i povezane informacije, radi nalaženja grešaka.

• Testiranje zahteva: Definisanje testova koji proveravaju da li su određeni zahtevi korisnika ostvareni

• Definisanje kriterijuma prihvatljivosti: Pitate korisnika da opiše kako bi on odredio da li je neko rešenje zadovoljava njegove potrebe i da je spremno za upotrebu.

• Simulacija zahteva: Postoje komercijalni alati koji omogućavaju razvojnom timu da simuliraju rad razvijenog prototipa sistema. Primenom prototipova, analitičar može, u interakciji sa korisnikom, da utvrdi prihvatljivost pojedinih rešenja.

5. Objasnite disciplinu inženjerstva zahteva koja se naziva upravljanje softverskim zahtevima?

Inženjerstvo zahteva se deli na razvoj zahteva i upravljanje zahtevima.

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.

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.

6. Objasnite pojam poslovnih zahteva.

Poslovni zahtevi (budiness requirements)

-objašnjavaju ZAŠTO organizacija primenjuje sistem, koje poslovne želi da ostvari.

Poslovni zahtevi se unose u Dokument o viziji i obuhvatu sistema. Podrazumeva se da je organizacija, pre početka rada na projektu razvoja sistema, utvrdila poslovne potrebe ili mogućnosti tržišta koje sistem treba da zadovolji.

7. Koji su akteri u razvoju softverskog sistema? Navedite i objasnite različite klase aktera.

Akteri u razvoju softverskog sistema su ljudi ili organizacije koje sistem direktno ili indirektno utiče, a istovremeno imaju posredan ili neposredan uticaj na sistemske zahteve.

Ovi akteri obuhvataju:

1. Krajnje korisnike sistema: Osobe ili organizacije koje direktno koriste softverski sistem kako bi obavljale određene zadatke.

2. Menadžere i druge uključene u organizacione procese: Osobe koje su odgovorne za organizacione procese koje sistem podržava, kao i menadžere koji imaju nadležnost nad sistemom.

3. Inženjere za razvoj i održavanje sistema: Stručnjaci odgovorni za razvoj, implementaciju i održavanje softverskog sistema.

4. Kupce sistema: Osobe ili organizacije koje planiraju koristiti sistem kako bi dobile određene usluge.

5. Eksterna tela: Vlade i organizacije koje imaju nadležnost u sertifikaciji, donošenju zakona ili regulisanju oblasti koje sistem pokriva.

8. Sta su procesni zahtevi? Navedite bar dva primera.

Procesni zahtevi u softverskom razvoju obuhvataju zahteve vezane za sam proces razvoja sistema, uključujući upotrebu standarda, CASE alata i generisanje izveštaja za menadžment. . Ovi zahtevi mogu biti nametnuti od strane organizacije za razvoj, kupca proizvoda ili treće strane, kao što su sigurnosni regulatori.

Oni se pojavljuju kada korisnici žele uticati na proces razvoja i mogu obuhvatiti:

1. Definisanje razvojnog procesa u skladu s ISO 9000 standardom.

2. Korišćenje određenog paketa CASE alata (na primer, XYZ) za razvoj sistema.

3. Generisanje izveštaja za menadžment koji prate napor uložen u razvoj svake identifikovane komponente sistema.

4. Specifikacija plana za otklanjanje katastrofalnih situacija pri razvoju sistema.

9. Na Sta se odnose eksterni zahtevi? Navedite bar dva primera.

Eksterni zahtevi mogu se klasifikovati kao zahtevi proizvoda ili procesni zahtevi, a potiču iz okruženja u kojem se sistem razvija. Oni obuhvataju informacije o aplikativnom domenu, organizacijskim mišljenjima, potrebama za integracijom sa drugim sistemima i relevantnim regulativama.

Primeri eksternih zahteva uključuju:

1. Sistem za edukaciju studenata: Format podataka o studentskom zapisu se mora podudarati sa onim koji se koristi u nacionalnom sistemu za evidenciju studenata.

2. Sistem medicinskih podataka: Službenik za zaštitu podataka u organizaciji mora da pre nego što se sistem stavi u rad, izda sertifikat da se svi podaci održavaju prema legislativi za zaštitu podataka.

Prvi od gornjih zahteva je zahtev koji proizilazi iz eksternog okruženja. Drugi od zahteva proizilazi iz potrebe da se sistem usaglasi sa legislativom za zaštitu podataka.

Eksterni zahtevi često nisu formulisani kao "sistem treba" ili "sistem ne sme", već opisuju okruženje koje se mora uzeti u obzir.

10. Koji sve načini mogu biti za prikupljanje zahteva?

Sve vrste zahteva treba prikupiti primenom različitih tehnika. Ovde se daju neke preporuke, na osnovu prakse.

**Definiši viziju projekta i njegov okvir:** Dokument o viziji i okviru projekta sadrži poslovne zahteve za proizvod. Iskaz o viziji daje svim akterima zajedničko razumevanje u proizvodu koji treba razviti. Okvir definiše granice sistema koji treba razviti, tj. šta treba, a šta ne treba. Na osnovu ovog dokumenta se vrši ocena predloženih zahteva.

**Utvrdi klase korisnika i njihove karakteristike:** Utvrdi grupe korisnike sistema, kojese razlikuju po frekvenciji korišćenja sistema, svojstvima koja koriste, nivou privilegija, ili iskustvu.

**Odaberi "šampiona proizvoda" za svako klasu korisnika:** Odaberi pojedinaca koji je tipičan za svoju klasu korisnika. On predstavlja potrebe svoje klase i donosi odluke u njeno ime.

**Vodite fokus grupe s tipičnim korisnicima:** Najčešće one ne donose nikakve odluke. Radi sa predstavnicima korisnika da bi utvrdio zahteve korisnika: Sa predstavnicima razgovarajte šta očekuju od softvera da bi uspešno obavili svoj posao i koju vrednost očekuju da postignu. Zahtevi korisnika se najšeće daju u formi slučajeva korišćenja, priča korisnika I scenarija.

**Utvrdi događaje u sistemu i moguće odgovore sistema:** Sačini listu spoljnih događaja i odgovora koje sistem treba da da na njih.

**Organizuj intervjue sa akterima:** Mogu biti pojedinačni i grupni. Oni su efikasni u prikupljanju zahteva, jer su fokusirani samo na zahteve određenog aktera.

**Osmatraj rad ljudi na poslu:** Napravi dijagram procesa radnog mesta.

**Raspodeli upitnike:** Korisni u slučaju velikih i distribuiranih grupa korisnika.

**Prouči raspoloživu dokumentaciju:** Prikupi potrebne informacije.

**Prouči prihvatljive postojeće zahteve, kao i izveštaje o problemima.**

11. Objasnite zahteve na tri nivoa: poslovne, zahteve korisnika i funkcionalne zahteve.

Poslovni zahtevi (budiness requirements)

-objašnjavaju ZAŠTO organizacija primenjuje sistem, koje poslovne želi da ostvari.

Poslovni zahtevi se unose u Dokument o viziji i obuhvatu sistema. Podrazumeva se da je organizacija, pre početka rada na projektu razvoja sistema, utvrdila poslovne potrebe ili mogućnosti tržišta koje sistem treba da zadovolji.

Zahtevi korisnika (user requirements)

-opisuju ciljeve ili zadatke koje korisnici moraju da izvrše na sistemu.

Zahtevi korisnika takođe uključuju opise atributa sistema ili njegove karakteristike koje su od važnosti za korisnika. Zahtevi korisnika se izražavaju u vidu slučajeva korišćenja (use cases), priča korisnika (user stories) i tabela događaj-odgovor (event-response table).

Funkcionalni zahtevi (functional requirements)

-određuju ponašanje sistema pod određenim uslovima.

Oni opisuju ŠTA inženjer razvoja treba da uradi da bi omogući korisniku da izvrši svoj zadatak.

Oni se obično specificiraju iskazima u kojima se korsiti reč "mora".

Analitičar poslovanja (business analyst) na osnovu razgovora sa korisnicima i inženjerima razvoja prevodi zahteve korisnika u funkcionalne zahteve i ciljeve kvaliteta. On definiše dokument: Specifikacija softverskih zahteva koji opisuje, koliko je to potrebno, očekivano ponašanje softverskog sistema.

12. Koji su problemi vezani za zahteve?

Razvoj softverskih sistema često se suočava s problemima isporuke sa zakašnjenjem i prekoračenjem planiranog budžeta. Greške u zahtevima sistema predstavljaju glavni uzrok tih problema, a mogu proizaći iz nedovoljnog uključenja korisnika, pogrešnog planiranja, dvosmislenih zahteva, dodavanja neplaniranih zahteva ili zanemarivanja određenih aktera razvoja.

Troškovi otklanjanja grešaka u zahtevima kasnije tokom razvoja znatno su veći u odnosu na otklanjanje istih grešaka u ranoj fazi.

Identifikacija aktera, tj. osoba ili organizacija pogođenih implementacijom sistema, ključna je za razumevanje različitih perspektiva i potreba koje treba zadovoljiti.

13. Objasnite mapu ekosistema kao tehniku za određivanje okvira.

Mapa ekosistema pokazuje sve sisteme povezane sa vašim sistemom koji su u međusobnoj interakciji i prikazuju prirodu ovih interakcija.

Mapa sistema predstavlja okvir prikazujući sve povezane sisteme.

Za razliku od dijagrama konteksta, mape ekosistema pokazuju i druge sisteme koji su povezani sa sistemom koji se razvija, uključujući i sisteme sa kojima on nema direktne interfejse.

Možete utvrditi eksterne sistema prateći podatke, tj. uključujući sisteme koje koriste podatke sistema koji se razvija. Kada dođete do sistema koji ne koristi podatke vašeg sistem, to znači da ste došli do granice vašeg sistema.

14. Sta je iskaz o viziji? Kako se formuliše?

Iskaz o viziji sumira dugoročnu svrhu i namenu proizvoda.

On treba da odražava uravnotežen pogled koji će da zadovolji očekivanja različitih aktera.

Iskaz o viziji treba da odražava uravnotežen pogled koji će da zadovolji očekivana različitih aktera projekta. On može da izgleda idealistički, ali on mora da se oslanja na realnost postojećih tržišta, arhitekture organizacije, strateške smernice organizacije, i na ograničenja resursa.

Sledeći uzorak ključnih reči može da se koristi za uspešnu formulaciju iskaza o viziji:

• Za (ciljni kupac)

• Koji (iskaz o potrebi ili prilici)

• Naziv proizvoda

• Je (kategorija proizvoda)

• Koji (glavne sposobnosti, ključna korist, neodoljiv razlog da se kupi ili upotrebi)

• Za razliku od (primarna konkurentna alternativa, sadašnji sistem, sadašnji poslovni proces),

• Naš proizvod (iskaz o primarnom razlikovanju i prednosti novog proizvoda).

15. Objasnite stablo svojstava kao tehniku za određivanje okvira

Stablo svojstava prikazuje logično grupisana svojstva sistema, hijerarhiјski podeljena na detaljnije nivoe prikazivanja.

Stablo svojstava obezbeđuje koncizan pogled na sva svojstva koja su planirana za projekat, Idealan je za prikaz direktorima koji žele kratak prikaz projekta.

Stablo svojstva može da prikaže tri nivoa detaljisanja svojstava, koji se označavaju sa Nivo 1, Nivo 2 i Nivo 3, tj. N1, N2 i N3.

Svako svojstvo ima svoju granu. Kada planirate neko izdanje ili iteraciju, možete definisati njegov okvir izborom skupa svojstava koje treba izvršiti. Svojstva se biraju u celina, a u slučaju svojstava N2 i N3 moguće ih je izabrati kao deo svojstava glavne grane. Ako se koriste boje, može se prikazati stablo svojstava svih izdanja ili iteracija, pri čemu jedna boja odgovara jednom izdanju ili iteraciji.

16. Koja je svrha dokumenta o viziji i okviru? Navedite i objasnite neka od najvaznijh poglavlja ovog dokumenta.

**Dokument o viziji i okviru (vision and scope document) sadrži poslovne ciljeve koji su osnova za naredne aktivnosti na projektu.**

Organizacije koje razvijaju komercijalni softverski proizvod često ovaj dokument zovu"dokument sa zahtevima marketinga" (marketing requirements document - MRD). MRD ulazi više u detalje ciljnog tržišnog segmenta i pitanjima koje se odnose na poslovni uspeh softvera. Vlasnik dokumenta o viziji i okviru je sponzor projekta, neko ko finansira projekat ili neko koji ima sličnu ulogu.

