

Zavod za primijenjeno računarstvo

Prof.dr.sc. Krešimir Fertalj



Razvoj informacijskih sustava



Zaštićeno Creative Commons licencom, <http://creativecommons.org>

O predmetu

☐ Kratki opis

- Obrađuju se postupci i tehnike strukturiranog i objektno usmjerenog pristupa razvoju informacijskih sustava namijenjenog krajnjem korisniku. Izlažu se tehnike analize i oblikovanja, analiza izvodljivosti, pristupi izradi informacijskog sustava, procedure provjere sustava, instalacija i konverzija te postupci održavanja sustava. Razmatra se utjecaj informatizacije na organizaciju.

☐ Kompetencije

- Predmet pruža znanja potrebna za sudjelovanje u svim fazama razvoja informacijskog sustava. Studenti će sudjelovati u projektiranju složenog, po mogućnosti stvarnog, informacijskog sustava. Studenti će biti osposobljeni za proučavanje postojećeg sustava, definiranje zahtjeva na novi sustav, predlaganje i evaluaciju alternativnih rješenja, oblikovanje složene programske podrške, izradu projektne dokumentacije te provedbu postupaka uvođenja i održavanja i potpore informacijskog sustava.

☐ Opterećenje = 3 h tjedno + 15 LiV + 90 samostalan rad

☐ ECTS = 5



FER \ Fertalj: Razvoj informacijskih sustava

2

Izvedba

☐ Predavači & izvođači

- prof. dr. sc. Krešimir Fertalj
- mr.sc. Boris Milašinović
- Marija Katić, prof.

☐ Administracija

- Zavod za primijenjeno računarstvo, III kat zgrada D
- tajnica Sonja Majstorović, Tel: 6129-915

☐ Obavijesti

- web-stranica predmeta <http://www.fer.hr/predmet/ris>
- ulaz u Zavod za primijenjeno računarstvo (3. kat zgrade D)

☐ Konzultacije i primjedbe

- ZPR, D/III, sjeverozapadno krilo
- e-mail nastavniku ili suradniku
 - To: ime.prezime@fer.hr
 - Subject: [RIS]
 - U potpisu navesti puno ime i prezime te JMBAG



FER \ Fertalj: Razvoj informacijskih sustava

3

Materijali

☐ Nastavni materijali -

- folije s predavanja, prateći primjeri
 - <http://creativecommons.org/licenses/by-nc-sa/2.5/hr/>
- podsjetnici, "šalabahteri", e-knjiga,
- <http://www.fer.hr/predmet/ris> - formalna stranica
- <http://bob.zpr.fer.hr/predmeti/RIS/> - kolaboracijski portal



☐ Literatura:

- J.A. Hoffer, J.F. George, J.S. Valacich: Modern Systems Analysis and Design, 4th ed., Prentice Hall, 2004. ISBN 0131454617
- A. Dennis, B.H. Wixom, D.Tegarden: Systems Analysis & Design: An Object-Oriented Approach with UML, Wiley & Sons, 2002. ISBN 0471348066
- G. B. Shelly, T. J. Cashman, H. J. Rosenblatt: Systems Analysis and Design, 5th ed, Course Technology, 2003. ISBN 0789566494



FER \ Fertalj: Razvoj informacijskih sustava

4

Sadržaj

- ☐ Osnove informacijskih sustava. Uvod u analizu i projektiranje sustava.
- ☐ Strategija organizacije. Identifikacija, selekcija, pokretanje i planiranje projekata. Modaliteti i alternative izgradnje.
- ☐ Prikupljanje informacija i definiranje problema. Specifikacija zahtjeva.
- ☐ Strukturirana analiza i dizajn. Modeliranje funkcija, procesa i podataka.
- ☐ Objektno usmjerena analiza i dizajn.
- ☐ Odabir arhitekture programske podrške.
- ☐ Oblikovanje korisničkog sučelja i izvješća.
- ☐ Konstrukcija, integracija i provjera ispravnosti te dokumentiranje sustava.
- ☐ Uvođenje sustava u primjenu. Postupci održavanja. Potpora korisnicima.
- ☐ Računalom podržano inženjerstvo. Obrnuto inženjerstvo. Preoblikovanje programske podrške.
- ☐ Metodologije razvoja informacijskih sustava.
- ☐ Osnove upravljanja projektom razvoja informacijskog sustava.
- ☐ Sustavi za upravljanje poslovanjem.

- ☐ Pozvano predavanje tvrtki koje se bave razvojem IS.



Vrednovanje rada



- ☐ Sudjelovanje na predavanjima
 - cijeni se aktivnost prema subjektivnom dojmu predavača
 - minimum je dolazak (i ostanak) na barem pola predavanja
- ☐ Domaće zadaće – individualni projekti
 - realizacija individualnog projekta (planiranje, analiza, projektiranje, ...)
 - isporuka rezultata obavlja se u tjednu zadavanja do nedjelje, 23:59:59
- ☐ Test na računalu, AHyCo blic
 - ocjenjivanje znanja "teorije"
 - 3 testa maksimalnog doprinosa po 10% bodova
 - svako od 20 generiranih pitanja vrijedi 0,5 % bodova
 - pogrešan odgovor odnosi 0,1% bodova
 - trajanje testa 60 minuta
 - generirani test se rješava pod nadzorom
 - student se treba samostalno i na vrijeme pripremiti
 - provjera korisničkog imena i lozinke, pokusni rad sa sustavom Ahyco



Međuispit(i), (ponovljeni) završni ispit

- ☐ Pismeni ispiti
 - rješavanje problema
 - znanje ključnih teorijskih koncepata



- ☐ Međuispit I
 - strukturirana analiza i oblikovanje
 - u tjednu 06.04.2009. – 10.04.2009.
- ☐ Međuispit II
 - objektno-orijentirana analiza i oblikovanje
 - u tjednu 18.05.2009. - 22.05.2009.
- ☐ Završni ispit
 - izrada, uvođenje, upravljanje projektom
 - u tjednu 22.06.2009. – 26.06.2009.
- ☐ Ponorljveni završni ispit
 - u tjednu 06.07.2009. – 10.07.2009.
 - mogu pristupiti samo studenti koji nisu ostvarili prolaz na završnom ispitu



Ocjenjivanje

- ☐ Za pozitivnu ocjenu (prolaz) treba ostvariti barem 50,00 % bodova svake pojedine komponente u tablici
 - Raspodjela prolaznih ocjena provodi se prema Gaussovoj raspodjeli
- ☐ Nemogućnost ispunjenja obveza treba dojaviti najkasnije tjedan dana od dana kada je nemogućnost nastupila
 - nema ispričnice "za cijeli semestar" na dan provjere znanja!!!

Aktivnost	Pojedinačno	Puti	Bodova	Udio
Predavanja	7	1	7	7.00%
Domaće zadaće	2-4	6	18	18.00%
Test (blic)	10	3	30	30.00%
I međuispit	15	1	15	15.00%
II međuispit	15	1	15	15.00%
Završni ispit	15	1	15	15.00%
		Ukupno	100	100.00%



Ostalo

☐ Ankete

- anonimne, služe za analizu i unaprjeđenje nastave
- ulazna anketa - tijekom prva tri tjedna predavanja
- završna anketa - tijekom zadnja tri tjedna predavanja
- obrazac će biti objavljen na webu FER – Ahyco

☐ Laboratorij programskog inženjerstva i informacijskih sustava 2

- Ukupno tjedno opterećenje 3 h
- Ukupno opterećenje temeljem predmeta RIS – 15 h
- Izrada prototipa projektiranog sustava

☐ Preporuča se dolazak na predavanja

- na kojima će se komentirati primjeri stvarnih projekata iz prakse, a što iz više razloga ne može biti stavljeno na folije ili stavljeno na web.

☐ Dostupni Primjeri: Posebno Označeni Na Folijama

☐ ???



Informacijski sustav



Podatak, informacija i znanje

☐ Informacija općenito

- obavijest, podatak o nečemu, predana novost ili znanje
- formalno, kolekcija simbola koji mogu imati određeno značenje

☐ Podatak, informacija i znanje

- podatak je sirova činjenica koja predstavlja neku istinu iz stvarnog svijeta
 - pojedinačni podaci sami za sebe znače malo ili nemaju neko značenje
- informacija je interpretacija podatka - pročišćen, organiziran i obrađen podatak u smislenom kontekstu
 - informacija je subjektivnog značenja, u kontekstu primatelja
- znanje se gradi na temelju novih informacija koje se nadovezuju na postojeće znanje
 - isti podaci mogu biti različito interpretirani od strane različitih ljudi u ovisnosti o njihovom znanju
- primjer:

-130000	???
-130000 HRK	?
na računu	!
broj 1313 (tj. Mojem)	!!!

☐ Informacijska znanost, obavijesna znanost → informatika (informatics, information theory).



Sustav

☐ Sustav = uređeni poredak međuzavisnih komponenti povezanih za postizanje zajedničke svrhe ili cilja.

☐ Fizički sustav – materijalne naravi

- skup jedinica, organizacijski ujedinjenih u cjelinu (npr. poduzeće)
- skup dijelova, povezanih općom funkcijom (npr. živčani)

☐ Apstraktni sustav – konceptualni, proizvod ljudske mašte

- oblik društvene organizacije (npr. državni)
- oblik, način ustrojstva, organizacija nečega (npr. izborni)
- uvjetovan planskim, pravilnim rasporedom dijelova (npr. sistem rada)



Elementi sustava

❑ Komponente - podsustavi

- nerazdvojni dijelovi ili agregacije dijelova koje pripadaju sustavu
 - npr. Nabava, Proizvodnja, Prodaja
 - npr. Studentska služba, Zavodi, Financijska služba
- mogu biti zamijenjeni bez potrebe za promjenom čitavog sustava (!?)
- organizacija, interakcija, međuzavisnost, integriranost

❑ Granica

- definira sadržaj i domašaj sustava

❑ Okolina

- sve što je izvan granica sustava, ali se još uvijek tiče sustava
- npr. MZOŠ i studentske udruge kao okolina visokog učilišta

❑ Ulazi i izlazi

- sve što ulazi u sustav iz okoline ili ga napušta

❑ Sučelja

- veze-dirališta između sustava i njegove okoline (zaštita, filtri)

❑ Ograničenja

- unutarnji i vanjski čimbenici koji određuju i ograničavaju funkcioniranje sustava
- npr. ograničenja prostora ili ljudstva, pravilnici, zakonske odredbe



Informacijski sustav

❑ Informacijski sustav (Information System) = skup međusobno povezanih komponenti koji prikuplja, obrađuje, pohranjuje i pruža informacije potrebne za obavljanje poslovne zadaće

- obavijesni sustav, informacijski sistem
- sustav za upravljanje sustavom ljudskih aktivnosti [Checkland, 1980]

❑ Svrha

- prikupljanje i pružanje informacija korisnicima u jednoj ili više organizacija → organizacijski IS (poslovni IS)

❑ Podrazumijeva

- sustave podržane računalom → računalni ("kompjuterizirani", "kompjuterski")
- sustave koji se ne oslanjaju na računala, ali obrađuju informacije

❑ Korisnici

- poslovodstvo, djelatnici (zaposlenici, osoblje), klijenti ...



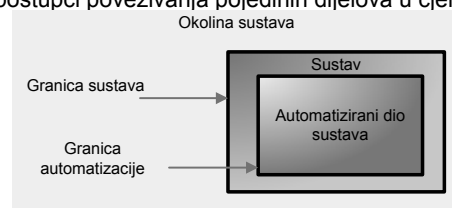
Informacijski sustav podržan računalom

❑ Informacijski sustav podržan računalom

- Uređenje ljudi, podataka, procesa, sučelja, poveznica (mreža) i tehnologije koji djeluju međusobno sa svrhom podrške i poboljšanja svakodnevnih poslovnih aktivnosti (obrada podataka), kao i podrške rješavanju problema i donošenju odluka (informacijski servisi). [Whitten et. al, 2000]

❑ Komponente IS

- programska oprema (software) – sistemska, namjenska
- računalna oprema (hardware) – računala i periferije
- osobe (lifeware) - korisnici, informatičari
- mrežna infrastruktura (netware) - mrežna oprema
- organizacija (orgware) – postupci povezivanja pojedinih dijelova u cjelinu
- ...



Poslovni i informacijski sustav

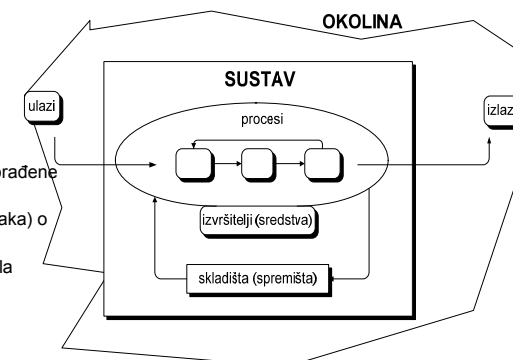
❑ Poslovni sustav

- materijalni ulazi i izlazi (sirovine, energija, proizvodi) i informacijski tokovi (poruke, dokumenti, ...)
- Ulaz u neki PS predstavlja izlaz iz nekog drugog PS
- procesi: obrada, prerada, proizvodnja
- Povratne veze - usporedba plana i realizacije
- izvršitelji: osobe, strojevi, alati
- skladišta: spremišta materijala

- I(P)S: nabave, prodaje, proizvodnje, financija, ljudskih resursa i plaća, osnovnih sredstava
- IS Subvencionirane Prehrane (ISSP)
- Integrirani PIS Visokih Učilišta (IPISVU)
- IS Personalnog Upravljanja MORH

❑ Informacijski sustav

- podsustav poslovnog sustava
- ulazi i izlazi: ulazne informacije, obrađene informacije
- procesi: obrada informacija (podataka) o stanjima stvarnog sustava
- izvršitelji: osobe, programi, računala
- skladišta: spremišta informacija (podataka)



Slojevi poslovnih informacijskih sustava

- ❑ **Transaction Processing System (TPS), Data Processing System**
 - evidencija i obrada podataka o poslovnim transakcijama
 - pr. uplate i isplate po računima, upis predmeta, prodaja, narudžba ili zaprimanje robe
- ❑ **Management Information System (MIS) - upravljački**
 - obrada transakcijskih podataka
 - izrada izvješća potrebnih za upravljanje i nadzor poslovanja
- ❑ **Decision Support System (DSS) – za potporu odlučivanju**
 - odlučivanje na temelju podataka iz različitih izvora – transakcijskih sustava ali i povijesnih podataka
 - interaktivno okruženje za donošenje odluka (što-ako, analiza hijerarhijski strukturiranih podataka – drill down, grafički modeli poslovnih procesa)
- ❑ **Executive Information System (EIS) – (pod)varijanta DSS za izvršne rukovoditelje**
 - koristi se za donošenje strateških odluka
 - informacije dolaze iz IS organizacije ali i vanjskih izvora, npr. novosti o konkurenciji, ekonomska predviđanja, izvješća s burza dionica



Upravljanje IS

❑ Upravljanje i razine IS

Informacijski sustav	Informacije, kada	Korisnici, tko	Upravljanje
transakcijski	obrada podataka, dnevno	niže poslovodstvo	operativno
upravljački	zbirne, periodički	srednje poslovodstvo	taktičko (trendovi)
za potporu odlučivanju	sintetizirane, “ad hoc”	više poslovodstvo, uprava	strategijsko (“što ako”,...)



