

Projekt izrade aplikacije

Specifikacija zahtjeva

2019/20.02

➤ Projekt

- Projekt je vremenski određeno nastojanje da se proizvede jedinstven proizvod, usluga ili rezultat. [PMBOK – Project Management Book of Knowledge, PMI]
- Projekt je niz jedinstvenih, složenih i povezanih aktivnosti koje imaju određeni cilj i koji se mora postići u zadanom vremenskom roku, u okviru zadanog proračuna i u skladu sa specifikacijama. [Wisocky, Beck and Crane]

➤ Vremenska određenost

- Svaki projekt mora imati jasno određen početak i kraj. Projekti mogu biti kratki ili trajati godinama, ali će svakako završiti.
- Projekt završava u trenutku kada postane jasno da su ciljevi projekta dostignuti ili kada se zaključi da ciljevi projekta ne mogu ili neće biti dostignuti.

➤ Jedinstvenost

- Projekt se odnosi na rad na nečemu što prije nije postojalo i što se razlikuje od rezultata nastalih sličnim projektima.

Upravljanje projektom

3

- Upravljanje, rukovođenje projektom (Project management)
 - Upravljanje projektima je primjena znanja, vještina, alata i tehnika u projektnim aktivnostima da bi se ispunili projektni zahtjevi. [PMI]
- Upravljanje projektom uključuje
 - Planiranje
 - Utvrđivanje zahtjeva
 - Postavljanje jasnih i ostvarivih ciljeva
 - Uravnoteženje zahtjeva na kvalitetu, doseg, vrijeme i trošak,
 - Prilagodbu interesima i očekivanjima zainteresiranih strana – dionika (eng. Stakeholders)
 - Organiziranje
 - Formiranje projektnog tima
 - Raspoređivanje obaveza
 - Tko što i kada treba napraviti
 - Usmjeravanje
 - Nadgledanje, omogućavanje izvršenja
 - Kontroliranje
 - Provjera učinka i rezultata

Uloge na projektu

4

- Korisnik, Korisnik usluga, Klijent (User, Customer, Client)
 - osoba ili grupa, naručilatelj ili krajnji korisnik
- Sponzor projekta (project sponsor)
 - Osoba ili grupa koja osigurava (financijske) resurse za projekt
- Voditelj projekta (project manager)
 - Osoba imenovana kako bi ostvarila ciljeve projekta
- Resursi projekta
 - Osobe, oprema, usluge, materijal, budžet ili druga sredstva.
- Projektna ekipa
 - Svi članovi ekipe, uključujući upravljačke, a u nekim slučajevima i sponzora
 - voditelj – upravljanje projektom
 - sistem analitičar – određivanje potreba, specifikacija zahtjeva i dizajna
 - projektant/arhitekt – uspostava osnovne arhitekture
 - razvojnici (developer, builder) – kodiranje, testiranje
 - administrator baza podataka – administriranje DBMS
 - sistem inženjer / sistem administrator – administriranje OS i mreže

Dokumentiranje projekta

5

- Povelja projekta (Project Charter)
 - Dokument kojim pokretač projekta ili sponzor odobrava projekt i ovlašćuje voditelja za primjenu organizacijskih resursa u provedbi projekta.
 - Plan projekta = Plan upravljanja softverskim projektom
 - IEEE Standard for Software Project Management Plans 1058-1998
 - dokument koji opisuje sveukupnu organizaciju projekta
 - Primjeri 📁 Prilozi\ul> - PlanProjekta.dot
 - Firma-PlanProjekta.doc
- Plan može sadržavati raspored
 - općenito: SoftwareDevelopment.mpp, PlanRazvoja.mpp
 - konkretno: Firma-PlanProjekta.mpp

1. Introduction

- 1.1 Project Overview
- 1.2 Project Deliverables
- 1.3 Evolution of the Software Project Management Plan
- 1.4 Reference Materials
- 1.5 Definitions and Acronyms

2. Project Organization

- 2.1 Process Model
- 2.2 Organizational Structure
- 2.3 Organizational Boundaries and Interfaces
- 2.4 Project Responsibilities

3. Managerial Process

- 3.1 Management Objectives and Priorities
- 3.2 Assumptions, Dependencies, and Constraints
- 3.3 Risk Management
- 3.4 Monitoring and Controlling Mechanisms
- 3.5 Staffing Plan

4. Technical Process

- 4.1 Methods, Tools, and Techniques
- 4.2 Software Documentation
- 4.3 Project Support Functions

5. Work Packages, Schedule, and Budget

- 5.1 Work Packages
- 5.2 Dependencies
- 5.3 Resource Requirements
- 5.4 Budget and Resource Allocation
- 5.5 Schedule

6. Additional Components

7. Index

8. Appendices

Izrada plana projekta

6

➤ Koraci izrade plana projekta

- Izrada liste zadataka
- Izrada hijerarhije zadataka (work breakdown structure)
- Procjena trajanja zadataka
- Izrada ovisnosti među zadacima
- Dodjela resursa

	Zadatak	Trajanje (u danima)	Prethodnici	Naziv resursa
1	Doseg	7		
2	Određivanje dosega	5		Voditelj projekta
3	Određivanje sponzorstva	1,5	2	Sponzor projekta; Voditelj projekta
4	Određivanje resursa	2	2	Voditelj projekta
5	Dovršetak dosega	0	3;4	
6	Analiza / Softverski zahtjevi	16		
7	Analiza potreba	5	5	Sistem analitičar
8	Prikupljanje informacija	6	7	Sistem analitičar
9	Prijedlog izvedbe sustava	5	8	Sistem analitičar
10	Dovršetak analize	0	9	
11	Dizajn	5		
12	Razvoj funkcionalnih specifikacija	5	10	Sistem analitičar; Projektant
13	Izrada baze podataka	5	10	Administrator baze podataka
14	Dovršetak dizajna	0	13;12	
15	Razvoj	35		
16	Izrada formi korisničkog sučelja	4	12	Razvojniki
17	Izrada funkcija za pohranu podataka	12	14	Razvojniki
18	Izrada funkcija za ispis izvještaja	15	14	Razvojniki
19	Izrada izvještaja	10	18	Razvojniki
20	Izrada funkcija izračuna	12	16	Razvojniki
21	Razvojno testiranje (debugiranje)	10	16;17;18;19;20	Razvojniki
22	Dovršetak razvoja	0	21	
23	Testiranje	12		
24	Izrada testova programskog cjelina prema specifikacijama proizvoda	5	14	Tester
25	Izrada plana integracijskog testiranja prema specifikacijama proizvoda	5	14	Tester
26	Testiranje komponenti prema specifikacijama proizvoda	7	24	Tester
27	Provjera integracije modula	7	25	Tester
28	Dovršetak testiranja	0	26;27	
29	Dokumentacija	5		
30	Razvoj specifikacija i sustava pomoći	5	22	Razvojniki
31	Dovršetak dokumentacije	0	30	Razvojniki
32	Uvođenje sustava i poduka korisnika	5		
33	Ugradnja programske potpore	5	28	Razvojniki; Sistem administrator
34	Dovršetak uvođenja sustava i poduke korisnika	0	33	