Biznis analitičar može da pomogne ovima u formulisanju i u pisanju dokumenta o viziji i okviru. Ulaz u definisanje poslovnih ciljeva treba da obezbede oni koji pokreću projekat i koji najbolje znaju razlog za to pokretanje. Pokretač projekta može biti kupac-naručilac softvera ili rukovodstvo organizacije koje planira da proizvod (softver) prodaje na tržištu ili vizionar projekta, menadžer projekta, ekspert u domenu primene softvera, ili član marketing tima.

17. Sta predstavljaju vizija proizvoda i okvir proizvoda?

Iskaz o viziji daje svim akterima zajedničko razumevanje u proizvodu koji treba razviti. Okvir definiše granice sistema koji treba razviti, tj. šta treba, a šta ne treba.

Iskaz o viziji sumira dugoročnu svrhu i namenu proizvoda. On treba da odražava uravnotežen pogled koji će da zadovolji očekivanja različitih aktera.

Okvir se definiše nizom funkcionalnih zahteva koji se planiraju za izvršenje (implementaciju) u okviru planiranog izdanja ili iteracije. Definisanje okvira narednih izdanja ima ograničenja i isključenja.

18. Kako se određuje poslovna korist od projekta i koji je njen značaj za razvoj projekta?

Poslovna korist treba da donese istinsku vrednost sponzorima projekta i korisnicima proizvoda.

Poslovni zahtevi postavljaju kontekst i omogućavaju merenje koristi koje biznis može da očekuje od projekta. Organizacije ne bi trebalo da započnu nijedan projekat bez jasnog razumevanja vrednosti koju projekat donosi biznisu, tj. uspešnosti poslovanja.

Postavite 4 merljive ciljeve u vidu poslovnih ciljeva i onda definišite matricu uspeha koja vam omogućava da merite do koje mere ste na putu da zadovoljite ove ciljeve.

Sponzori projekta, koji ga finansiraju, rukovodstvo organizacije, menadžeri marketinga i vizionari proizvoda treba da odrede poslovne ciljeve projekta. Međutim, obično je izazov da se utvrde poslovne koristi koje projekat treba da ostvari, jer mogu postojati različiti stavovi aktera. Zbog toga, biznis analitičar treba da ima pripremljena pitanja i da zna koga treba da pita, da bi mogao da formuliše poslovnu korist koju projekt treba da omogući organizaciji.

Poslovna korist treba da donese istinsku vrednost sponzorima projekta i korisnicima proizvoda.

19. Sta se dogadja ako su poslovni zahtevi suprotstavljeni? Na koji način se moze rešiti problem suprotstavijenih poslovnih zahteva?

**Suprotstavljeni poslovni interesi:** Poslovni akteri, poput proizvođača, trgovca i kupca, mogu imati suprotstavljene interese u projektu, kao što je slučaj sa željom kupca za brzom kupovinom i prodavca za zadržavanjem kupca u trgovini.

**Razrešavanje konflikta pre biznis analize:** Pre nego što biznis analitičar počne rad na detaljnim softverskim zahtevima, donosioci odluka u projektu trebaju rešiti potencijalne konflikte između poslovnih interesa kako bi se osigurala jasna usmerenost projekta.

**Fokus na poslovnoj vrednosti:** Ključni fokus treba biti na obezbeđivanju najveće poslovne vrednosti za sve aktere, prioritetno za krajnje korisnike (agente), uzimajući u obzir njihove osnovne poslovne ciljeve.

**Razlučivanje konflikata na biznis nivou:** Konfliktne situacije ne treba rešavati isključivo na nivou softverskog tima. Biznis analitičar uočava konflikte, ali rešavanje tih situacija spada u domen biznisa, zahtevajući donošenje odluka na biznis nivou.

**Kontrola širenja okvira projekta:** Kontrola nad širenjem okvira projekta zbog interesa pojedinih aktera ključna je kako bi se održala usklađenost sa poslovnim ciljevima i sprečilo nepotrebno proširivanje opsega.

**Promene u poslovnim zahtevima:** Ukoliko razrešavanje konflikta dovodi do promene poslovnih ciljeva, neophodno je zatražiti i promene u planu projekta, uključujući rokove, budžet i resurse.

**Važnost povratka na početne poslovne zahteve:** Kod dužih projekata, gde se češće javljaju promene, povratak na početne poslovne zahteve može biti ključan za usklađivanje sa novonastalim situacijama.

**Prilagodba funkcionalnih i nefunkcionalnih zahteva:** U skladu sa promenama u poslovnim ciljevima, funkcionalni i nefunkcionalni zahtevi takođe trebaju biti prilagođeni kako bi odražavali nove uslove i prioritete.

20. Koje pripremne aktivnosti se mogu sprovesti pre razvoja softvera u odredjenom domenu primene?

Svi akteri u projektu, a posebno analitičari poslovanja moraju da prođu odgovarajuću obuku za razvoj softvera u određenom domenu primene. Obično neki članovi projektnog tima dobiju ulogu analitičara poslovanja na projektu, iako često i nemaju formalno obrazovanje za vršenje te uloge.

Analitiza poslovanja je specijalizovana i izazovna uloga. Pored neophodne obuke za izvršenje ovakve uloge, potrebno je da analitičar ima dobre komunikacione sposobnosti, da je zainteresovan za ovu ulogu i da u što većoj meri poznaje i oblast primene softvera, te da razume jezik korisnika, da bi razumeo i njihove potrebe.

Potrebno je realizovati sledeće akcije:

• Obuka analitičara poslovanja (business analysts): U većoj ili manjoj meri, svi članovi tima vrše i ulogu analitičara poslovanja. Analitičar poslovanja trebalo bi da ima bar nekoliko dana obuke u različitim aktivnosti poslovne analize. To je osnova za izgradnju sopstvenog iskustva i dalje usavršavanje. On mora da bude strpljiv, da je dobro organizovan, da ima efektivnu veštinu komunikacije sa drugima, kao i da razume domen primene softvera.

• Obuka aktera o zahtevima: Najefektivnija obuka obuhvata više funkcionalnih disciplina, ne samo rad sa zahtevima. Akteri treba da imaju dan-dva obuke za rad sa zahtevima, da bi razumeli terminologiju, ključne koncepte i prakse, da bi shvatili zašto je inženjerstvo zahteva klučno za uspeh projekta razvoja softvera. Ova obuka je dobra i za razvoj sposobnosti tima i povezivanje članova tima.

• Obuka inženjera razvoja o domenu primene softvera: Organizujte seminar za inženjere razvoja na kome će se upoznati sa domenom primene softvera, sa aktivnosti kupaca, sa njihovom terminologijom, i sa ciljevima zbog kojih se realizuje projekat. To je i prilika da se upoznaju sa predstavnicima korisnika sa kojima će sarađivati tokom trajanja projekta. Definisanje procesa inženjerstva zahteva: Dokumentujete sve aktivnosti procesa inženjerstva zahteva. Dajte uputstva kako da se one sprovedu da bi analitičari uspešno odradili svoj posao.

• Kreiranje rečnika termina: Rečnik termina sadrži potrebne termine iz oblasti domena primene sistema. Time se minimiziraju nerazumevanja. Uključite sinonime, skraćenice i dr. Takav rečnik postaje stalna vrednost organizacije i služi i u sledećim projektima. Može da posluži novim članovima tima da se upoznaju sa problematikom i terminologijom, što je neophodno da bi se uključili uspešno u projekat.

**4, 5, 6. NEDELJA**

**==============**

21. Šta se opisuje preduslovima i postuslovima slučajeva korišćenja?

Preduslovi (preconditions) definišu šta mora da bude ispunjeno da bi sistem počeo da izvršava slučaj korišćenja. Sistem mora da ima testove za utvrđivanje ispunjenosti preduslova. Preduslovi mogu da opišu stanja sistema ("ATM mora da u sebi ima novac, da bi mogao da pokrenem slučaj povlačenje novca iz automata"), ali ne opisuju nameru korisnika ("korisniku je potreban novac").

Kada sistem dobije događaj koji ga pokreće na izvršenje nekog uslova korišćenja, on onda ide na proveru preduslova, i ako su zadovoljeni, počinje sa izvršavanjem slučaja korišćenja. Pokretač nije preduslov.

Postuslovi (postconditions) opisuju stanje sistema posle uspešnog izvršenja slučaja korišćenja. Postuslovi mogu da obuhvate:

· nešto što korisnik može da opazi

· fizički rezultat

· promenjeno je unutrašnje stanje sistema

Svi izlazi iz slučaja korišćenja bi trebalo da obezbede određenu vrednost korisniku, te su svi oni opažljivi korisniku. Ono što je neophodno da bude postuslov, a što korisnik ne mora da opazi, to mora biznis analitičar da stavi u postuslove, ako je to uslov da sistem radi ispravno.

22. Objasnite intervju kao tehniku za utvrđivanje zahteva.

Intervju može biti individualni ili intervjui sa malim grupama korisnika.

-Jednostavnije ih je organizovati od radionica, koje su rad sa većom grupom korisnika.

-Intervjui se dobri za izazivanje poslovnih zahteva od rukovodioca koji imaju malo vremena za razgovor.

Saveti za vođenje intervjua:

· Uspostavite odnos: Predstavite se, upoznajte sagovornika sa sadržajem razgovora, ukažite mu na ciljeve razgovora i pitajte da li on ima neka pitanja.

· Držite se okvira: Usmeravajte razgovor ka njegovim ciljevima.

· Pripremite pitanja i model sagovornika: Pripremite se za razgovor pripremom pitanja i možda neke konstatacije. To daje dobar početak razgovora.

· Sugerišite ideje: Na osnovu odgovora koje dobijete, kreirajte ideje i alternative za vreme izazivanja zahteva. Ako se korisnik teško izražava o svojim potrebama, onda idite na tehniku osmatranja njegovog rada.

· Aktivno slušajte: Nagnite se napred, pokažite strpljenje, verbalno reagujte kada treba i pitajte kada vam nešto nije jasno. Rezimirajte šta mislite da je korisnik rekao, ako vam nije najjasnije, da bi sa njim to usaglasili.

Možete snimati razgovor da bi preslušali nejasne delove razgovora, a i da bi imali i dokaz šta vam je sagovornik rekao, mada je njegova potvrda beleške merodavnija. Odmah posle intervjua, dok vam sećanje ne izbledi, sačinite belešku sa intervjua i dostavite osobi sa kojom ste razgovarali, da bi proverili da li ste njene stavove i odgovore pravilno protumačili i zapisali. Tražite i da sagovornik to potvrdi potpisom na dostavljen dokument - beleška sa razgovora.

23. Ko rešava problem sukobljenih zahteva?

Odluke treba donositi što niže u hijerarhiji organizacije od strane dobro

informisanih ljudi koji su bliski problemima.

Na početku projekta odredite ko će biti donosioci odluka u vezi sa zahtevima.

Ako nije jasno ko je odgovoran za donošenje tih odluka ili ako ovlašćeni pojedinci odustanu od svojih odgovornosti, odluke će pasti na programere ili analitičari podrazumevano. Međutim, većina njih nema potrebno znanje i perspektivu za donošenje najboljih poslovnih odluka.

24. Koje su preporuke za planiranje izazivanja zahteva? Kako se pripremiti za aktivnost izazivanja zahteva?

Plan izazivanja uključuje tehnike koje koristite, kada planirate da ih koristite i sa kojom svrhom.

Na početku projekta, poslovni analitičar treba da isplanira pristup projektu zahtevima. Čak i jednostavan plan akcije povećava šansu za uspeh i postavlja realna očekivanja zainteresovane strane. Samo ako ste izričito zauzeti za resurse, raspored i rezultate, možete izbeći da se učesnici povuku da rade druge poslove. Plan privlačenja uključuje tehnike koje ćete koristiti, kada ih planirate da koristite i u koje svrhe. Kao i bilo koji plan, koristite ga kao vodič i podsetnik tokom celog projekta, ali shvatite da ćete možda morati da menjate plan tokom celog projekta.