Ostale vrste IS

- ❑ **Expert System (ES) – ekspertni sustav**
 - sustav pravila i mehanizam zaključivanja
 - ugrađena stručna znanja za korištenje od strane manje
- ❑ **Office Automation System (OAS), Office support systems (OSS)**
 - potpora uredskom poslovanju
 - upravljanje elektroničkom poštom, kolanje dokumenata, dijeljenje kalendara
- ❑ **Group Support System (GSS), Groupware**
 - potpora grupnom radu
 - kolaboracija, obavješćivanje, zaključavanje dokumenata
- ❑ **Supply Chain Management (SCM)**
 - upravljanje lancem nabave
 - integrira razvoj proizvoda, nabavu proizvoda, proizvodnju i upravljanje zalihama.
- ❑ **Customer relationship management (CRM)**
 - upravljanje odnosom s kupcima
 - podržava marketing, prodaju i servis, uključujući interakciju s korisnicima



Primjer poslovnog IS (1)

- ❑ **Financijsko računovodstvo**
 - Upravljanje glavnom knjigom.
 - Bilanca i Račun dobiti i gubitka (financijski izvještaji).
 - Analitika kupaca.
 - Analitika dobavljača.
 - Računovodstvo osnovnih sredstava.
 - Računovodstvo zaliha.
 - Porezno računovodstvo
- ❑ **Upravljačko računovodstvo (Kontroling)**
 - Računovodstvo troškovnih centara.
 - Računovodstvo profitnih centara.
 - Računovodstvo internih naloga.
 - Računovodstvo profitabilnosti.
- ❑ **Prodaja i distribucija**
 - Upravljanje nalogima za prodaju.
 - Upravljanje ugovorima.
 - Fakturiranje.
- ❑ **Nabava, skladištenje i logistika**
 - Zahtjevnice.
 - Obrada narudžbenica.
 - Evidencija primki materijala.
 - Ovjera faktura (likvidacija).
 - Upravljanje ugovorima.
 - Upravljanje zalihama i skladištem, uključujući inventuru i fizički promet robe.
- ❑ **Upravljanje ljudskim potencijalom**
 - Kadrovska evidencija.
 - Obračun plaća.
- ❑ **Proizvodnja**
 - Planiranje proizvodnje.
 - Izvršenje proizvodnje.
 - Proizvodnja po narudžbi.
 - Procesna proizvodnja.
- ❑ **Upravljanje kvalitetom**
 - Ispitivanje kvalitete.
- ❑ **Izvještaji**



Primjeri IS (2), (3)

❑ IS biljnih vrsta

- Taksonomija
- Herbar
- Nalazišta
- Narodna imena
- Sinonimi
- Literatura
- Ekonomska upotrebljivost
- Zemljopisna rasprostranjenost
- Ugroženost
- ...

❑ IS personalnog upravljanja

- Osnovni podaci o osobama (osobine, civilna i vojna povijest)
- Ustroj (postrojbe, dužnosti, ustrojna mjesta)
- Tijek službe (prijam, služba, posebna stanja, izlaz)
- Dodjela činova i redova, ocjenjivanje
- Dodjela odličja i medalja
- Školovanje (povijest, izobrazba u HV)
- Evidencija prekršaja, mjere
- Ostalo (naknade, doplatci, ...)



Značajke informacijskih sustava

❑ Značajke informacijskih sustava

- složena okolina, koju je teško u potpunosti definirati
- složeno sučelje prema okolini, koje uključuje različite ulaze i izlaze
- složene veze između ulaza i izlaza (strukturno, algoritmički)
- obujam i složenost podataka

→ Problem projektiranja, izrade i održavanja

❑ Važnost IS

- Informacija je postala upravljački resurs jednake važnosti kao što su vlasništvo (osnovna sredstva), ljudski resursi i kapital.
 - IS sadrži/predstavlja znanje (know-how) organizacije koju podržava.
 - IS i aplikacije pokazuju se prijeko potrebnim za održanje konkurentnosti ili postizanje kompetitivnog prestiža organizacije.
- Organizacije u kojima se IS primjenjuju visoko su ovisne o stalnoj raspoloživosti IS kroz dulje vrijeme.



Projektiranje i izgradnja IS



Razvoj informacijskih sustava

❑ The CHAOS Report [Standish Group, 1994], <http://www.standishgroup.com>

- Prosječni trošak projekta
 - velike kompanije: 2,32 M\$
 - srednje kompanije: 1.33 M\$
 - male kompanije: 434 K\$
- Prosječno prekoračenje troškova 189%
- Prosječno prekoračenje rokova 222%
- Projekti završeni na vrijeme, u okviru predviđenih sredstava, sa svim predviđenim funkcijama - 16.2%
- Projekti završeni i u funkciji, ali uz veće troškove, dulje trajanje i/ili reduciranu funkcionalnost - 52.7%
- Prekinuti projekti - 31.1%

❑ Standish Group, 2002:

- 34% uspješnih projekata
- 17% potpunih neuspjeha

❑ Ciljevi razvoja IS

- izgraditi sustav koji radi i koji je pouzdan, unutar zadanih granica
- izgraditi sustav koji zadovoljava poslovne ciljeve, prema zahtjevima korisnika
- izgraditi sustav u prihvatljivom vremenu i po opravdanoj cijeni



Neki primjeri neuspješnih projekata/sustava

❑ London Ambulance System (1992)

- Nakon uvođenja sustav se 2 puta "raspao" radi niza pogrešaka, naročito onih zbog upravljanja. Neposredni trošak bio je relativno nizak (£9m), ali se vjeruje da su neki ljudi umrli jer se do njih nije stiglo na vrijeme.
 - dispečerski centar i ambulantne jedinice (vozila) na terenu
 - preopterećenje već u normalnom pogonu
 - produljenje odziva u hitnim slučajevima na 7h, radi krutosti sustava
 - niz propusta nabave, projektiranja, ugradnje i primjene sustava

❑ Taurus (1993)

- Projekt sustava automatiziranih transakcija Londonske burze prekinut je nakon 5 godina razvoja i troška od £75m te posljedični gubitak klijenata od £450m. Ukupni gubici smatra se da su neizračunljivi.
 - strateški projekt željeno najveće burze na svijetu, "de-materijalizacija" dionica
 - složenost, prostorno raspršene ekipe, loša arhitektura, nepridržavanje metodologije, loše upravljanje projektom i kontrola kvalitete

❑ Denver Airport (1994)

- Nepouzdanost softvera za upravljanje prtljagom uzrokovala je odgodu otvaranja zračne luke u trajanju od 16 mjeseci uz troškove od 1.1 M\$/dan.
 - uočeni mehanički problemi rješavani su dodavanjem upravljačke opreme
 - sustav je prvotno pušten u pogon unatoč saznanju da ne radi dobro



Uzroci neuspjeha

❑ Informacije o neuspješnim projektima u RH se rjeđe objavljuju. Među najčešćim uzrocima su [Kalpić, Fertalj & Mornar, 2001]

- Loša organizacija i vođenje projekata
 - oslonac na vanjske voditelje i savjetnike, delegirano upravljanje projektima, nerealno planiranje i pretjerani optimizam, formalno izvješćavanje o napretku, formalni nadzor nad projektom, podcijenjena uloga vlastitih stručnjaka
- Loša izvedba projekata
 - neodgovarajuća analiza sustava, pogreške u dizajnu i kontroli kvalitete, neodgovarajuća CASE pomagala i krivo korištenje, pa čak i svojevrsna zloraba CASE pomagala

❑ Mnogi sustavi su propali ili su bili odbačeni jer su se izvođači trudili napraviti lijepa programska rješenja a nisu razumjeli suštinu organizacije i poslovanja.



Poboljšanje uspješnosti IS

❑ Poboljšanje uspješnosti IS

- Projektiranje IS
- Planiranje, upravljanje provedbom, praćenje napretka
- Uključivanje korisnika
 - Korisnik poznaje(?) poslovni proces i zna(?) odrediti potrebe.
 - Informatičar upoznaje(?) poslovanje i zna(?) kako izraditi IS.
 - Važno je da u procesu izgradnje sudjeluje i poslovodstvo, da bi se upoznalo sa stvarnim mogućnostima i koristima uvođenja IS, posebice jer donosi konačne (za poduzeće sudbonosne) odluke.

❑ Sažetak načela razvoja informacijskih sustava

- Korisnici i vlasnici sustava moraju biti aktivno uključeni
- Treba koristiti pristup koji vodi k rješavanju problema
- Uspostaviti faze i aktivnosti
- Uspostaviti standarde za konzistentan razvoj i dokumentiranje
- Opravdati izgradnju sustava kao kapitalnu investiciju
- Ne oklijevati ako treba revidirati doseg ili otkazati projekt
- Projektirati za rast i promjene



Programska potpora

❑ Programska oprema/podrška/potpora, softver (software)

- dio računalnog sustava koji nema fizikalnih dimenzija
- opći pojam za sve vrste programa, programskih jezika itd
- Skup elemenata ili objekata u jedinstvenoj "konfiguraciji" koju čine računalni programi + podaci + dokumentacija
- svojstva:
 - složenost, podložnost pogreškama,
 - ne troši se, teško mjerljiv,
 - stari, dugo se koriste,
 - lako se kopira (zajedno s pogreškama)

❑ Primijenjena programska potpora = Računalna aplikacija (application)

- namjenski program, primjenska programska oprema
- računalom podržano rješenje jednog ili više poslovnih problema ili potreba

❑ Informacijski sustav = sustav aplikacija za upravljanje ljudskim aktivnostima



Programsko inženjerstvo

□ Programsko inženjerstvo (software engineering)

- "Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is **the application of engineering to software**. (2) The study of approaches as in (1)." [IEEE Std 610.12-1990, 1994]
 - IEEE Std 610.12-1990 "Standard Glossary of Software Engineering Terminology", IEEE, New York, 1994, p. 67.
 - IEEE = Institute of Electrical and Electronic Engineers, <http://www.ieee.org>
- **sistematičan, discipliniran i mjerljiv pristup razvoju, primjeni i održavanju softvera**
- **primjena inženjerskog pristupa na programsku opremu**
- Programsko inženjerstvo je inženjerska disciplina koja obuhvaća sve aspekte izrade programske opreme. [Sommerville, 2004]
- „Programsko inženjerstvo... ima za cilj ekonomičan razvoj visokokvalitetne programske opreme“ [Pagel / Six 1994, str. 49]

□ Područje programskog inženjerstva

- poslovi kojima se oblikuje i razvija programska oprema
- sustavna primjena prikladnih alata i tehnika na čitav proces razvoja programske potpore



Informacijsko inženjerstvo

□ Informacijsko inženjerstvo (information engineering, IE)

- Inženjerski pristup izgradnji IS i upravljanju informacijama
 - IS mora biti projektiran, kao što se to čini s drugim proizvodima
 - razvoj IS kao strukturiran i planiran proces podržan računalom
- Sustavna primjena prikladnog skupa metoda, tehnika i alata u procesu razvoja informacijskih sustava.

□ Metoda (method), metodologija (methodology)

- Smišljen i organiziran skup procedura izrade (modela, softvera), uz korištenje dobro definirane notacije
 - sugerira proces obavljanja i način bilježenja
 - definira primjenu tehnika i njihovo povezivanje (pr. ERD-DFD)

□ Tehnika (technique)

- definira način provođenja postupka i dokumentiranja rezultata tog postupka
- npr. tehnika intervjuiranja, strateško planiranje, strukturirano programiranje

□ Pomagalo, alat (tool)

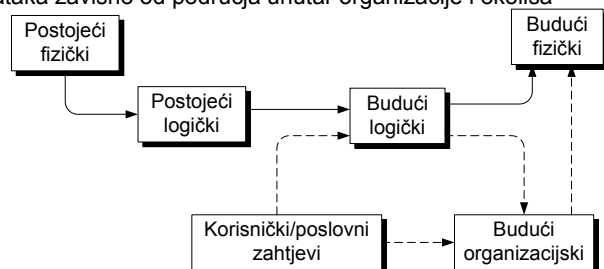
- instrument, sredstvo (artefact) koje se koristi u razvoju IS
- npr. aplikacija za upravljanje projektima, CASE pomagalo, generator



Projektiranje IS

□ Slijed izrade fizičkog i logičkog modela postojećeg IS, a zatim logičkog i fizičkog modela budućeg IS na temelju poslovnih zahtjeva i zahtjeva krajnjih korisnika.

