Oblikovanje programske potpore

2012./2013. grupa P01

Inženjerstvo zahtjeva u oblikovanju programske potpore

Prof.dr.sc. Vlado Sruk



Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva



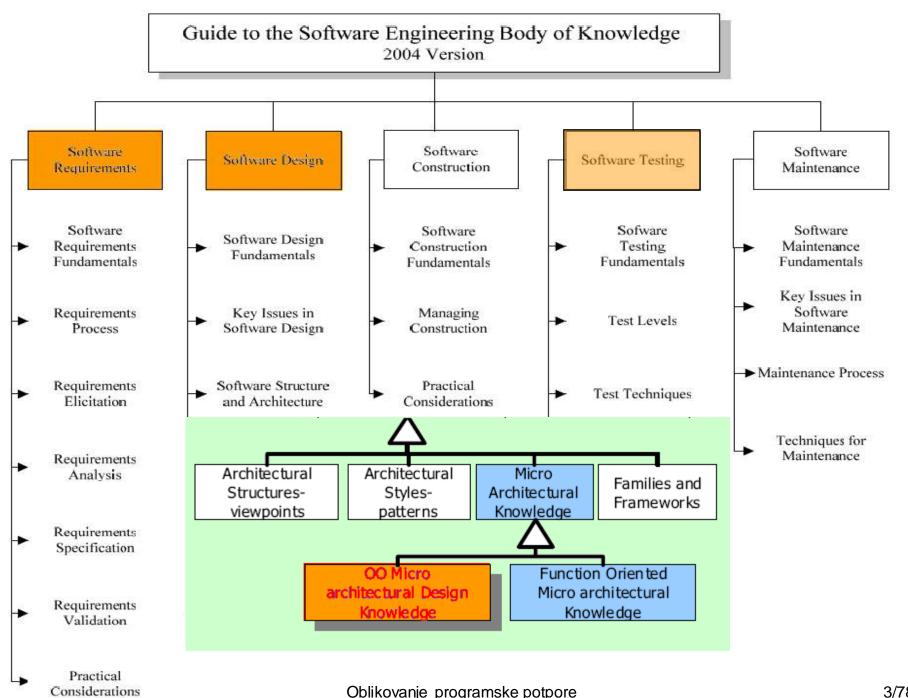
Zavod za elektroniku, mikroel., računalne i inteligentne sustave

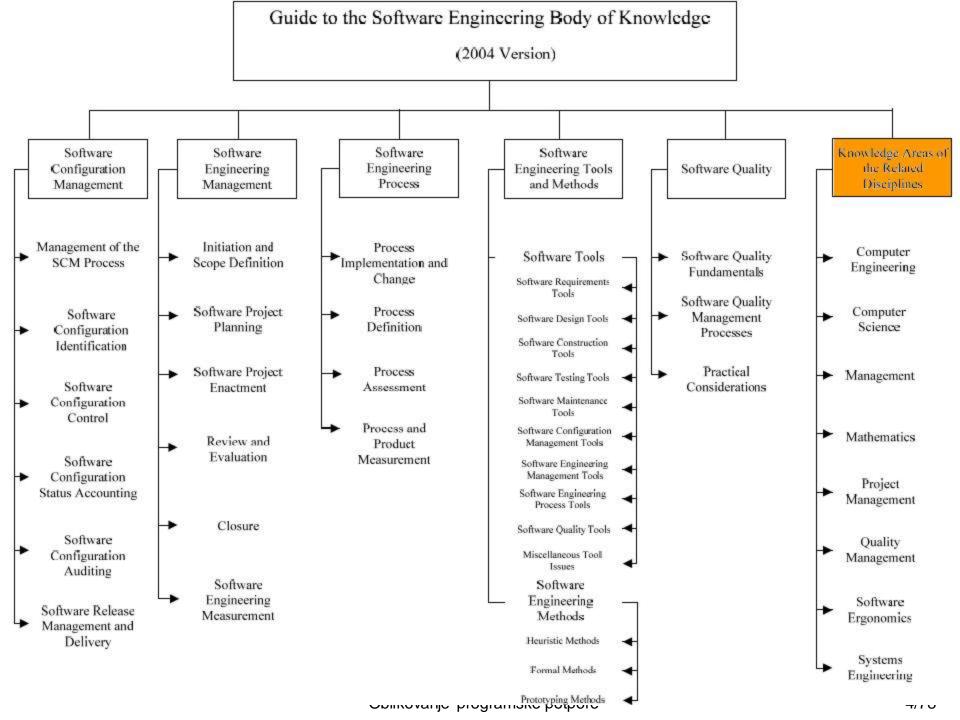


Osvrt



Sadržaj programskog inženjerstva







Tema



Inženjerstvo zahtjeva (engl. requirements engineering)

- Koncept korisničkih zahtjeva i zahtjeva sustava
- Tipovi zahtjeva
- Organizacija dokumenata zahtijeva programske potpore
- Proces programskog inženjerstva
- Temeljne inženjerske aktivnosti generiranja i dokumentiranja zahtjeva.
- Tehnike izlučivanja i analize zahtjeva.



Literatura



- Sommerville, I., Software engineering, 8th ed., Addison-Wesley, 2007.
- Grady Booch, James Rumbaugh, Ivar Jacobson: *Unified Modeling Language User Guide*, 2nd Edition, 2005
- Simon Bennett, John Skelton, Ken Lunn: Schaum's
 Outline of UML, Second Edition, 2005
- WWW
- Rational Software Architect
 - http://publib.boulder.ibm.com/infocenter/rtnlhelp/v6r0m0/in dex.jsp?topic=/com.ibm.xtools.modeler.doc/topics/cextend. html



Inženjerstvo zahtjeva



- engl. requirements engineering, requirements gathering, requirements capture, requirements specification
- To je postupak pronalaženja, analiziranja, strukturiranja, dokumentiranja i provjere korisnički zahtijevanih usluga sustava, te ograničenja u uporabi.
 - fokus: utvrđivanje ciljeva, funkcija i ograničenja sklopovlja i programske podrške.
- Zahtjevi sami za sebe su opisi usluga sustava i ograničenja koja se generiraju tijekom procesa inženjerstva zahtjeva.
 - klasifikacija zahtjeva obzirom na razinu detalja i sadržaj.