Vaš plan treba da sadrži sledeće stavke:

· Ciljevi za pronalaženje Planirajte ciljeve za prikupljanje zahteva za ceo projekat i ciljeve za svaku planiranu aktivnost izvlačenja.

· Strategija za izvlačenje i planirane tehnike Odlučite koje ćete tehnike koristiti sa različitim interesnim grupama. Možete koristiti neku kombinaciju upitnika, radionica, poseta kupcima, individualnih intervjua i drugih tehnika, u zavisnosti od pristupa koji imate zainteresovanim stranama, vremenskih ograničenja i vašeg znanja o postojećem sistemu.

· Procena rasporeda i resursa Identifikujte i kupca i učesnike u razvoju za različite aktivnosti prikupljanja zahteva, zajedno sa procenama potrebnog napora i kalendarskog vremena. Možda ćete moći samo da identifikujete korisničke klase, a ne neke konkretne pojedince, ali to će menadžerima omogućiti da započnu planiranje nadolazećih potreba za resursima. Procenite vreme BA, uključujući vreme za pripremu za izuzeće i za obavljanje naknadnih analiza.

· Dokumenti i sistemi potrebni za nezavisno prikupljanje podataka Ako sprovodite analizu dokumenata, sistemskog interfejsa ili korisničkog interfejsa, identifikujte materijale potrebne da biste ih osigurali kada su vam potrebni.

· Rad na očekivanom prikupljanju zahteva Znajući da ćete kreirati listu slučajeva upotrebe, SRS, analize rezultata upitnika ili specifikacije kvaliteta, pomažu vam da ciljate na prave aktere projekta, teme i detalje tokom iznošenja.

· Rizici utvrđivanja zahteva: Identifikujte faktore koji bi mogli da ometu vašu sposobnost da dovršite aktivnosti utvrđivanja zahteva kako je planirano. Procenite ozbiljnost svakog rizika i odlučite kako ga možete ublažiti ili kontrolisati.

25. Objasnite fokus grupu kao tehniku za utvrđivanje zahteva.

Fokus grupa je reprezentativna grupa korisnika koja se saziva radi usmeravane aktivnosti izazivanja zahteva. Posmatranjem otkrivate neizrečene zahteve.

One su interaktivne i očekuje se da svi učesnici govore. Fokus grupe su zgodne za izražavanje stavova korisnika, impresija, preferencija, i potreba. Važne su pri razvoju komercijalnog proizvoda kada nemate direktne buduće korisnike.

Izbor članova grupe treba pažljivo uraditi. Ili ćete držati jednu grupu sa predstavnicima svih klasa korisnika, ili ćete držati više fokus grupa, sa svakom klasom korisnika posebno.

Radom fokus grupe se rukovodi, ali na način da svako može slobodno da govori. Možete snimati diskusiju. One daju subjektivna mišljenja koja kasnije morate da analizirate i da im određujete prioritet. Učesnici fokus frupa obično ne donose nikakve odluke.

Posmatranja Neki učesnici iz raznih razloga ne iskazuje sve što znaju i žele. Zato ih treba posmatrati dok rade. Posmatranja traju te ih treba pažljivo koristiti. Da ne bi ometali u radu ljude, ograničite posmatranja na najviše dva sata. Izaberite važne i visokorizične zadatke za posmatranje.

Osmatranja mogu biti tiha (kada se korisici ne mogu uznemiravati dok rade), a mogu biti i interaktivna. (kada je to mogućne radi razjašnjavanja zadataka koji radi). Dokumentujte vaša opažanja tokom posmatranja. Možete i snimiti rad zaposlenog, ako je to dozvoljeno.

Upitnici (questionairs) su jeftini i dobri u slučaju velikog broja ispitanika. Njavažnije je dobro pripremiti pitanja. Evo nekih preporuka:

• Ponudite odgovore koji pokrivaju sve opcije mogućih odgovora.

• Ponuđeni odgovori ne smeju da budu međusobno isključivi.

• Ne postavljajte pitanja na koje se može samo potvrdno odgovoriti.

• Ako koristite ocene, onda ih koristite u svim pitanjima.

• Koristite pitanja sa mogućim odgovorima ako rezultat ide u statističku obradu i analizu.

• Možete koristiti savete stručnjaka za pripremu pitanja, kako bi postavili prava pitanja pravim ljudima.

• Uvek testirajte pitanja pre nego što ih distribuirate.

• Ne pitajte mnogo, ako želite da ljudi odgovore na upitnik.

26. Objasnite vezu između slučajeva korišćenja i poslovnih pravila.

Poslovna pravila mogu da utiču na naredne korake u normalnom scenariju definišući ispravne ulazne vrednosti ili određujući kako će se izvršavati obračun.

Slučajevi korišćenja i poslovna pravila su isprepletani. Neka poslovna pravila ograničavaju uloge svih ili pojedinih delova nekog slučaja korišćenja. Na primer, pravilo da samo korisnici sa posebnom dozvolom mogu da izvršavaju alernativne scenarije. To znači da ovo pravilo zahteva da se postavi dati preduslov i da se on testira pre nego što se korisniku dozvoli dalje izvršavanje. Poslovna pravila mogu da utiču na naredne korake u normalnom scenariju definišući ispravne ulazne vrednosti ili određujući kako će se izvršavati obračuni.

Pretpostavimo da jedan avio prevoznik naplaćuje dodatak na kartu putnika ako on želi da rezerviše posebno sedište u avionu. Kad putnik izvrši slučaj korišćenja u kome on rezerviše sedište u avionu, odgovarajuća poslovna pravila se aktiviraju i menjaju cenu karte putnika, ako je on izabrao određena sedišta sa višom tarifom.

Kada specificirate neko poslovno pravilo, zapišite indentifikator onoga ko to poslovno pravilo koje utiče na slučaj korišćenja postavio i zapišite koji deo slučaja korišćenja svako poslovno pravilo zahvata.

Kada istražujete slučajeve korišćenja, možete otkriti značajna poslovna pravila. Kada hemičar koji učestvuje u izazivanju zahteva za Chemist Tracking System, bilo je zahteva da jedan korisnik ne bi trebalo da vidi pravila drugog, i obrnuto. Tako su došli do novog pravila: "Korisnik može da vidi samo pravila za zahteve koje je on postavio".

Ponekad, vi otkrijete pravila za vreme izazivanja zahteva i njihove analize, ponekad unapredite postojeća pravila, a ponekad vi već znate postojeća pravila koja će sistem morati da poštuje.

27. Objasnite vezu izmedu slučajeva korišćenja i funkcionalnih zahteva.

Slučajevi upotrebe opisuju perspektivu korisnika, sagledavajući spoljno vidljivo ponašanje sistema. Oni ne sadrže sve informacije potrebne programeru.

Programeri softvera ne primenjuju poslovne zahteve ili zahteve korisnika. Oni implementiraju funkcionalne zahteve, specifične bitove ponašanja sistema.

Slučajevi upotrebe opisuju perspektivu korisnika, sagledavajući spoljno vidljivo ponašanje sistema. Oni ne sadrže sve informacije potrebne programeru da bi napisao softver. Korisnik bankomata ne zna za bilo kakvu pozadinsku obradu, poput komunikacije sa računarom banke. Ovaj detalj je korisniku nevidljiv, ali programer mora znati za njega. Programeri koji primaju čak i potpuno opisane slučajeve upotrebe često imaju mnogo pitanja. Da biste smanjili ovu nesigurnost, razmislite o tome da BA (Business Analyst) ima izričito specificiranje funkcionalnih zahteva neophodnih za implementaciju svakog slučaja upotrebe.

Mnogi funkcionalni zahtevi ispadaju iz dijaloških koraka između aktera i sistema. Neke su očigledni, kao što je „Sistem će svakom zahtevu dodeliti jedinstveni redni broj.“ Nema smisla da ih duplirate negde drugde ako su to jasni zahtevi iz slučaja upotrebe. Ostali funkcionalni zahtevi se ne prikazuju u opisu slučaja upotrebe. Na primer, način na koji se dokumentiraju slučajevi upotrebe obično ne precizira šta sistem treba da radi ako nije ispunjen preduslov.

Ovo je primer kako slučajevi upotrebe često ne pružaju sve potrebne informacije programeru da zna šta da gradi. BA mora izvući te nedostajuće zahteve i dostaviti ih programerima i ispitivačima. Analiza dobijanja pogleda programera na zahteve na osnovi pogleda korisnika na zahteve, treba da izvrši BA i time da doda vrednost projektu

Sistem praćenja hemikalija koristio je slučajeve upotrebe prvenstveno kao alat za otkrivanje potrebnih funkcionalnih potreba. Analitičari su napisali samo povremene opise manje složenih slučajeva upotrebe. Zatim su izvukli sve funkcionalne zahteve koji bi, kada se primene, omogućili akteru da izvrši slučaj upotrebe, uključujući alternativne tokove i rukovaoce izuzetaka. Analitičari su ove funkcionalne zahteve dokumentovali u SRS-u, koji je organizovan kao svojstvo proizvoda.

Funkciju povezanu sa slučajem korišćenja možete dokumentovati na više načina. Nijedna od sledećih metoda nije savršena, pa odaberite pristup koji najbolje odgovara načinu na koji želite da dokumentujete i upravljate softverskim zahtevima vašeg projekta.

28. Koje su zamke pri radu sa slućajevima koriscenja?

Previše slučajeva korišćenja:

· Nemojte praviti poseban slučaj za svaki mogući scenario. Fokusirajte se na ključne interakcije umesto da ih pretrpavate.

Kompleksni slučajevi korišćenja:

· Izbegavajte izuzetno složene slučajeve korišćenja jer postaju teški za razumevanje. Odaberite jednostavan osnovni tok i koristite alternativne tokove za opisivanje varijacija.

Projektovanje u slučajevima korišćenja:

· Ograničite se na opisivanje šta korisnik očekuje od sistema, izbegavajte detalje o tome kako će sistem izvršiti zadatke ili kako će izgledati interfejs.

Definisanje podataka u slučajevima korišćenja:

· Izuzmite definiciju podataka iz slučajeva korišćenja. Definišite podatke posebno u rečniku i modelu podataka.

Nepoznatost korisnicima:

· Pišite slučajeve korišćenja na način koji korisnici lako razumeju. Pitajte korisnike da pregledaju i daju povratne informacije. Održavajte jednostavnost kako bi korisnici lakše ostvarili svoje ciljeve.

29. Prednosti korišćenja zahteva koncentrisanih na korisnika, umesto onih koji su koncentrisani na sistem?

???

30. Opišite cikličan način razvoja zahteva.

Vi prvo vršite utvrđivanje zahteva, analizirate šta ste naučili i onda zapisujete neke zahteve. Tada i utvrđujete šta vam od informacija nedostaje i onda vršite dodatno utvrđivanje zahteva i tako se ciklus utvrđivanje-analiza-specifikacija zahteva ponavlja.

To je težak posao koji traje. Ne očekujte da ćete posle par radionica završiti proces utvrđivanja zahteva, tj. da ćete ih prikupiti, otkriti, izvući i definisati. Morate to uraditi primenom više različitih tehnika utvrđivanja zahteva.

31. Šta treba uraditi posle sesije prikupljanja zahteva?

Treba da organizujete i podelite beleške sa sesija, da dokumentujete otvorena pitanja, da klasifikujete prikupljene informacije.

Organizovanje i deljenje beleški sa sesija

Odmah posle završene sesije izazivanja zahteva, dok vam su informacije još sveže, napravite belešku o održanoj sesiji. Ubeležite sve predloge. Dajte na recenziju beleške učesnicima sesije (intervjua, radionice, fokus grupe, i dr.) i u skladu sa tim izvršite izmene u beleški. Sačuvajte i sirove beleške, koje ste vodili na sastanku. Možda će vam trebati kasnije, da se na njih pozovete. Što podelite pripremljenu belešku na recenziju učesnicima, tražeći od njih da potvrde tačnost onoga što su izjavili i što je raspravljano na sesiji. Ako je potrebno, razjasnite razgovorom sa pojedinim učesnicima neka neslaganja. Konačnu verziju beleški pošaljite i ostalim akterima.