- fizički model (ugradbeni, implementacijski, tehnički) – opisuje kako je sustav fizički i tehnički ugrađen, te tko, gdje i kada nešto radi
- logički model (esencijalni, konceptualni, poslovni) – što je sustav, što radi, što su podaci
- operativni (budući fizički) sustav – što, tko i kada, ali ne i gdje
- opcionalno, može se razmatrati organizacijska razina – različito značenje podataka zavisno od područja unutar organizacije i okoliša



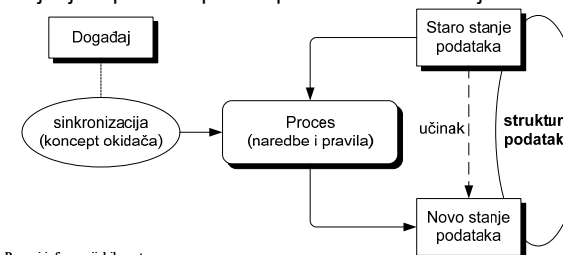
Modeliranje IS

□ Izrada modela koji odgovaraju dijelovima stvarnog sustava

- model: apstrakcija ili reprezentacija dijela stvarnog svijeta
- Kada otprije ne postoji sustav, potrebno je odrediti "surogat" postojećeg sustava po ugledu na istovrsne sustave u drugim organizacijama ili razvoj započeti s izradom logičkog modela.

□ Tehnike oblikovanja dijagramima

- Izradom modela nastoji se opisati situacija u kojoj:
 - događaj iz vanjskog svijeta pokreće (okida) proces
 - proces ima određeni učinak na podatke u nekom stanju
 - obavljanjem procesa podaci prelaze u novo stanje.



Tehnike modeliranja

- ❑ **Modeliranje podataka / oblikovanje podataka (data modelling)**
 - model podataka – ŠTO su podaci, odnosno što opisuju podaci
 - konceptualni model - opisuje podatke i veze između podataka, najčešće model entiteti-veze (entity-relationship model)
 - logički model – opisuje strukturu podataka i logičkih datoteka, najčešće relacijski model podataka (relational data model)
- ❑ **Modeliranje procesa/funkcija (process modelling, functional decomposition)**
 - model funkcija i procesa – KAKO se prikupljaju, obrađuju i distribuiraju podaci
 - model funkcija - oblikuje se razlaganjem (dekompozicijom) funkcija, iterativno od vrha prema dolje (od globalnih funkcija do osnovnih procesa)
 - model procesa – opisuje obradu podataka promatranog sustava, najčešće dijagram toka podataka (data-flow diagram)
- ❑ **Modeliranje objekata**
 - uočavanje podataka i procesa u objekte
 - objektni modeli – dijagrami koji dokumentiraju strukturu objekata i njihove interakcije (UML)
- ❑ **Modeliranje događaja**
 - model događaja – KADA se podaci obrađuju
 - razmatranje učinka koji događaji imaju na procese i podatke te opis stanja, npr. dijagram promjene stanja (state transition diagram)
- ❑ **Modeliranje resursa/sredstava**
 - izvršitelji - TKO obrađuje podatke, GDJE se nalaze podaci, GDJE se obrađuju podaci
- ❑ **Modeliranje programa**
 - struktura (programskih) modula IS, primjerice strukturnim kartama, UML dijagramima



Ključne aktivnosti i sudionici

- ❑ **Ključne aktivnosti # skupni sinonim ponekad informacijsko inženjerstvo**
 - Sistemska analiza, Analiza sustava (Systems analysis)
 - proučavanje poslovanja s ciljem preporuke poboljšanja i specifikiranja zahtjeva na rješenje
 - Sistemski dizajn, Dizajn sustava (Systems design)
 - specifikacija ili konstrukcija računalom podržanog rješenja identificiranih poslovnih zahtjeva
- ❑ **Dionici, interesni sudionici (stakeholders)**
 - Korisnik, Klijent (User, Customer, Client) – osoba ili grupa za koju se gradi IS
 - Korisnik sustava – krajnji korisnik
 - neposredno koristi IS pri obavljanju svakodnevnih poslova ili koristi informaciju dobivenu iz IS
 - Vlasnik sustava – naručitelj (stvarni vlasnik ili upravitelj organizacije)
 - naručuje i plaća razvoj i održavanje IS, određuje prioritete i politiku njegovog korištenja
 - Projektant, Dizajner sustava (Designer)
 - tehnički stručnjak koji oblikuje sustav tako da zadovolji zahtjeve korisnika
 - prevodi poslovne zahtjeve i ograničenja u tehničko rješenje, modeliranjem
 - Graditelj sustava (Builder, Developer) - programer, razvojnici
 - izgrađuje sustav, provjerava njegovu ispravnost te ga isporučuje u primjenu
 - konstruira komponente sustava na temelju specifikacija koje rade projektanti sustava



Ključne uloge

- ❑ **Sistem analitičar [Whitten et. al, 2000]**
 - Sistem analitičar potpomaže proučavanje problema i potreba poslovanja radi određivanja kako poslovni sustav i informacijska tehnologija mogu najbolje riješiti problem i postići unaprjeđenje poslovanja. Plodovi ove aktivnosti mogu biti poboljšani poslovni procesi, poboljšani informacijski sustavi te nove ili poboljšane aplikacije, a često sve zajedno.
 - identifikacija tehnoloških poboljšanja poslovnih procesa, oblikovanje poslovnih procesa, projektiranje IS, osiguranje sukladnosti IS standardima
- ❑ **Poslovni analitičar**
 - analiza poslovanja, identifikacija koristi, oblikovanje (novih) procesa i procedura
- ❑ **[drugi analitičari, npr. infrastrukturni, upravljanja promjenama]**
- ❑ **Upravitelj projekta**
 - upravljanje ekipom analitičara, razvojnika i drugih stručnjaka
 - izrada plana, raspodjela resursa i nadzor provedbe
- ❑ **Analitičar ?= konzultant**



Životni ciklus razvoja sustava

Systems Development Life-Cycle (SDLC)



Životni ciklus razvoja

□ Općeniti proces razvoja sustava

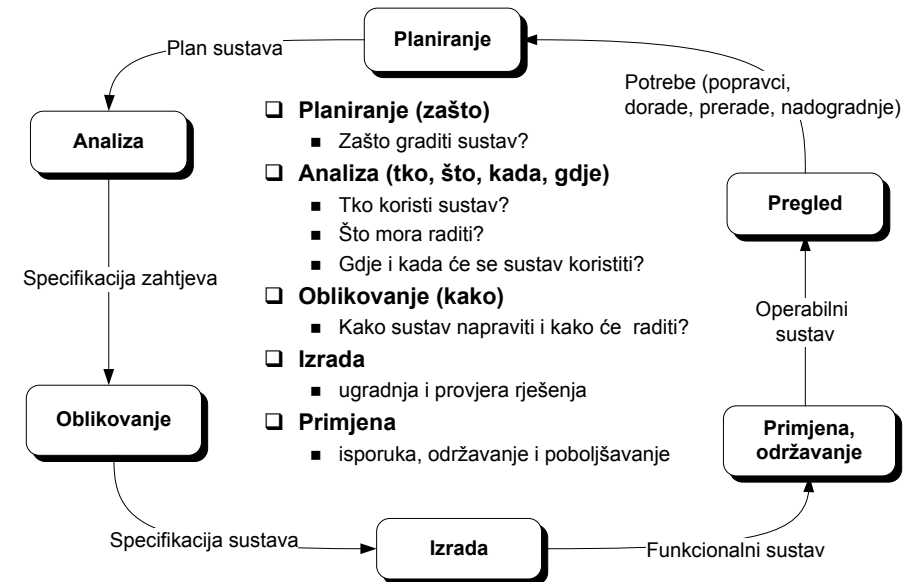
- Ustanovljavanje problema
- Proučavanje i razumijevanje problema
- Identifikacija zahtjeva na rješenje ili očekivanja
- Procjena alternativnih rješenja i odabir "najboljeg" smjera
- Oblikovanje odabranog rješenja
- Ugradnja odabranog rješenja
- Vrednovanje rezultata. Ukoliko problem nije riješen povratak na korak 1 ili 2

□ SDLC = software/systems development life-cycle

- konzistentan i standardizirani način razvoja IS
- model razvojnog procesa - unaprijed propisan proces razvoja
- definira faze i zadatke (aktivnosti) koje treba obaviti tijekom razvoja
- ciklus osigurava "kontrolne točke" za praćenje napretka, procjenu postignutih rezultata i donošenje odluka o daljnjim koracima



Faze životnog ciklusa



Planiranje sustava

- Traženje odgovora na pitanje zašto graditi sustav

□ Inicijacija projekta

- Zahtijevanje sustava (system request), Inicijalna strategija (initial strategy)
 - određivanje poslovnih ciljeva, problema i ideja njihovog rješavanja
 - snimka stanja, sažetak poslovnih potreba
- Studija izvodljivosti (feasibility study)
 - analiza problemskog područja i određivanje (granica) projekata
 - procjena ključnih aspekata predloženog projekta
 - tehnička izvodljivost (može li se izgraditi)
 - ekonomska izvodljivost (hoće li se isplatiti)
 - organizacijska izvodljivost (da li će se koristiti)
- Posebno povjerenstvo (steering committee, approval committee) odobrava projekte

□ Upravljanje projektom

- po odobrenju projekta
- Izrada plana rada, kadrovanje projekta, upravljanje i nadzor projekta

→ Poslovni cilj i projekti, plan sustava, plan informatizacije (glavni projekt)



Faza analize

- Traženje odgovora na pitanje tko će koristiti sustav, što će sustav raditi te gdje i kada će biti korišten.
- Proučavaju se postojeći sustavi, ustanovljavaju moguća poboljšanja i razvija koncept novog sustava

□ Strategija analize (analysis strategy)

- vodič kroz analizu postojećeg (as-is) i projektiranje novog (to-be) sustava

□ Analiza zahtjeva (requirements analysis): detaljna analiza kojom se preciziraju granice projekta i poslovni zahtjevi

- detaljna analiza postojećeg sustava (problema i poslovnih zahtjeva)
- modeliranje (postojećeg) sustava

□ Specifikacija zahtjeva (requirements specification): detaljni opis zahtjeva na IS, oblikovan tako da ga razumiju projektanti

- modeliranje (budućeg) sustava
- definiranje zahtjeva na budući IS

→ Poslovni model sustava, prijedlog sustava



Oblikovanje sustava

- određuje kako će sustav funkcionirati, sa stanovišta hardvera, softvera i mrežne infrastrukture, korisničkog sučelja, programa te spremišta podataka
- ❑ **Strategija projekta (design strategy)**
 - odluka da li graditi vlastitim snagama (insourcing), naručiti razvoj po mjeri (outsourcing) ili kupiti gotovo rješenje
- ❑ **Dizajn sustava (system design)**
 - modeliranje cjelokupnog sustava (pogled projektanta)
 - odabir tehničke arhitekture sustava
 - konceptualno modeliranje podataka
 - opći dizajn sučelja
- ❑ **Detaljni dizajn (detailed design)**
 - razrada rješenja, izrada tehnološkog modela IS (pogled izvođača)
 - dizajn sučelja, pohrane podataka i programa
- ➔ **Tehnička specifikacija sustava**
 - kolekcija specifikacija arhitekture, sučelja, pohrane podataka i programa
- ❑ **Po potrebi se revidira studija izvodljivosti i plan projekta**



Izrada sustava

- Provedba - ugradnja ili nabava oblikovanih rješenja
- ❑ **Izrada, implementacija (implementation), konstrukcija**
 - ugradnja baze podataka,
 - kodiranje procesa (funkcija) novog IS
- ❑ **Testiranje (testing)**
 - provjera komponenti
- ❑ **Integracija i provjera sustava (system integration, evaluation)**
 - udruživanje dijelova i provjera cjeline, da bi se dokazalo da sustav radi (da je ispravno napravljen) te da radi ono što je zahtijevano (da je napravljen pravi sustav, koji ispunjava zahtjeve)
- ➔ **Funkcionalni sustav i tehnološki opis sustava**

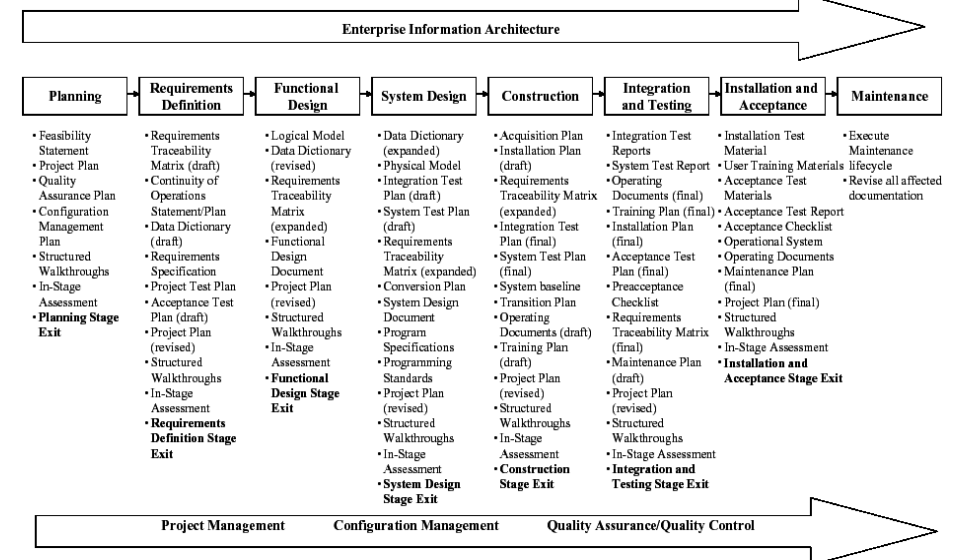


Primjena sustava

- ❑ **Uvođenje u primjenu i održavanje**
 - Uvođenje u primjenu, primjena (operation, production)
 - prijenos (konverzija) podataka sa starog na novi sustav
 - uspostava novih radnih aktivnosti
 - Održavanje (maintenance)
 - održavanje sustava u upotrebi
 - izmjene sustava radi poboljšanja radnih karakteristika (performanci), poboljšanja ili prilagodbe načina uporabe
 - Potpora (support)
 - dobavljača opreme, tehničkog osoblja korisnicima
- ➔ **Informacijski sustav u primjeni, plan održavanja**
- ❑ **Pregled (review)**
 - revizija, preispitivanje čitavog sustava kada su potrebne veće izmjene uslijed promjena u poslovanju ili promjena poslovnih ciljeva
- ➔ **Novi projekti, novi razvojni ciklus**
- ❑ **Navedene su tipične faze životnog ciklusa, bez implikacije da se radi o diskretnim i/ili slijednim aktivnostima!**



Primjer: IS LC Stages and Deliverables, DOE Software Development Methodology, <http://cio.doe.gov/sqse>



Razvoj projekta po stadijima

❑ U nastavku su navedene tipične aktivnosti i rezultati aktivnosti po fazama razvoja, neznatno modificirano prema [McConnel, 1998].

