

Specifikacija zahtjeva

Programsko inženjerstvo

Ciljevi poglavlja

- Predstaviti pojam softverskih zahtjeva, te procese koji se koriste za otkrivanje i dokumentiranje tih zahtjeva
- Razumjeti koncept:
 - korisničkih zahtjeva
 - zahtjeva sustava
 - funkcionalnih zahtjeva
 - ne-funkcionalnih zahtjeva
- Objasniti kako se zahtjevi organiziraju u dokument specifikacije zahtjeva
- Opisati proces prikupljanja, analize, validacije i upravljanja zahtjevima

Specifikacija zahtjeva

- engl. *Requirements engineering* ili *Software Requirements Specification* – SRS
- Proces utvrđivanja (pronalaženja, analize, dokumentiranja, provjere) usluga koje korisnik zahtjeva od sustava te ograničenja pod kojima taj sustav mora raditi se naziva **specifikacija zahtjeva**.



Što je zahtjev?

- Zahtjevi predstavljaju opis usluga koje pruža sustav i ograničenja u radu sustava.
- Problem je što se pojam zahtjev u softverskoj industriji ne koristi na konzistentan način, pa može biti:
 - vrlo apstraktna definicija osnovnih funkcionalnosti (npr. zahtjevi na osnovu kojih se raspisuje natječaj)
 - detaljna matematička specifikacija ili detaljan opis svih funkcionalnosti (npr. zahtjevi na osnovu kojih se razvija softver).



Tipovi zahtjeva

- **Korisnički zahtjevi** - pišu se za korisnika i opisani su na način da ih razumije korisnik.
 - Treba opisati korištenjem prirodnog jezika, tablica, dijagrama a trebaju biti takvi da ih razumiju svi korisnici.
- **Zahtjevi sustava** – namijenjeni su razvojnom timu. To je dokument sa detaljnim opisom svih funkcija, usluga i ograničenja sustava, u potpunosti strukturiran. Može se koristiti za potpisivanje ugovora.



MHC-PMS - primjer zahtjeva

Definicija korisničkih zahtjeva

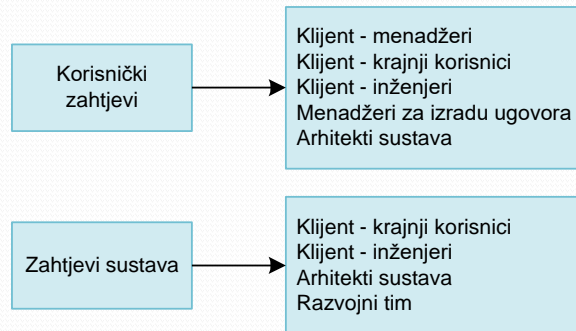
1. Sustav generira mjesečne izvještaje koji sadrže cijenu lijekova koje je prepisala svaka od klinika toga mjeseca.

Specifikacija zahtjeva sustava

- 1.1. Zadnji radni dan mjeseca generira se izvještaj cijene svih prepisanih lijekova, i klinike koja ih je prepisala.
- 1.2. Sustav automatski u 17:30 zadnji radni dan u mjesecu priprema izvještaj za ispis.
- 1.3. Treba napraviti izvještaj za svaku kliniku, a treba sadržavati naziv lijeka, ukupan broj izdanih recepata, broj prepisanih doza i ukupnu cijenu.
- 1.4. Za svaki lijek koji dolazi u različitim dozama, treba generirati odvojeni izvještaj za svaku od doza.



Tko čita koji tip zahtjeva?



Zainteresirane strane

- Zainteresirane strane (engl. stakeholders) su osobe ili organizacija na koju sustav utječe na bilo koji način.
- Vrste zainteresiranih strana
 - Krajnji korisnici
 - Menadžeri
 - „Vlasnici” sustava
 - Vanjske zainteresirane strane

MHC- PMS – primjer zainteresiranih strana

- Pacijenti čiji se podaci nalaze u sustavu.
- Doktor koji su odgovorni za te pacijente.
- Sestre koje se konzultiraju sa doktorima i također vode brigu o pacijentima.
- Sestra na prijemnom šalteru koja ugovara pacijentove dolaske.
- IT stručnjaci zaduženi za razvoj sustava.
- Medicinski etičar koji vodi brigu o tome da se ne naruši etički kodeks unutar sustava.
- Medicinski menadžeri koji dobivaju informacije iz sustava.
- Ostalo medicinsko osoblje.



Agilne metode i zahtjevi

- Većina agilnih metoda smatra da pisanje specifikacije zahtjeva gubljenje vremena jer se zahtjevi prebrzo mijenjaju i dokument vrlo brzo zastari.
- Metode poput XP-a koriste inkrementalno opisivanje zahtjeva u obliku scenarija.
- To je prihvatljivo za poslovne sustave, ali je problematično za sustave koji zahtijevaju opširnu analizu prije same isporuke (npr. kritični sustavi) ili sustave na kojima radi više timova.