Dokumentujte otvorena pitanja

Za vreme izvršenja aktivnosti izazivanja zahteva, primetili ste neka nerazjašnjena pitanja koja treba kasnije razjasniti da bi se popunio jaz u informacijama. Zbog toga, možete pripremiti i nova pitanja. Zapišite ih u alatu za praćenje otvorenih pitanja. Za svako pitanje, dajte vezu sa odgovarajućom beleškom. Zapisujte napredak u razmatranju i rešavanju otvorenih pitanja, Evidentirajte vlasnika otvorenog pitanje, tj. odgovornog da to pitanje reši, i rok do kada se pitanje mora razjasniti. Najbolje da koristite isti alat za praćenje otvorenih pitanja koje koristi tim za razvoj i testiranje softvera.

32. Objasnite radionice kao tehniku za utvrđivanje zahteva.

Radonica je strukturisan sastanak na kome grupa aktera i eksperata rade zajedno na definisanju, kreiranju, poboljšanju i postizanju saglasnosti o zahtevima korisnika.

Rad u grupi je efikasniji u rešavanju konfliktnih situacija, nego individualni rad. Takođe, primena radionica ubrzava proces izazivanja zahteva.

Neki saveti za efektivno vođenje i olakšavanje rada radionice:

- Postavite pravila: Dogovorite osnovna pravila, kao što su vreme početka i kraja, vraćanje na posao na vreme, isključenje mobilnih uređaja, aktivno učešće svakog, i konstruktivno izražavanje mišljenja.

- Raspodelite uloge: Dodelite uloge članovima tima, kao što su beleženje, vođenje vremena, održavanje reda, primena pravila i osiguravanje da svi budu uključeni.

- Planirajte dnevni red: Unapred obavestite učesnike o planu i temama radionice.

- Držite se teme: Proverite da li diskusije odgovaraju poslovnim zahtevima i ostanite fokusirani.

- Zabeležite ideje: Zapišite ideje za kasnije razmatranje.

- Postavite vremenska ograničenja: Odredite vreme za svaku temu i na kraju sumirajte diskusiju pre nego što pređete na sledeću.

- Formirajte male grupe: Optimalno je imati 5-6 učesnika i razmotrite rad u manjim grupama sa različitim korisnicima.

- Podstičite angažovanje: Osigurajte da svi učesnici budu angažovani. Pitajte tiše učesnike za mišljenje i obezbedite da svako ima priliku da izrazi svoje ideje.

33. Objasnite analizu interfejsa sistema kao tehniku za utvrđivanje zahteva.

34. Objasnite upitnik kao tehniku za utvrđivanje zahteva.

35. Dajte preporuke za utvrđivanje slučajeva korišćenja.

36. Kako znate da ste završili sa utvrđivanjem zahteva?

37. Kako klasifikovati informacije dobijene od kupca posle aktivnosti prikupljanja zahteva?

38. Objasniti primenu slučajeva korišćenja i korisničkih priča.

**7, 8, 9. NEDELJA**

39. Koji su načini za dokumentovanje poslovnih pravila?

1. **Katalog pravila**: Poslovna pravila se dokumentuju u formi kataloga, koji služi kao centralizovano skladište za sva pravila. Katalog sadrži detaljne informacije o svakom pravilu, uključujući definicije, primenu, identifikacioni kod i razlog postojanja pravila (izvor). Jednostavan i lak za upotrebu.
2. **Softvera za upravljanje zahtevima**: Poslovna pravila se unose u softver za upravljanje zahtevima kao poseban tip zahteva. Ovaj pristup omogućava da se pravila jasno povežu sa specifičnim zahtevima, funkcijama ili slučajevima upotrebe u softverskim projektima. Lakši za praćenje i ažuriranje
3. **Baze sa poslovnim pravilima:** Baze podataka omogućavaju skladištenje, pretragu, i upravljanje kompleksnim i obimnim skupovima pravila koji se primenjuju u različitim aplikacijama i poslovnim procesima. Efikasniji u rukovanju velikim brojem pravila i sposobnost.

40. Koje preporuke za pisanje zahteva možete izdvojiti?

1. **Jasnost i preciznost:** Svaki čitalac treba da interpretira zahtev na isti način kako ga je autor zamislio.
2. **Perspektiva pisanja:** Odlučite da li ćete pisati iz perspektive sistema (šta sistem treba da uradi) ili korisnika (šta korisnik treba da uradi) i držite se te perspektive.
3. **Praćenje strukture:** Uslov => Događaj pokretanja => Odgovor sistema. Npr. Ako korisnik ima novac na računu i kupi nešto, sistem oduzima novac sa računa.
4. **Nivo detalja:** Pisati samo onoliko detalja koliko je potrebno da se zahtev primeni.
5. **Izbegavanje dvosmislenosti:** Koristite konzistentne pojmove, definišite ih u rečniku projekta i izbegavajte dvosmislene ili neprecizne termine.

41. Objasnite atribute kvaliteta i uslove internacionalizacije.

**Atributi kvaliteta:**

1. **Upotrebljivost**: Opisuje koliko je softver "user-friendly". Obuhvata jednostavnost korišćenja, učenja, pamćenja i efikasnost interakcija.
2. **Performansa**: Odnos između ostvarenog rada sistema i vremena potrebno da se izvrši. Obuhvata kratko vreme odziva, brzinu prenosa podatka i visok protok podataka.
3. **Bezbednost**: Bezbednost se odnosi na čuvanje sistema od neovlašćenog pristupa i korišćenja. Uključuje fizičku bezbednost, bezbednost podataka i softvera.
4. **Sigurnost:** Odnosi se na prevenciju gubitka ili oštećenja, bilo ljudske ili materijalne, koja može proizići iz upotrebe proizvoda. Obuhvata zaštitne mere, opasne radnje koje treba sprečiti, i sigurnosne protokole.

**Uslovi internaionalizacije:** Uslovi koji moraju biti zadovoljeni da bi se softver mogao koristiti u različitim državama. Zahtevi pokrivaju razlike u valutama, formatiranju datuma i brojeva, mernim jedinicama, jeziku i pravopisnim konvencijama, vremenskim zonama, međunarodnim propisima i zakonima i slično.

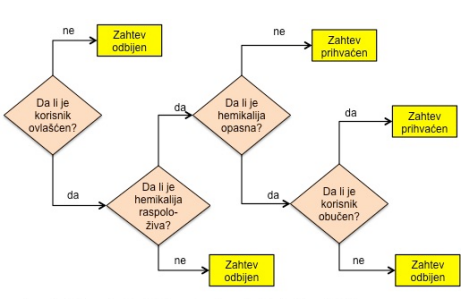
42. Koje preporuke za obeležavanje zahteva možete izdvojiti?

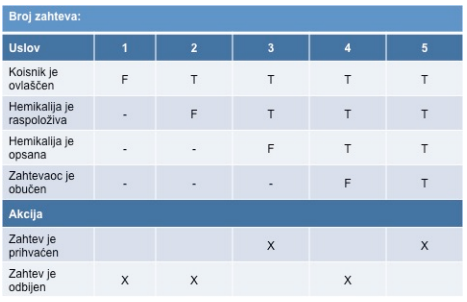
1. **Jedinstveni identifikator za svaki zahtev**: Svaki zahtev bi trebalo da ima svoj jedinstveni identifikator. Ovaj identifikator se koristi pri svakom referenciranju odgovarajućeg zahteva, što olakšava kolaboraciju unutar tima.
2. **Kreiranje sadržaja dokumenta**:Sadržaj omogućava brz i efikasan pregled i navigaciju kroz dokument.
3. **Brojevi tabela i slika**: Dodeljivanje brojeva svim tabelama i slikama u dokumentu sa odgovarajućim naslovima.
4. **Povezivanje referenci**: Koristiti funkciju povezivanja referenci na druge delove dokumenta ili linkove za relevantne elektronske dokumente.

43. Koji su mogući načini za obeležavanje zahteva?

1. **Broj redosleda**: Jednostavan način obeležavanja zahteva je dodeljivanje broja po redosledu pojavljivanja. Na primer, oznake mogu biti UV-9 za slučajeve korišćenja ili DR-29 za funkcionalne zahteve, gde prefiks označava tip zahteva.
2. **Hijerarhijsko dodeljivanje brojeva**: Zahtevi se mogu obeležavati hijerarhijski, na primer FR3.2 za glavni funkcionalni zahtev, a 3.2.1, 3.2.2, 3.2.2.1 itd. za podzahteve. Ova metoda pomaže u organizaciji i praćenju međusobne povezanosti zahteva, međutim, ako se briše neki zahtev, svaka naredna brojčana oznaka se smanjuje za 1, što može dovesti do zabune.
3. **Hijerarhijski tekstualni tagovi:** Hijerarhijski tekstualni tagovi takođe funkcionišu hijerarhijski, ali koriste imena umesto brojeve, kao *Product.Cart.Discount*. Ne menjaju se pri dodavanju, brisanju ili pomeranju drugih zahteva.

44. Objasnite stablo odlučivanja i tabelu odlučivanja.

**Stablo odlučivanja** je grafički alat koji se koristi za prikazivanje procesa donošenja odluka. Stablo se sastoji od čvorova i grana. Čvorovi predstavljaju tačke odlučivanja ili uslova, dok grane predstavljaju moguće opcije ili ishode tih odluka. Svaki čvor sadrži pitanje ili uslov, a grane iz čvora predstavljaju moguće odgovore ili akcije. Proces se nastavlja do krajnjih čvorova, koji predstavljaju konačne odluke ili akcije.

**Tabela odlučivanja** je strukturirani alat koji se koristi za prikazivanje procesa donošenja odluka. Tabela se sastoji iz dva glavna dela: uslova i akcija. Uslovi se obično nalaze na vrhu tabele, dok su akcije prikazane u donjem delu.Za svaki mogući skup uslova, tabela definiše odgovarajuću akciju ili niz akcija koje treba preduzeti.

45. Kako se sve mogu predstaviti i opisati funkcije sistema?

1. **Dijagrami toka podataka (DFD - Data Flow Diagrams)**: Kako podaci teku kroz sistem.
2. **Dijagrami toka procesa**: Kako koraci ili aktivnosti teku unutar specifičnog procesa.
3. **Mape dijaloga (Dialog Maps)**: Interakcija korisnika sa sistemom.
4. **Stabla odlučivanja i tabele odlučivanja**: Vizualizuju proces donošenja odluka.
5. **UML State Diagrams**:Prikazuju različita stanja kroz koja sistem ili njegov deo prolazi.
6. **UML Use Case Diagrams**: Ilustruju kako korisnici (akteri) interaguju sa sistemom.
7. **UML Class Diagrams**: Prikazuju strukturu sistema prikazivanjem klasa, njihovih atributa, metoda i odnosa među njima.
8. **UML Interaction Diagrams**: Kako objekti međusobno interaguju kako vreme prolazi.

46. Koje su prepoznate karakteristike odličnih zahteva?

1. **Kompletnost**: Svaki zahtev treba da sadrži dovoljno informacija potrebnih za razumevanje od strane korisnika.Nedostajuće informacije trebaju biti unete pre početka izrade softvera.
2. **Tačnost**: Zahtevi moraju tačno opisivati sposobnosti sistema da zadovolje potrebe aktera. Tačnost zahteva proveravaju oni koji su zahtev postavili.
3. **Ostvarljivost**: Zahtevi moraju biti izvodljivi u okviru projektnih ograničenja kao što su rokovi, budžet i resursi.
4. **Potreba**: Zahtevi treba da se usklađuju sa poslovnim ciljevima, standardima i regulativom.
5. **Usklađenost sa Prioritetima**: Zahtevi treba da imaju dodeljen prioritet u skladu sa njihovim značajem za projekat.
6. **Nedvosmislenost**: Zahtevi moraju biti formulirani tako da svi učesnici projekta imaju isto razumevanje njihovog značenja.
7. **Proverljivost**: Zahtevi moraju biti definisani na način da se mogu proveriti, idealno testiranjem.

47. Objasnite klasifikaciju poslovnih pravila.

1. **Činjenice**: Činjenice su iskazi o poslovanju koji su istiniti u određenom vremenskom trenutku.

*Primer:* “Svaki kontejner hemikalija ima jedinstveni bar kod za identifikaciju.”

1. **Ograničenja**: Ograničenja su izjave koje ograničavaju radnje koje sistem ili njegovi korisnici mogu obavljati. Ova pravila mogu biti vezana za organizacione politike, zakonske regulacije ili industrijske standarde.