➤ Zadaci – osnovni gradbeni elementi svakog projekta

- predstavljaju posao koji se mora obaviti da bi se postigao cilj projekta
- opisuju tijek događaja, trajanja i zahtjeva za resursima na projektu
- primitivni zadaci
 - zadaci koji se dekompozicijom ne mogu podijeliti na jednostavnije zadatke
- skupni zadaci (summary tasks)
 - zbrajaju trajanje i troškove primitivnih zadataka
 - trajanje, datum te izračunate vrijednosti se automatski izvode iz skupa primitivnih zadataka
- prekretnice ili miljokazi (milestones)
 - ključni događaj ili krajnji rok odnosno cilj koji treba postići
 - trajanja 0
 - služe za provjeru stupnja dovršenosti drugih zadataka
 - pomak ključnog događaja ima za posljedicu vremenski preraspored

Izrada hijerarhije zadataka

8

- Faza - grupa povezanih zadataka koji se odnose na fazu projekta
 - Zbirni zadaci se odnose na faze
- WBS (work breakdown structure)
 - hijerarhijska lista faza, zadataka i prekretnica
 - osnova za pregledni raspored projekta
- Dva su pristupa razvoju zadataka i faza:
 - Planiranje s vrha prema dolje (Top-down)
 - pristup od općeg prema specifičnom
 - identificira glavne faze i rezultate projekta prije dodavanja zadataka potrebnih za završetak tih faza
 - složeni projekti mogu imati nekoliko slojeva faza
 - Planiranje s dna prema gore (Bottom-up)
 - pristup od specifičnog prema općem
 - identificira što više zadataka najnižeg sloja prije grupiranja u faze

Procjena trajanja zadataka

9

➤ Trajanje zadatka

➤ očekivana količina vremena za završetak zadatka

➤ minute (m), sati (h), dani (d), tjedni (w), mjeseci (mo)

	Zadatak	Trajanje (u danima)	Prethodnici	Naziv resursa
1	Doseg	7		
2	Određivanje dosega	5		Voditelj projekta
3	Određivanje sponzorstva	1,5	2	Sponzor projekta; Voditelj projekta
4	Određivanje resursa	2	2	Voditelj projekta
5	Dovršetak dosega	0	3;4	
6	Analiza/Softverski zahtjevi	16		
7	Analiza potreba	5	5	Sistem analitičar
8	Prikupljanje informacija	6	7	Sistem analitičar
9	Prijedlog izvedbe sustava	5	8	Sistem analitičar
10	Dovršetak analize	0	9	
11	Dizajn	5		
12	Razvoj funkcionalnih specifikacija	5	10	Sistem analitičar; Projektant
13	Izrada baze podataka	5	10	Administrator baze podataka
14	Dovršetak dizajna	0	13;12	

Međuzavisnost zadataka

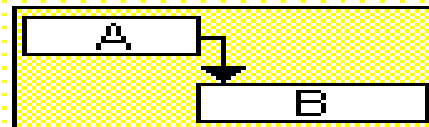
10

- Projekt može zahtijevati da zadaci budu napravljeni u određenom redoslijedu
 - Niz – iza jednog slijedi drugi zadatak
 - Zavisnost – sljedbenik (successor) može biti izvršen ako je dovršen prethodnik (predecessor)
 - Bilo koji zadatak može biti prethodnik jednom ili više sljedbenika

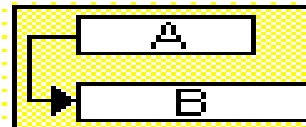
- Odnosi između zadataka:

- Finish-to-start (FS) - završni datum prethodnika jest početni sljedbenika
- Start-to-start (SS) - početni datum prethodnika utvrđuje početni sljedbenika
- Finish-to-finish (FF) - završni datum prethodnika utvrđuje završni sljedbenika
- Start-to-finish (SF) - početni datum prethodnika utvrđuje završni sljedbenika

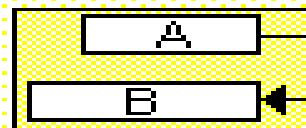
Finish-to-start (FS)



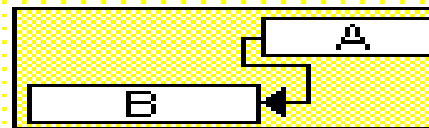
Start-to-start (SS)



Finish-to-finish (FF)



Start-to-finish (SF)

