TEHNIČKO VELEUČILIŠTE U ZAGREBU
POLYTECHNICUM ZAGRABIENSE

Objektno orijentirani razvoj programa

ISVU: 130938/19970

Dr. sc. Danko Ivošević, dipl. ing.
Predavač

Ak. god. 2021./2022.
Ljetni semestar

OBJEKTNO ORIJENTIRANI RAZVOJ PROGRAMA

INŽENJERSTVO ZAHTJEVA

„How To Engineer Software”

Why Should We Care?

- 18% of projects fail to deliver any usable software
- Of projects that do deliver, average
  - 42% late
  - 35% over budget
  - 25% under scope
  - Abundance of delivered defects
- 2019 US software budget ~$340 billion
  - ~$61 billion in cancellations
  - ~$72 billion in cost overruns
  - ~$41 billion in scope under-runs
  - → Funders expected to pay only ~$166 billion for functionality actually delivered!

See also:
“Top Five Challenges in Software Development”
Lunch & Learn

Reference: [Standish13]

[Izvor: Software Engineering 101 | Steve Tockey – YouTube]

„Surova stvarnost”

- *Standish* skupina →
- Izvješća *CHAOS* ≡ *Comprehensive Human Appraisal for Originating Software*
- https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/

„Surova stvarnost”

- https://qracorp.com/wp-content/uploads/2020/10/Leveraging-NLP-in-Requirements-Analysis.pdf (James Martin)

„How To Engineer Software”

Top Five Root Causes of Poor Performance

- Data supports
  - (1) Vague, ambiguous, incomplete requirements
  - (2) Inadequate project management
- Professional experience suggests
  - (3) Uncontrolled design, code complexity
  - (4) Over-dependence on testing
  - (5) “Self-documenting code” is myth

See also:
“Top Five Challenges
in Software Development”
Lunch & Learn

Izvor: How to Engineer Software, Part 1: Semantic Models | Steve Tockey – YouTube

Višedimenzijska klasifikacija zahtjeva

- Apstraktne izjave o usluzi koju bi programski sustav trebao ponuditi?

ili

- Detaljna formalna definicija funkcije programskog proizvoda?

Višedimenzijska klasifikacija zahtjeva

Primjer 4.1. Ako neka tvrtka želi ponuditi natječaj o izradi velikog, složenog programskog projekta na otvoreno tržište, tada ona nudi apstraktnu specifikaciju koja u najopćenitijim crtama opisuje funkciju konačnog programskog proizvoda. Tada se na natječaj javljaju ponuđači koji predlažu načine na koje će riješiti zadani problem. Nakon što pregleda prijedloge ponuđača, tvrtka odabire izvođača projekta. Zatim taj izvođač piše detaljniju specifikaciju sustava koju tvrtka vrednuje prije nego što se potpisuje ugovor i kreće u izradu projekta.

- ➔ I jedno i drugo smatra se specifikacijom zahtjeva.

Podjela zahtjeva prema razini detalja

(od manje prema većoj razini detalja)

- **korisnički zahtjevi** (engl. *user requirements*)
- **zahtjevi sustava** (engl. *system requirements*)
- **specifikacija programske potpore** (engl. *software specification*)

Podjela zahtjeva prema razini detalja

Korisnički zahtjevi

1. Programski sustav za ministarstvo zdravstva generirat će mjesečna izvješća za upravu u kojima će predočiti cijenu lijekova koji se propisuju za svaku kliniku tijekom tog mjeseca.

Zahtjevi sustava

1.1 Na zadnji radni dan u mjesecu, generirat će se sažetak o propisanim lijekovima, njihovoj cijeni i klinikama koje su ih propisale.

1.2 Sustav će automatski generirati izvješće spremno za ispis nakon 17:30 na zadnji radni dan u mjesecu.

1.3 Izvješće će biti napravljeno i za svaku kliniku posebno i popisat će pojedinačne nazive lijekova, ukupan broj recepata, broj propisanih doza i ukupnu cijenu propisanih lijekova.

1.4 Ako su lijekovi dostupni u različitim dozama (npr. 10 mg, 20 mg), zasebna izvješća će se napraviti za svaku postojeću dozu.

1.5 Pristup svim izvješćima o cijenama bit će ograničen na sve ovjerene korisnike koji se nalaze na kontrolnoj listi s razinom pristupa: "uprava".