*Primer:* “Veb aplikacije ne smeju sadržati HTML oznake koje su zastarele prema HTML 5 standardu”.

1. **Omogućivači Akcija**: Omogućivači akcija su pravila koja pokreću određenu akciju kada su ispunjeni specifični uslovi. Obično imaju format “ako – onda”.

*Primer:* “Ako je postignut rok trajanja hemijskog kontejnera, onda obavestiti osobu koja je trenutno u posedu tog kontejnera”

1. **Zaključci**: Zaključci stvaraju nove činjenice iz postojećih činjenica. Obično se pišu u obliku “ako – onda”, ali bez preduzimanja radnje, već samo stvarajući novo znanje.

*Primer:* “Ako uplata nije primljena u roku od 30 dana nakon dospeća, račun je delinkventan.”

1. **Obračuni:** Obračuni su procesi u kojima se postojeći podaci transformišu u nove podatke koristeći matematičke formule ili algoritme.

*Primer:* “Ukupna cena neke narudžbine je zbir cene stavki, sa umanjenjem zbog popusta na količinu”

48. Šta su poslovna pravila? Kakva je primena poslovnih pravila u praksi?

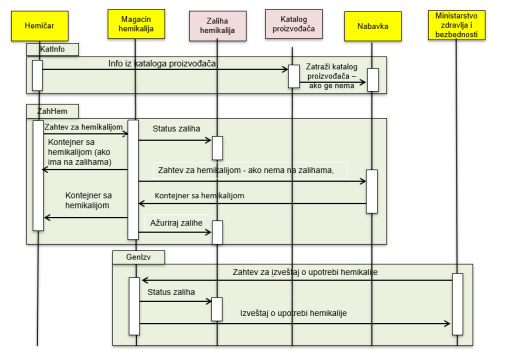
**Poslovna pravila** su formalni iskazi koji definišu principe poslovanja unutar organizacije. Ona uključuju zakone i regulacije, poslovne ciljeve, kao i industrijske standarde. Poslovna pravila nisu direktno softverski zahtevi, ali pružaju informacije potrebne za razvoj sistema koji treba da prati ta pravila.

Poslovna pravila se **primenjuju** tako što oblikuju kako poslovni procesi funkcionišu i kako se informacioni sistemi razvijaju i koriste. Poslovna pravila se moraju primeniti kroz različite softverske aplikacije i procese, moraju biti pravilno dokumentovana i upravljana i ne smeju ući u sukob jedan s drugim.

49. Objasnite UML sekvencijalni dijagram.

**UML sekvencijalni dijagram** je vrsta vizualnog alata za prikazivanje interakcija između različitih aktera i entiteta unutar sistema tokom vremena.

Vreme teče odozgo nadole, tako da su gornji događaji na dijagramu oni koji se dešavaju prvi. Dijagram uključuje aktere (obično ljudske korisnike ili eksterne sisteme) i entitete (obično objekti unutar sistema). Predstavljeni su vertikalnom linijom, poznatom kao "životna linija".

Horizontalne strelice predstavljaju poruke ili interakcije između aktera i entiteta. Pravougaonici duž životnih linija označavaju period kada entitet obavlja neku aktivnost. Ovi pravougaonici, poznati kao "aktivacione šipke", pokazuju kada i koliko dugo se odvija određena akcija.

.

50. Kako se pišu zahtevi iz perspektive korisnika, a kako iz perspektive sistema?

Kada se zahtevi pišu iz **perspektive korisnika**, fokus je na tome šta korisnik treba ili želi da uradi sa sistemom. Opisuju se korisničke interakcije sa sistemom, npr. "Korisnik treba da dobije potvrdu o uspešnoj kupovini nakon obavljanja transakcije."

Kada se zahtevi pišu iz **perspektive sistema**, fokus je na funkcionalnostima koje sistem treba da pruži. Ovo uključuje tehničke aspekte sistema, kao što su procesiranje podataka, odgovori sistema i interne operacije. Opisuje se kako sistem interno funkcioniše, npr. "Nakon primanja zahteva za pretragu, sistem treba da vrši pretragu u bazi podataka i prikaže rezultate."

51. Objasnite zahteve za podatke i zahteve za spoljne interfejse.

**Zahtevi za podatke** se odnose na to kako sistem koristi podatke. To uključuje kako sistem prikuplja, obrađuje, čuva i izbacuje podatke. Ideja je da se jasno definiše koje vrste podataka sistem će koristiti, kako će oni biti strukturirani i kako će biti upravljani tokom vremena.

**Zahtevi za spoljne interfejse** se fokusiraju na opisivanje kako sistem interaguje sa drugim sistemima, uređajima ili korisnicima. Može se raditi o interakcijama sa drugim softverskim aplikacijama, hardverskim uređajima, ili korisničkim interfejsima.

**10, 11, 12. NEDELJA**

**==============**

52. Šta su atributi kvaliteta? Kako se klasifikuju?

**Atributi kvaliteta** su karakteristike proizvoda ili sistema koje određuju njegov kvalitet. Oni su bitni za razvoj i održavanje softverskih proizvoda, jer omogućavaju timovima da se fokusiraju na ključne aspekte koji su najvažniji za uspeh projekta. Atributi se klasifikuju na spoljašnje i unutrašnje atribute kvaliteta.

**Spoljašni atributi** su vidljivi korisnicima tokom izvršenja softvera. Oni su direktno povezani sa korisničkim iskustvom i zadovoljstvom. Primeri uključuju performanse, pouzdanost, upotrebljivost, i sigurnost.

**Unutrašnji atributi kvaliteta su** više usmereni na razvoj i održavanje softvera, i nisu direktno vidljivi korisnicima. Oni uključuju aspekte poput proverljivosti, izmenljivosti sistema i ponovne upotrebe komponenata.

53. Na šta se odnosi integritet, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Integritet** se odnosi na sposobnost sistema da očuva tačnost i doslednost podataka tokom obrade i skladištenja. Ovaj atribut je ključan za osiguranje da su podaci zaštićeni od slučajnih grešaka, oštećenja, neovlašćenih izmena, gubitaka ili napada.

Primer: *INT-1 – Potvrda sigurosne kopije:* “Sistem će, nakon pravljenja sigurnosne kopije neke datoteke, uporediti kopiju sa originalnom datotekom.”

54. Kakvi su prototipovi za bacanje?

**Prototipovi za bacanje** su privremeni modeli ili verzije softverskog sistema koji se koriste u ranim fazama razvoja. Cilj ovih prototipova je da brzo i efikasno istraže ideje, koncepte, dizajn, funkcionalnosti i korisničke interakcije. Čak nije ni potrebno dizajnirati softver, mogu se koristiti papiri sa interfejsovima gde inženjer glumi ulogu sistema, menjajući “ekrane” kako korisnik interaguje sa sistemom.

Prototipovi za bacanje su posebno korisni u situacijama kada postoje nesigurnosti, dvosmislenosti, ili nekompletnosti u zahtevima projekta.

55. Zbog čega se definišu prioriteti zahteva?

Dok bi idealno softver trebao da sadrži sve attribute kvaliteta, u praksi uvek postoji ograničenje resursa, bilo da se radi o vremenu, budžetu, ili radnoj snazi. Postavljanje **prioriteta** omogućava timu da se fokusira na najvažnije aspekte projekta koji će najviše doprineti vrednosti proizvoda i zadovoljstvu korisnika.

Kada su zahtevi rangirani po važnosti, tim može efikasnije da upravlja svojim vremenom i resursima, prvo se baveći onim delovima projekta koji imaju najveći uticaj na krajnji proizvod.

56. Kakvi su evolucijski prototipovi?

**Evolucijski prototip** je dizajniran da se putem niza iteracija razvije u konačni proizvod. Svaka iteracija prototipa dodaje nove funkcionalnosti ili poboljšava postojeće, postepeno se približavajući krajnjem cilju.

Iako nisu funkcionalni kao krajnji proizvod, evolucijski prototipovi su korisni i mogu vršiti realne zadatke. Korisnici testiraju prototip i daju povratne informacije koje oblikuju dalji razvoj, što osigurava da proizvod ostane usklađen sa očekivanjima korisnika.

57. Kako se procenjuje vrednost prototipa i koji su rizici pri pravljenju jednog?

**Procena vrednosti** prototipa uključuje testiranje upotrebljivosti i interakcije korisnika sa prototipom. Pritom se posebna pažnja posvećuje načinu na koji korisnici intuitivno koriste prototip, uključujući njihove reakcije, izraze lica, i poteškoće sa kojima se suočavaju.

Prilikom pravljenja prototipa, **rizici** se mogu odnositi na niz izazova i potencijalnih problema. Jedan je da korisnik misli da je prototip bliži finalnom proizvodu nego što zaista jeste. Ili da se korisnici fokusiraju previše na detalje korisničkog interfejsa, zanemarujući osnovne funkcionalnosti sistema. Takođe, performanse prototipa mogu stvoriti pogrešne pretpostavke o performansama konačnog proizvoda.

58. Šta su tehnike postavljanja prioriteta zahteva? Navedite neke.

Tehnike postavljanja prioriteta zahteva uključuju različite metode koje pomažu u rangiranju zahteva prema njihovoj važnosti za projekat. Neke su:

1. **Binarno Odlučivanje:** Zahtev je prioritet ili nije prioriter.
2. **Poređenje u Parovima:** Svaki zahtev se upoređuje sa svakim drugim u parovima da bi im se odredio relativni prioritet.
3. **Trostepena Skala:** Zahtevi su grupisani u tri kategorije - visok, srednji i nizak prioritet.
4. **MoSCoW Metod:** Zahtevi se razdvajaju u četiri kategorije: Must have (Mora), Should have (Treba), Could have (Može) i Won't have (Neće).
5. **100 Dolara Metoda:** Članovi tima dobijaju 100 imaginarnih dolara koje dodeljuju različitim zahtevima. Što se više dodeli nekom specifičnom zahtevu, to je veći prioritet.
6. **Prioritizacija Zasnovana na Vrednosti, Troškovima i Riziku:** Ova analitička metoda ocenjuje zahteve na osnovu njihove vrednosti za kupce, troškova implementacije i povezanih rizika. Cilj je identifikovati zahteve koji pružaju najveću vrednost uz najmanje troškove i rizike.

59. Na šta se odnosi dostupnost, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Dostupnost** se odnosi na sposobnost sistema da bude operativan i dostupan za korišćenje kada je to potrebno. Visoka dostupnost je ključna za sisteme koji moraju biti konstantno online.

Primer: AVA-1. Online prodavnica garantuje 99.9% dostupnost, osiguravajući da kupci mogu pristupiti i obaviti kupovinu u bilo koje vreme.

60. Na šta se odnosi lakoća instalacije, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Lakoća instalacije** se odnosi na jednostavnost sa kojom se softver može instalirati i konfigurisati. Cilj je smanjiti tehničke barijere i potrebu za stručnim znanjem.

Primer: INS-1. Softver za grafički dizajn dizajniran je tako da korisnici mogu završiti instalaciju u tri jednostavna koraka, bez potrebe za dodatnim konfiguracijama.

61. Na šta se odnosi interopabilnost, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Interoperabilnost** se odnosi na sposobnost sistema da se integriše i komunicira sa drugim sistemima, razmenjujući podatke i funkcionalnosti.

Primer: INTOP-1. Aplikacija za upravljanje projektima omogućava uvoz podataka iz različitih vanjskih kalendara, poboljšavajući koordinaciju tima.

62. Na šta se odnosi performansa, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Performansa** se odnosi na brzinu isa kojom sistem može obraditi zadatke. Visoke performanse su ključne za korisničko iskustvo.

Primer: PER-1. Veb aplikacija za strimovanje video sadržaja omogućava brzo učitavanje i reprodukciju videa bez kašnjenja, čak i na sporijim internet konekcijama.

63. Na šta se odnosi pouzdanost, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Pouzdanost** je mera sposobnosti sistema da pouzdano funkcioniše tokom određenog vremenskog perioda.

Primer: REL-1. Kritični sistem za upravljanje letom u avijaciji funkcioniše bez grešaka tokom dugih letova, obezbeđujući pouzdane informacije pilotima.

64. Na šta se odnosi robusnost, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

**Robusnost** se odnosi na sposobnost sistema da se nosi sa greškama ili neočekivanim uslovima bez prekida rada.

Primer: ROB-1. Finansijski softver održava operacije uprkos greškama u unosu podataka, sprečavajući prekide u transakcijama.

