

INŽENJERSTVO ZAHTJEVA U OBLIKOVANJU PROGRAMSKE POTPORE

(*engl. requirements engineering - RE*)

Inženjerstvo zahtjeva je proces izrade specifikacije programskog produkta. To je prva generička aktivnost tijekom procesa programskog inženjerstva.

Ciljevi ove prezentacije:

- Uvesti koncepte korisničkih zahtjeva i zahtjeva sustava.
- Opisati funkcionalne i ne-funkcionalne zahtjeve.
- Objasniti kako se zahtjevi programske potpore organiziraju i opisuju u odgovarajućim dokumentima.

1

INŽENJERSTVO ZAHTJEVA (RE)

- To je **postupak** pronalaženja, analiziranja, dokumentiranja i provjere **zahtijevanih usluga** sustava, te ograničenja u uporabi.
- **Zahtjevi** sami za sebe su **opisi usluga sustava i ograničenja (specifikacija)** koja se generira tijekom procesa inženjerstva zahtjeva.
- Obzirom na **razinu detalja** razlikujemo:
 - **Specifikacija visoke razine** apstrakcije (obično u okviru ponude za izradu programskog produkta) = **korisnički zahtjevi**. Pišu se u **prirodnom jeziku i grafičkim dijagramima**. Moraju biti razumljivi ne-tehničkom osoblju.
 - Vrlo **detaljna specifikacija** (uobičajeno nakon prihvatanja ponude, a prije sklapanja ugovora) = **zahtjevi sustava**. Pišu se **strukturiranim prirodnim jezikom, posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom**.
 - **Specifikacija programske potpore** je najdetaljniji opis i objedinjuje korisničke i zahtjeve sustava.

2

TKO ČITA ZAHTJEVE (za koga se rade):

Korisnički zahtjevi

Klijenti (rukovoditelji – manageri)
 Krajnji korisnici sustava
 Klijenti (inženjeri)
 Rukovoditelji za pisanje ugovora
 Specijalisti za oblikovanje sustava (arhitekti)

Zahtjevi sustava

Krajnji korisnici sustava
 Klijenti (inženjeri)
 Specijalisti za oblikovanje sustava (arhitekti)
 Specijalisti za razvoj programske potpore

Specifikacija programske potpore

Klijenti (inženjeri) – možda ?
 Specijalisti za oblikovanje sustava (arhitekti)
 Specijalisti za razvoj programske potpore

3

Zahtjevi **obzirom na sadržaj**: FUNKCIONALNI, NEFUNKCIONALNI I ZAHTJEVI DOMENE

(odnose se kako **na korisničke tako i zahtjeve sustava**)

- Funkcionalni zahtjevi

Izjave o uslugama koje sustav mora pružati, kako će sustav reagirati na određeni ulazni poticaj, te kao bi se sustav trebao ponašati u određenim situacijama.

- Nefunkcionalni zahtjevi

Ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja, (ne)usvojeni standardi, ograničenja u procesu razvoja i oblikovanja itd.

- Zahtjevi domene primjene

Zahtjevi koji proizlaze iz domene primjene sustava kao i oni koji karakteriziraju tu domenu.

4

Primjer korisničkih funkcionalnih zahtjeva, sustav LIBSYS

(Hipotetski knjižničarski sustav koji pruža jedinstveno sučelje prema bazama članaka u različitim knjižnicama. Korisnik može pretraživati, spremati, ispisivati članke za osobnu potrebu.)

- Korisnik mora moći pretraživati početni skup baza ili podskup.
- Sustav mora sadržavati **odgovarajuće preglednike** koji omogućuju čitanje članaka u knjižnici.
- Svako narudžbi mora se alocirati jedinstveni identifikator (ORDER_ID) koji korisnik mora moći kopirati u svoj korisnički prostor.

5

Poteškoće u navedenom primjeru: Prirodni jezik

- Nedostatak jasnoće (preciznost nije lako postići bez detaljiziranog ali naporno čitljivog dokumenta).
- Miješaju se funkcionalni i nefunkcionalni zahtjevi.
- Nenamjerno objedinjavanje više zahtjeva u jednom.

Nejasno postavljene zahtjevi mogu biti različiti interpretirani od korisnika i razvojnih timova što dovodi do problema u procesu razvoja i kršenju ugovora.

Npr. u LIBSYS sustavu nejasno je : “odgovarajući preglednik”:

Intencija korisnika –	više preglednika posebne namjene za svaki tip dokumenta.
Intencija razvojnog tima –	samo preglednik teksta kao bitnog sadržaja dokumenta.