Masifikacija zahtjeva prema razini detalja

Korisnički zahtjevi

- specifikacija visoke razine apstrakcije u okviru ponude za izradu programskog produkta
- pišu se u prirodnom jeziku i grafičkim dijagramima.
 - Klijenti (rukovoditelji manager)
 - Krajnji korisnici sustava
 - Klijenti (inženjeri)
 - Rukovoditelji za pisanje ugovora
 - Specijalisti za oblikovanje sustava (arhitekti)

Zahtjevi sustava

- vrlo detaljna specifikacija uobičajeno nakon prihvaćanja ponude, a prije sklapanja ugovora
- pišu se strukturiranim prirodnim jezikom, posebnim jezicima za oblikovanje sustava, dijagramima i matematičkom notacijom.
 - Krajnji korisnici sustava
 - Klijenti (inženjeri)
 - Specijalisti za oblikovanje sustava (arhitekti)
 - Specijalisti za razvoj programske potpore

Specifikacija programske potpore

- najdetaljniji opis koji objedinjuje korisničke i zahtjeve sustava
 - Klijenti (inženjeri) možda ?
 - Specijalisti za oblikovanje sustava (arhitekti)
 - Specijalisti za razvoj programske potpore





KORISNIČKI ZAHTJEVI

Klasifikacija zahtjeva obzirom na sadržaj

- Odnosi se na korisničke zahtjeve i zahtjeve sustava
- Funkcionalni zahtjevi (engl. Functional requirement)
 - 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.
 - specifikacija rezultata rada: engl. "system shall do <requirement>"
- Nefunkcionalni zahtjevi (engl. Non-functional requirement)
 - ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja,
 (ne)usvojeni standardi, ograničenja u procesu razvoja i oblikovanja i sl.
 - naglasak na karakteristikama: engl. "system shall be <requirement>"
- Zahtjevi domene primjene
 - zahtjevi (funkcionalni i nefunkcionalni) koji proizlaze iz domene primjene sustava kao i oni koji karakteriziraju tu domenu.



Primjer funkcionalnih zahtjeva: Sustav LIBSYS



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

Izvor: Sommerville, I., Software engineering

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

Primjer: http://www.libsys.co.in/offerings-libsys7.html



Poteškoće: Prirodni jezik



- Zahtijeva socijalne i komunikacijske vještine
- Nedostatak jasnoće
 - preciznost nije lako postići bez detaljiziranog i teško čitljivog dokumenta
- Miješaju se funkcionalni i nefunkcionalni zahtjevi.
- Nenamjerno objedinjavanje više zahtjeva u jednom.
- Nejasno postavljeni zahtjevi mogu biti različito interpretirani od korisnika i razvojnih timova
 - uzrokuje probleme u procesu razvoja i kršenje ugovora
- Npr. u LIBSYS sustavu: "odgovarajući preglednik":
 - korisnik preglednici posebne namjene za svaki tip dokumenta.
 - razvojni tim samo preglednik teksta kao bitnog sadržaja dokumenta.



Primjer:



- Zahtjev potporne rešetke (engl. grid) u grafičkom uređivaču:
- Za precizno postavljanje entiteta na crtež, korisnik može uporabom upravljačkog panela uključiti rešetku u centimetrima ili inčima. Inicijalno je rešetka isključena. Rešetka se može uključiti i isključiti kao i mijenjati mjerne jedinice tijekom rada s editorom. Biti će osigurana opcija smanjivanja slike kako bi stala na zaslon, ali broj prikazanih crta rešetke će biti reduciran kako ne bi popunile manje slike.
- Miješaju se tri različita tipa zahtjeva:
 - Koncepcijski
 - potreba za rešetkom
 - Nefunkcionalni
 - mjerne jedinice rešetke
 - Nefunkcionalni ulazno izlazni zahtjevi
 - prebacivanje između tipova rešetki

Izvor: Sommerville, I., Software engineering

🥯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 složenih sustava.



Problemi dokumentiranja zahtjeva



- Različitost izražavanja specifikacije
 - različiti stilovi pisanja
 - različita razina iskustva
 - različiti formati
 - predetaljni ili nedovoljno specificirani zahtjevi
- Moguća poboljšanja:
 - jedan zahtjev po stavci
 - grupiranje (hijerarhijsko) povezanih zahtjeva
 - numeriranje
 - uporaba standarda



🥟 Klasifikacija nefunkcionalnih zahtjeva

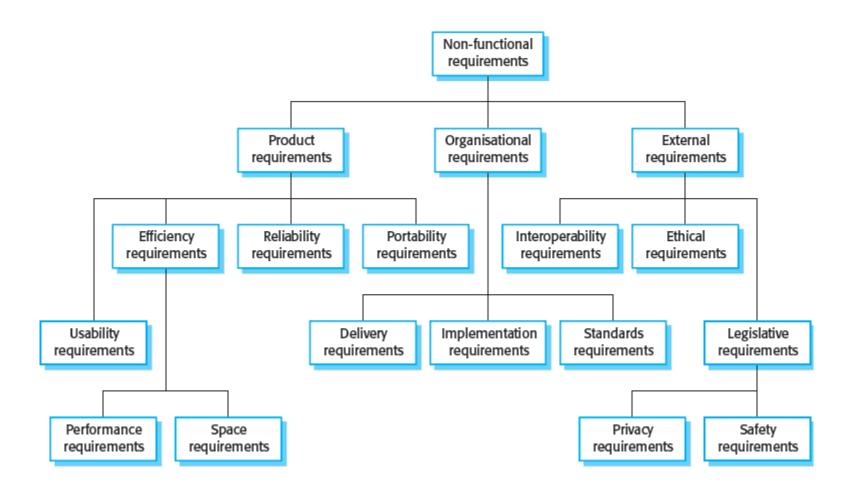


- Zahtjevi programskog produkta:
 - zahtjevi koji specificiraju da se isporučeni produkt mora ponašati na osobit način (npr. vrijeme odziva).
- Organizacijski zahtjevi:
 - zahtjevi koji su rezultat organizacijskih pravila i procedura (npr. uporaba propisanog standardnog procesa razvoja, DoD ADA).
- Vanjski zahtjevi:
 - zahtjevi koji proizlaze izvan sustava i razvojnog procesa (međusobna operabilnost, legislativni zahtjevi i sl.).
- Moraju biti mjerljivi!