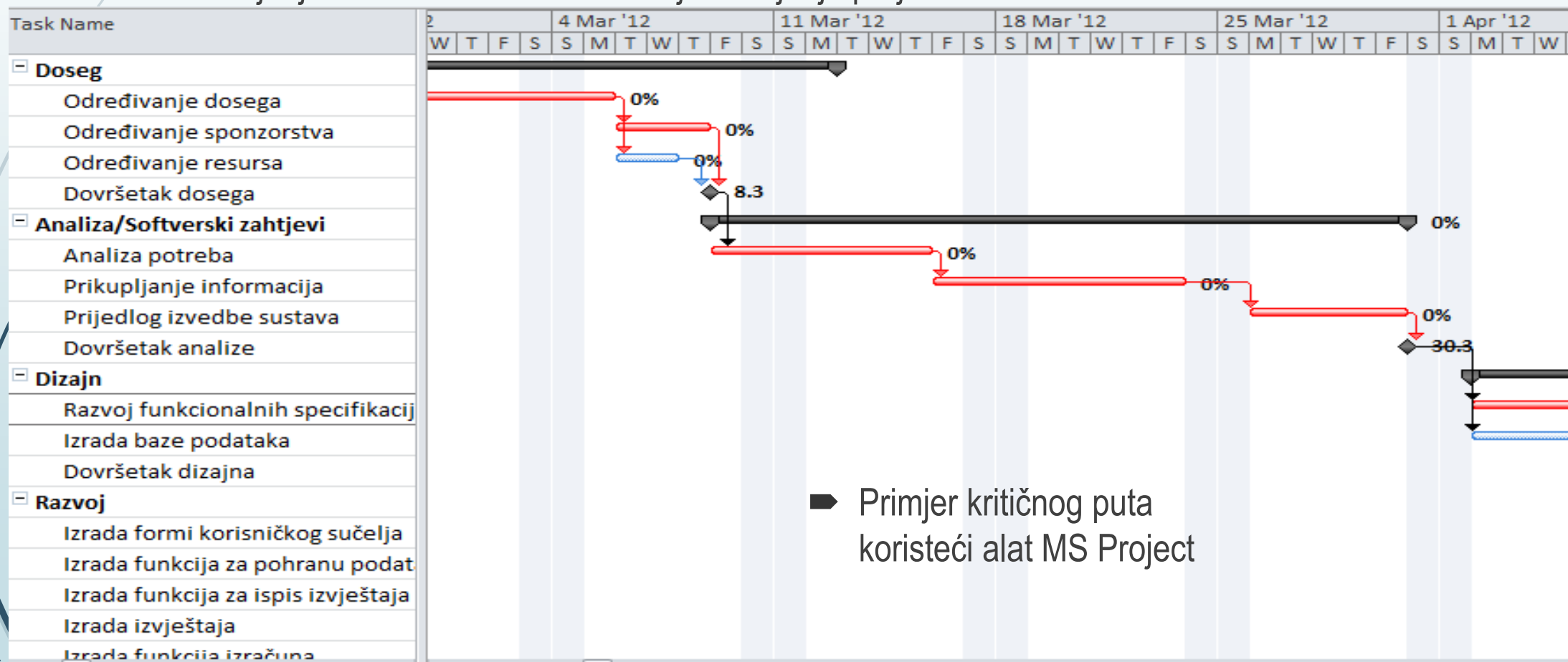


Kritični put

11

➤ Kritični put

- niz zadataka koji moraju završiti na vrijeme da bi projekt završio na vrijeme
- svaki zadatak na kritičnom putu je kritični zadatak
- kašnjenje kritičnih zadataka uzrokuje kašnjenje projekta



- Primjer kritičnog puta koristeći alat MS Project

➤ Resursi - sredstva

- ljudi, oprema i materijal potrebni za obavljanje zadataka

➤ Vrste resursa:

➤ Resursi rada (work resources)

- ljudi (ograničeno vrijeme rada)
- oprema (neograničeno vrijeme rada)

➤ Resursi materijala (material resources)

- potrošni materijal koji predstavlja projektni utržak
- daje informaciju o brzini konzumiranja resursa

➤ Dva važna pogleda na resurse:

- Raspoloživost - u koje vrijeme određeni resurs može raditi na zadatku i koliko posla može obaviti
- Trošak – koliko novca će biti potrošeno na resurse

- Unos resursa i pratećih podataka (dostupnost, trošak)
- Maksimalne jedinice (max. units)
 - prikazuju vrijednosti raspoloživosti resursa u postocima
 - 100% predstavlja jednog čovjeka punog radnog vremena
 - 300% predstavlja tri čovjeka punog radnog vremena
- Osnova po kojoj je resursu dodijeljeno obavljanje posla
- Za pojedinačnu prilagodbu uvažavaju se radni i neradni dani resursa
 - Primjer: ako kalendar evidentira radno vrijeme samo četvrtkom i petkom 13-17 sati, 100% raspoloživosti nekog resursa ne znači 40 satno tjedno radno vrijeme, nego 8 sati rada tjedno

- Dodjelom resursa zadatku, dodjeljuje mu se određeni posao
- Razlikovati posao (work) od trajanja (duration)
 - Posao je stvarni rad i odnosi se na
 - zadatak - stvarni rad potreban za završetak zadatka
 - dodijeljeni zadatak - stvarni rad nekog resursa na nekom zadatku
 - resurs - ukupni rad neke osobe na svim zadacima
 - Trajanje = Posao / Jedinice ($\text{Duration} = \text{Work} / \text{Units}$)
 - Ovisi o kalendaru rada
- Raspoređivanje temeljem napora (effort-driven scheduling)
 - metoda planiranja koja se koristi kod ažuriranja resursa zadatka
 - trajanje obrnuto količini resursa