65. Na šta se odnosi sigurnost a šta bezbednost, kao atributi kvaliteta? Dajte po jedan primer ovih zahteva.

**Sigurnost** je zaštita sistema od neovlašćenih pristupa i cyber napada.

Primer: SEC-1. Korporativni email sistem koristi napredne algoritme za enkripciju i višefaktorsku autentifikaciju, štiteći osetljive informacije od neovlašćenog pristupa.

**Bezbednost** podrazumeva zaštitu korisnika i podataka od štetnih posledica grešaka u softveru.

Primer: SAF-1. Automobilski softver za autonomnu vožnju detektuje i sprečava potencijalne kolizije, štiteći putnike i pešake.

66. Na šta se odosi interopabilnost, kao atribut kvaliteta? Dajte jedan primer ovog zahteva.

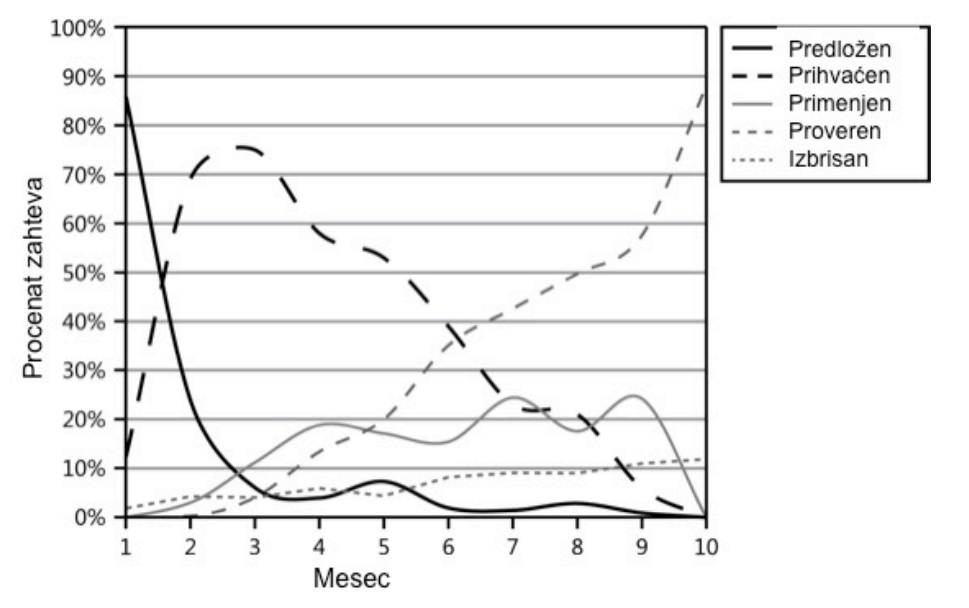
**Upotrebljivost:** Odnosi se na lakoću korišćenja i razumevanja sistema od strane korisnika. Primer: USE-1. Mobilna aplikacija za naručivanje hrane ima jednostavan i intuitivan interfejs koji omogućava korisnicima brzo naručivanje hrane sa nekoliko dodira.

**13, 14, 15. NEDELJA**

**==============**

**UPDATE QUESTION ENUMERATION**

67. Kako se sprovodi praćenje statusa zahteva tokom vremena?



Broj zahteva se smanjuje kada se funkcionalnost ukloni sa osnovne linije a povecava kako se opseg izdanja (planirana verzija softvera) menja.

Ovo je vizuelni prikaz pomocu koga se nadgleda status skupa zahteva tokom hipotetickog desetomesecnog projekta. Svaki zahtev ima vrednost statusa na kraju svakog meseca. Linija pokazuje kako se vremenom priblizavamo verifikaciji svih odobrenih zahteva. Projekat je zavrsen kada svi zahtevi imaju status verifikovan, izbrisan ili odlozen.

68. Objasnite problem "puzećih" promena okvira projekta.

Kada radimo na nekom projektu neizbezno je da ce doci do promene u zahtevima. Novi zahtevi koji naidju mogu doneti skroz novu funkcionalnost ili znacajne modifikacije nakon sto je set zahteva za projekat vec utvrdjen. Dodavanje novih zahteva je legitiman, neizbezan pa cak u nekim trenucima i povoljan proces.

Medjutim stalne promene u projektu bez prilagodjavanja resursa, ciljeva ili rasporeda moze biti stetno po sam projekat.

Pravi problem nije u tome sto se zahtevi menjaju, nego sto kasne promene mogu imati veliki uticaj na vec gotove funckionalnosti. Ako se svaki novi zahtev odobri, akterima u projektu se moze ciniti da ce razvoj projekta trajati zauvek, a mozda i hoce.

Za resavanje ovog problema neophodno je proceniti svaki predlozen zahtev. Isto je odlicno i angazovati samog kupca kako bi se smanjili zahtevi koji se previde. Koriscenje prototipova kao i inkementalni razvoj softvera dosta pomazu u specifikaciji zahteva i izbegavanju puzecih promena.

69. Kako upravljati promenama u zahtevima? Koje atribute treba da ima zahtev za promenu?

Kompanija koja se bavi razvojem projekta mora da osigura da:

* Da se predlozene promene zahteva promisljeno procene pre nego sto se obavezu
* Odgovarajuci pojedinci donose informisane poslovne odluke o zahtevanim promenama
* Aktivnost promene postaje vidljiva pogodjenim zainteresovanim stranama
* Odobrene promene se saopstavaju svim zahvacenim stranama

Promena u zahtevima je neizbezno da se desi. Prakticno je nemoguce definisati sve zahteve na pocetku projekta. Svet se menja kako napreduje razvoj projekta, donose se novi zakoni, propisi sa kojima se mora uskladiti.

**Atributi zahteva** se mogu predstaviti kao objekt koji cini kontekst i pozadinu zahteva.Preko atributa se zahtevi mogu lako pretraziti.

Svaki zahtev se moze zamisliti kao objekat sa svojim svojstvima pomocu koga se on identifikuje i razlikuje od drugih zahteva. Vrednosti atributa se unose u neku tabelu, dokument, bazu podataka ili nesto slicno gde vodimo evidenciju o zahtevima, a najbolje bi bilo cuvati ih u neki alat za upravljanje zahtevima.

Neki od atributa zahteva mogu biti:

* Datum kreiranja
* Trennutni broj verzije zahteva
* Autor
* Prioritet
* Status

70. Čemu služe formalne recenzije zahteva? Zbog čega se ističe inspekcijski proces, kao jedna vrsta formalne recenzje?

Najbolja utvrdjena vrsta formalne recenzije se naziva inspekcija. Formalne recenzije sluze da daju izvestaj koji identifikuje ispitivani materijal, recenzente i ocenu tima koji je imao zadatak da validira zahteve projekta. Cilj formalne recenzije je rezime utvrdjenih nedostataka i problema koji su se pojavili tokom pregleda zahteva.

Inspekcija je jedna od najkvalitetnijih tehnika za utvrdjivanje kvaliteta softvera. Moze se ustedeti veliko vreme ovom metodom, cak i 10 sati za svaki ulozeni sat. Detaljna inspekcija zahteva je zamoran i dugotrajan proces, ali se svaki sekund ulozen u inspekciju isplati.

71. Navedite i objasnite korake procedure za praćenje zahteva.

Preporucuju se sledeci koraci za primenu trazenja zahteva (preko tragova) u projektu:

* - **Obuka tima**: Edukujte tim i rukovodstvo o praćenju zahteva, ciljevima, mestima čuvanja podataka i tehnikama definisanja veza. Obezbedite angažovanje svih učesnika.
* - **Izbor veza:** Odaberite vezu koju želite definisati, ali ne pokušavajte sve odjednom.
* **- Izbor matrice:** Odaberite vrstu matrice sledljivosti i mehanizam čuvanja podataka (tekstualni dokument, Excel, alat za upravljanje zahtevima).
* - **Definisanje delova proizvoda:** Identifikujte ključne funkcije ili delove proizvoda za praćenje informacija o sledljivosti.
* **- Odgovorne osobe:** Postavite pojedince koji će dostavljati informacije o vezama i koordinatora za praćenje.
* - **Ažuriranje procedura:** Promenite razvojne procedure da podstaknete ažuriranje veza nakon primene zahteva ili promena.
* - **Označavanje:** Definišite konvencije označavanja za jedinstvene identifikatore elemenata sistema.
* - **Kontinuirano prikupljanje:** Podstičite tim da prikuplja informacije o tragovima tokom razvoja, ne samo na kraju projekta.
* - **Provera informacija:** Redovno proveravajte informacije o tragovima kako biste osigurali njihovu tačnost i aktuelnost.

72. Šta je kontrolna lista neispravnosti i čemu služi?

Kontrolne liste neispravnosti su alati koji pomažu recenzentima da sistematično pregledaju dokumente ili proizvode kako bi identifikovali uobičajene greške ili nedostatke. Ove liste služe kao smernice koje usmeravaju pažnju recenzenata na ključna područja ili tačke fokusa tokom pregleda.

73. Objasnite aktivnosti validacije i verifikacije zahteva.

**Verifikacija zahteva:**

Verifikacija zahteva je proces potvrđivanja da su napisani zahtevi ispravni, tačni i u saglasnosti sa definisanim standardima. Ova aktivnost ima za cilj osiguranje da je dokumentacija zahteva pravilno napisana i da zadovoljava određene standarde kvaliteta.

Ključne aktivnosti u verifikaciji zahteva uključuju:

* Proučavanje i Pregled Dokumenata:

Recenzenti pregledavaju dokumentaciju zahteva kako bi identifikovali potencijalne greške, nedostatke ili nejasnoće.

Proverava se konzistentnost, preciznost i punoća napisanih zahteva.

* Primena Standarda:

Proverava se da li su zahtevi napisani u skladu sa standardima, smernicama ili metodologijama koje je tim odabrao.

Ocenjuje se da li su zahtevi izraženi na odgovarajući način i da li su ispunili kriterijume za kvalitet.

* Provera Sledljivosti:

Utvrđuje se da li se svaki zahtev može pratiti i povezati sa izvorom zahteva, obično poslovnim ciljevima ili drugim izvorima.

* Analiza Osiguranja Kvaliteta:

Vrši se analiza da bi se osiguralo da dokumentacija zahteva zadovoljava kriterijume kvaliteta postavljene za dokumente u razvojnom procesu.

**Validacija zahteva:**

Validacija zahteva je proces koji se fokusira na ocenu da li napisani zahtevi zaista ispunjavaju potrebe i ciljeve poslovnog okruženja. Ova aktivnost ima za cilj osiguravanje da su postavljeni zahtevi relevantni i adekvatni za ostvarivanje poslovnih ciljeva.

Ključne aktivnosti u validaciji zahteva uključuju:

* Komunikacija sa Stakeholderima:

Aktivno se komunicira sa relevantnim akterima kako bi se razjasnili zahtevi i potrebe.

Proverava se da li su zahtevi u skladu sa očekivanjima i prioritetima stakeholdersa.

* Provera Sposobnosti Ispunjenja Potreba:

Utvrđuje se sposobnost zahteva da zadovolji stvarne potrebe korisnika i poslovnih zahteva.

Analizira se kako će ispunjenje svakog zahteva doprineti postizanju poslovnih ciljeva.

* Provera Sposobnosti Izvedbe:

Razmatra se da li je ostvarljivo implementirati i ispitati svaki zahtev u praksi.

Analizira se tehnička izvodljivost i realnost implementacije.

* Proučavanje Uticaja na Poslovanje:

Procenjuje se kako će svaki zahtev uticati na poslovne procese i da li će doneti vrednost organizaciji.

* Iterativni Pregledi:

Iterativno se pregledavaju i ažuriraju zahtevi kako bi se osiguralo da i dalje odgovaraju dinamičnom poslovnom okruženju.

74. Šta je sledljivost zahteva?

**Sledljivost zahteva** (ili treacability) predstavlja sposobnost praćenja životnog ciklusa nekog zahteva od njegovog nastanka pa sve do implementacije i kasnijih promena. Ova karakteristika omogućava projektantima, razvojnim timovima, i drugim relevantnim akterima da prate kako se svaki pojedinačni zahtev razvija i utiče na ostale delove sistema tokom vremena. Sledljivost čini da svaki korak u procesu razvoja može biti pratljiv unazad ili unapred, omogućavajući efikasno upravljanje promenama i održavanje jasne veze između zahteva i drugih elemenata projekta.