Primjer klasifikacije





Izvor: Sommerville, I., Software engineering



Primjer: Nefunkcionalni zahtjevi



- Analiza primjera LIBSYS:
- Zahtjevi programskog produkta:
 - korisničko sučelje LIBSYS sustava biti će implementirano kao jednostavni HTML bez uporabe okvira ili Java apleta.
- Organizacijski zahtjevi:
 - proces razvoja sustava i isporučeni dokumenti moraju slijediti standard XYZCo-SP-STAN-95.
- Vanjski zahtjevi:
 - sustav neće operatorima otkriti osobne informacije o klijentima (osim njihovog imena i referentnog broja).



Zahtjevi domene primjene



- Zahtjevi domene primjene mogu biti novi funkcionalni zahtjevi ili ograničenja na postojeće zahtjeve.
- 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).
- Npr. LIBSYS zahtjevi domene primjene:
 - zbog ograničenja 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.



Primjer: Standard ISO 9126



- ISO/IEC 9126 Software engineering -Definira osnovna svojstva programske potpore
- Funkcionalnost (engl. functionality)
 - suitability
 - accuracy
 - interoperability
 - security
 - functionality Compliance
- Pouzdanost (engl. reliability)
 - maturity
 - fault Tolerance
 - recoverability
 - reliability Compliance
- Uporabljivost (engl. usability)
 - understandability
 - learnability
 - operability
 - attractiveness
 - usability Compliance

- Efikasnost (engl. efficiency)
 - Time Behaviour
 - Resource Utilisation
 - Efficiency Compliance
- Održavanje (engl. Maintainability)
 - Analyzability
 - Changeability
 - Stability
 - Testability
 - Maintainability Compliance
- Prenosivost (engl. Portability)
 - Adaptability
 - Installability
 - Co-Existence
 - Replaceability
 - Portability Compliance



Primjer



- Tablični zapis funkcijskih zahtjeva
- tablični zapis nefunkcijskih zahtjeva

Zahtjev	Opis
F1	
F2	

Zahtjev	Opis
NF1	
NF2	
D1	





ZAHTJEVI SUSTAVA



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 prikazati nekim od modela sustava.
 - model procesa programskog inženjerstva je apstraktna reprezentacija procesa.
- Odnos zahtjeva sustava i oblikovanja
 - 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 radit u sinergiji s drugim sustavima koji generiraju zahtjeve na oblikovanje.
 - Uporaba specifičnog oblikovanja može biti zahtjev domene primjene.



Izražavanje zahtjeva 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, teorija skupova i sl.)
 - notacija zasnovana na matematičkom konceptu. Najstrože definirana specifikacija. Korisnici je ne vole jer je ne razumiju.



Izražavanje zahtjeva sustava



- Strukturirani prirodni jezik
 - koristi se u zadavanju projekta.
- Jezici za opis oblikovanja (npr. SDL)
 - 1968 ITU study of stored program control systems
 - 1988 Blue Book SDL (SDL-88) Effective tools. Syntax well defined - formal definition. Language much as 1984.
 - 1999 SDL-2000, MSC-2000 podrška za OO modeliranje (UML), poboljšana podrška implementacija.
- Grafička notacija
 - UML Unified Modeling Language
- Matematička specifikacija
 - kripke strukture, logika, vremenska logika.





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.

Outputs CompDose – the dose in insulin to be delivered

Destination Main control loop

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

Izvor: Sommerville, I., Software engineering

🚾 ahtjevi sustava: specifikacija sučelja 🚉

- Potrebno je specificirati sučelje prema korisniku i prema drugim sustavima. Postoje tri tipa sučelja:
 - proceduralno sučelje
 - skup usluga kroz sučelje primjensko programsko sučelje engl. Application Programming Interface - API .
 - strukture podataka koje se izmjenjuju s drugim sustavima
 - npr. Entity-Relation-Attribute
 - predstavljanje podataka značenje pojedinih podataka
- Formalna notacija je vrlo efikasan način specifikacije sučelja. Npr.:

```
interface PrintServer {

// defines an abstract printer server
// requires: interface Printer, interface PrintDoc
// provides: initialize, print, displayPrintQueue, cancelPrintJob, switchPrinter

void initialize ( Printer p );
void print ( Printer p, PrintDoc d );
void displayPrintQueue ( Printer p );
void cancelPrintJob (Printer p, PrintDoc d);
void switchPrinter (Printer p1, Printer p2, PrintDoc d);
}//PrintServer

Oblikovanje programske potpore
```



Standardizacija zahtjeva



- IEEE 830 Recommended Practice for Software Requirements Specifications
 - http://standards.ieee.org/reading/ieee/std_public/description/se/index.html
- Definira općenitu strukturu dokumenta
 - uvod
 - opći opis sustava
 - specifičnosti zahtjeva
 - prilozi
 - indeks
- Primjer



Struktura dokumenta zahtjeva



Predložak

- predgovor
- uvod
- pojmovnik
- definicija zahtjeva korisnika
- arhitektura sustava
- specifikacija zahtjeva sustava
- model sustava
- razvoj sustava
- prilozi
- indeks



Zaključci o zahtjevima



- Zahtjevi postavljaju što sustav treba raditi i definiraju ograničenja u implementaciji i radu sustava.
 - ne postoji jedinstveni standard
- Korisnički zahtjevi su izjave na višoj apstraktnoj razini što bi sustav trebao raditi.
 - prirodni jezik, tablice, dijagrami
- Zahtjevi sustava su detaljne specifikacije o funkcijama sustava.
 - strukturirani prirodni jezik, specifični jezici oblikovanja, grafička notacija, matematička specifikacija
- Funkcionalni zahtjevi definiraju usluge koje sustav osigurava.
- Nefunkcionalni zahtjevi 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.