Podjela zahtjeva prema razini detalja

- Dionici koji čitaju ili dorađuju zahtjeve:
  - Klijentski inženjeri
  - Klijentski rukovoditelji i menadžeri
  - Krajnji korisnici sustava
  - Rukovoditelji za pisanje ugovora
  - Specijalisti za oblikovanje sustava – arhitekti
  - Specijalisti za razvoj programske potpore
  - Stručnjaci iz domene primjene sustava

Zahtjevi prema razini detalja

Dionici / Vrsta zahtjeva	Korisnički zahtjevi	Zahtjevi sustava	Specifikacija programske potpore
Klijentski inženjeri			
Klijentski rukovoditelji i menadžeri			
Krajnji korisnici sustava			
Rukovoditelji za pisanje ugovora			
Specijalisti za oblikovanje sustava – arhitekti			
Specijalisti za razvoj programske potpore			
Stručnjaci iz domene primjene sustava			

Zahtjevi prema razini detalja

Dionici / Vrsta zahtjeva	Korisnički zahtjevi	Zahtjevi sustava	Specifikacija programske potpore
Klijentski inženjeri	×	×	×
Klijentski rukovoditelji i menadžeri	×		
Krajnji korisnici sustava	×	×	
Rukovoditelji za pisanje ugovora	×		
Specijalisti za oblikovanje sustava – arhitekti	×	×	×
Specijalisti za razvoj programske potpore		×	×
Stručnjaci iz domene primjene sustava	×		

Podjela zahtjeva prema sadržaju

- **funkcionalni zahtjevi** (engl. *functional requirements*)
- **nefunkcionalni ili ostali zahtjevi** (engl. *non-functional*, *other requirements*)
- **zahtjevi domene primjene** (engl. *domain requirements*)

Funkcionalni zahtjevi

- izjave o:
  - uslugama koje programski proizvod mora pružati,
  - kako će sustav reagirati na određeni ulazni poticaj,
  - kako bi se trebao ponašati u određenim situacijama.

- ➔ potpuni i dosljedni

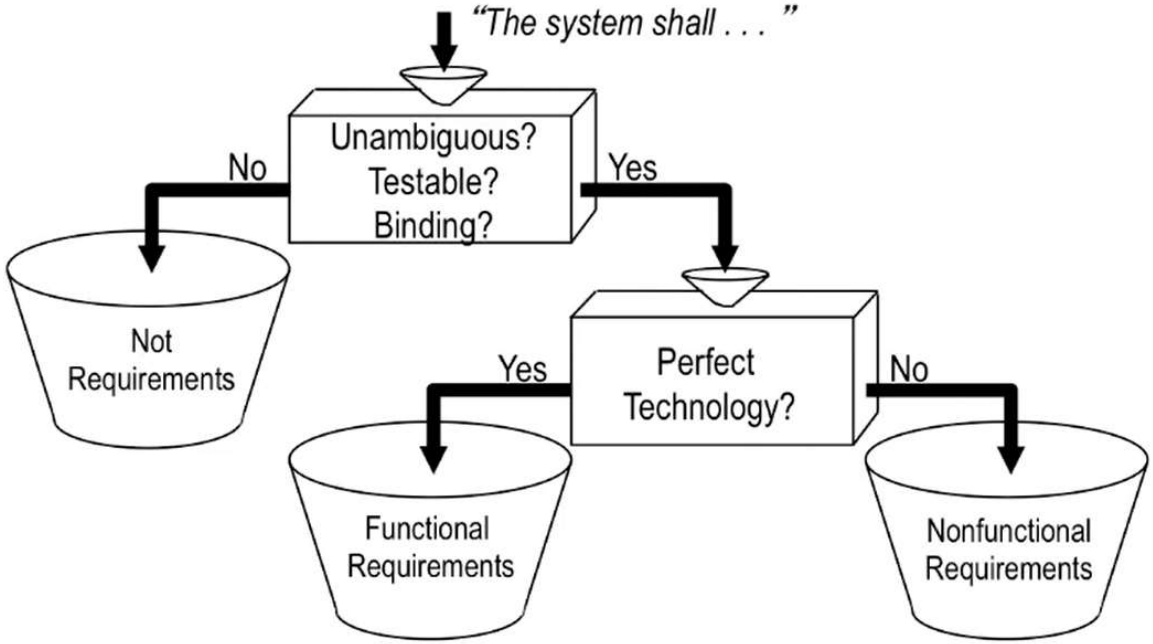
  („kompletni” i „konzistentni”)

Nefunkcionalni ili ostali zahtjevi

- ograničenja u uslugama i funkcijama programske potpore (kvantitativno, ako može):
  - **zahtjevi programske potpore** – npr. vrijeme odziva, podržani sustav, komunikacijski protokol i sl.
  - **organizacijski zahtjevi** – npr. uporaba propisanog normiranog procesa razvoja, korištenje uvijek točno određenog programskog jezika pri razvoju.
  - **vanjski zahtjevi** – npr. postizanje međusobne operabilnosti s drugim sustavima, zakonski zahtjevi i drugo.