Kako bi zahtevi bili sledljivi, neophodno je:

* **Jedinstveno Označavanje Zahteva**: Svaki zahtev treba da ima jedinstvenu i trajnu oznaku koja ga identifikuje tokom celog projekta. Ova oznaka olakšava referenciranje zahteva i omogućava praćenje.
* **Logička Veza sa Drugim Elementima**: Zahtevi treba da imaju jasno definisane logičke veze sa drugim elementima projekta, kao što su dizajn, testiranje, implementacija, ili promene. Ovo omogućava transparentnost u pogledu toga kako svaki zahtev doprinosi celokupnom projektu.
* **Kratki i Jednostavni Zahtevi**: Zahtevi treba da budu izraženi na jasan i jednostavan način kako bi bili "sitnozrnati" (fine-grained). Ovo olakšava razumevanje svakog zahteva i smanjuje potrebu za tumačenjem složenih paragrafa.
* **Praćenje Promena**: Sledljivost takođe podrazumeva praćenje promena u zahtevima tokom vremena. Ažuriranje statusa, dodavanje novih informacija ili izmena treba da bude dokumentovano i pratioce informacije u vezi sa zahtevom.

66. Zašto upravljati promenama u zahtevima?

Upravljanje zahtevima je ključno zbog nekoliko razloga koji doprinose uspehu projekta:

* **Jasno Razumevanje Ciljeva Projekta**:

Upravljanje zahtevima omogućava definisanje i dokumentovanje jasnih ciljeva projekta kroz zahteve. To pomaže svim akterima, uključujući članove tima, klijente i druge zainteresovane strane, da razumeju šta se od njih očekuje da postignu.

* **Prevencija Gubitka Resursa**:

Ako se zahtevi ne upravljaju efikasno, postoji rizik od gubitka vrednih resursa, uključujući vreme, novac i napore. Efikasno upravljanje zahtevima pomaže u sprečavanju potrošnje resursa na nejasne, nepotrebne ili loše definisane zahteve.

* **Smanjenje Rizika Od Nepotrebnih Promena**:

Jasan proces upravljanja zahtevima pomaže identifikaciji i odobravanju promena u zahtevima. To smanjuje rizik od nepotrebnih promena tokom razvoja projekta, što može dovesti do nepotrebnih troškova i povećanja kompleksnosti.

* **Održavanje Sledljivosti:**

Upravljanje zahtevima doprinosi održavanju sledljivosti, tj. sposobnosti praćenja svakog zahteva od njegovog nastanka do implementacije. Ovo olakšava praćenje promena, analizu uticaja i održavanje transparentnosti tokom celog razvojnog procesa.

* **Smanjenje Jaza U Očekivanjima**:

Jasno definisani i dokumentovani zahtevi pomažu u smanjenju jaza u očekivanjima između članova tima, klijenata i drugih zainteresovanih strana. Svi su informisani o trenutnom stanju zahteva, što smanjuje nesporazume i konflikte.

* **Povećanje Efikasnosti Razvojnog Procesa**:

Upravljanje zahtevima povećava efikasnost razvojnog procesa. Definisanje, dokumentovanje i praćenje zahteva omogućava bolje planiranje, upravljanje resursima i praćenje napretka.

* **Povećanje Zadovoljstva Klijenata**:

Jasno definisani i zadovoljavajući zahtevi doprinose zadovoljstvu klijenata. Upravljanje zahtevima omogućava izgradnju proizvoda koji odgovara stvarnim potrebama i očekivanjima korisnika.

* **Održavanje Fokusiranosti na Poslovne Ciljeve**:

Upravljanje zahtevima pomaže održavanju fokusiranosti na poslovne ciljeve projekta. Osigurava da svi radovi i napore usmeravaju ka postizanju ciljeva, čime se povećava šansa za uspeh projekta.

67. Da li je praćenje zahteva sprovodijivo? Da li je potrebno? Argumentujte odgovore.

Praćenje zahteva može biti izazovan proces, ali je često nužan korak u razvoju softverskih sistema, posebno u složenim projektima gde je potrebno održavati sledljivost od zahteva do krajnjeg rešenja. Evo nekoliko argumenata koji podržavaju potrebu i izvedivost praćenja zahteva:

* **Osiguranje Kvaliteta:** Praćenje zahteva omogućava osiguranje kvaliteta softvera tako što se prati da li su svi zahtevi implementirani i da li softver zadovoljava očekivanja korisnika. To pomaže u sprečavanju propusta, poboljšava pouzdanost i smanjuje rizik od grešaka u kasnim fazama razvoja.
* **Sledljivost i Transparency:** Praćenje zahteva obezbeđuje sledljivost od početnih korisničkih potreba do konačnog rešenja. Ovo omogućava transparentnost u procesu razvoja, olakšava pronalaženje i rešavanje problema, te omogućava praćenje promena tokom vremena.
* **Zadovoljenje Regulatornih Zahteva:** U nekim industrijama, poput avioindustrije ili medicinskih uređaja, regulatorni zahtevi mogu zahtevati sledljivost od zahteva do projektnog rešenja. Ignorisanje ovih zahteva može dovesti do pravnih problema i nespojivosti sa standardima.
* **Upravljanje Rizicima:** Praćenje zahteva pomaže u upravljanju rizicima identifikacijom i praćenjem ključnih tačaka u razvoju softvera. Ovo omogućava pravovremeno reagovanje na eventualne probleme i promene u zahtevima.
* **Efikasnost Tima:** Dok je implementacija praćenja zahteva može biti resursno zahtevna, dugoročno može dovesti do efikasnijeg rada tima. Jasno definisani zahtevi olakšavaju komunikaciju među članovima tima i smanjuju nesporazume.
* **Korisničko Zadovoljstvo:** Pažljivo praćenje zahteva može dovesti do boljeg zadovoljenja korisničkih potreba i očekivanja, što doprinosi boljem prihvatanju softvera od strane korisnika.

68. Kako se sprovodi upravljanje zahtevima u projektima agilnog razoja softvera?

Agilni projekti se prilagođavaju promenama kroz serije razvojnih iteracija. Akteri dogovaraju prioritete za implementaciju priča u svakoj iteraciji, a nove priče dodate tokom iteracija imaju prioritet. Ako se pojave nove priče, mogu zameniti one sa nižim prioritetom, ali tim može zadržati originalni raspored isporuke ako to želi. Cilj je uvek raditi na pričama sa najvišim prioritetom kako bismo brže isporučili maksimalnu vrednost kupcima.

69. Kako drugi akteri u razvoju zahteva mogu doprineti razvoju zahteva?

Ako ste poslovni analitičar ili rukovodilac projekata, važno je da jasno komunicirate očekivanja sa akterima kako biste osigurali uspešan razvoj proizvoda. Evo ključnih tačaka:

* **Prihvatanje i testiranje sistema:** Očekujte od aktera da precizno definišu zahteve korisnika i funkcionalne zahteve. Jasna definicija očekivanog ponašanja softvera pomaže testerima u izvršavanju testova prihvatanja.
* **Konstruisanje:** Zahtevi su osnova za projektovanje i kodiranje softvera. Pregledajte projektno rešenje kako biste osigurali usklađenost sa zahtevima. Koristite jedinično testiranje kako biste proverili da li softver ispunjava specifikacije.
* **Korisnička dokumentacija:** Zahtevi proizvoda su ključni za pripremu korisničke dokumentacije. Akteri trebaju pružiti jasne i pravovremene zahteve kako bi podržali kvalitet korisničke dokumentacije.
* **Komunikacija sa grupom za razvoj:** Kao poslovni analitičar ili rukovodilac, definirajte jasne komunikacione interfejse između grupe za razvoj i ostalih funkcionalnih jedinica. Obezbedite usaglašenost interfejsa vezanih za specifikaciju zahteva sistema, dokumente o zahtevima tržišta i korisničke priče.

70. Kako se sprovode naknadne promene zahteva?

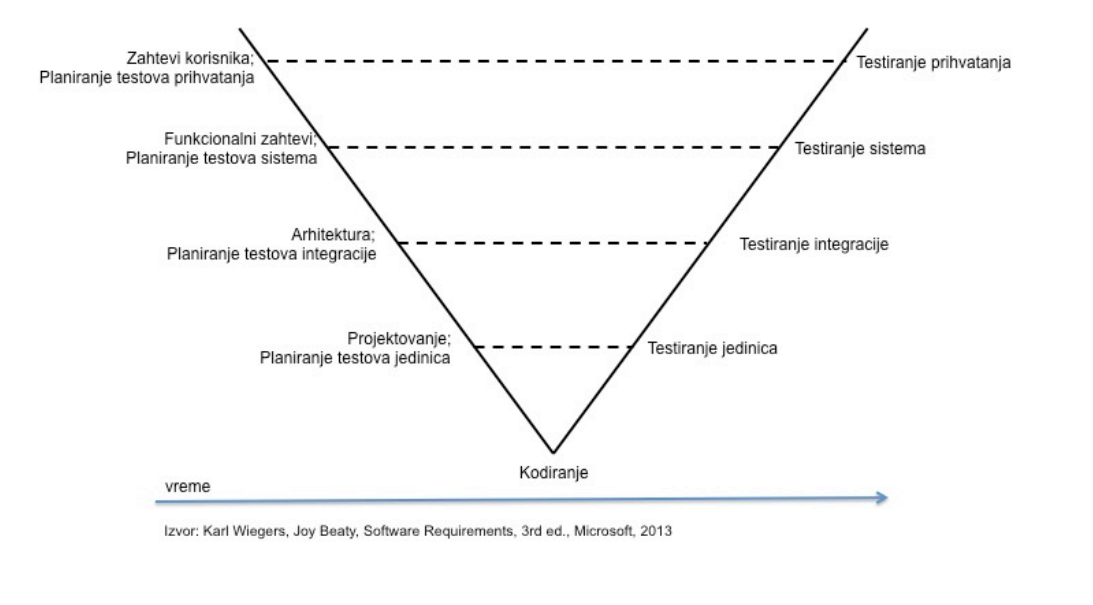
Smeštanje zahteva u alat za upravljanje zahtevima olakšava identifikaciju i upravljanje promenama u određenoj osnovnoj liniji. Razvojni tim koji prihvati predložene promene može naići na izazove u ispunjavanju postojećeg rasporeda i obaveza o kvalitetu. Rukovodilac projekta treba pregovarati o izmenama obaveza sa menadžerima, kupcima i drugim zainteresovanim stranama.

Projekat može prilagoditi nove ili izmenjene zahteve na različite načine, kao što su određivanje zahteva sa nižim prioritetom za kasnije iteracije, povećanje osoblja ili proširenje rasporeda isporuke. Nijedan pristup nije univerzalno ispravan, i izbor treba zavisiti od poslovnih ciljeva projekta i prioriteta koje su ključni akteri postavili na početku projekta.

Važno je prihvatiti realnost prilagođavanja očekivanja i obaveza kada je to potrebno. To je bolje nego pretpostavljati da će sve nove funkcije biti ugrađene bez prekoračenja budžeta, izgaranja članova tima ili kompromisa kvaliteta.

71. Opišite V model razvoja softvera.

V model razvoja softvera prikazuje ispitne aktivnosti koje se odvijaju paralelno sa odgovarajućim razvojnim aktivnostima. Na slici 1, vidimo da su testovi prihvatanja vezani za potrebe korisnika, testovi sistema se baziraju na funkcionalnim zahtevima, a testovi integracije proizlaze iz arhitekture sistema. Ovaj model se primenjuje bez obzira na to da li se testiraju aktivnosti razvoja softvera u kontekstu celog proizvoda, posebnog izdanja ili pojedinačnog priraštaja razvoja.



72. Kada i zašto se primenjuje praćenje i merenje uloženog rada na zahtevima?

Vreme koje se ulaže u aktivnosti vezane za zahteve predstavlja investiciju u uspeh projekta, a ne trošak. Vaš projektni plan treba sadržavati zadatke i resurse za aktivnosti upravljanja zahtevima kako biste osigurali efikasnost i kvalitet razvoja. Praćenje uloženog rada omogućava procenu da li je bilo dovoljno, adekvatno ili previše truda uloženo u razvoj i upravljanje zahtevima.