PROCESI INŽENJERSTVA ZAHTJEVA



Proces programskog inženjerstva



- Proces predstavlja strukturiran (organiziran) skup aktivnosti koji vodi nekom cilju.
- Proces inženjerstva zahtjeva je skup aktivnosti koje generiraju i dokumentiraju zahtjeve.
- Ciljevi:
 - opisati temeljne inženjerske aktivnosti i njihove odnose u generiranju i dokumentaciji zahtjeva.
 - upoznati se s tehnikama za izlučivanje i analizu zahtjeva.
 - opisati validaciju zahtjeva i ulogu recenzenta.
 - analizirati upravljanje zahtjevima (engl. requirements management) kao potporu procesu inženjerstva zahtjeva.

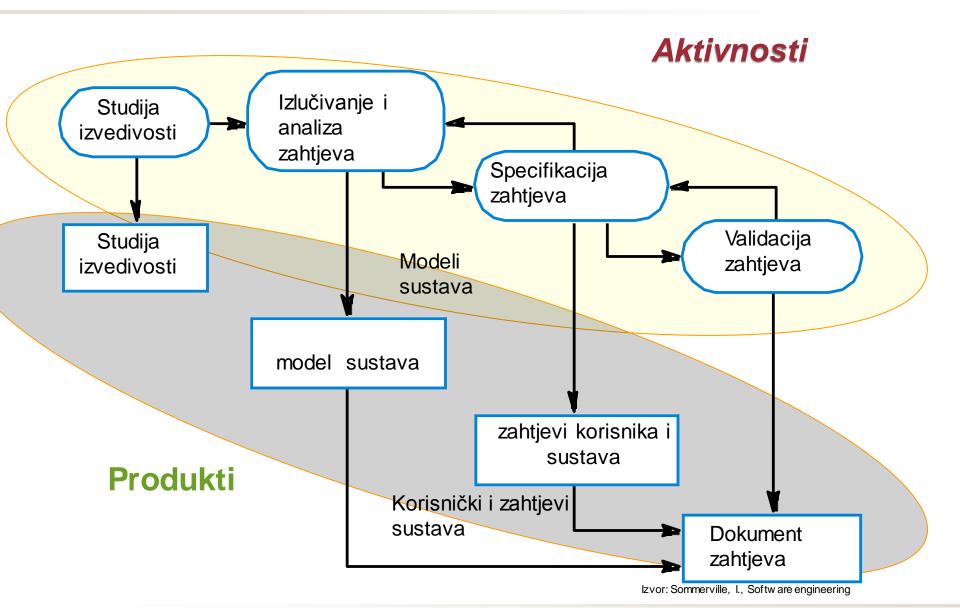


Procesi inženjerstva zahtjeva



- Procesi koji su u upotrebi u inženjerstvu zahtjeva razlikuju se ovisno o domeni primjene, ljudskim resursima i organizaciji koja oblikuje zahtjeve.
 - nema jedinstvenog procesa inženjerstva zahtjeva
- Dva uobičajena modela procesa inženjerstva zahtjeva:
 - klasični i spiralni
- Generičke aktivnosti zajedničke aktivnostima inženjerstva zahtjeva:
 - studija izvedivosti (engl. feasibility study)
 - izlučivanje zahtjeva (engl. requirements elicitation) i analiza i specifikacija zahtjeva
 - validacija zahtjeva
 - upravljanje zahtjevima

Wasični model procesa inženjerstva zahtjeva



Spiralni model procesa inženjerstva zahtjeva



- Tro-stupanjska aktivnost
 - specifikacija, validacija, izlučivanje.
- Promatra proces inženjerstva zahtjeva kroz iteracije.
- U svakoj iteraciji je različit intenzitet aktivnosti
 - u ranim iteracijama fokus na razumijevanju poslovnog modela.
 - u kasnijim modeliranje sustava zahtjeva
- Zahtjevi se u pojedinim iteracijama specificiraju s različitom razinom detalja.







Generičke aktivnosti inženjerstva zahtjeva:

STUDIJA IZVEDIVOSTI



Studija izvedivosti



- engl. feasibility study
- Na početku procesa inženjerstva zahtjeva određuje da li se predloženi sustav isplati
 - ulaz predstavljaju preliminarni zahtjevi
 - da li je vrijedan uloženih sredstava?
- Kratka fokusirana studija koja provjerava:
 - doprinose sustava ciljevima organizacije u koju se uvodi.
 - mogućnosti ostvarenja postojećom tehnologijom i predviđenim sredstvima.
 - mogućnosti integracije predloženog sustava s postojećim sustavima organizacije u koju se uvodi.



Provedba studije izvedivosti



- Temelji se na određivanju koje informacije su potrebne za studiju, prikupljanje informacija i pisanju izvješća.
- Pitanja za korisnike:
 - što ako se sustav ne implementira?
 - koji su trenutni problemi procesa organizacije?
 - kako bi predloženi sustav pomogao u poboljšanju procesa?
 - koji se problemi mogu očekivati pri integraciji novoga sustava?
 - da li je potrebna nova tehnologija ili nove vještine?
 - koje dodatne resurse organizacije traži implementacija novoga sustava?





Generičke aktivnosti inženjerstva zahtjeva:

IZLUČIVANJE I ANALIZA ZAHTJEVA



Izlučivanje i analiza zahtjeva



- engl. Requirements elicitation
- najznačajnija aktivnost u procesu inženjerstva zahtjeva
- Poznato i kao otkrivanje zahtjeva (engl. discovery).
- Uključuje stručno tehnički obrazovano osoblje koje u zajedničkom radu s kupcima i korisnicima:
- razjašnjava domenu primjene;
- definira usluge koje sustav treba pružiti;
- određuje ograničenja u radu sustava.
- Može uključivati:
- krajnje korisnike sustava, rukovoditelje, inženjere uključene u održavanje sustava, eksperte domene primjene, predstavnike sindikata i sl.
 - ≡ dionici (engl. stakeholders)



Izlučivanja zahtjeva



- Osnovne aktivnosti izlučivanja
 - razumjevanje domene primjene
 - Opće znanje područja
 - razumjevanje problema
 - Detalji specifičnog problema korisnika
 - razumjevanje konteksta
 - Interakcije sustava i ukupnog cilja
 - razumjevanje potreba i ograničenja korisnika

- Aktivnosti izlučivanja zahtjeva
 - Identificiranje aktora
 - Tipovi korisnika, uloge, sustavi
 - Utvrđivanje scenarija
 - Interakcija korisnika i sustava
 - Utvrđivanje obrazaca uporabe
 - Poboljšanje obrazaca uporabe
 - Utvrđivanje odnosa obrazaca
 - Utvrđivanje nefunkcionalnih zahtjeva
 - Performanse, sigurnost, ...