- Iako slični metodologiji, riječ o skupu praktičnih preporuka.
- Pojedine aktivnosti mogu biti opcionalne, ovisno o tipu projekta.

❑ Resursi

- Kontrolne liste za provjeru i predlošci projektne dokumentacije nalaze se na <http://www.construx.com>
- Prilagodljivi razvojni proces (Adaptable Process Model) opisan je na <http://www.rsps.com>
- Prije spomenuti <http://cio.doe.gov/sqse>



Pokretanje projekta

❑ Aktivnosti

- postavljanje upravitelja projekta
- postavljanje ciljeva projekta
- izrada vizije projekta
- analiza isplativosti projekta
- studija izvodljivosti
- plan kontrole promjena
- sastavljanje liste rizika
- otvaranje dnevnika projekta
- otvaranje dnevnika problema
- angažman 2–3 iskusna razvojnika

❑ Rezultati

- donesena odluka o pokretanju projekta (ako se isplati)
- kreirani i provjereni dokumenti:
 - vizija projekta (svrha i ciljevi)
 - studija izvodljivosti i isplativosti
 - plan kontrole promjena
 - lista mogućih rizika
 - dnevnik softverskog projekta
 - dnevnik problema



Analiza

❑ Aktivnosti

- prikupljanje informacija
- stvaranje vodiča za stil sučelja
- kreiranje i revizija prototipa korisničkog sučelja
- razrada procjena
- priprema prijedloga o korištenju postojećih dijelova softvera, odnosno nabavi novih
- izrada i revizija plana razvoja softvera (prvo preliminarni, a potom i detaljni)
- analiza i klasifikacija zahtjeva
- okupljanje i suradnja svih zainteresiranih strana s ciljem otklanjanja problema povezanih sa zahtjevima
- razrješavanje nejasnih i dvosmislenih zahtjeva
- kreiranje i revizija korisničkog priručnika / specifikacija zahtjeva
- izrada i revizija plana testiranja
- revizija i ažuriranje rizika projekta
- revizija i ažuriranje dnevnika projekta
- revizija i ažuriranje prototipa sučelja i vodiča za stil sučelja
- revizija i ažuriranje dnevnika problema
- kontrola kvalitete svih proizvoda

❑ Rezultati

- detaljan prototip korisničkog sučelja
- kreirani i provjereni dokumenti:
 - matrica praćenja zahtjeva
 - korisnički priručnik / specifikacija zahtjeva
 - plan za osiguranje kvalitete softvera
 - detaljan plan razvoja softvera
 - plan testiranja
 - rječnik podataka
 - procjene
- ažurirani ostali dokumenti



Oblikovanje

❑ Aktivnosti

- zapošljavanje većine razvojne ekipe
- priprema preliminarnih dijagrama dizajna
- priprema PDL-a za funkcije/objekte dizajna visoke razine
- razrješavanje problema preostalih iz faze razvoja zahtjeva
- zapošljavanje većine QA osoblja
- definiranje dizajna visoke razine (arhitekture sustava)
- kreiranje i revidiranje dokumenta o arhitekturi softvera
- ažuriranje procjena
- ažuriranje rizika
- ažuriranje dnevnika projekta
- ažuriranje korisničkog priručnika / specifikacije zahtjeva
- ažuriranje dnevnika problema
- ažuriranje plana razvoja
- definiranje dizajna niske razine
- kreiranje i revizija plana isporuke po stadijima
- izrada plana izrade za prvi stadij
- kreiranje i revizija testova za prvi stadij
- kontrola kvalitete svih proizvoda

❑ Rezultati

- kreirani i revidirani slijedeći dokumenti:
 - dokument o arhitekturi softvera
 - plan gradnje softvera po stadijima
 - dokument detaljnog dizajna
 - testovi za 1. stadij
 - plan isporuke po stadijima
- ažurirani i revidirani svi ostali dokumenti



Ugradnja

□ Aktivnosti

- zapošljavanje cijele razvojne ekipe i QA osoblja
- korekcija pogrešaka iz prethodnih faza
- kreiranje i revizija testova za sljedeći stadij
- kreiranje i revizija plana konstrukcije za trenutni stadij
- kreiranje liste mini koraka za trenutni stadij
- postavljanje standarda kodiranja
- pisanje programskog koda
- testiranje i integracija cjelina
- tehnička revizija
- ažuriranje i revizija dokumenta detaljnog dizajna
- ažuriranje liste rizika
- ažuriranje dnevnika projekta
- ažuriranje korisničkog priručnika / specifikacije zahtjeva
- ažuriranje dnevnika problema
- ažuriranje plana razvoja
- ažuriranje procjena
- kreiranje i revizija plana testiranja
- kreiranje i revizija testova sustava
- izrada plana alfa i beta testiranja
- izrada nacrt testova prihvatljivosti
- izrada riješenih problemskih primjera (Test case)
- kontrola kvalitete svih proizvoda

□ Rezultati

- kreirani i revidirani slijedeći dokumenti:
 - plan integracijskih i sistemskih testiranja
 - plan alfa i beta testiranja
 - nacrt testova za prihvaćanje softvera
- napisana lista mini koraka
- napisan programski kod
- obavljena testiranja komponenti
- obavljena integracija komponenti
- ažurirani svi ostali dokumenti



Integracija i provjera

□ Aktivnosti

- provjera sustava, alfa i beta testiranja
- sastavljanje izvještaja o rezultatima testiranja
- popravljavanje i prepravljavanje programskog koda
- ažuriranje korisničkog priručnika
- identificiranje kandidata za ponovo uporabe programske knjižnice
- kreiranje plana testiranja za prihvaćanje konačnog proizvoda
- kreiranje konačnog plana konverzije
- kreiranje nacrt plana održavanja
- priprema za provjeru prihvatljivosti
- ažuriranje plana projekta
- ažuriranje procjena
- ažuriranje dnevnika projekta
- ažuriranje dnevnika problema
- kontrola kvalitete svih proizvoda

□ Rezultati

- izvještaji napravljeni na temelju svih obavljenih testiranja
- plan testova za prihvaćanje finalnog proizvoda
- instalacijski plan
- softver koji radi kako treba (testirani sistemski kod i datoteke podrške)



Instalacija i prihvaćanje

□ Aktivnosti

- ažuriranje dnevnika projekta
- ažuriranje procjena
- izrada konačne verzija plana održavanja
- kompletiranje korisničke dokumentacije
- izrada kontrolne liste isporuke
- finalno testiranje prije isporuke (mjerilo prihvaćanja finalne verzije)
- izrada izvještaja o rezultatima testova za prihvaćanje softvera
- izrada dokumenata o završetku (obrasci, suglasnost o primopredaji)
- izrada instalacijske inačice
- ažuriranje dnevnika problema

□ Rezultati

- instalacijska verzija softvera
- izvještaj o rezultatima testova za prihvaćanje softvera
- konačna verzija korisničke dokumentacije
- ažurirani ostali dokumenti



Isporuka i održavanje

□ Aktivnosti

- isporuka funkcionalnog proizvoda
- arhiviranje projekta (izvornog koda, razvojnih pomagala, dokumenata, ...)
- finalno ažuriranje dnevnika projekta
- kreiranje i revizija povijesnih dokumenata
- uspostava distribucije
- upravljanje distribucijom (kome isporučiti, kada, kako)
- uspostavljanje službe (centra) za suradnju i pomoć korisnicima
- održavanje, nadgradnja ili izdavanje novih verzija
- nove verzije se prije isporuke testiraju i u skladu s tim korigira se korisnički priručnik

□ Rezultati

- isporuka potpuno dovršenog i ispravnog proizvoda
- arhivirani projektni mediji
- dokument o povijesti projekta
- instalacijski paketi
- korisnicima omogućeno da predlažu dalje korekcije softvera ili da se obrate za pomoć ako im zatreba



Statistika razvoja softvera (Mynatt)

1. Tipični projekt razvoja softvera traje
a) 1 - 5 b) 6 - 11 c) 12 - 23 d) 24 - 48 mjeseca.
2. Za programski sustav srednje veličine,
a) manje od 10 b) 10 - 20 c) 21 - 30 d) više od 30
redaka izvornog programskog koda se tipično proizvede dnevno po osobi, tijekom cijelog razdoblja razvoja sustava.
3. Približan broj pogrešaka koje se pronađu u svakih 1000 redaka izvornog izvršnog programskog koda tijekom razvoja programskog sustava jest:
a) manje od 30 b) 30 - 40 c) 40 - 50 d) 50 - 60
4. Približan broj pogrešaka koje se pronađu u svakih 1000 redaka izvornog izvršnog programskog koda u isporučenom sustavu jest:
a) manje od 4 b) 4 - 8 c) 8 - 12 d) više od 12
5. Koji je približan postotak programskih sustava čiji je razvoj započeo, a da budu i konačno dovršeni?
a) 90 - 100% b) 80 - 90% c) 70 - 80% d) 60 - 70%

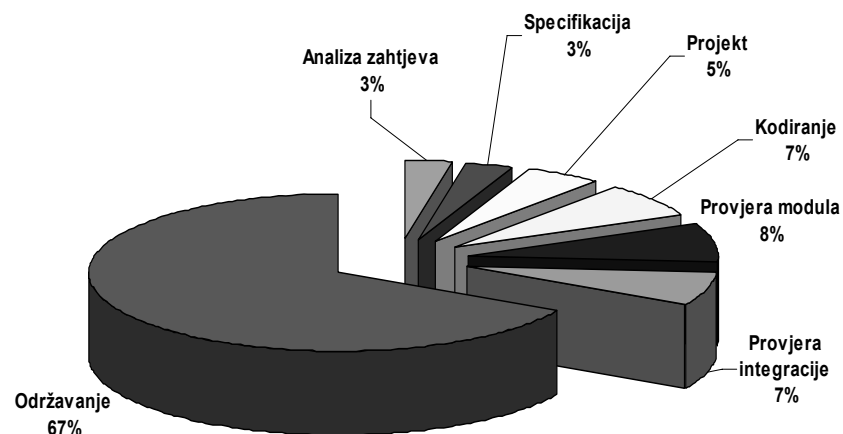


Statistika u svezi razvoja softvera (Mynatt)

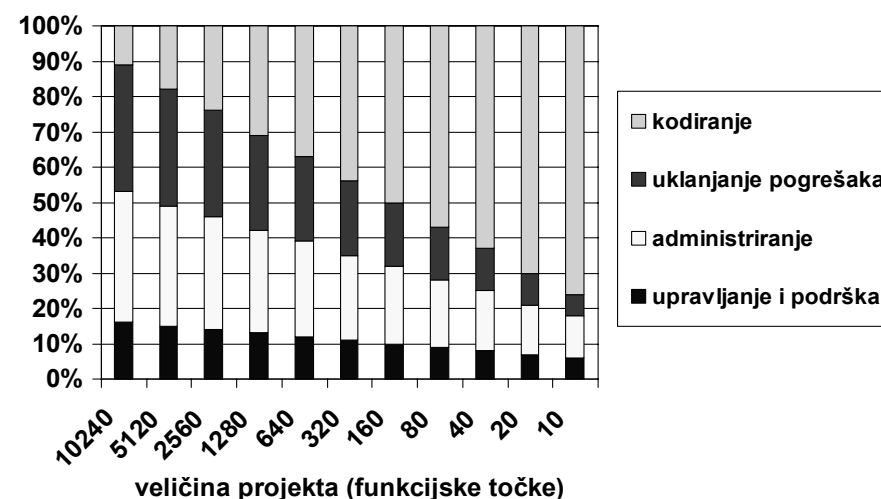
6. Trošak posjedovanja i održavanja softvera je tipično
a) za polovicu b) jednako c) dvostruko
puta veći nego što je bio trošak razvoj softvera.
7. Većina pogrešaka koje korisnici pronađu u softveru su rezultat
a) programerove pogreške
b) problema u oblikovanju zadatka ili njegova razumijevanja
c) administrativnih pogrešaka
d) pogrešaka u projektu
8. Koliki je udio programiranja u razvoju softvera?
a) 99% b) 70% c) 50% d) 20% e) 10%
9. Prema današnjem stanju razvoja softvera, sustav se smatra velikim ako sadrži barem
a) 2.000 b) 10.000 c) 100.000 d) 1000.000
redaka izvornog koda