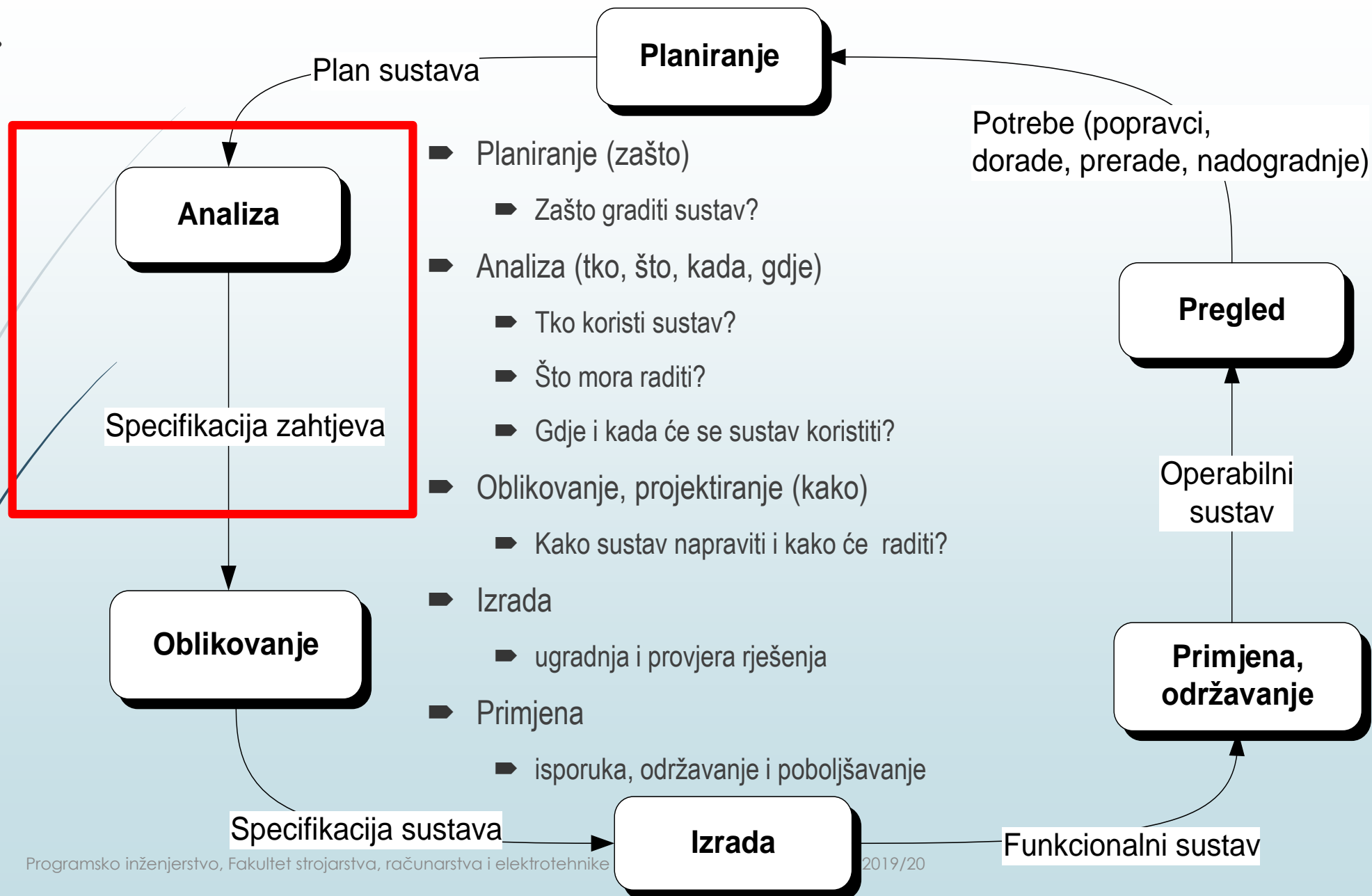
➡ Rješavanje problema:

- ➡ Pomaknuti rokove (trajanje)
- ➡ Dodati dodatne resurse
- ➡ Produžiti radno vrijeme
- ➡ Povećati jedinice posla
- ➡ Smanjiti količinu posla

Specifikacija zahtjeva

Životni ciklus programske potpore

17

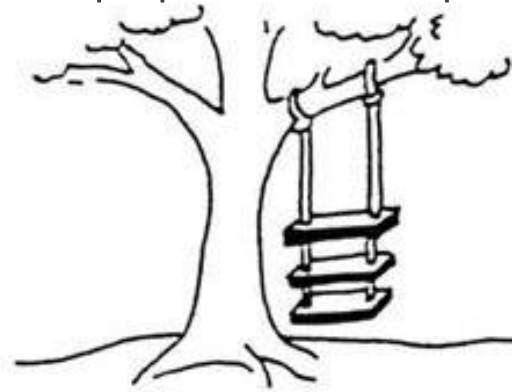


- ISO/IEC/IEEE 24765:2010 Systems and software engineering—Vocabulary:
 - uvjet ili sposobnost koje korisnik treba da bi riješio problem ili ostvario cilj.
 - uvjet ili sposobnost koji mora posjedovati ili zadovoljiti sustav, komponenta sustava, proizvod ili usluga da bi zadovoljila ugovor, standard, specifikacije ili neki drugi ugovoreni dokument.
 - Dodatno, prema PMBOK zahtjevi uključuju nabrojane i dokumentirane potrebe, želje i očekivanja sponzora, korisnika i ostalih dionika u projektu
- ISO/IEC/IEEE 29148:2011 Systems and software engineering--Life cycle processes--Requirements engineering
 - izjava kojom se prevodi ili izražava potreba i potrebi pridružena ograničenja i uvjeti

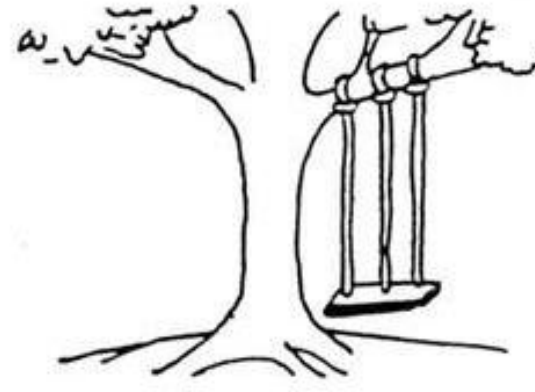
Pogrešno postavljeni zahtjevi za posljedicu imaju neispunjena očekivanja

19

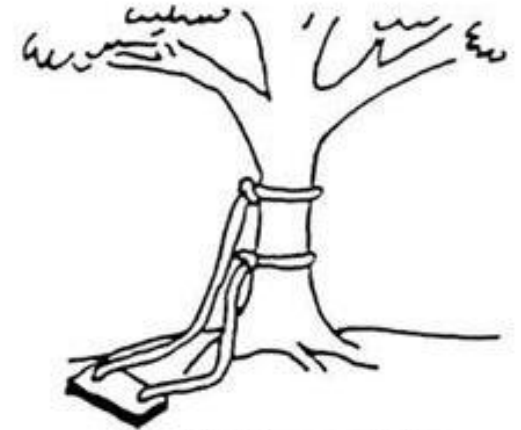
► ... a naknadne prepravke su „skupe”