Metode izlučivanja zahtjeva



- Intervjuiranje kao metoda izlučivanja
- Scenarij kao metoda izlučivanja
- Izlučivanje i specificiranje zahtjeva obrascima uporabe (UML "use cases")
- Specificiranje dinamičkih interakcija u sustavu (UML sekvencijski dijagrami)
- Promatranje rada
- Izrada prototipa
- ...

🌑 iralni model izlučivanja i analize zahtjev 🎕

- 4 osnovne aktivnosti
- Izlučivanje/Otkrivanje zahtjeva
 - interakcija s dionicima s ciljem otkrivanja njihovih zahtjeva. Zahtjevi domene primjene se također definiraju na ovom stupnju. Izvori informacija su dokumenti, dionici, slični sustavi.
- Klasifikacija i organizacija zahtjeva
 - grupiraju se srodni zahtjevi i organiziraju u koherentne grozdove (klastere).
- Ustanovljavanje prioriteta i pregovaranje
 - zahtjevi se razvrstavaju po prioritetima i razrješuju konflikti.
- Dokumentiranje zahtjeva
 - zahtjevi se dokumentiraju i ubacuju u slijedeći ciklus spirale.





Analiza zahtjeva



Provjere

- neophodnosti zahtjeva
- konzistentnosti i kompletnosti
- mogućnosti ostvarenja

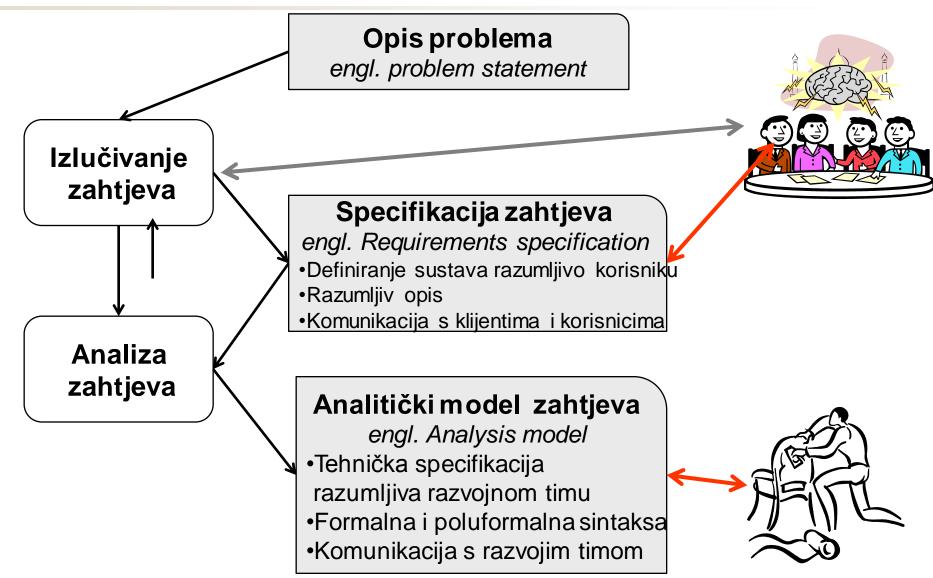
Problemi:

- dionici ne znaju što stvarno žele.
- dionici izražavaju zahtjeve na različite, njima specifične načine.
- različiti dionici mogu imati konfliktne zahtjeve.
- organizacijski i politički faktori mogu utjecati na zahtjeve.
- zahtjevi se mijenjaju za vrijeme procesa analize.
- pojavljuju se novi dionici uz promjenu poslovnog okruženja.



Rezultati procesa izlučivanja







Pogledi



- engl. Viewpoints
- Način strukturiranja zahtjeva tako da oslikavaju perspektivu i fokus različitih dionika
 - dionici se mogu razvrstati po različitim pogledima.
- Ova više-perspektivna analiza je značajna jer ne postoji jedan jedinstveni ispravan način u analizi zahtjeva sustava i omogućava razrješavanje konflikata.
- Tipovi pogleda:
 - pogledi interakcije
 - Ljudi i drugi sustavi koji izravno komuniciraju sa sustavom.
 - indirektni pogledi
 - Dionici koji ne koriste sustav izravno, ali utječu na zahtjeve.
 - pogledi domene primjene
 - Karakteristike domene i ograničenja koja utječu na zahtjeve.



Primjer: Bankomat



Bankomat

- čitač mag. Kartica
- tastatura, zaslon
- utor za umetanje omotnica
- spremnik novčanica
- pisač za ispis potvrda
- ključ za uključivanje/isključivanje
- komunikacija s bankom



Opis rada



- Posluživanje jednog korisnika
- Ubacivanje kartice + identifikacija PIN-om, podaci se šalju banci na validaciju tijekom svake transakcije
- Korisnik može obaviti jednu ili više transakcija
- Kartica se zadržava u bankomatu sve dok korisnik obavlja transakcije, nakon završetka kartica se vraća (postoji iznimke)



Usluge bankomata



- Korisnik može podići novce s računa kartice.
 Podizanje novac odobrava banka.
- Korisnik može uložiti novac na račun kartice (gotovina/ček)
 - korisnik upisuje uloženi iznos
 - operator ručno verificira iznos
 - banka odobrava prihvaćanje uplate
- Korisnik može prebacivati novce između računa
- Korisnik može pregledati stanje računa



Bankomat: dionici



- Dionici sustava bankomata:
 - bankovni klijenti
 - predstavnici drugih banaka
 - bankovni rukovoditelji
 - šalterski službenici
 - administratori baza podataka
 - rukovoditelji sigurnosti
 - marketing odjel
 - inženjeri održavanja sustava
 - sklopovlja i programske potpore
 - regulatorna tijela za bankarstvo



Bankomat: pogledi



Pogledi interakcije

- klijenti
- predstavnici banaka

Indirektni pogledi

- rukovoditelji
- osoblje zaduženo za sigurnost
- šalterski službenici
- administratori podataka
- odjel marketinga