Funkcionalni i ne-funkcionalni zahtjevi

Programsko inženjerstvo

11

Vrste zahtjeva

- Osnovne vrste zahtjeva za softverske sustave su:
 - **Funkcionalni zahtjevi**
 - Popis usluga koje pruža sustav, kako sustav treba reagirati na određene ulaze, te kako se sustav ponaša u određenoj situaciji, u nekim slučajevima u funkcionalnim zahtjevima može pisati i što sustav ne bi trebao raditi.
 - **Ne-funkcionalni zahtjevi**
 - Ograničenja na usluge ili funkcionalnosti sustava (vremenska ograničenja, ograničenja razvojnih procesa, standarda, performanse, ...)
 - **Zahtjevi domene**
 - Ograničenja iz područja domene koja utječu na rad sustava.



Programsko inženjerstvo

12

Funkcionalni zahtjevi

- Opisuju što bi sustav trebao raditi, tj. opisuju funkcionalnosti ili usluge sustava.
- Mogu biti vrlo apstraktne izjave koje opisuju što sustav treba raditi ili u detalje opisivati usluge koje će pružiti taj sustav ovisno o tome kome su namijenjeni.
- Ovisе o vrsti softvera, korisnicima i tipu sustava na kojem se softver koristi.



Programsko inženjerstvo

13

MHC-PMS - primjer funkcionalnih zahtjeva

1. Korisnik bi trebao biti u mogućnosti pretraživati popis dogovorenih pregleda za sve klinike.
2. Za svaku kliniku sustav treba generirati popis pacijenata koji su taj dan naručeni.
3. Svi zaposlenici klinike se trebaju identificirati preko 8-znamenkastog broja zaposlenika ukoliko žele koristiti sustav.



Programsko inženjerstvo

14

Nepreciznost zahtjeva

- Neprecizno ili nejasno napisani zahtjevi uzrokuju probleme - korisnici i razvojni tim mogu na različite načine interpretirati nejasne zahtjeve.
- Npr. u prethodnom primjeru MHC-PMS sustava moguće je krivo interpretirati zahtjev 1:
 - *Namjera korisnika* – traži se ime pacijenta u popisu dogovorenih pregleda za sve klinike.
 - *Interpretacija programera* – pretraživanje u listi pacijenata pojedine klinike, tj. korisnik odabere kliniku i tek onda traži.



Programsko inženjerstvo

15

Konzistentnost i potpunost zahtjeva

- U principu zahtjevi bi trebali biti:
 - **Potpuni** – Trebaju uključiti opis svih zahtjeva koje su naveli korisnici.
 - **Konzistentni** – Ne smije biti konflikta ili kontradikcija kod opisa sustava.
- U praksi je nemoguće stvoriti potpun i konzistentan dokument zahtjeva zbog:
 - grešaka i propusta koji se događaju kod pisanja dokumenta,
 - različiti ljudi unutar firme daju kontradiktorne informacije.



Programsko inženjerstvo

16

Ne-funkcionalni zahtjevi

- To nisu zahtjevi koji su direktno povezani s funkcionalnošću sustava, oni se odnose na:
 - svojstva i ograničenja sustava npr. pouzdanost, brzina izvršavanja, zahtjevi za pohranom, ...
 - ograničenja U/I uređaja ili prezentacije podataka na sučeljima u sustavu, ...
 - korištenje određenog CASE sustava, programskog jezika ili razvojne metode.
- Ne-funkcionalni zahtjevi mogu nekada biti važniji za prihvaćanje sustava od funkcionalnih. Ukoliko se ne ostvare sustav je beskoristan.



Programsko inženjerstvo

17

Implementacija ne-funk. zahtjeva

- Ne-funkcionalni zahtjevi najčešće utječu na cjelokupnu arhitekturu sustava, a ne na pojedinačne komponente.
 - Npr. ukoliko je važna brzina izvođenja, sustav treba organizirati tako da se minimizira komunikacija među komponentama.
- Jedan ne-funkcionalni zahtjev, poput sigurnosti, može generirati čitav niz povezanih funkcionalnih zahtjeva.