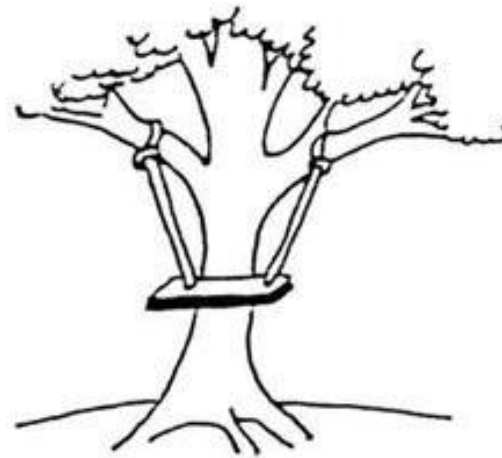
*As proposed by
the project sponsors*



*As specified in
the project request*



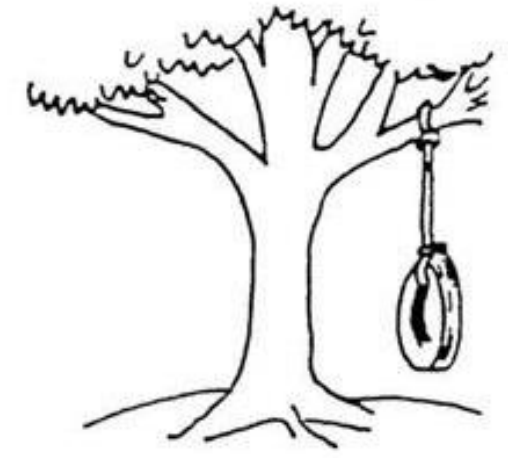
*As designed by
the senior analyst*



*As produced by
the programmers*



*As installed at
the user's site*



*What the user
wanted*

➤ Poslovni zahtjevi (zašto)

- ciljevi organizacije ili zahtjevi rukovodstva, poslovni problemi ili potrebe
- opisani u viziji i doseg projekta

➤ Korisnički zahtjevi (zahtjevi krajnjih korisnika)

- opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijama
- sadržani u opisima slučajeva korištenja, tj. opisima scenarija rada

➤ Funkcionalni zahtjevi (što)

- očekivana softverska funkcionalnost potpore korisničkih zadataka
- mogućnost programa (feature) – skup funkcionalnih zahtjeva za ispunjenje poslovnih

➤ Nefunkcionalni zahtjevi (kako ili kako dobro)

- sistemski zahtjevi, atributi kvalitete, ostali nefunkcionalni zahtjevi
- standardi, pravila i ugovori, opisi vanjskih sučelja

➤ Odgovaraju na pitanje *zašto* (se radi neki sustav)

- predstavljaju ciljeve organizacije ili korisničke zahtjeve na višoj razini i ukratko opisuju problem koji treba riješiti
- sadržani u dokumentima u kojima se opisuje vizija i opseg projekta

➤ Primjeri:

- poboljšanje usluge postojećim klijentima tvrtke i pridobivanje novih
- evidencija članstva i automatizacija postupka primanja novih članova neke udruge
- praćenje finansijskih podataka udruge i njenih članova
- poboljšanje procesa prodaje
- omogućavanje internetske prodaje
- podrška organiziranju natjecanja i okupljanja

Primjeri zahtjeva vlasnika sustava

22

➤ Očekivana novčana ušteda

- Sustav mora biti tako koncipiran da prava na subvencioniranu prehranu može koristiti samo student koji ih je stekao i da ih može koristiti samo u svrhu prehrane.

➤ Primjer: Sustav mora onemogućiti:

- korištenje subvencije od strane osoba koje nemaju na to pravo
- zaradu ilegalnih posrednika
- korištenje subvencije za druge svrhe osim prehrane
- naplatu usluga koje nisu pružene

➤ U idealnom slučaju zahtjevi vlasnika podudaraju se s poslovnim ciljevima!

➤ Zahtjevi krajnjih korisnika

- opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijama
- sadržani u opisima slučajeva korištenja, tj. opisima scenarija rada
- obično se izražavaju u obliku „Korisnik želi/treba/mora moći obaviti...”.

➤ Primjeri:

- *Korisnik mora moći ostvariti pravo na prehranu kod bilo kojeg pružatelja usluge* - Novi sustav mora omogućiti da student ostvaruje svoje pravo kod bilo kojeg pružatelja usluge subvencionirane prehrane. Dosadašnja praksa je bila da svaki pružatelj usluga izdaje svoje bonove koji se mogu koristiti samo u određenim restoranima
- *Korisnik treba plaćati obroke nakon korištenja pojedinog obroka.* - Treba izbjeći bilo kakvo plaćanje od strane studenata za potrebe ostvarivanja prava, a posebice unaprijed.
- *Korisnik mora moći prijaviti gubitak kartice* – Potrebno je smanjiti rizik gubitka ostvarenih prava te sustav mora onemogućiti zlorabu stečenih prava.

Funkcionalni zahtjevi

24

- Odgovaraju na pitanje što (se može/mora napraviti koristeći sustav)
 - definiraju softversku funkcionalnost (očekivano ponašanje i operacije koje sustav može izvoditi) koju treba ugraditi u proizvod da bi omogućio korisnicima obavljanje njihovih zadataka
 - posebno zanimljiva mogućnost programa (feature) – skup logički povezanih funkcionalnih zahtjeva koje korisniku omogućuju ispunjavanje poslovnih zahtjeva
- Primjeri:
 - Nakon što se studentu jednom zavedu prava na matičnoj ustanovi, sustav mora proslijediti informaciju svim pružateljima usluga, odnosno omogućiti distribuirane upite
 - Sustav na dnevnoj bazi mora kreirati izvještaje sa statistikom prehrane po pružateljima usluge i vrsti obroka.

