Domaći zadatak broj 8
DZ08

Troškovi održavanja softvera

Tekst zadatka:

Opis aplikacije: Aplikacija ima funkciju da automatizuje zakazivanje termina korisnicima kod svog omiljenog frizera. Korisnik ima pregled termina kod svog omiljenog frizera gde vidi pregled slobodnih i zakazanih termina kod frizera. Korisnik dalje odabira jedan od slobodnih termina koji mu odgovara i šalje zahtev za zakazivanje termina. Frizer dobija obaveštenje o zahtevu korisnika i može da odgovori na njega – da prihvati ili da odbije pružanje usluge. U slučaju prihvatanja pružanja usluge termin prelazi iz stanja slobodan u zakazan, a korisnik dobija poruku o uspešno zakazanom terminu. U slučaju odbijanja zahteva za pružanje usluge, korisnik dobija poruku o nemogućnosti zakazivanja sa opcionim dodatnim tekstom o obrazloženju.

U okviru domaćeg zadatka potrebno je:

1. Predstaviti troškove održavanja softvera
2. Definisati moguće razloge za povećanje troškova održavanja softvera
3. Simulirati i opisati proces reinženjeringa softvera

Rešenje:

Predviđanje troškova održavanja softvera mora dati odgovore na dva pitanja:

1. Koliki će biti troškovi održavanja u toku životnog veka sistema?
2. Koliki će biti troškovi održavanja sistema sledeće godine?

Prilikom analize **Opisa problema** iz zadatog domaćeg zadatka, odredio sam Korisničke role u aplikaciji i Funkcionalne i Nefunkcionalne zahteve.

Korisničke role:

- Korisnik (kupac usluge)
- Frizer (davalac usluge)

Funkcionalni zahtevi – korisnička rola Korisnik (kupac usluge):

1. Korisnik treba da ima mogućnost pregleda termina,
2. Korisnik treba da ima mogućnost pregleda slobodnih termina,
3. Korisnik treba da ima mogućnost pregleda zakazanih termina,
4. Korisnik treba da ima mogućnost odabira jednog od slobodnih termina,
5. Korisnik treba da ima mogućnost slanja zahteva za zakazivanje termina korisniku Frizeru,
6. Korisnik treba da ima mogućnost da dobije poruku o uspešno zakazanom terminu,
7. Korisnik treba da ima mogućnost da dobije poruku o nemogućnosti zakazivanja termina,
8. Korisnik treba da ima mogućnost da vidi dodatni tekst o obrazloženju o nemogućnosti zakazivanja termina.

Funkcionalni zahtevi – korisnička rola Frizer:

1. Frizer treba da ima mogućnost da dobije obaveštenje o zahtevu korisnika za rezervacijom termina,
2. Frizer treba da ima mogućnost da odgovori na zahtev korisnika,
3. Frizer treba da ima mogućnost da prihvati obavezu pružnja usluge,
4. Frizer treba da ima mogućnost da odbije pružanje usluge ,
5. Frizer treba da ima mogućnost pisanja dodatnog teksta sa obrazloženjem o nemogućnosti zakazivanja termina,
6. Frizer treba da ima mogućnost slanja poruke sa dodatnim tekstom o obrazloženju o nemogućnosti zakazivanja termina.

Funkcionalni zahtevi – Sistem:

1. Sistem mora da ima bazu podataka.

Nefunkcionalni zahtevi:

- Zahtevi softverskog proizvoda:
    - Sistem treba da bude izgrađen tako da radi kao web aplikacija, mobilna Android aplikacija i mobilna IOS aplikacija.
    - Sistem treba da bude prilagođen radu na verzijama Android OS-a od verzije 5.0 Lollipop do verzije 8.0 Oreo
    - Sistem treba da bude prilagođen radu na verzijama IOS operativnog sistema od verzije 7 do verzije 11.
- Zahtevi organizacije:
    - Podaci koje korisnici unose, kao i podaci o izvršenim rezervacijama termina se čuvaju u bazi podataka.

Proces korektivnog održavanja softvera obuhvata 5 faza sa svojim osnovnim aktivnostima. Svaka faza održavanja softvera u domaćem zadatku je procenjena u radnim satima koje će članovi tima utrošiti na sprovođenje aktivnosti u okviru tih faza.