Razdioba troškova u životnom ciklusu softvera



Učešće aktivnosti kao funkcija veličine projekta



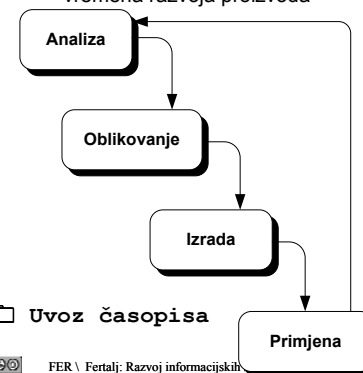
Modeli razvoja



Vodopadni (waterfall) model

❑ Klasični vodopadni model

- slijedno napredovanje iz faze u fazu
- nisu dozvoljene naknadne promjene rezultata prethodnih faza
- zahtjevi za kvalitetom proizvoda su naglašeniji od zahtjeva za smanjivanjem troškova projekta i vremena razvoja proizvoda



❑ Primjenjivost

- dobro definirano i stabilno okruženje, gdje postoje razrađene procedure ručne obrade ili računalni sustav koji treba unaprijediti
- veliki projekti (investicije)
- zaposlenici neiskusni ili posjeduju ograničeno tehničko znanje

❑ Nedostaci

- uvođenje prema gore (bottom up): moduli, podsustavi, sustav
- sustav nije upotrebljiv dok nije gotov u potpunosti
- problem predodžbe o proizvodu na temelju pisane specifikacije
- korisnici prekasno uoče nedostatke
- problem u slučaju pogrešaka ili novih/promijenjenih zahtjeva



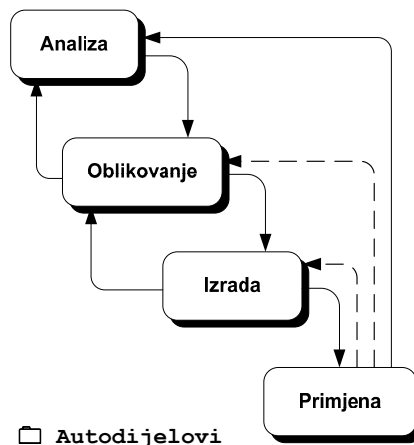
Pseudostrukturni i radikalni vodopadni model

❑ Pseudostrukturni model

- povratna veza i mogućnost promjene rezultata prethodnih faza
- primjena tehnika strukturiranog programiranja
- uvođenje prema dolje: moduli na višim, pa na nižim razinama

❑ Strukturni (radikalni)

- aktivnosti različitih faza mogu se obavljati istovremeno
- korištenje rječnika podataka, 4GL i generatora aplikacija
- prikladan kada se unaprijed ne zna konačni izgled sustava
- u konačnici mora nastati (papirnat) model sustava



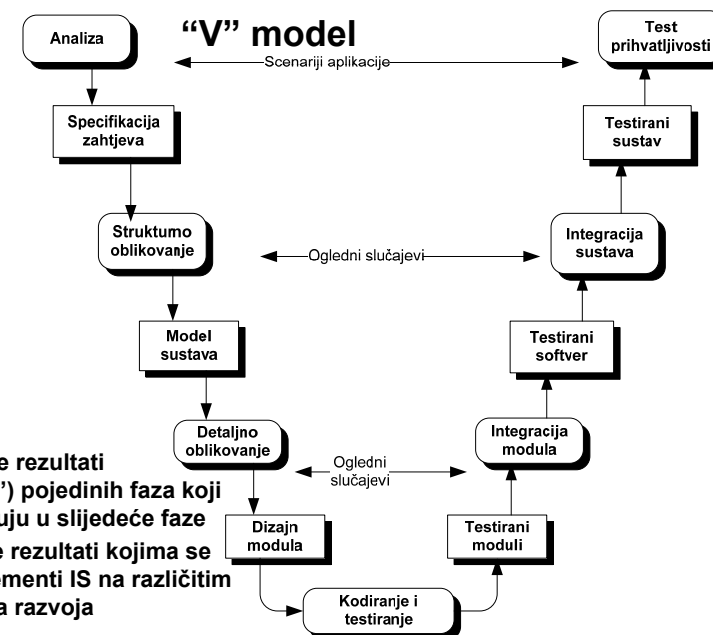
❑ Autodijelovi



❑ Validacija

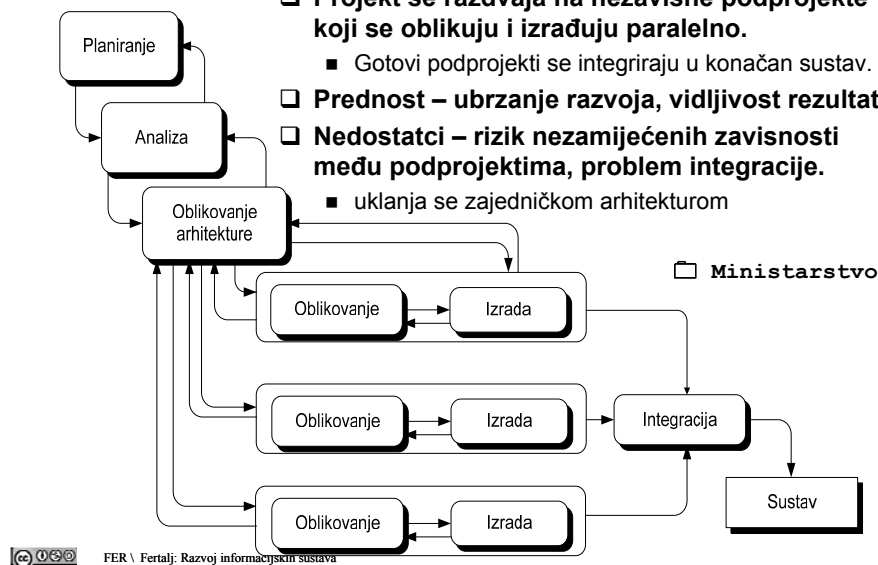
❑ Verifikacija

- definiraju se rezultati ("proizvodi") pojedinih faza koji se proslijeđuju u slijedeće faze
- određuju se rezultati kojima se testiraju elementi IS na različitim stupnjevima razvoja



Paralelni razvoj po podprojektima

- ❑ Projekt se razdvaja na nezavisne podprojekte koji se oblikuju i izrađuju paralelno.
 - Gotovi podprojekti se integriraju u konačan sustav.
- ❑ Prednost – ubrzanje razvoja, vidljivost rezultata
- ❑ Nedostaci – rizik nezamijećenih zavisnosti među podprojektima, problem integracije.
 - uklanja se zajedničkom arhitekturom

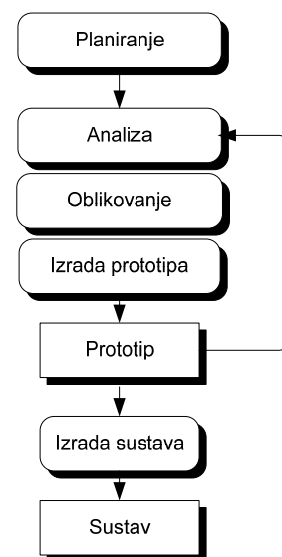


Prototipski model razvoja

- ❑ Prototip = model koji se radi da bi se isprobale neke mogućnosti
- ❑ Vrste prototipova
 - Model oponašanja (mock-up, model u naravnoj veličini)
 - jednoekranski ili višeekranski model kojim se prikazuje kako će izgledati dio sustava (npr. sučelje)
 - Istraživački model (research model)
 - istraživanje dijelova sustava kako bi se provjerile neke ključne postavke (npr. provjera performanci određenog modula)
 - Ugradbeni model (implementation model)
 - traženje načina na koje se sustav može izraditi (npr. koji sustav za upravljanje BP, programski jezik, računala...)

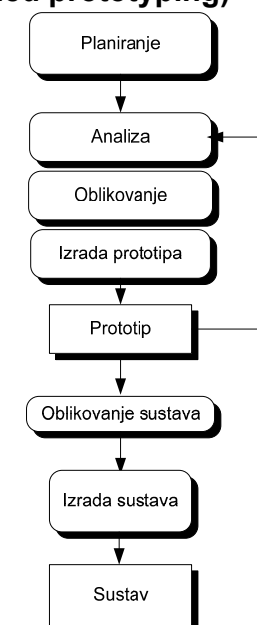
Brzo prototipiranje (rapid prototyping)

- ❑ Funkcionalni prototip
 - nedovršeni proizvod koji simulira rad pravog sustava
- ❑ Evolucijsko prototipiranje (sinonim)
 - razvoj na osnovi povratnih informacija
 - prototip inkrementalnom doradom – "bistrenjem" (stepwise refinement) postaje dio završnog IS
 - kada prototip postane dovoljno dobar, završava se preostali posao i prototip postaje konačan proizvod
 - prikladno za male projekte (tzv. one-man) s promjenjivim zahtjevima
- ❑ Prednosti
 - stječe se bolja slika o zahtjevima korisnika
 - uklanjaju se moguća iznenađenja na kraju razvoja
 - moguće su promjene zahtjeva korisnika
 - povećanje kreativnosti i brzine razvoja
- ❑ Nedostaci
 - "zaboravljanje" da prototip nije pravi sustav
 - mogući neuspjeh zamjene prototipa sustavom
 - dokumentacija proizlazi iz izrade, uz opasnost da pisane specifikacije nikad neće biti napravljene
 - nemogućnost ispravne procjene i planiranja resursa



Ograničeno prototipiranje (constrained prototyping)

- ❑ Nefunkcionalni prototip
 - prikaz izgleda
 - prototip nije i ne postaje radni sustav
- ❑ Prototipiranje kao sredstvo određivanja zahtjeva
 - prototipom se smanjuje rizik lošeg određivanja zahtjeva
 - po dovršetku analize prototip se odbacuje (throwaway prototyping)
 - slijedi faza oblikovanja sustava
- ❑ Prednost
 - postoji dokumentacija
 - moguće je planiranje

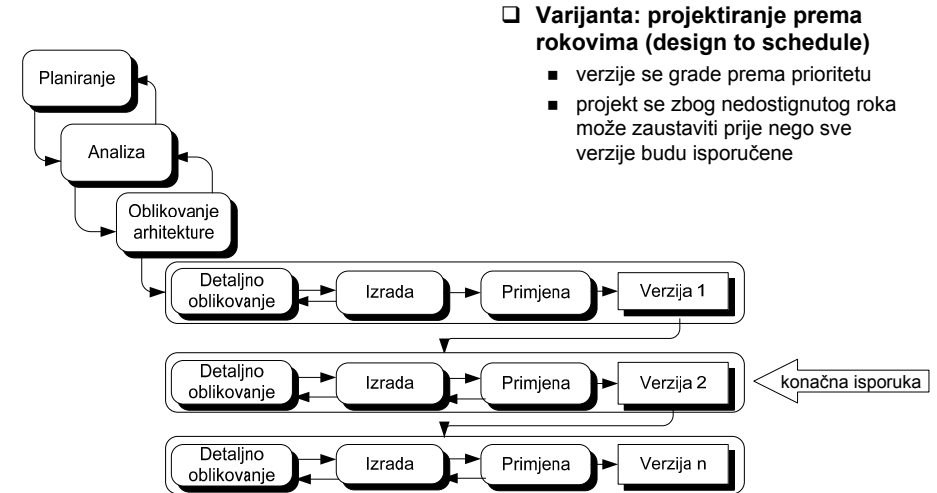


Fazni razvoj, model postupne isporuke

- ❑ **Fazni razvoj, model postupne isporuke (Staged Delivery), evolucijski razvoj, evolucijsko prototipiranje, inkrementalni razvoj**
 - predstavnik brzog razvoja aplikacija, RAD (Rapid Application Development)
 - sustav je podijeljen po verzijama koje se izrađuju slijedno
 - svaki od slijedova vodi proizvodu koji se isporučuje i generira zahtjeve
 - u ranim fazama sličan vodopadnom modelu, dok se kasnije faze obavljaju zasebno za pojedinu isporuku projekta
- ❑ **Prednosti**
 - omogućuje isporuku dijelova proizvoda prije konačnog završetka projekta
 - brža uporaba - ranije vidljiva vidi korist sustava
 - korisnici brže identificiraju dodatne zahtjeve
- ❑ **Nedostaci**
 - korisnici upotrebljavaju sustav koji je "namjerno" nedovršen
 - potreba za pažljivim planiranjem (raspodjela resursa, ovisnosti između pojedinih faza isporuke) - ključno je da najpotrebnije i najkorisnije mogućnosti budu ugrađene prve



Evolucijski model, model postupne isporuke

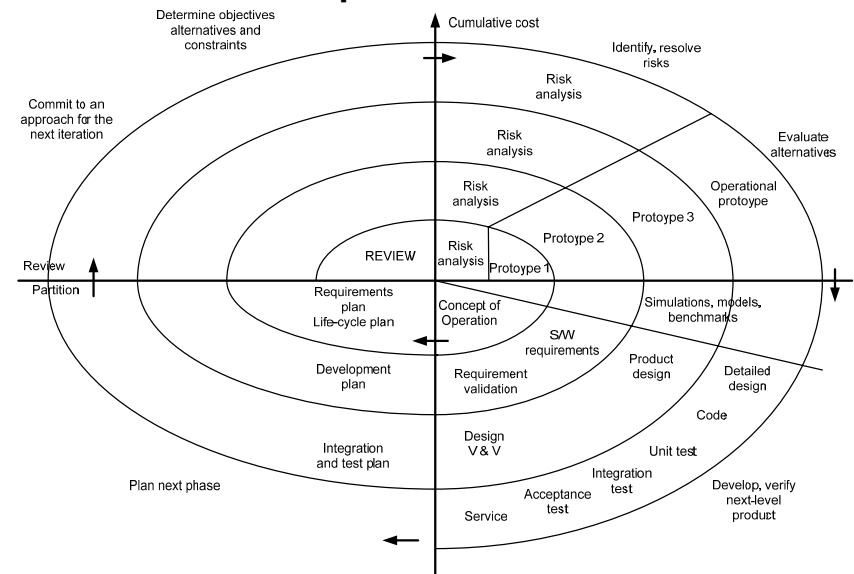


Spiralni model

- ❑ **Iterativno se analizira rizik i planira naredna iteracija od 6 koraka:**
 - Određivanje ciljeva, alternativnih metoda i ograničenja,
 - Određivanje i pronalaženje rješenja za rizike koji se mogu pojaviti,
 - Procjenjivanje alternativnih rješenja,
 - Razvoj izlaznih proizvoda iteracije i potvrđivanje njihove točnosti,
 - Planiranje sljedeće iteracije,
 - Pokretanje sljedeće iteracije (ako se odluči krenuti u novu iteraciju).
- ❑ **Svaka iteracija dovodi projekt bliže kraju.**
 - U slučaju da je rizik prevelik, projekt se obustavlja ili prekida.
 - Radijalna koordinata predstavlja kumulativni trošak
 - Zadnja iteracija predstavlja klasični ciklus
- ❑ **Primjena**
 - izgradnja velikih sustava (troškovi izrade malih sustava su mali)
 - kada provođenje analize rizika ne predstavlja preveliki relativni trošak
 - najčešće na interne projekte (naručitelj i izvođač iz iste organizacije)
- ❑ **Prednosti**
 - praćenje napretka, sigurnost
- ❑ **Nedostaci**
 - složenost, trošak