Nefunkcionalni zahtjevi

25

- Odgovaraju na pitanje kako (ili kako dobro sustav mora raditi)
 - posljedica standarda, pravila i ugovora kojih se proizvod mora pridržavati
 - opisi vanjskih sučelja
 - zahtjevi na performanse
 - ograničenja na dizajn i implementaciju te svojstva kvalitete (preciziraju opis proizvoda navodeći karakteristike proizvoda u različitim dimenzijama koja su važne ili korisniku, ili graditelju).
- Primjer
 - U sustavu prehrane nefunkcionalni zahtjevi mogu biti vezani za oblik korisničke kartice, protokol povezivanja, obvezu fiskalizacije, zaštitu osobnih podataka, itd...
 - Svaki zapis sadrži identifikator korisnika i datum unosa ili promjene

Primjer nepotpunog zahtjeva

26

- "Proizvod će dostaviti statusnu poruku u redovitim intervalima ne manjim od 60 sekundi."
 - Što je statusna poruka i pod kojim uvjetima će biti dostavljena?
 - Koliko dugo ostaje vidljiva?
 - Koji dio proizvoda će dostaviti poruku?
 - Koliko dosljedni intervali moraju biti?
- Zahtjev, preciznije i detaljnije
 - Modul za nadzor će ispisivati statusnu poruku u za to određeni dio sučelja.
 - Poruka će se ažurirati svakih 60 sekundi (plus minus 10 sekundi) nakon što započne izvođenje pozadinskog zadatka i bit će vidljiva cijelo vrijeme.
 - Ukoliko se pozadinski zadatak izvodi normalno, modul za nadzor će ispisivati postotak obavljenog posla.
 - Modul za nadzor će ispisati "Zadatak obavljen." nakon što se zadatak obavi.
 - Modul će ispisati poruku o pogrešci ukoliko dođe do zastoja u izvođenju.
 - Problem je rastavljen u više zahtjeva jer će svaki zahtijevati posebno testiranje.
 - Ukoliko je više zahtjeva grupirano u jedan lakše je previdjeti neki od njih tijekom izrade ili testiranja.
- Primijetiti da u zahtjevu nije detaljno opisano kako će se poruka i gdje ispisivati. To će biti odlučeno tijekom dizajna !

Primjer neostvarivog zahtjeva

27

- "Proizvod će se trenutno prebaciti između ispisivanja i skrivanja znakova koji se ne mogu tiskati."
 - Računala ništa ne mogu napraviti trenutno te je ovaj zahtjev neostvariv.
 - Da li programska podrška sama odlučuje kad će se prebaciti iz jednog stanja u drugo ili je to inicirano akcijom korisnika?
 - Na koji dio teksta će se primijeniti promjena prikaza: da li samo označeni tekst, cijeli dokument ili nešto treće?
 - Nejednoznačnost: da li su "znakovi koji se ne mogu tiskati" skriveni znakovi, posebne oznake ili kontrolni znakovi?
- Bolji zahtjev:
 - "Korisnik će posebno dogovorenom akcijom, odabrati da li će se HTML oznake u trenutno otvorenom dokumentu prikazivati ili ne."
 - Sad je jasno da je riječ o HTML oznakama te da korisnik mora obaviti nekakvu akciju, ali nije točno navedeno kakvu (npr. kombinacija tipki), što se prepušta dizajnerima.

Primjer neodređenog zahtjeva

28

- "Parser će brzo generirati izvješće o pogreškama HTML oznaka, koje omogućava brzi ispravak pogrešaka kada program koriste početnici u HTML-u."
 - Riječ "brzo" je neodređena.
 - Nije definirano što i kada se tvori izvješće i to čini zahtjev nekompletnim.
 - Kako se ovjerava zahtjev? Pronaći nekoga tko se smatra početnikom u HTML-u i zatim vidjeti kako brzo će, uz pomoć izvješća, ispraviti pogreške?
 - Kada se generira izvješće?
- Bolje:
 - Nakon što je HTML analizator obradio datoteku generirat će izvješće koje sadrži broj linije i tekst pronađenih HTML pogrešaka, te opis svake pogreške.
 - Ukoliko nema pogrešaka prilikom analize, neće se generirati izvješće.

Postavljanje prioriteta

29

- Nužno svojstvo - Da li korisnik nešto stvarno mora imati?
 - Postoji tendencija da se previše zahtjeva proglasi nužnim!
 - Po definiciji, ako sustav ne uključuje nužne zahtjeve, taj sustav ne može ispuniti svoju svrhu.
 - Treba testirati svaki zahtjev koji se smatra nužnim i probati ga rangirati.
 - Ako se zahtjev može rangirati onda nije obavezan!
 - Potpuno obvezni zahtjevi se ne mogu rangirati jer su nužni za prvu verziju sustava!
- Poželjno svojstvo - Funkcije koje korisnik želi na kraju imati
 - Ranije verzije sustava mogu pružiti (ne potpunu) funkcionalnost bez tih zahtjeva.
 - Poželjni zahtjevi mogu i trebaju biti rangirani.
- Neobvezna svojstva - Proizvoljni zahtjevi
 - Svojstva i mogućnosti bez kojih se može (npr. ostvarivanje ostalih prava iz studentskog standarda iz primjera zahtjeva krajnjih korisnika)
 - Iako bi ih lijepo bilo imati, to nisu pravi zahtjevi.
 - Ovi zahtjevi također mogu biti rangirani.

Postupci prikupljanja informacija (1)

30

- *Prikupljanje informacija se ne odnosi samo na fazu analize, već i planiranja i oblikovanja*
- Intervjui s ključnim korisnicima
 - uključiti predstavnike svih grupa korisnika i lokalne informatičare
 - zatvoreni (korisnici odgovaraju na unaprijed pripremljena pitanja) ili otvoreni
- Repertoar i vrste pitanja
 - Pitanja zatvorenog tipa: Koliko ...?, Kako se ... ?
 - Pitanja otvorenog tipa: Što mislite o ... ?, Koji su najveći problemi ... ?
 - „Probna” pitanja: Zašto?, Primjer?, Objašnjenje za ...
- Klasične poteškoće:
 - specifična terminologija nerazumljiva programskim inženjerima
 - ispuštanje informacija dobro poznatim korisnicima, ali ne i ostalima
 - razlučiti činjenice od mišljenja, provjeriti moguće kontradikcije
 - iznošenje „službene” istine, odnosi među ljudima i organizacijama
- Prikupljene informacije dopuniti iz drugih izvora: postojeća baza podataka, dokumentacija i (papirnat) obrasci, literatura, zakoni, propisi, ...