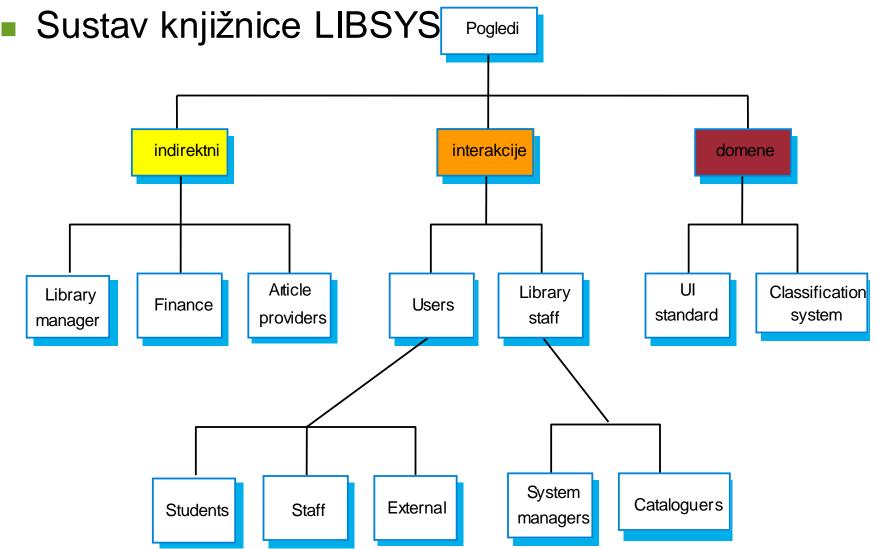
Pogledi domene primjene

standardi u komunikaciji između banaka



Primjer pogleda:LIBSYS





Izvor: Sommerville, I., Software engineering





Generičke aktivnosti inženjerstva zahtjeva:

METODE IZLUČIVANJA ZAHTJEVA

Metode u izlučivanju i analizi zahtjeva

- Intervjuiranje kao metoda izlučivanja
- Scenarij kao metoda izlučivanja
- Izlučivanje i specificiranje zahtjeva obrascima uporabe (UML "use_cases")
- Specificiranje dinamičkih interakcija u sustavu (UML sekvencijski dijagrami)



Intervjuiranje



- U formalnom i neformalnom intervjuiranju tim zadužen za inženjerstvo zahtjeva ispituje dionike o sustavu koji trenutno koriste te o novo predloženom sustavu.
- Tipovi intervjua:
- Zatvoreni intervju
 - odgovara se na skup prije definiranih pitanja.
- Otvoreni intervju
 - ne postoje definirana pitanja, već se niz pitanja otvara i raspravlja s dionicima.
- U praksi intervjui često ne daju dobre rezultate za zahtjeve domene primjene
 - inženjeri zahtjeva često ne razumiju specifičnu terminologiju domene
 - eksperti domene toliko poznaju te zahtjeve da ih ne artikuliraju dobro.



Scenariji



- Primjeri iz stvarnog života o načinu korištenja sustava.
- Sadržaj scenarija:
 - opis početne situacije.
 - opis normalnog/standardnog tijeka događaja.
 - opis što se eventualno može dogoditi krivo.
 - informaciju o paralelnim aktivnostima.
 - opis stanja gdje scenarij završava.
- Izlučivanje zahtjeva
 - dionici diskutiraju i kritiziraju scenarij



Primjer scenarija za sustav LIBSYS (1):



Početno stanje

 korisnik se prijavio u LIBSYS sustav, pronašao časopis u kojem se nalazi željeni članak.

Normalan rad

- korisnik odabire članak za kopiranje.
 - Sustav traži od korisnika informaciju o njegovim pravima (tip pretplate) ili načinu plaćanja.
 - Opcije plaćanja su kreditna kartica ili račun organizacije koja ima pretplatu.
- sustav traži da korisnik potpiše formular o pravima na kopiranje i ostali detaljima transakcije. To se daje LIBSYS sustavu.
- formular o pravima na kopiranje se provjerava, i ako je dozvoljeno članak u PDF formatu se prosljeđuje do korisničkog računala u sklopu LIBSYS sustava. Korisnik treba odabrati pisač i kopija članka se ispisuje. Ukoliko je članak tipa "samo za ispis", članak se briše sa korisničkog računala nakon što korisnik potvrdi da je ispis završen.



Primjer scenarija za sustav LIBSYS (2):



Greške u sustavu

- korisnik nije ispravno popunio formular o pravima na kopiranje. U tom slučaju formular se mora ponovo dati korisniku na ispravak. Ako je ponovljeni formular krivo ispunjen, zahtjev korisnika se odbacuje.
- sustav može odbaciti način plaćanja. Korisnikov zahtjev se odbacuje.
- prijenos članka na korisnikovo računalo nije ispravno izveden. Treba ponavljati dok prijenos ne bude uspješan ili dok korisnik ne prekine transakciju.
- članak nije moguće ispisati. Ako članak nije tipa "samo za ispis" drži ga se u radnom prostoru LIBSYS sustava, a iu suprotnom članak se izbriše i korisnik kreditira u visini cijene članka.

Paralelne aktivnosti

prijenos i obrada zahtjeva drugih korisnika LIBSYS sustava.

Završno stanje

korisnik i dalje prijavljen na sustav. Ako je članak "samo za ispis" briše se.



Metode izlučivanja zahtjeva



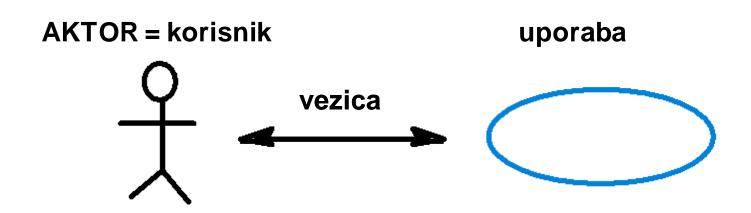
- Intervjuiranje kao metoda izlučivanja
- Scenarij kao metoda izlučivanja
- Izlučivanje i specificiranje zahtjeva obrascima uporabe (UML use cases)
- Specificiranje dinamičkih interakcija u sustavu (UML sekvencijski dijagrami)



Obrasci uporabe



- engl. Use cases
- Obrasci uporabe predstavljaju tehniku preuzetu iz UML standarda.
 Opisuju se AKTORI u interakciji s uslugama sustava.
- Skup obrazaca uporabe opisuje sve moguće interakcije sustava.
- Uz obrasce uporabe, dodatno se mogu koristiti i drugi dijagrami sekvenci za detaljan opis tijeka događaja.
- Tri temeljna elementa u modelima obrazaca uporabe su: obrasci uporabe, aktori i odnosi (relacije, engl. relations) među njima.