Spiralni model



Objektno usmjereni pristup razvoju

Objektno usmjereni pristup razvoju

- mogao bi se provoditi bilo kojim od klasičnih modela razvoja
 - najviše sliči faznom razvoju
 - ključna razlika je u pristupu dekompoziciji
 - tradicionalno, proces dekompozicije je usmjeren procesima (process-centric) ili podacima (data-centric)
 - OO dekompozicija je usmjerena na objekte koji sadrže i podatke i procese
- ### OO pristup razvoju treba biti [Booch, Jacobson, Rumbaugh]
- Vođen slučajevima primjene (Use-Case Driven)
 - UC kao glavno pomagalo za definiranje ponašanja sustava
 - Inherentno jednostavni, opisuju po jednu aktivnost (pa nema potrebe za ugniježđenom dekompozicijom i ulančanim dijagramima)
 - Usmjeren arhitekturi, s arhitekturom u središtu (Architecture-Centric)
 - arhitektura vodi specifikaciju, konstrukciju i dokumentiranje, kroz tri pogleda
 - funkcionalni – opisuje vanjsko ponašanje sustava, pogled korisnika
 - statički – struktura sustava (atributi, postupci, razredi, veze)
 - dinamički – opisuje unutarnje ponašanje razmjenom poruka između objekata i promjenama njihovog stanja (dijagramima ponašanja)
 - Iterativan i inkrementalan (Iterative and Incremental)
 - ponavljani postupci, postupno povećanje dovršenosti

69

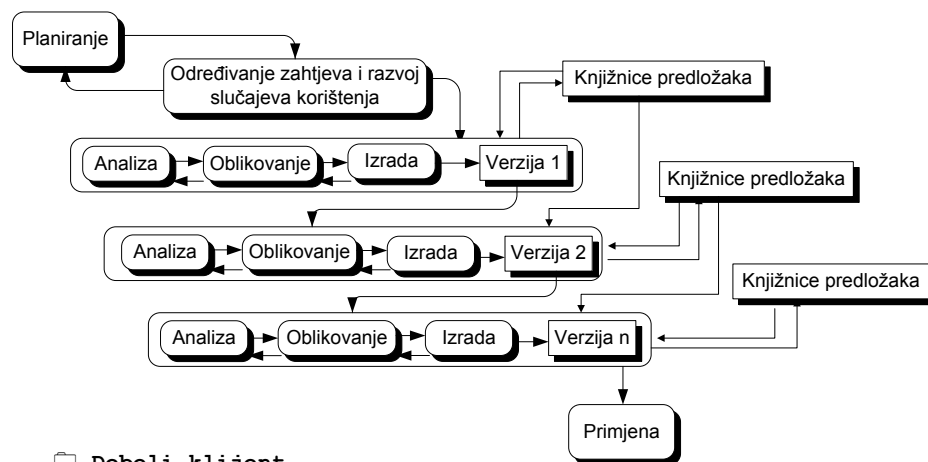
Osnovni objektno orijentirani model

Minimalist Object-Oriented Systems Analysis & Design (MOOSAD)

- generički pristup OOSAD
 - zasnovan na konceptima ujedinjenog procesa (Unified Process) i ekstremnog programiranja (eXtreme Programming)
 - UML kao sredstvo dokumentiranja
- ### Provode se gradnje sustava
- gradnja (build) – inačica softvera u razvoju
 - gradnje se temelje na slučajevima korištenja po prioritetima
 - svaka gradnja predstavlja inkrementalni napredak završetku
 - svaka gradnja obuhvaća analizu, oblikovanje i izradu
 - svaka izgradnja vraća povratnu informaciju o utvrđivanje zahtjeva, isporučuje funkcionalni sustav za sljedeći izgradnju i skup predložaka koji se mogu dodavati u programske knjižnice
 - gradnje omogućuju upravljanje projektom zasnovano na čvrstim rokovima (timeboxing), pri čemu su vremenski okviri pojedinog timeboxa od jednog do dva tjedna ili mjeseca ovisno o veličini i složenosti projekta

70

MOOSAD pristup



- ☐ **Debeli klijent**
- ☐ **Višeslojni poslovni sustav**

71

Nabava gotovih proizvoda

Alternativa razvoju je kupnja postojećeg proizvoda.

- Off the Shelf, Commercial-Off-The-Shelf (COTS), Shrink-wrap – općenito
- Enterprise Resource Planning (ERP) – sustav za podršku poslovanju

Postojeći proizvodi rijetko zadovoljavaju svu potrebnu funkcionalnost, ali nabava može imati i prednosti.

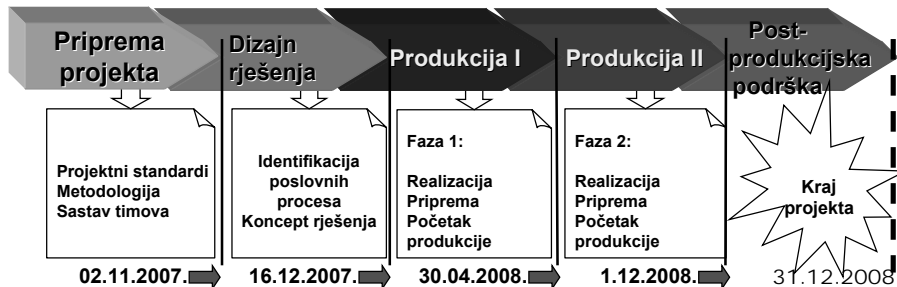
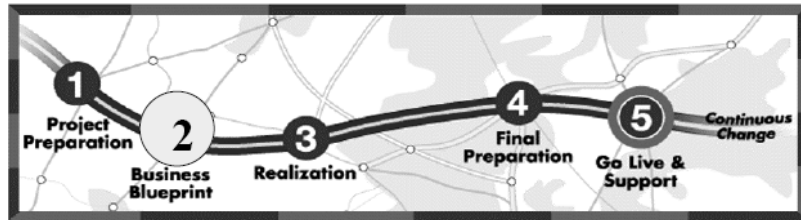
- Proizvod je vidljiv i može se ocijeniti (!?)
- Proizvod je "odmah" dostupan po nabavi (!?)
 - Ovo vrijedi uglavnom za pakete namijenjene širokom krugu nepoznatih korisnika.
- U slučaju ERP, treba obaviti prilagodbu koja može potrajati mjesecima ili godinama.
 - npr. SAP: 20k tablica, 20k zaslona, ABAP (Advanced Business Application Programming)

Životni ciklus ERP proizvoda

- Analiza (Project Planning, Project Preparation)
- Specifikacija zahtjeva (Business Blueprint)
- Implementacija i testiranje (Realization)
- Kontrola kvalitete i poduka (Final Preparation)
- Uvođenje u primjenu i održavanje (Going Live i podrška)

72

Životni ciklus ERP proizvoda



NAČELA IS



Uključenost sudionika i pristup rješavanju problema

- ❑ **Korisnici i vlasnici sustava moraju biti aktivno uključeni**
 - Za uspješnu izgradnju IS apsolutno je nužan angažman vlasnika i korisnika.
 - Osobe odgovorne za razvoj IS moraju naći vremena za vlasnike i korisnike, inzistirati na njihovom sudjelovanju i tražiti suglasnost od njih za sve odluke koje ih se tiču.
- ❑ **Treba koristiti pristup koji vodi k rješavanju problema**
 - Korištenjem metodologija smanjuje se rizik zbog korištenja prečaca i pogrešaka.
 - Metodologije proizvode potpunu i konzistentnu dokumentaciju, iz jednog projekta u drugi. Metodologija je, prije svega i najviše, postupak rješavanja problema prilikom izgradnje sustava!
 - Klasični pristup rješavanju problema je sljedeći:
 - Proučiti i shvatiti problem u kontekstu sustava,
 - Utvrditi zahtjeve koje mora ispuniti prikladno rješenje,
 - Prepoznati moguća rješenja i odabrati "najbolje",
 - Projektirati i/ili ugraditi rješenje,
 - Promatrati i vrednovati utjecaj izrađenog rješenja te u skladu s time dotjerivati rješenje.
 - Kad se nema iskustva, postoji težnja da se preskoče ili skrate neki od koraka u rješavanju problema. Ishod se kreće u rasponu:
 - rješava se krivi problem,
 - krivo se rješava problem,
 - izabere se krivo rješenje.



Uspostava životnog ciklusa i standarda razvoja

- ❑ **Uspostaviti faze i aktivnosti**
 - Većina životnih ciklusa i metodologija sastoji se od faza,
 - U najjednostavnijem obliku, životni se ciklus sustava sastoji od četiri faze: pregled, analiza, projekt, ugradnja.
 - Peta aktivnost, potpora sustavu, poboljšava rezultat, iterirajući kroz prethodne četiri faze, ali sada na umanjnim problemima i sa ciljem dotjerivanja i poboljšanja sustava.
 - Faze se obično razlome na aktivnosti i zadatke kojima se može lako upravljati i lako ih riješiti,
 - Faze projekta treba dovršavati u slijedu, od vrha do dna.
- ❑ **Uspostaviti standarde za konzistentan razvoj i dokumentiranje**
 - Standardi za razvoj sustava obično opisuju:
 - aktivnosti,
 - odgovornosti,
 - naputke ili zahtjeve za dokumentiranje,
 - provjere kvalitete.
 - Potreba za dokumentacijskim standardima ukazuje na uobičajenu pogrešku mnogih analitičara – njihov propust da izrađuju dokumentaciju kao trajnu aktivnost tijekom životnog ciklusa.



Opravdanje investicije u sustav

❑ Opravdati izgradnju sustava kao kapitalnu investiciju

- Informacijski sustavi jesu kapitalna investicija,
- U kapitalnim investicijama, treba razmatrati:
 - za svaki problem najvjerojatnije postoji nekoliko mogućih rješenja,
 - nakon što se prepoznaju varijante rješenja, analitičar treba vrednovati svako od mogućih rješenja glede njegove izvodljivosti, a pogotovo ekonomičnosti.
 - Ekonomičnost se definira kao odnos između dobiti ostvarene sustavom i troškova njegove izgradnje i pogona.



Ne oklijevati ako treba projekt otkazati

❑ Ne oklijevati ako treba projekt otkazati

- Značajna prednost faznog pristupa razvoju sustava jest u tome što stvara nekoliko prilika da se ponovno ocijeni izvodljivost.
- Dugoročno gledano, otkazani projekti su jeftiniji od ugrađenih promašaja.
- Većina analitičara propusti podesiti procjenu troškova i rokova prilikom proširenja područja projekta. Kao rezultat, analitičar često nepotrebno prihvaća odgovornost za premašaj troškova i rokova.
- Pristup “puzećeg” preuzimanja obveza:
 - Višestruke provjere izvodljivosti su ugrađene u metodologiju razvoja sustava.
 - U bilo kojoj točki provjere, svi se već učinjeni troškovi smatraju nepovratnima i nevažnima za daljnje odlučivanje.
 - Projekt treba u svakoj od točaka ponovno vrednovati da se utvrdi da li je još uvijek izvodljiv.
- U svakoj točki provjere analitičar treba razmotriti:
 - ukidanje projekta ako više nije izvodljiv,
 - ponovnu procjenu troškova i rokova ako se opseg projekta povećava,
 - sužavanje opsega projekta ukoliko su proračun i rokovi fiksirani, ali nedovoljni za ostvarenje svih ciljeva projekta.



Projektiranje i planiranje sustava

❑ Podijeli pa vladaj

- Svi su sustavi dijelovi većih sustava (nadsustava),
- Praktički svi sustavi sadrže manje sustave (podsustave),
- Sustav se podijeli u barem dva podsustava da bi se njime lakše ovladalo i sagradilo veći sustav,
- Podjelom većeg problema (sustava) u manje dijelove (podsustave) kojima je lakše upravljati, analitičaru se pojednostavljuje postupak rješavanja problema.

❑ Projektirati sustav tako da prati rast i promjene

- Mnogi analitičari upadnu u zamku izradivši sustav koji odgovara isključivo trenutnim zahtjevima,
- Entropijom se naziva prirodno i neizbježno kvarenje bilo kojeg sustava,
- Kada tijekom faze podrške sustavu, troškovi održavanja premaše troškove izgradnje novog sustava, sustav je zastario.