(Ne)funkcionalni zahtjevi

Semantic Model is Technology-Free



Izvor: How to Engineer Software, Part 2: A Deeper Dive into Semantic Models | Steve Tockey – YouTube

Zahtjevi domene primjene

- pojavljuju se kasnije u procesu otkrivanja zahtjeva:
  - razumljivost vs. implicitnost
- mogu biti novi funkcionalni zahtjevi ili ograničenja na postojeće zahtjeve

Razlikovanje vrste zahtjeva

Primjer 4.3. Hipotetski sustav LIBSYS

Opis: Knjižničarski sustav koji pruža jedinstveno sučelje prema bazama članaka u različitim knjižnicama. Korisnik može pretraživati, spremati i ispisivati članke za osobne potrebe.

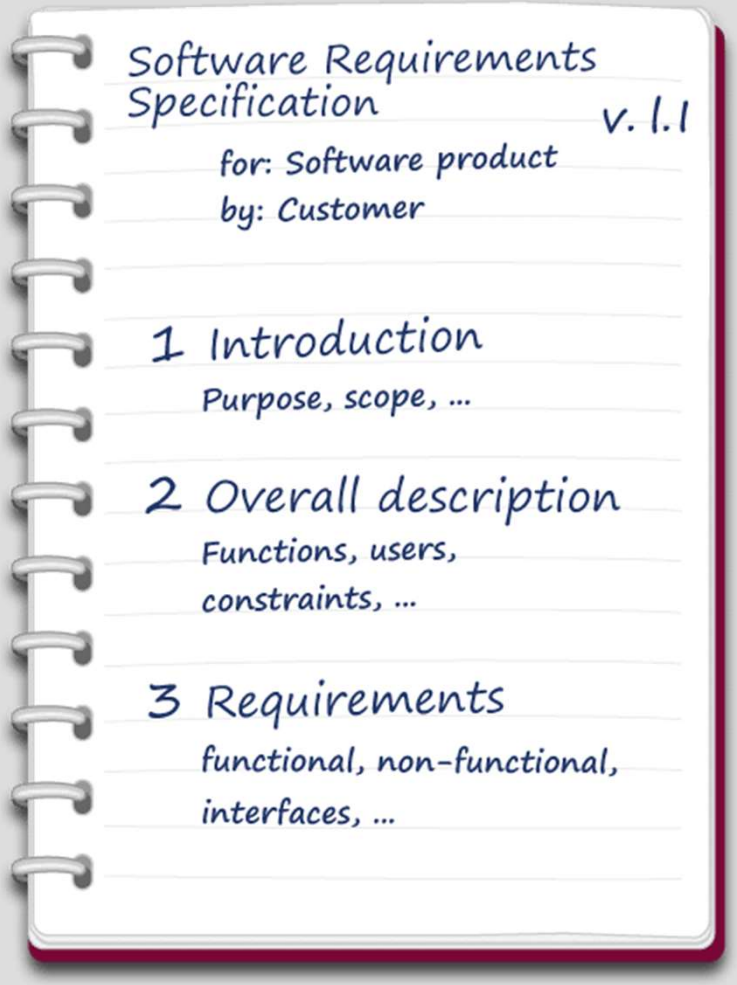
Zahtjevi: Korisnik mora moći pretraživati početni skup baza članaka ili njihov podskup. Sustav mora sadržavati odgovarajuće preglednike koji omogućuju čitanje članaka u knjižnici. Korisničko sučelje LIBSYS sustava treba biti implementirano kao jednostavni HTML bez uporabe okvira ili Java "appleta". Svakoj narudžbi mora se alocirati jedinstveni identifikator (ORDER_ID) koji korisnik mora moći kopirati u svoj korisnički prostor.

Načini specifikacije zahtjeva

- Rečenice prirodnog jezika
- Strukturirani prirodni jezik
- Specifični jezik za opis oblikovanja
- Grafička notacija
- Matematička specifikacija

Dokument zahtjeva

- Kakav treba biti?
- Što treba uključivati?