Postupci prikupljanja informacija (2)

31

➤ Radne sjednice

- zajednička analiza korisnika i analitičara, *brainstorming*

➤ Upitnici i ankete

- Nadomjestak za intervju ili prikupljanje informacija o resursima.

➤ Analiza dokumentacije

- Treba prikupiti sve dokumente značajne za poslovanje (pravila, strukture podataka)

➤ Promatranje (etnografski pristup)

- Neposredni uvid u poslovne procese promatranjem radnih sredina

➤ Ostale tehnike

- Prosudba postojećih aplikacija ili evidencija na računalu • analiza funkcionalnosti, strukture podataka te podataka koje treba “spasiti”
- Prototipiranje – kada nema uzora ili korisnik ne zna što hoće

➤ Postupak analize mora biti prilagođen korisniku !


Zahtjevi – što, a ne kako!

32

- Prijedlog rješenja nije zahtjev!
- Zahtjevi ne smiju sadržavati detalje dizajna ili implementacije
 - ali ako su nepotpuni onemogućuju planiranje projekta i praćenje promjena
- Usmjeriti se na ono što je potrebno obaviti, a ne na način realizacije
- Istražiti zašto korisnik predlaže određenu ugradnju – razumijevanje potrebe
- Ne izjednačavati „tako se uvijek radi” s „tako treba raditi”!
 - Može dovesti do cementiranja loših navika

Dokumentiranje analize (zahtjeva)

33

- Definicija zahtjeva (Requirements Definition)
 - izjava o stanju i ograničenjima sustava te potrebama
 - narativni dokument namijenjen korisniku ili ga piše korisnik
 - poslovni i korisnički zahtjevi te njihovi prioriteti
 - uočeni problemi, ključne pretpostavke i preporuke rješenja
- Primjeri:
 -  Firma-Zahtjevi.doc (SpecifikacijaZahtjeva.dot)

Dokumentiranje analize (zahtjeva) (2)

34

➤ Specifikacija zahtjeva (Requirements Specification)

- često se naziva i funkcionalnom specifikacijom
- strukturirani dokument s detaljnim opisom očekivanog ponašanja sustava
- namijenjen ugovarateljima i izvoditeljima razvoja
- ugradbeno nezavisan pogled na sustav
 - funkcionalni i nefunkcionalni zahtjevi te njihovi prioriteti
 - model organizacijske strukture (strukturni dijagrami)
 - opis protoka dokumenata (dijagrami toka)
 - model procesa (dijagram toka podataka)
 - konceptualni model podataka (dijagram entiteti-veze)

Predložak dokumenta specifikacije zahtjeva

35

1. Uvod

- 1.1 Namjena
- 1.2 Konvencije dokumenta
- 1.3 Tko treba čitati dokument i savjeti za čitanje dokumenta
- 1.4 Opseg proizvoda
- 1.5 Reference

2. Sveobuhvatni pregled

- 2.1 Kontekst proizvoda
- 2.2 Funkcije proizvoda
- 2.3 Kategorije korisnika i svojstva
- 2.4 Okruženje u kojem se izvodi proizvod
- 2.5 Ograničenja dizajna i ugradnje
- 2.6 Pretpostavke i ovisnosti

3. Zahtjevi za sučeljem

- 3.1 Korisničko sučelje
- 3.2 Hardversko sučelje
- 3.3 Softversko sučelje
- 3.4 Komunikacijsko sučelje

4. Svojstva sustava

- 4.x Svojstvo X
 - 4.x.1 Opis i prioriteti
 - 4.x.2 Nizovi pobuda/odziv
 - 4.x.3 Funkcijski zahtjevi

5. Ostali nefunkcionalni zahtjevi

- 5.1 Zahtjevi za performansama sustava
- 5.2 Zahtjevi za sigurnošću korisnika
- 5.3 Zahtjevi za sigurnošću podataka
- 5.4 Kvaliteta programske podrške
- 5.5 Poslovna pravila
- 5.6 Korisnička dokumentacija

6. Ostali zahtjevi

Dodatak A: Rječnik

Dodatak B: Modeli i dijagrami

Dodatak C: Lista nedovršenih/neodređenih zahtjeva

- Uzastopni brojevi
 - novi zahtjev dobiva raspoloživi serijski broj (npr. Z01, Z02, ... ili FZ01,...).
 - nepregledno, teško za pratiti veći broj zahtjeva, nepraktično brisanje
- Hijerarhijsko numeriranje (X.Y.Z)
 - jednostavno uređivanje (Word), problem brisanja koje uzrokuje posmak
 - poboljšanje: tekstovne oznake unutar brojčanih hijerarhija
 - Primjer: "3.2. - Funkcije editora", sa zahtjevima "ED-01", "ED-02", itd.
- Hijerarhijske tekstovne oznake
 - „objektno”, npr. ISPIS.KOPIJE.POTVRDA
 - prednost - strukturiranost, jednostavnost, nedostatak: nezgrapnost
- Potpunost
 - nijedan zahtjev ili informacija ne smiju nedostajati – teško uočljivo
 - kontrola „praćenjem” poslovnog procesa
 - nepotpune posebno označiti (npr. TBD - "to be determined")
 - • razriješiti prije ugradnje

- Scenarij – neformalni opis zamišljene interakcije korisnika i sustava
 - Korisničke priče kao jednostavna vrsta scenarija
- Tipični sadržaj
 - polazne pretpostavke koje vrijede na početku interakcije
 - normalni tok događaja
 - varijabilni (alternativni) ili iznimni tok događaja
 - popis drugih istovremenih aktivnosti koje mogu utjecati na trenutnu interakciju
 - stanje sustava nakon što interakcija završi