Resursi

- Association for Information Systems, <http://home.aisnet.org/>
- Kontrolne liste za provjeru i predlošci projektne dokumentacije nalaze se na <http://www.construx.com>
- Prilagodljivi razvojni proces (Adaptable Process Model) opisan je na <http://www.rsps.com>
- <http://www.itjungle.com/tfh/tfh053105-story03.html>



Reference

- Avison, D.E. & Fitzgerald, G. Information Systems development: methodologies, techniques and tools. 2nd. ed. McGraw-Hill, 1998.
- Awad A.M. System Analysis and Design, Irwin, 1985.
- Blum, B.I. A taxonomy of software development methods. Comm. of the ACM, vol. 37(11), 1994, 82-94.
- Boehm, B.W. Seven Basic Principles of Software Engineering, Journal of Systems & Software, 3(1) | March 1983, pp. 3-24.
- Brooks, F.P. The Mythical Man Month. Addison Wesley, 1975.
- Brooks, F.P. No silver bullet: essence and accidents of software engineering. Computer vol.20 (no 4), 1987, 10-19.
- Davis W.S, Yen D.C. The Information System Consultant's Handbook: Systems Analysis and Design. CRC Press, 1999.
- Dennis A., Wixom B.H., Tegarden D. Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach, 2nd Edition, 2005



Reference

- IEEE Std 610.12-1990 "Standard Glossary of Software Engineering Terminology" in IEEE Software Engineering Standards Collection, 1994 Edition, IEEE, New York, 1994, p. 67.
- Harel, D. Biting the Silver Bullet: Toward Brighter Future for System Development, Computer, 25 (1), 1992, 8-20.
- Hirschheim, R. and Klein, H.K. Four paradigms of information systems development, Comm. of the ACM, 32 (10), 1989, 1199 - 1216.
- Maciaszek L. Requirements Analysis and System Design: Developing Information Systems with UML, 3/ed, Addison Wesley Higher Education, 2007.
- McConnell S. Software Project Survival Guide Microsoft Press, 1998.
- McConnell S. : Rapid Development: Taming Wild Software Schedules. Redmond, Wa.: Microsoft Press, 1996.



Strategija organizacije Identifikacija i selekcija projekata Pokretanje i planiranje projekata

Corporate strategy
Project Identification and Selection
Project Initiation and Planning

Strateško planiranje poslovanja

- **Strateško planiranje poslovanja**
 - Definiranje poslovne strategije, dugoročno planiranje resursa i akcija
 - Uprava definira viziju i misiju organizacije, tj. strateške ciljeve.
 - misija – svrha (postojanja) organizacije, određuje postojeće stanje i procese
 - vizija – namjeravano, očekivano stanje u daljoj budućnosti
 - primjer: <http://www.zaba.hr> (O nama)
 - Na temelju toga definiraju se poslovni ciljevi, poslovni procesi i zadaci kojima će organizacija u planiranom razdoblju ispuniti svoju misiju.
 - što se želi postići (vrijednosti): prepoznatljivost, kvaliteta, prihodi, ...
 - kako to postići: promjenom organizacije, poboljšanjem sustava administracije, ...
 - povećanje proizvodnosti ulaganjem u nove proizvodne tehnologije uz istovremeno održavanje kakvoće proizvoda
 - osiguranje zadovoljstva i radnih sposobnosti zaposlenika doškolovanjem i mogućnostima napredovanja
- **Čimbenici koji utječu na postavljanje ciljeva**
 - ograničenja (organizacijska, financijska, zakonska, ...)
 - potrebe i želje uprave, posloводства, zaposlenika (ugled, utjecaj, ...)
- **Vremenski okviri, razdoblje [Awad, 1985]:**
 - kratkoročno, obično manje od 2 godine
 - srednjeročno, 2-5 godina
 - dugoročno, više od 5 godina



Planiranje informacijskog sustava

❑ Traženje odgovora na pitanja:

- Čime se organizacija bavi (grana, proizvodi, tržište, konkurencija)?
- Koji su problemi, zadaće i ciljevi poslovnog sustava?
- Koja je željena uloga IS u postizanju postavljenih ciljeva?
- Koje aplikacije postoje, čemu i kako služe, koje i kakve podatke sadrže?
- Postoje li aplikacije čiji je razvoj u tijeku? U kojem su stadiju razvojni projekti?
- Koje su potrebne aplikacije?
- Koji su raspoloživi resursi (osoblje, tehnička sredstva, tehnologija, ...)?

❑ Razlozi zbog kojih treba planirati IS

- U Organizaciji sastavljenoj od više cjelina, kao što su Uprava, Financije, Proizvodnja i Informatika postoji više sustava ili aplikacija (informatičkih otoka) uz posljedicu:
 - umnožavanje informacija uz različito tumačenje u različitim dijelovima
 - nepotpunost informacija, naročito kada svaka cjelina prikuplja samo njoj važne informacije
 - problem povezivanja informacija pri pokušaju cjelovite interpretacije
 - problem različite arhitekture, različitog posluživanja, razmjene podataka i održavanja



Strateško planiranje IS

❑ Tradicionalno planiranje IS

- provodi se odvojeno od poslovnog planiranja ili
- provodi se kao reakcija na promjene u poslovnoj politici

❑ Strateško planiranje IS

- procjena alternativnih usluga, aplikacija i infrastrukture
- uspostava smjera i prioriteta usklađivanja informacijskih servisa sukladno misiji, viziji i ciljevima organizacije
- planiranje IS sukladno strategiji razvoja organizacije, tako da informatizacija bude potpora promjeni organizacije i poslovnih procesa
- primjena metoda analize i dizajna na istraživanje poslovnog sustava s ciljem definiranja općeg (sveobuhvatnog) plana i arhitekture IS čiji razvoj slijedi

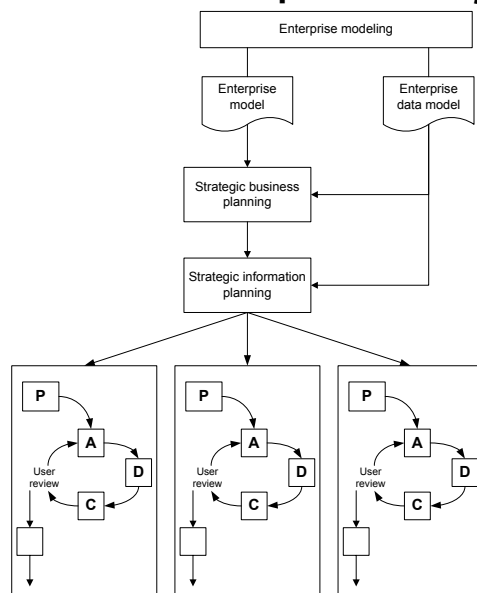
❑ U praksi ...

- Umjesto prema strateškom planu, organizacija se "dovodi u red" tijekom informatizacije i s pomoću informatizacije.
- Analizom sustava evidentiraju se problemi i slabosti poslovnih procesa.
- Dizajn sustava predlažu se ili nameću poboljšanja.

❑ Primjer: Pilot projekt informatizacije visokih učilišta



Okosnica plana razvoja sustava



R. McLeod, E. Jordan (2002).
Systems Development: A Project
Management Approach, ISBN: 0-
471-22089-2, Wiley Higher
Education



Primjer strategije: Društvo (1)

❑ Poslovna strategija

- Ključne vrijednosti su nematerijalne:
 - Tradicija, Ugled, Povjerenje, Iskustvo, Znanje, Motivacija, IS
- Potpora IT za konkurentnost na tržištu
 - ne može se kupiti (od/za izravnog konkurenta)

❑ Orijentacija ka klijentu (CRM)

- Ponuda proizvoda
 - ponuda usmjerena na klijente
- Zadovoljavati stalno promjenjive potrebe klijenata
 - visoko personalizirani proizvodi
 - djelotvorna prodaja putem kanala distribucije, uključivši mobilnu prodaju

❑ Razvoj poslovne inteligencije (BI)

- pronalaženje skrivenih zakonitosti (znanja)
- potpora poslovnom odlučivanju:
 - informacije: pravovremene, pouzdane, relevantne
- doprinos povećanju konkurentnosti i dinamičnosti poslovanja



Primjer strategije: Društvo (2)

❑ Infrastruktura

- Stanje: prikladno odabrana i adekvatna potrebama
- Budućnost: zadržavanje + proširenje ~ potrebe
- Poboljšanja: upravljanje IT infrastrukturom
 - ugradnja koncepata i praksi za upravljanje i nadzor informacijskih servisa
 - koristi: definiranost i efikasnost procesa IT potpore
 - npr. uspostava jedinstvenog kontaktnog centra (service desk) za pomoć korisnicima IT sustava te pozivnog (call) centra za krajnje korisnike

❑ Usvajanje nadolazećih regulativa

- Financijska industrija sve više regulirana (Basel II, Solvency II)
- Značajni zahtjevi na infrastrukturu i ostale dijelove IS

❑ Sigurnost i zaštita informacijskog sustava

- uvesti SABSA, ISO (COBIT)
- zaštita od neovlaštenog pristupa
- oporavak nakon incidenta
- osiguranje kontinuiteta poslovanja, rezervna računalna lokacija
- formirati odbor za informacijsku sigurnost



FER \ Fertalj: Razvoj informacijskih sustava

89

Primjer strategije: Društvo (3)

❑ Poslovne aplikacije

- Zadovoljavajuća funkcionalnost
 - prijedlozi konkretnih poboljšanja – unaprjeđenje, zamjena, integracija ...
- Nedostatna integriranost
 - poteškoće u iskoristivosti, prepreka daljem razvoju (npr. CRM, jedinstvena autentifikacija/autorizacija korisnika)
 - temeljem procjene
 - » Service Oriented Architecture (SOA)
 - » Event Driven Architecture (EDA)
 - » Enterprise Service Bus (ESB)
 - integraciju temeljiti na matičnim podacima i standardnim sučeljima

❑ Arhitektura IS

- prelazak sa dvoslojne na višeslojne web aplikacije
- odabir razvojnog alata uvažavajući strategiju proizvođača RDBMS
- odabir i migracija razvojne platforme - pilot projektima
- upravljanje projektima i razvojem aplikacija – agilnost, modeliranje procesa



FER \ Fertalj: Razvoj informacijskih sustava

90

Primjer strategije: Društvo (4)

❑ Upravljanje informacijskim sustavom

- uskladiti ciljeve i djelovanje Informatike sa strategijom i ciljevima poslovanja
- Problem - neuređenost odnosa Informatike i poslovanja
 - zahtjevi na Informatiku dolaze kaotično
 - odluke bez Informatike
- Osnovati Odbor za informatiku (IT Steering Committee)
 - istaknuti predstavnici poslovnih područja
 - usmjeravanje, prioritizacija i nadzor poslova Informatike
- Reorganizacija Sektora informatike
 - Ured za upravljanje projektima
 - Centar kompetencije za poslovnu inteligenciju
 - Služba za poslovne procese
 - Upravljanje IT servisima prema ITIL smjernicama

❑ Ljudski resursi

- Za standardne funkcije (financije, kadrovi, ...) moguća nabava standardnih aplikacija
- Vlastiti razvoj ili potpuno upravljanje nabavljenim softverom za osnovnu djelatnost
- Outsourcing, insourcing, multisourcing kao modaliteti angažmana vanjskih usluga
 - kriterij: tehnička, tehnološka i podatkovna kompatibilnost
- nedostatak resursa - postupno udvostručiti broj stručnjaka
 - stimulativno zapošljavanje i zadržavanje osoblja



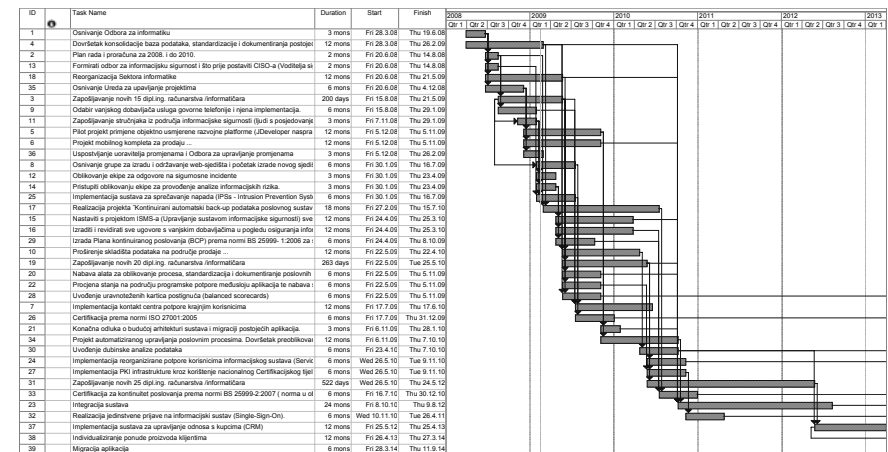
FER \ Fertalj: Razvoj informacijskih sustava

91

Primjer strategije: Društvo (5)

❑ Plan informatizacije za razdoblje 2008-2012

■ Društvo-Strategija.mpp



FER \ Fertalj: Razvoj informacijskih sustava

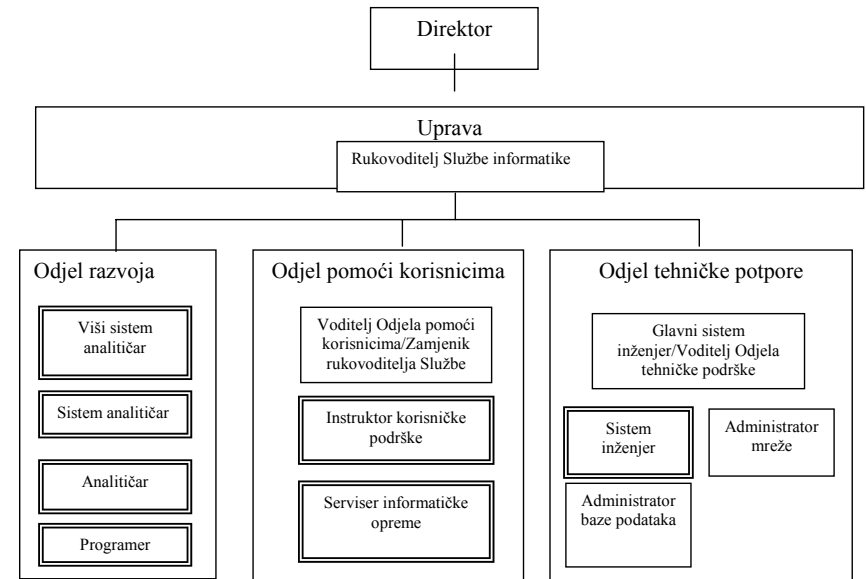
92

Primjer strategije: Poduzeće (1)