Modeliranje obrascima uporabe



- Model obrazaca uporabe je pogled koji ističe ponašanje sustava kako ga vide vanjski korisnici.
- Model obrazaca uporabe razdjeljuje funkcionalnost sustava u
 - transakcije ("obrasce uporabe")
 - razumljive korisnicima ("aktorima").
- Pogodno za modeliranje:
 - korisničkih zahtjeva.
 - scenarija ispitivanja sustava (engl. test scenarios).

Modeliranje dinamičkih interakcija u sustav<u>i</u>

- Metoda izlučivanja zahtjeva modeliranjem dinamičkih interakcija u sustavu
- Modeliranje ponašanja, engl. behavioral modeling
- Detaljniji razvoj i prikaz scenarija u izlučivanju, analizi i dokumentiranju zahtjeva:
 - obrasci uporabe identificiraju individualne interakcije u sustavu.
 - dodatne informacije u inženjerstvu zahtjeva uz obrasce uporabe slijede iz dijagrama interakcije koji pokazuju aktore involvirane u interakciji, entitete (objekte, instancije) s kojima su u interakciji i operacije pridružene tim objektima.
- Osnovni tipovi dijagrama interakcija:
 - sekvencijski
 - kolaboracijski.



Interakcije



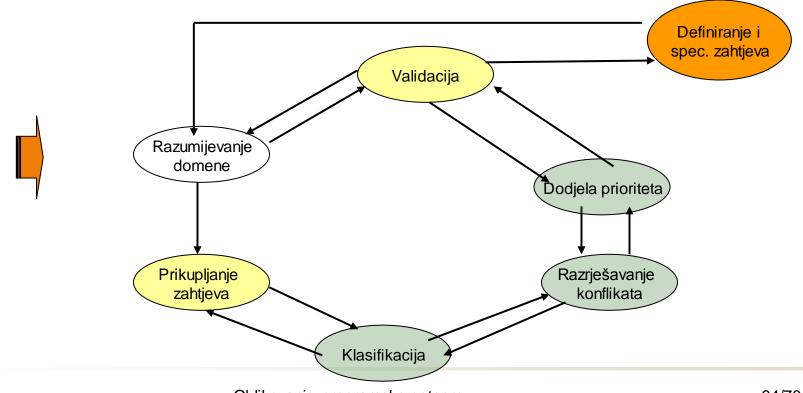
- Skup komunikacija između instancija elemenata sustava.
 - npr. u objektno usmjerenoj arhitekturi:
 - Pozivi operacija (procedura)
 - Kreacije instancija (objekata).
 - Destrukcije instancija (objekata).
 - komunikacije su djelomično uređene u vremenu.
- Modeliranje interakcije omogućava:
 - specificiranje međudjelovanje između elemenata sustava.
 - olakšava identifikaciju sučelja.
 - utvrđivanje potreba za raspodjelom zahtjeva.



Analiza zahtjeva



- Otkrivanje funkcije sustava, struktura sustava, ponašanje sustava.
- Cilj analize zahtjeva je utvrđivanje problema, nekompletnosti, i nejednoznačnosti u izlučenim zahtjevima.
 - dionici ih razrješavaju pregovorima.
- Analiza je isprepletena s procesom izlučivanja zahtjeva





Konflikti i utvrđivanje prioriteta



- Utvrđivanje interakcije zahtjeva
 - pomaže pronalaženju problema
 - tablica
- Zahtjevi visokog prioriteta engl. Core requirements
 - obavezno razmotriti pri analizi, oblikovanju i impl.
 - neophodno demonstrirati klijentu pri preuzimanju
- Srednji prioritet engl. Optional requirements
 - obavezno razmotriti pri analizi i oblikovanju
 - uobičajeno se implementira u drugim iteracijama projekta
- Nizak prioritet engl. Fancy requirements
 - analizira se u obliku naprednih mogućnosti
 - pogodno za prikaz mogućnosti budućeg razvoja





Generičke aktivnosti inženjerstva zahtjeva:

VALIDACIJA ZAHTJEVA



Validacija zahtjeva



- Cilj validacije je pokazati da dokument zahtjeva predstavlja prihvatljiv opis sustava koji naručitelj doista želi.
 - naknadno ispravljanje pogreške u zahtjevima može biti višestruko skuplje od ispravljanja pogrešaka u implementaciji
- Tehnike validacije:
 - recenzija zahtjeva
 - sistematska ručna analiza
 - izrada prototipa
 - provjera na izvedenom sustavu
 - generiranje ispitnih slučaja
 - razvoj ispitnih sekvenci za provjeru zahtjeva
- Rezultati validacije
 - lista problema (pojašnjenja, nedostaci, konflikti, neostvarivost)
 - lista utvrđenih akcija za razrješavanje problema



Elementi provjere zahtjeva



Razumljivost

Da li je dokument jasno napisan?

Kompletnost

Da li sustav uključuje sve funkcije koje je korisnik tražio?

Konzistencija

Da li postoji konflikt u zahtjevima?

Valjanost

Da li sustav osigurava funkcije koje podupiru potrebe korisnika?

Realnost

Da li se sve funkcije mogu implementirati uz danu tehnologiju i proračun?

Provjerljivost

Da li se svi zahtjevi mogu provjeriti?

Sljedivost

Da li je naveden izvor dokumenta i razlozi uvrštavanja zahtjeva?

Adaptabilnost

Mogu li se zahtjevi mijenjati bez velikog utjecaja na druge zahtjeve?