Primjer scenarija korištenja

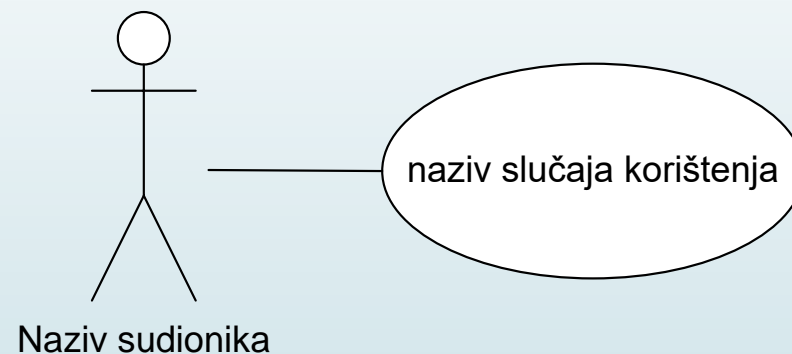
38

- Plaćanje proizvoda iz košarice (korisnička priča:Korisnik mora moći platiti odabrane proizvode karticom)
- Preduvjet: Kupac je dodao proizvode u košaricu te se odlučuje za plaćanje
- Normalni tok:
 1. Kupac unosi podatke o otpremi
 2. Sustav prikazuje punu informaciju o cijeni, uključujući otpremu
 3. Kupac unosi podatke o kreditnoj kartici
 4. Sustav autorizira kupnju
 5. Sustav potvrđuje prodaju interaktivno
 6. Sustav šalje elektroničko pismo potvrde
- Alternativa: Neuspješna autorizacija
 - U koraku 4, sustav ne uspijeva autorizirati karticu
 - Dozvoliti kupcu ponovni unos podataka o kartici
- Alternativa: Postojeći kupac
 - 1a. Kupac se prijavljuje te sustav prikazuje informaciju o otpremi i cijeni te zadnje četiri znamenke broja kreditne kartice
 - Kupac potvrđuje prikazane pretpostavljene podatke ili unosi nove
- **Primijetiti da scenarij ne sadrži implementacijske detalje**

Dijagrami slučajeve korištenja

39

- engl. *use-case diagram*
- Jedan slučaj obično predstavlja jednu interakciju između sustava i okoline (sudionika)
 - Sudionik, glumac, akter, ...
- Svaki slučaj korištenje opisuje se scenarijem ili dijagramom slijeda (engl. *sequence diagram*)



- Pojedini slučajevi mogu
 - uključivati («include») zajedničke dijelove ponašanja
 - biti proširenje («extend»), tj. specijalni slučaj nekog postojećeg slučaja

40

-
- ```
graph TD
 Kupac((Kupac))
 Administrator((Administrator))
 Prijava(Prijava korisnika)
 Registracija(Registracija korisnika)
 Placanje(Plaćanje)
 Izracun(Izračun troškova dostave)
 Pregled(Pregled narudžbi)

 Kupac --> Prijava
 Kupac --> Registracija
 Kupac --> Placanje
 Administrator --> Pregled

 Registracija -.->|<<extend>>| Prijava
 Placanje -.->|<<extend>>| Registracija
 Placanje -.->|<<include>>| Izracun
```



- Provjera dokumentacije sa zahtjevima
  - jedna od najkorisnijih softverskih tehnika
  - mikro ekipa (analitičari, projektanti, ... , korisnici) ispituje specifikacije i modele
- Pisanje testova, probnih slučajeva
  - prototipiranje, izvođenje scenarija korištenja – provjera očekivanog ponašanja
  - slučajeve povezati s funkcionalnim zahtjevima da se osigura potpunost
- Pisanje korisničkog priručnika
  - skica korisničkog priručnika - kao specifikacija ili dio analize
  - dobre upute opisuju svu vidljivu funkcionalnost – ostalo je u SRS
- Definiranje kriterija prihvatljivosti
  - definiranje (s korisnicima) uvjeta pod kojima će proizvod zadovoljiti zahtjeve
  - na osnovu slučajeva/scenarija korištenja

# Upravljanje zahtjevima (1)

42

- Definiranje postupka za promjenu zahtjeva
  - postupak kojim se novi zahtjev ili promjena postojećeg analizira i prihvća
  - predložene promjene moraju slijediti unaprijed definiranu proceduru
- Uspostava odbora za promjene (*Change Control Board, CBB*)
  - ključni članovi projekta
  - odlučuje o usvajanju zahtjeva te postavlja prioritete i rokove
- Analiza utjecaja promjena zahtjeva
  - procjenjuje se utjecaj promjene na organizaciju, raspored ili drugo
  - služi donošenju dobrih (ispravnih, mogućih) odluka o (ne)prihvćanju promjena
- Praćenje promjena zahtjeva na svim proizvodima
  - za prihvaćenu promjenu se kroz matricu praćenja zahtjeva pronalaze sve ovisne komponente (izvorni kod, testni slučajevi, neki drugi zahtjev,...).

# Upravljanje zahtjevima (2)

43

- Uspostava vremenske osnovice (baseline) i kontrole verzija
  - definiranje slike dogovorenih zahtjeva u određenom trenutku
  - nakon osnovice, zahtjevi prolaze postupak promjena
  - verzioniranje specifikacije – uklanjanje neodređenosti
- Praćenje povijesti promjena zahtjeva
  - zapis o vremenu, vrsti i sadržaju, razlozima, verziji i autoru promjene
- Praćenje statusa zahtjeva
  - status (predložen, odobren, ugrađen, provjeren) + statistika
- Mjerenje stabilnosti zahtjeva
  - brojanje promjena – mjera da je problem shvaćen, opseg definiran
- Korištenje alata za upravljanje zahtjevima
  - evidencija zahtjeva, međusobne povezanosti i statusa
  - verzioniranje zahtjeva i dokumentacije
- **Inženjerstvo zahtjeva** (requirements engineering) = razvoj zahtjeva (određivanje zahtjeva, analiza, specificiranje (dokumentiranje) i verifikacija) + upravljanje zahtjevima.