[Izvor: https://www.microtool.de/en/knowledge-base/what-is-a-software-requirements-specification]

Dokument zahtjeva

- Prema normi instituta IEEE (1998.):

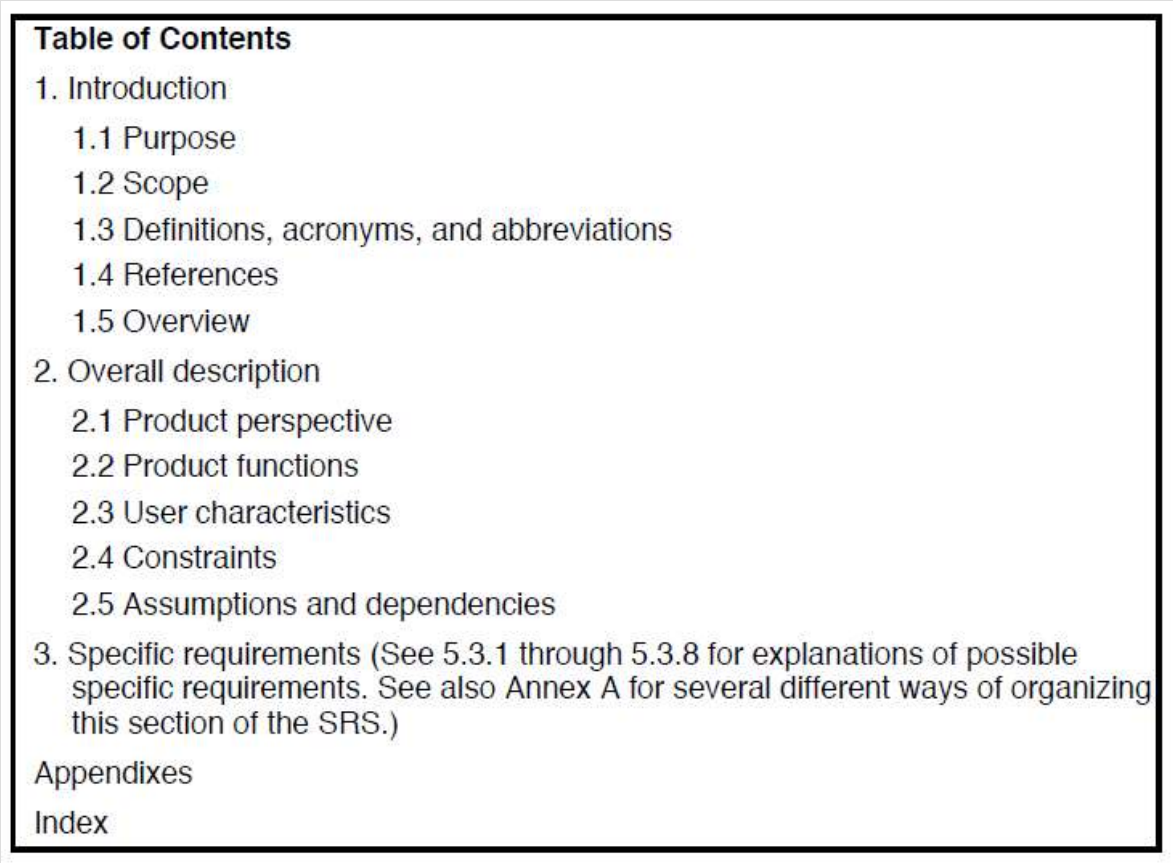


Table of Contents

[http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf]

Dokument zahtjeva

- I. Sommerville, Software Engineering, 9th ed., Addison-Wesley, Harlow, England, 2011.:
  1. Predgovor
  2. Uvod
  3. Rječnik pojmova
  4. Definicija korisničkih zahtjeva
  5. Specifikacija zahtjeva sustava
  6. Arhitektura sustava
  7. Modeli sustava
  8. Evolucija sustava
  9. Dodaci
  10. Indeks (pojmova, dijagrama, funkcija).

OBJEKTNO ORIJENTIRANI RAZVOJ PROGRAMA

LITERATURA

REFERENCE I LITERATURA

- A. Jović, N. Frid, D. Ivošević: Procesi programskog inženjerstva, 3. izdanje, Sveučilište u Zagrebu, FER ZEMRIS, 2019.
- I. Sommerville, Software engineering, 8th ed., Addison Wesley, 2007.
- G. Overgaard, B. Selic, C. Bock, OMG, 2000.
- S. Tockey: How to Engineer Software, Wiley-IEEE Computer Society Press, 2019.