6

Primjer korisničkih zahtjeva:**Zahtjev na potpunu rešetku u grafičkom editoru:**

*Kako bi se olakšalo postavljanje entiteta na crtež, korisnik može preko upravljačkog panela **uključiti rešetku** u **centimetrima ili inčima**. Inicijalno je **rešetka isključena**. Rešetka se može **uključiti i isključiti** kao i **izmijeniti centimetre i inče** u svako vrijeme rada s editorom. Biti će osigurana opcija smanjivanja slike kako bi stala na zaslon, ali broj prikazanih crta rešetke će biti reducirane kako ne bi popunile manje slike.*

Miješaju se tri različita tipa zahtjeva:

- Konceptijski (potreba za rešetkom).
- Nefunkcionalni (mjerne jedinice rešetke).
- Nefunkcionalni ulazno izlazni zahtjevi (prebacivanje između tipova rešetki).

Zaključak: Prirodni jezik u specifikaciji korisničkih zahtjeva mora se koristiti vrlo pažljivo (Dom. zadaće i Lab. vježbe ?).⁷

KOMPLETNOST I KONZISTENCIJA ZAHTJEVA

- **Kompletni zahtjevi:**
Sadrže opise svih zahtijevanih mogućnosti.
- **Konzistentni zahtjevi:**
Ne smiju sadržavati konflikte ili kontradikcije u opisima zahtijevanih mogućnosti.
- U praksi je **nemoguće postići kompletan i konzistentan** dokument o zahtjevima.

Npr. svemirska letjelica:

Smanjivanje energije traži sklopove s malom potrošnjom. Sklopovi male potrošnje imaju manji faktor integracije pa ih treba više. Više sklopova rezultira u većoj težini koja iziskuje više pogonske energije.

NEFUNKCIONALNI ZAHTJEVI - klasifikacija

- Zahtjevi programskog produkta:

Zahtjevi koji specificiraju da se isporučeni produkt mora ponašati na osobit način (npr. vrijeme odziva).

- Organizacijski zahtjevi:

Zahtjevi koji su rezultat organizacijskih pravila i procedura (npr. uporaba propisanog standardnog procesa razvoja, DoD ADA).

- Vanjski zahtjevi:

Zahtjevi koji proizlaze izvan sustava i razvojnog procesa (međusobna operabilnost s drugim sustavima, legislativni zahtjevi isl.).

9

Nefunkcionalni zahtjevi – primjer LIBSYS

Zahtjevi programskog produkta:

Npr.: Korisničko sučelje LIBSYS sustava biti će implementirano kao jednostavni HTML bez uporabe okvira ili Java “appleta”.

Organizacijski zahtjevi:

Npr.: Proces razvoja sustava i isporučeni dokumenti moraju slijediti standard XYZCo-SP-STAN-95.

Vanjski zahtjevi:

Npr.: Sustav neće operatorima otkriti osobne informacije o klijentima (osim njihovog imena i referentnog broja).

10

ZAHTJEVI DOMENE PRIMJENE

Zahtjevi domene primjene mogu biti **novi funkcionalni** zahtjevi ili **ograničenja** na postojeće zahtjeve.

Npr. LIBSYS zahtjevi domene primjene:

Zbog restrikcija u pravima kopiranja neki dokumenti se po dolasku moraju odmah izbrisati.

Ovisno o zahtjevu korisnika dokumenti se mogu ispisati lokalno kako bi se ručno dostavili korisniku.

Problemi zahtjeva domene:

Razumljivost: programeri ne razumiju domenu primjene i traže detaljan opis zahtjeva.

Implicitnost: Specijalisti domene poznaju primjenu tako dobro da podrazumijevaju zahtjeve (koje tada eksplicitno ne određuju).

11

Detaljna specifikacija = ZAHTJEVI SUSTAVA

- Detaljnija specifikacija funkcija sustava, njegovih usluga i ograničenja nego zahtjevi korisnika.
- Uloga tih specifikacija je definiranje oblikovanja sustava.
- Mogu se uključiti u ugovor o isporuci sustava.
- Zahtjevi sustava mogu se definirati ili ilustrirati nekim od **modela** sustava.

ZAHTJEVI SUSTAVA I OBLIKOVANJE

U principu zahtjevi određuju **ŠTO** sustav mora raditi, a oblikovanje (dizajn) određuje **KAKO** će se to ostvariti.

U praksi su **zahtjevi i oblikovanje neodvojivi**.

- Arhitektura sustava strukturira zahtjeve.
- Sustav često mora raditi u sinergiji s drugim sustavima koji generiraju zahtjeve na oblikovanje.
- Uporaba specifičnog oblikovanja može biti zahtjev domene primjene.

12

IZRAŽAVANJE ZAHTEVA SUSTAVA

Strukturirani prirodni jezik

Definiranje standardnih formulara i obrazaca u kojima se izražavaju zahtjevi (definicije, ulazni podaci i izvori, prethodni i posljedični uvjeti, popratni efekti, ...). Prednost ovakve specifikacije je u zadržavanju izražajnosti prirodnog jezika, ali uz nametnutu izvjesnu uniformnost. Nedostatak je ograničena terminologija. U praksi **nema usvojene globalne standardizacije**.