Merenje rada zahteva promenu kulture i disciplinu za evidentiranje dnevnih radnih aktivnosti, ali to pruža korisne uvide. Praćenje rada omogućava uvid u to kako članovi tima stvarno provode vreme u poređenju s njihovim očekivanjima. Ovo praćenje pomaže u pokazivanju da li tim sprovodi planirane aktivnosti vezane za zahteve.

Važno je razumeti da uloženi rad nije isto što i proteklo kalendarsko vreme. Prekidi i interakcije s drugima mogu uticati na trajanje zadataka. Praćenje vremena koje posvećuje poslovni analitičar (BA) i drugi učesnici u projektu pomaže u planiranju budućih projekata.

Broj sati provedenih u aktivnostima razvoja potreba treba zabeležiti, uključujući planiranje, održavanje radionica, pisanje specifikacija, kreiranje modela analize, ocenjivanje prototipova i pregled zahteva. Takođe, trud posvećen upravljanju zahtevima, kao što je konfigurisanje alata, podnošenje promena, procena predloženih promena i komunikacija sa zainteresovanim stranama, treba takođe pratiti. Ove informacije pružaju osećaj ukupnih troškova aktivnosti na projektu.

73. Ko su učesnici procesa inspekcije i koje su njihove uloge?

Učesnici inspekcije trebalo bi da predstavljaju četiri perspektive:

• **Autor radnog proizvoda i možda vršnjaci autora:** Poslovni analitičar koji je napisao dokument sa zahtevima pruža tu perspektivu. Ako možete, uključite još jednog iskusnog BA jer on zna koje vrste grešaka u pisanju zahteva da traži.

• **Ljudi koji su izvor informacija koji se ubacuju u predmet koji se pregledava:** Ovi učesnici mogu biti stvarni predstavnici korisnika ili autori prethodne specifikacije. U nedostatku specifikacije višeg nivoa apstrakcije, inspekcija mora da uključi predstavnike kupaca, kao što su šampioni proizvoda, kako bi se osiguralo da zahtevi pravilno i potpuno opisuju njihove potrebe.

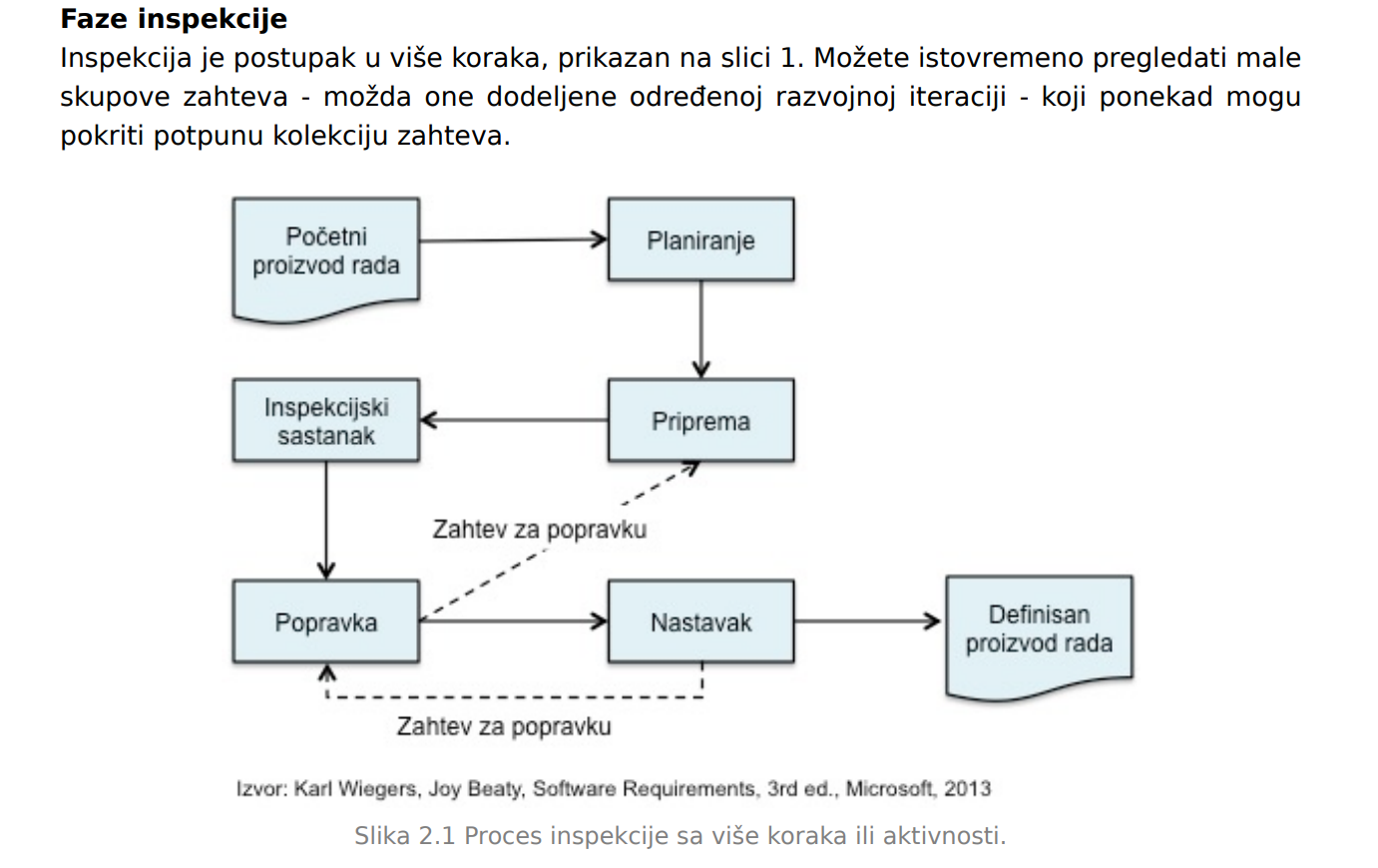
**Ljudi koji će obavljati posao na osnovu materijala koji se pregledava:** Za SRS možete uključiti programera, testera, menadžera projekata i pisca dokumentacije za korisnika, jer će otkriti različite vrste problema. Tester će najverovatnije ukazati na zahtev koji se ne može verifikovati; programer može uočiti zahteve koji su tehnički neizvodljivi.

• **Ljudi koji su odgovorni za povezivanje sistema na koje će uticati materijal koja se pregledava:** Ovi inspektori će potražiti probleme sa zahtevima spoljnog interfejsa. Takođe mogu da primete efekte talasanja, pri čemu promena zahteva u SRS koji se pregledava utiče na druge sisteme

74. Kako se planira proces Inspekclje? Kako se sprovodi proces Inspekcije?

Proces inspekcije se planira kroz sledeće korake:

* **Partnerstvo između autora i moderatora:** Autor i moderator zajedno planiraju inspekciju. Razmatraju ko će učestvovati, pripremaju materijale koje će inspektori dobiti pre sastanka, određuju ukupno vreme potrebno za pregled materijala i zakazuju inspekciju.
* **Razmatranje brzine pregleda:** Broj pregledanih stranica u satu ima značajan uticaj na efikasnost inspekcije. Praktične smernice sugerišu optimalni opseg od dve do četiri stranice na sat. Izbor brzine pregleda zavisi od faktora kao što su prethodni podaci o efikasnosti tima, količina teksta na stranicama, složenost zahteva, verovatnoća i uticaj neotkrivenih grešaka, kritičnost materijala i nivo iskustva autora zahteva.
* **Prilagođavanje brzine pregleda:** Brzinu pregleda treba prilagoditi na osnovu prethodnih iskustava tima, složenosti materijala, važnosti otkrivanja grešaka i nivoa iskustva autora. Optimalna brzina pregleda obično zavisi od ovih faktora.
* Pregled slika i grafova: Slika 2.2 pokazuje vezu između broja pronađenih defekata i brzine pregleda. Analizom ove veze, tim može odabrati odgovarajuću brzinu pregleda kako bi postigao optimalan balans između efikasnosti i otkrivanja grešaka.
* **Kontinuirano praćenje i prilagođavanje:** Praćenje efikasnosti inspekcije je ključno. Ako se primećuje da brzina pregleda utiče na kvalitet pregleda ili otkrivanje grešaka, tim treba kontinuirano prilagoditi brzinu pregleda kako bi optimizovao proces inspekcije tokom vremena.



75. Kako se mogu poboljšati procesi razvoja i upravljanja zahtevima?

?

76. Objasnite upotrebu matrice sledljivosti.

Matrica sledljivosti zahteva je alat koji prikazuje kako su različiti elementi softverskog sistema povezani unazad sa zahtevima. Ova matrica omogućava pratnju veza između funkcionalnih zahteva, slučajeva upotrebe i drugih elemenata sistema tokom celog razvoja softvera. Evo objašnjenja upotrebe matrice sledljivosti:

* **Povezivanje sa slučajevima upotrebe:** Matrica sledljivosti omogućava povezivanje svakog funkcionalnog zahteva unazad sa određenim slučajem upotrebe. Na taj način, možete jasno videti koji zahtevi podržavaju određene funkcionalnosti sistema i kako su oni vezani za očekivane scenarije korišćenja.
* **Povezivanje sa sistemskim elementima:** Matrica pruža pregled kako svaki funkcionalni zahtev utiče na elemente projektnog rešenja, kao što su arhitektonske komponente, kod i ispitivanje. Ovaj nivo detalja pomaže u tačnom lociranju softverskih elemenata koji su povezani sa specifičnim zahtevima.
* **Praćenje napretka:** Praznine u matrici ukazuju na neispunjene veze ili nezavršen rad. Praćenjem matrice sledljivosti možete lako identifikovati nedostatke, nepokrivenost ili elemente koji nisu usaglašeni sa zahtevima. To pomaže u vođenju evidencije o napretku projekta i identifikaciji oblasti koje zahtevaju dodatnu pažnju.
* **Precizno popunjavanje podataka:** Matrica se popunjava tek kada je rad na određenom elementu sistema završen, a ne prema planu. Ovaj pristup obezbeđuje tačne informacije o statusu veza između zahteva i sistema, jer se podaci popunjavaju samo kada je povezani rad stvarno završen.
* **Definisanje odnosa između zahteva:** Pored matrice sledljivosti zahteva, mogu se koristiti i druge matrice koje definišu različite odnose između parova zahteva. Ovi odnosi, poput "specificira/je određeno od", "zavisi od" ili "ograničava/ograničava od", pružaju dodatne informacije o međusobnim vezama između različitih vrsta zahteva.

77. Šta predstavlja osnovna lista zahteva? Kako se formira i za šta se koristi?

Osnovna lista zahteva predstavlja skup zahteva koji su odobreni i postavljeni pod upravljanje konfiguracijom, često definišući sadržaj određenog planiranog izdanja ili iteracije razvoja. Poslovni analitičar (BA) ima ključnu ulogu u upravljanju zahtevima. Evo pojednostavljenog objašnjenja procesa upravljanja zahtevima:

* **Postavljanje mehanizama skladištenja zahteva:** BA postavlja sisteme za skladištenje zahteva kako bi omogućio efikasno praćenje, organizaciju i upravljanje zahtevima tokom razvoja softvera.
* **Definisanje atributa zahteva:** BA definiše karakteristike i atribute svakog zahteva, uključujući poslovne zahteve, zahteve korisnika, funkcionalne i nefunkcionalne zahteve, rečnik podataka i modele analiza.
* **Koordinacija statusa zahteva:** BA prati status svakog zahteva, identifikujući da li su odobreni, u razvoju, čekaju reviziju ili su završeni. Ova koordinacija pomaže u održavanju preglednosti nad stanjem svakog zahteva.
* **Ažuriranje podataka u tragovima:** BA redovno ažurira podatke u matrici sledljivosti zahteva, prateći veze između zahteva, slučajeva upotrebe i sistema. Ovi ažurirani podaci omogućavaju jasnu evidenciju o vezi između zahteva i sistema tokom vremena.
* Nadgledanje i promena aktivnosti: Po potrebi, BA nadgleda i menja aktivnosti vezane za zahteve. To uključuje identifikaciju i rešavanje problema, kao i prilagodbe planova kako bi se odražavale promene u zahtevima ili okolnostima projekta.