Generičke aktivnosti inženjerstva zahtjeva:

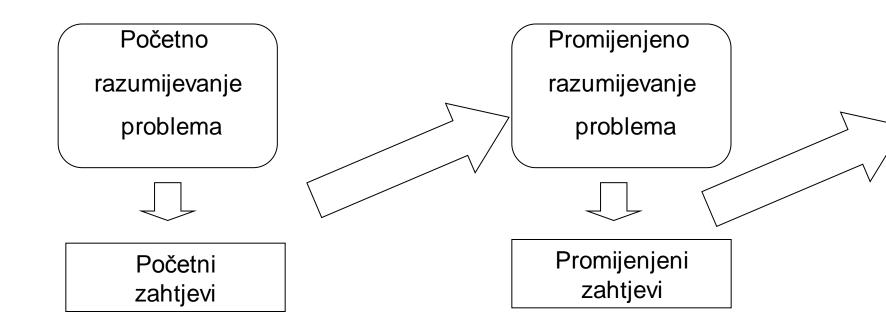
UPRAVLJANJE PROMJENAMA ZAHTJEVA



🍉 Upravljanje promjenama zahtjeva



- Upravljanje ili rukovanje engl. requirements management
- Promjene nastupaju zbog promijenjenog modela poslovanja, boljeg razumijevanja procesa tijekom razvoja ili konfliktnim zahtjevima u različitim pogledima.
- Evolucija zahtjeva:





Klasifikacija promjena zahtjeva



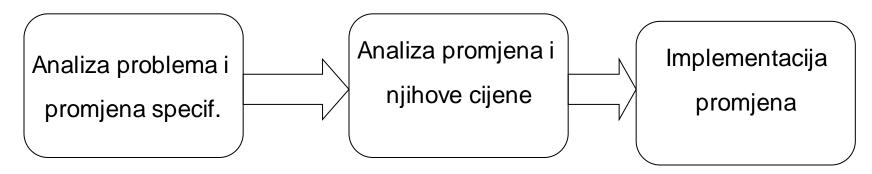
- Okolinom promijenjeni zahtjevi
 - promjena zbog promjene okoline u kojoj organizacija posluje (npr. bolnica mijenja financijski model pokrivanja usluga).
- Novonastali zahtjevi
 - zahtjevi koji se pojavljuju kako kupac sve bolje razumije sustav koji se oblikuje.
- Posljedični zahtjevi
 - zahtjevi koji nastaju nakon uvođenja sustav u eksploataciju, a rezultat su promjena procesa rada u organizaciji nastalih upravo uvođenjem novoga sustava
- Zahtjevi kompatibilnosti
 - zahtjevi koji ovise o procesima drugih sustava u organizaciji; ako se ti sustavi mijenjaju to traži promjenu zahtjeva i novo uvedeni sustav



Upravljanje procesom promjena



- Planiranje:
 - identifikacije zahtjeva (individualno identificiranje zahtjeva).
- Upravljanje procesom promjena
 - proces koji slijedi kada se utvrdi potreba za promjenom



- Sljedivost
 - koje informacije su potrebne za povezivanje zahtjeva
- Izbor CASE alata
 - automatiziranje skladištenja, unošenja promjena i povezivanje dokumenata





SOCIJALNI I ORGANIZACIJSKI ČIMBENICI



Socijalni i organizacijski čimbenici



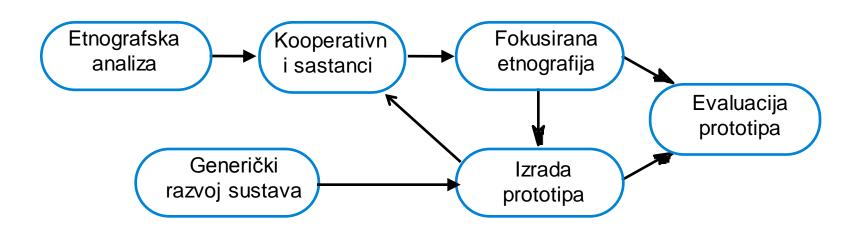
- Programski produkti se uvijek koriste u socijalnom i organizacijskom kontekstu.
 - to uvelike utječe, a ponekad i dominira na zahtjeve sustava.
- Socijalni i organizacijski čimbenici nisu jedinstven pogled, već utjecaj na sve poglede.
 - oblikovanje programske potpore mora to uvažiti, ali trenutno ne postoji sistematski postupak kako se to može uključiti u analizu zahtjeva.
- Ne postoje jednostavni modeli za opis obavljanja nekog posla.
 - ljudima je često teško precizno opisati što rade!?
 - postojeći modeli zasnovani na prošlim, a ne obrascima ponašanja na novom poslu.
 - djelomično se tome može doskočiti izradom prototipa, te praćenjem rada na njemu.
- Za razumijevanje procesa korisna znanja iz područja društvenih znanosti



Etnografija



- Kvalitativno promatranje i opis ponašanja ljudi u društvu.
- Zahtjevi izvedeni temeljem istraživanja kako ljudi stvarno rade
 - ne kako definicija poslovnog procesa formalno propisuje!!!
- Zahtjevi izvedeni temeljem kooperacije i uzimajući u obzir aktivnosti drugih ljudi.
- Fokusirana etnografija (etnografija + prototip):



Izvor: Sommerville, I., Software engineering



Zaključak



- Proces inženjerstva zahtjeva uključuje:
 - studiju izvedivosti, izlučivanje zahtjeva i analizu;
 - specifikaciju zahtjeva i rukovanje (upravljanje) zahtjevima.
- Izlučivanje i analiza zahtjeva je iterativan proces koji uključuje razumijevanje domene primjene, prikupljanje, klasifikaciju, strukturiranje, sastavljanje prioriteta i validaciju zahtjeva.
 - sustavi imaju više dionika s različitim zahtjevima.
 - socijalni i organizacijski čimbenici utječu na zahtjeve sustava.
- Validacija zahtjeva bavi se valjanošću, konzistentnošću, kompletnošću, realizmom u izvedbi i provjerljivošću.
- Promjene načina poslovanja nužno mijenjaju zahtjeve
 - kontinurani proces
- Rukovanje zahtjevima uključuje planiranje i upravljanje promjenama (engl. Requirements management) zahtjevima.



Stvarnost









lzvor: http://www.projectcartoon.com



Diskusija