Osnovne aktivnosti koje treba obaviti tokom procesa korektivnog održavanja softvera su:

1) Identifikacija i analiza – obuhvata aktivnosti identifikovanja i analize grešaka u softveru i novih zahteva ukoliko oni postoje. Predviđeno vreme za izvođenje ove faze je 32 radna sata, odnosno 4 radna dana.
2) Dizajn – podrazumeva projektovanje novih modula koji se trebaju promeniti ili dodati. Predviđeno vreme za izvođenje ove faze je 40 radnih sati, tj. 5 radnih dana,
3) Implementacija – je faza u kojoj se popravljaju uočene greške i dodaju se nove funkcionalnosti. Predviđeno vreme trajanja ove faze je 80 radnih sati, tj. 10 dana.
4) Testiranje – je faza koja se vrši kako bi se nakon implementacije proverilo da izvršene promene nisu izazvale nove greške. Predviđeno vreme trajanja ove faze je 40 radnih sati, tj. 5 radnih dana.
5) Isporuka – klijentima se isporučuje nova verzija softvera sa ispravljenim greškama ili dodatnim funkcionalnostima. Predviđeno vreme trajanja ove faze je 2 radna časa.

Troškovi održavanja softvera

U softverskom proizvodu opisanom u domaćem zadatku, troškovi održavanja softvera bi mogli da obuhvate:

1. Prilagođavanje okruženju usled mogućeg upgrade-a operativnog sistema na kome radi aplikacija. Ovo je veoma čest slučaj kod mobilnih aplikacija (Android i IOS)
2. Zadovoljenje novih zahteva u slučaju dodavanja novih funkcionalnosti. Neke od novuh funkcionalnosti koje bi se mogle dodati u aplikaciju za zakazivanje termina kod frizera bi mogle biti: Izbor željenog tretmana kod fizera (na primer, posebne kategorije samo za šišanje, samo za pranje kose i feniranje, … i slično).
3. Popravljanje greški, eventualno ako se u toku eksploatacije sistema ispostavi da postoje neke greške u kodu ili strukturi sistema koji mogu da utiču na tačnost rada sistema.

Razlozi za povećanje troškova održavanja

Osnovni razlozi za povećanje troškova održavanja sistema predstavljenom u tekstu domaćeg zadatka, mogli bi da se nađu među razlozima kao što su:

1. Stabilnost tima – gde softver obično održava tim koji nije razvijao softver od početka. To može da bude problem jer timu za održavanje obično treba vremena da se upozna sa sistemom i kodom na kome je sistem napisan.
2. Loša praksa razvoja – gde je obično ugovor o održavanju sistema nezavistan od ugovora o razvoju sistema. Neretko firma koja je softver razvila, gleda da što više smanji cenu razvoja, pa time nekad prouzrokuje veće troškove održavanja.
3. Starost programa i njegova struktura – često se dešava da su programi pisani na zastarelom programskom jeziku ili su pravljeni sa starim i prevaziđenim strukturama.
4. Veštine zaposlenih – obično ljudi koji održavaju softver su manje stručni od ljudi koji su pravili softver, pa to može dovesti do povećanog vremena potrebnog za korekciju grešaka i održavanje softvera, što direktno povećava cenu održavanja.

Reinženjering softvera

Reinženjering softvera je promena strukture arhitekture sistema, ponovno pisanje kodova primenom modernih programskih jezika, promena strukture i vrednosti podataka sistema kao i promena strukture i vrednosti sistemskih promena. U reinženjeringu softvera funkcionalnost softvera se ne menja.

Proces reinženjeringa softvera opisanog u domaćem zadatku mogao bi da obuhvati:

1. Promenu jezika na kom je pisana Android mobilna aplikacija. Što bi značilo da se mobilna Android aplikacija prvobitno napisana u Java programskom jeziku sada napiše iz početka u Kotlin programskom jeziku.
2. Takođe, moguće je da se mobilna IOS aplikacija prvobitno napisana primenom Objective C programskog jezika sada ponovo napiše pomoću Swift programskog jezika.

Proces reinženjeringa starog softvera sadrži niz aktivnosti koje poboljšavaju performanse starog softvera. Aktivnosti u procesu reinženjeringa su:

- Prevođenje izvornog koda – na primer iz Java u Kotlin ili iz Objective C u Swift,
- Poboljšanje strukture programa – analizira se i menja struktura programa,
- Povratno inženjerstvo – analiza i dokumentovanje izvornog koda,
- Modularizacija programa – eliminiše se redudantnost koda iz sistema,
- Reinženjering podataka – u skladu sa promenama u sistemu, menjaju se i podaci, vrši se i promena šeme baze podataka, konverzija baze u novu strukturu.

MSc Branislav Manojlović
Novi Sad 19.05.2018.