- Jezik za opis oblikovanja (npr. SDL)

Poput programskog jezika, ali s više apstraktnih obilježja, definira se operacijski model sustava.

- Grafička notacija (npr. UML)

Grafički jezik proširen tekstom.

- Matematička specifikacija (FSM, logika isl.)

Notacija zasnovana na matematičkom konceptu. Najstrože definirana specifikacija. Kupci je ne vole jer je ne razumiju.

13

Primjer zahtjeva sustava strukturiranim prirodnim jezikom

Insulin Pump/Control Software/SRS/3.3.2

Function	Compute insulin dose: Safe sugar level
Description	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.
Inputs	Current sugar reading (r2), the previous two readings (r0 and r1)
Source	Current sugar reading from sensor. Other readings from memory. (otkuda ulazi)
Outputs	CompDose Š the dose in insulin to be delivered
Destination	Main control loop (kamo izlazi)
Action:	CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.
Requires	Two previous readings so that the rate of change of sugar level can be computed.
Pre-condition	The insulin reservoir contains at least the maximum allowed single dose of insulin.
Post-condition	r0 is replaced by r1 then r1 is replaced by r2
Side-effects	None

14

Izražavanje zahtjeva sustava u predmetu

“Oblikovanje programske podrške”

Strukturirani prirodni jezik

Koristit će se u zadavanju domaćih zadaća i laboratorija.

Jezici za opis oblikovanja (npr. SDL)

Neće se posebno razmatrati u ovom kolegiju.

Grafička notacija (UML)

Biti će detaljno prikazani u nastavku predavanja.

Matematička specifikacija

U okviru kolegija razmatrat će se **Kripke struktura**, **logika**, **vremenska logika**.

15

ZAHTJEVI SUSTAVA: SPECIFIKACIJA SUČELJA

Specifikacija sučelje prema korisniku i prema drugim sustavima. Postoje tri tipa sučelja:

- **Proceduralno sučelje** (skup usluga kroz sučelje – API: “Application Programming Interface” = primjensko programsko sučelje).
- **Strukture podataka** koje se izmjenjuju (najčešće opisane grafičkom notacijom, npr. Entity-Relation-Attribute, ERA model).
- **Predstavljanje podataka** (što znače pojedini bitovi u riječi).

Formalna notacija jednoznačno opisuje sučelje, ali traži specijalističko znanje

Npr.:

```
interface P rintSe rver {
    // defines an abstract printer server
    // requires:      interface P rinter, interface P rintDoc
    // provides: initialize, print, d isplayP rintQu eue, can celP rintJob , switchP rinter

    void initialize ( P rinter p ) ;
    void p rint ( P rinter p , P rintDoc d ) ;
    void d isplayP rintQu eue ( P rinter p ) ;
    void can celP rintJob ( P rinter p , P rintDoc d ) ;
    void s witchP rinter ( P rinter p1 , P rinter p2 , P rintDoc d ) ;
} //P rintSe rver
```

16

IEEE STANDARD:

ŠTO DOKUMENT ZAHTJEVA MORA SADRŽAVATI

1. Predgovor
2. Uvod
3. Rječnik pojmova
4. Definicije korisničkih zahtjeva
5. Specifikacija zahtjeva sustava
6. Arhitektura sustava
7. Modeli sustava
8. Evolucija sustava
9. Dodaci
10. Indeks

17

ZAKLJUČCI:

- Zahtjevi jednoznačno postavljaju što sustav treba raditi i definiraju ograničenja u implementaciji i radu sustava. Ne postoji usvojeni jedinstveni standard za izražavanje zahtjeva.
- **Korisnički zahtjevi** su izjave na višoj apstraktnoj razini što bi sustav trebao raditi. Pišu se u prirodnom jeziku, tablicama ili dijagramima.
- **Zahtjevi sustava** su detaljne specifikacije o funkcijama sustava. Izražavaju se **strukturiranim prirodnim jezikom**, specifičnim **jezikom za oblikovanje** (npr. SDL), **grafičkom notacijom** (npr. ERA, UML), i **matematičkom specifikacijom** (npr. vremenska logika).
- **Funkcionalni zahtjevi** (korisnički i zahtjevi sustava) definiraju usluge koje sustav osigurava.
- **Ne-funkcionalni zahtjevi** (korisnički i zahtjevi sustava) postavljaju ograničenja na sustav ili na proces oblikovanja sustava.
- **Dokument zahtjeva** programskog produkta je usklađen skup izjava o svim zahtjevima na sustav.
- **IEEE standard** je korisna početna točka za definiranje detaljiziranog specifičnog načina pisanja dokumenta zahtjeva.

18