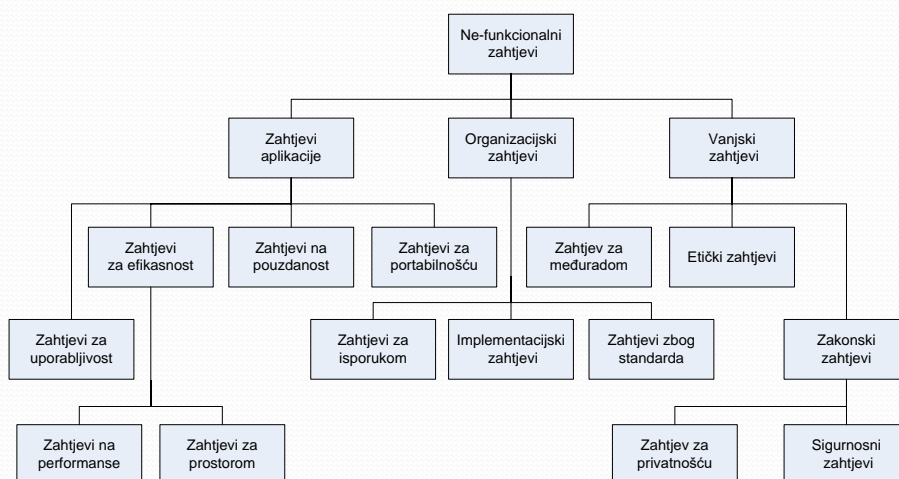
Programsko inženjerstvo

18

Klasifikacija ne-funkcionalnih zahtjeva

- **Zahtjevi aplikacije**
 - Zahtjev da se aplikacija ponaša na određeni način (npr. brzina izvođenja, pouzdanost, ...)
- **Organizacijski zahtjevi**
 - zahtjevi koji su posljedica organizacijske politike i procedura (npr. procesi koji se standardno koriste, implementacijski zahtjevi, ...)
- **Vanjski zahtjevi**
 - Zahtjevi koji se javljaju se kao posljedica faktora van samog sustava (npr. zakonski okviri, suradnja s drugim sustavima, etički zahtjevi, ...)

Vrste ne-funkcionalnih zahtjeva



MHC-PMS - primjer ne-funkcionalnih zahtjeva

- **Zahtjevi aplikacije**
 - Sustav mora biti dostupan svim klinikama tijekom normalnog radnog vremena. Vrijeme nedostupnosti sustava u tom periodu ne smije prijeći 5 sekundi.
- **Organizacijski zahtjevi**
 - Korisnici sustava moraju se identificirati korištenjem svoje zdravstvene knjižice.
- **Vanjski zahtjevi**
 - Sustav mora provesti odredbe o zaštiti privatnosti pacijenta kao što je navedeno u zakonu.



Programsko inženjerstvo

21

Ciljevi i zahtjevi

- Ne-funkcionalni zahtjevi mogu biti jako komplicirani za precizan opis, a neprecizni zahtjevi su teški za provjeru.
- Zbog čega se često izražavaju kroz **ciljeve sustava**
 - Cilj opisuje generalnu namjeru korisnika (npr. jednostavnost korištenja).
- Dokazivanje ne-funkcionalnih zahtjeva
 - Izjava koja koristi neku mjeru koja se može koristiti za objektivnu provjeru cilja (npr. prosječni korisnik nakon jednog sata treninga ne smije proizvoditi više od 5 grešaka dnevno).
- Ciljevi su korisni razvojnom timu jer oni opisuju namjere korisnika.



Programsko inženjerstvo

22

MHC-PMS – primjer ciljeva i zahtjeva

- ***Način na koji menadžeri izražavaju zahtjeve:***
 - Sustav treba biti jednostavan za korištenje i organiziran na takav način da se minimiziraju korisničke greške.
- ***Izražavanje zahtjeva na način da se mogu provjeriti:***
 - Medicinsko osoblje treba biti u stanju koristiti sustav nakon 4-satnog treninga. Nakon toga ne smiju generirati više od 2 greške po satu.



Mjerenje nefunkcionalnih zahtjeva

Svojstvo	Mjera
Brzina	obavljene transakcije/s korisnik/vrijeme odziva vrijeme osvježavanja ekrana
Veličina	MBajt-ima Broj ROM čipova
Jednostavnost korištenja	Vrijeme potrebno za trening Broj stranica za pomoć
Pouzdanost	Prosječno vrijeme rada bez greške Vjerojatnost da su usluge nedostupne Brzina događanja grešaka Dostupnost
Robusnost	Vrijeme potrebno da se sustav ponovno podigne nakon pada Postotak grešaka koje uzrokuju pad sustava Vjerojatnost grešaka na podacima u slučaju pada sustava
Portabilnost	Broj ciljanih sustava



Interakcija među zahtjevima

- Vrlo čest neki od ne-funkcionalnih zahtjeva su u konfliktu s drugim funkcionalnim ili ne-funkcionalnim zahtjevima.
- Npr. kod MHC-PMS sustava:
 - Zahtjev da se medicinsko osoblje prijavljuje u sustav pomoću identifikacijske kartice zahtjeva postojanje čitača kartica.
 - Postoji zahtjev da se medicinsko osoblje može prijaviti u sustav s mobilnih uređaja (nemaju na raspolaganju čitač kartica), što znači da trebaju postojati i alternativne metode prijave u sustav.
- Npr. kod nogometnog stadiona:
 - Potrebno je dobro osvjetljenje.
 - Traži se što manja potrošnja energije.
 - Koji je zahtjev važniji?



Zahtjevi domene

- Zahtjevi koje nameće aplikacijska domena sustava, a ne određene potrebe korisnika.
- Mogu biti novi funkcionalni zahtjevi, ograničenja na funkcionalne zahtjeve ili upute kako se provode određeni proračuni.
- Problem je što osobe koje pišu specifikaciju zahtjeva ne razumiju osnovne karakteristike domene za koju se aplikacija radi i ne mogu reći jesu li neki od zahtjeva domene izostavljeni ili u sukobu s drugim zahtjevima.



Specifikacija zahtjeva

Kraj

Programsko inženjerstvo

76