- ❑ **Prosinac 2005.**
 - Tridesetak aplikacija
 - Informacijski otoci
 - Različite tehnologije
 - Zastarjela tehnologija
 - Raspršeno osoblje
 - Zastarjela podjela posla
- ❑ **Primjena**
 - Funkcionalnost je dobra uz 3 (od 17) izuzetka.
 - Aplikacije su izrazito potrebne.
 - Korisnicima je lako koristiti aplikacije.
 - Većina aplikacija je dosta fleksibilna, samo dvije su izuzetak.
 - Točnost je prilična.
 - Stabilnost zadovoljavajuća.
 - Brzina vrlo dobra.
 - Dostupnost i adekvatnost opreme više nego vrlo dobra uz samo 1 izuzetak.
 - Korisnička dokumentacija je osrednja.
 - Obučenost korisnika je dobra, ali bi mogla biti i bolja.
 - Podrška je svagdje dobra do odlična.
- ❑ **Definiranje smjernica razvoja i organizacije informatičke podrške**
 - sistematizacija radnih mjesta u informatici i reorganizacija službe
 - poduka informatičkih djelatnika
- ❑ **Definiranje smjernica razvoja i arhitekture računalnog sustava**
 - integracija sustava nad zajedničkim modelom podataka
 - razvoj aplikacija
- ❑ **Smjernice upravljanja IS-om**
- ❑ **Smjernice upravljanja sigurnosti IS-a**
- ❑ **Taktički plan provedbe projekta izgradnje IS**



Primjer strategije: Poduzeće (2)



Primjer strategije: Poduzeće (3)

- ❑ **Plan za razdoblje 2006-2009.**
 - Poduzeće-Strategija.mpp

ID	Task Name	Duration	Start	Finish	5	2
1	Održavanje aplikacija	900 days	Mon 21.11.05	Fri 1.5.09	H2	I
2	Faza pripreme	274 days	Mon 16.1.06	Thu 1.2.07		
51	Vladanje informacijskim sustavom	188 days	Mon 13.2.06	Wed 1.11.06		
126	Izgradnja imeničke infrastrukture	53 days	Wed 5.4.06	Fri 16.6.06		
49	Sistemski poslovi	800 days	Mon 17.4.06	Fri 8.5.09		
94	Osiguranje IT sustava na razini cjelokupne infrastrukture	17 days	Mon 19.6.06	Tue 11.7.06		
108	Izgradnja sigurnosnih politika/procedura/uputa usklađeni	40 days	Wed 12.7.06	Tue 5.9.06		
146	Implementacija sustava za upravljanje IT infrastrukturom	34 days	Wed 6.9.06	Mon 23.10.06		
16	Aplikacije osnovne djelatnosti	334 days	Fri 2.2.07	Wed 14.5.08		
28	Poslovne i računovodstvene aplikacije	476 days	Fri 2.2.07	Fri 28.11.08		
50	Skladište podataka	6 mons	Mon 1.12.05	Fri 15.5.09		



Primjer strategije: Poduzeće (4)

- ❑ **Ažurirani plan pripreme 2009.**

ID	Task Name	Duration	Start	Finish	Qtr 1, Jan
1	Faza pripreme	80 days	Mon 26.1.09	Fri 22.5.09	
2	Uspostava infrastrukture projekta (TFS, SP)	5 days	Mon 26.1.09	Fri 30.1.09	
3	Školovanje za modeliranje podataka	5 days	Mon 2.2.09	Fri 6.2.09	
4	Školovanje za SQL	5 days	Mon 16.2.09	Fri 20.2.09	
5	Školovanje za razvojno okruženje	15 days	Mon 2.3.09	Fri 20.3.09	
6	Formiranje ekipa	30 days	Mon 23.3.09	Fri 8.5.09	
7	Metodologija razvoja	30 days	Mon 26.1.09	Fri 6.3.09	
8	Razvoj programske okosnice	60 days	Mon 26.1.09	Fri 24.4.09	
9	Poduka primjene metodologije i okosnice	10 days	Mon 11.5.09	Fri 22.5.09	



Odabir i pokretanje projekata



Identifikacija projekta

- ❑ **Sponzor projekta = osoba ili grupa koja ustanovi poslovne potrebe**
 - najčešće netko iz poslovnog područja (Financije, Studentska, Proizvodnja)
- ❑ **Pokretači promjena**
 - korisnici - nezadovoljstvo aplikacijama i/ili podacima (nepouzdanost, nedostupnost, manjkavost), prijedlog novih mogućnosti
 - npr. nestabilnost aplikacija, potrebe za podacima, potreba za novim funkcijama
 - reorganizacija, utjecaji okoline – promjene organizacijske strukture, promjene poslovnih procesa
 - npr. promjene na Sveučilištu uzrokovane novim Zakonom, restrukturiranje studija, smjerova i zavoda
 - pokazatelji poslovanja
 - npr. pad prodaje, uska grla proizvodnje, neplanirano i nejasno povećanje troškova
 - zastarjela tehnologija – zahtjev za novim tehnologijama
 - npr. razvojni alati (problem održavanja), korisnička sučelja (Internet), baze podataka, neintegriranost sustava (informacijski otoci)

Predlaganje projekata

- ❑ **Predlaganje projekata**
 - temeljem (prethodno napravljenog) plana informatizacije
 - planu prethodi analiza i prijedlog strategije
 - temeljem zahtjeva na sustav unutar organizacije
 - sponzor projekta interno dostavlja zahtjev na sustav (system request) uočavanjem poslovnih potreba
 - provodi se snimka stanja i izrada internog prijedloga - povelje projekta
 - temeljem prijava na natječaj organizacije/institucije koja traži rješenje
 - u slučaju natječaja otvorenog tipa vanjski predlagatelj navodi potencijalno tržište ili korisnike te očekivanu korist
 - primjeri:
 - Program hrvatskog inovacijskog tehnološkog razvitka (HITRA)
 - Razvoj na znanju utemeljenih poduzeća (RAZUM)
 - natječaj MZOŠ za prijavu i-projekta (iprojekti.mzos.hr)
- ❑ **Odabir projekta (project selection)**
 - povjerenstvo/odbor za odabir (steering committee, approval comitee) odobrava projekte, nakon čega slijedi dublja analiza i ponovna procjena

Zahtjev na sustav

□ Zahtjev na sustav

- sažetak projekta (sponzor, naziv, cilj, svrha)
 - npr. odjel Financija, dopredsjednik Marketinga
- poslovne potrebe – poslovni razlozi za pokretanje
 - npr. povećanje prodaje, povećanje udjela na tržištu, poboljšanje pristupa podacima, poboljšanje usluge krajnjim korisnicima
- poslovni zahtjevi – poboljšane mogućnosti poslovanja
 - npr. pristup putem interneta, prikupljanje podataka o kupcima, izrada upravljačkih izvješća
- očekivana korist – poslovna vrijednost
 - npr. povećanje prodaje 3%, povećanje udjela na tržištu 1%, uštede u održavanju sustava
- posebnosti i ograničenja - argumenti relevantni za odobrenje
 - npr. rok do kraja poslovne godine, kompatibilnost s politikom sigurnosti i zaštite

□ Primjer: \Planiranje\ZahtjevZaUslugamaIS



Pokretanje projekata (project initiation)

□ Snimka stanja, početno istraživanje (survey phase, initial study, preliminary investigation)

- prethodno istraživanje, tj. istraživanje koje prethodi projektu
- pokreće se na zahtjev za sustav
- prepoznavanje problema i potreba
- traženje odgovora na pitanje "Da li je projekt vrijedan pažnje?"
- rezultat je povelja projekta (project charter) i početni (grubi, okvirni) plan projekta
- povjerenstvo odobrava, odbacuje ili odgađa prijedlog

□ Analiza izvodljivosti, Faza proučavanja problema (study phase)

- produbljivanje snimke
- uočavanje problema, uzroka i mogućih posljedica
- određivanje doseg
- preciziranje ciljeva, prijedlozi rješenja, procjena izvodljivosti
- "Da li su problemi vrijedni rješavanja?", "Da li je gradnja isplativa?"
- povjerenstvo odobrava, odbacuje ili reducira projekt

□ Upravljanje projektom (→ Organizacija i upravljanje projektom)

- Izrada plana rada
- Ustroj projektne ekipe
 - uključivanje sudionika iz različitih djelatnosti, npr. djelatnici različitih poslovnih područja ili organizacijskih jedinica, uprava, vanjski konzultanti
 - važno je osigurati predanost dionika zajedničkom cilju (commitment)
- Uspostava upravljanja i nadzora projekta



Snimka stanja

□ Snimka stanja

- brzo istraživanje i procjena problema, mogućih prilika i direktiva
 - problem: neželjena situacija koja sprječava potpuno ispunjenje svrhe, postizanje ciljeva, obavljanje zadaća
 - moguća prilika: mogućnost pozitivne promjene, čak i kada ne postoji problem
 - direktiva: zahtjev ili ograničenje nametnuti poslovnom politikom (npr. pravilnik) ili vanjskim utjecajem (npr. zakon)
- određivanje početnog dosega projekta i početnog plana projekta

□ Snimka poslovnog sustava

- pregled poslovnih planova, ako takvi postoje ili nisu "tajna"
- prikupljanje informacija, najčešće intervjuiranjem vlasnika i viših rukovoditelja
- evidentiranje problema i prijedloga

□ Snimka postojećeg (postojećih) IS

- identifikacija korisnika i opsega postojećeg IS
- uočavanje problema i nedostataka postojećeg IS
- procjena potreba za nadgradnjom (poboljšanjima)
- procjena potreba za izmjenama (prilagodbom i popravcima)
- procjena potreba za izradom novih IS ili podsustava IS

□ Primjeri: \Planiranje\ISFER-Snimka-*

- temeljem jednog kruga intervjua, prikupljanja dokumentacije i provjere



Primjer: postojeći problemi, prijedlozi rješenja

Kratko obrazloženje problema, mogućnosti ili direktive	Hitnost	Vidljivost	Korist	Prioritet	Predloženo rješenje
1. Vrijeme odgovora na narudžbu mjereno od vremena zaprimanja narudžbe do isporuke klijentu se povećalo na prosječno 15 dana	HITNO	Visoka	175,000	2	Novi razvoj
2. Nedavno preuzimanje kompanija: <i>Privatna predstava i Filmsko platno</i> nametnuto je povećanje zahtjeva za protokom informacija i dokumenata.	6 mjeseci	Srednja	75,000	2	Novi razvoj
3. Trenutno 3 različita sustava za unos narudžbi servisiraju odjele za audio, video i video igre. Svaki sustav ima vlastito sučelje prema različitim skladišnom sustavu, pa treba objediniti skladišnu evidenciju.	6 mjeseci	Srednja	515,000	2	Novi razvoj
4. Postoji nedostatak pristupa informacijama nužnim za upravljanje i donošenje odluka. Ovo će se još pogoršati preuzimanjem dva dodatna sustava za obradu narudžbi (iz <i>Privatna predstava i Filmsko platno</i>).	12 mjeseci	Niska	15,000	3	Po razviku novog sustava, pružiti korisnicima lako svladive alate za pisanje izvještaja.
5. Izražena je nedosljednost (nekonzistentnost) između podataka u evedencijama članova i narudžbi.	3 mjeseca	Visoka	35,000	1	Brza ispravka, a zatim novi razvoj
6. Sustavi datoteka u <i>Privatna predstava i Filmsko platno</i> nisu kompatibilni s onim u <i>Zvučna pozornica</i> . Problemi s podacima obuhvaćaju nedosljednosti u podacima i nedostatak upravljanja ulazom i izmjenama.	6 mjeseci	Srednja	nepoznato	2	Novi razvoj, dodatna ocjena koristi može povećati žurnost
7. Postoji mogućnost uvođenja sustava naručivanja putem Interneta, ali su sigurnost i kontrola pristupa problematični.	12 mjeseci	Niska	nepoznato	4	Buduće verzije tek razvijenog sustava
8. Postojeći sustav unosa narudžbi nije kompatibilan s planiranim sustavom za automatsku identifikaciju (štipčasti kod) koji se razvija za skladište.	3 mjeseca	Visoka	65,000	1	Brza ispravka, a zatim novi razvoj

•Vidljivost: U koliko mjeri će rješenje ili novi sustav biti dostupni korisnicima.

•Korist: Pausalna procjena koliko bi rješenje povećalo dobit ili smanjilo trošak u jednoj godini.



Početno planiranje projekta

❑ Određivanje cilja i svrhe projekta (goal, objective)

- izdvajanje zadataka koji su sukladni poslovnim ciljevima, a mogu biti informatizirani
- Koji se probleme nastoji riješiti projektom?
- Koje su očekivane koristi?

❑ Doseg i razgraničenje projekata ili pod-projekata (Scope)

- System boundary, Constraints, Objectives, Permissions, End products
- Koje su granice sustava? Koji će zahtjevi biti ispunjeni?
- Koja su ograničenja (tehnoška, organizacijska, financijska)?
 - Što ne može biti napravljeno? Što neće biti napravljeno?
- Što će se postići? Čemu će služiti?
- Tko će, kako i pod kojim uvjetima moći koristiti rješenje?
- Kako se mjeri ili određuje dovršenost i uspjeh (neuspjeh)?

❑ Vremensko planiranje

- određivanje prioriteta zadataka i okvirnih rokova prioriteta



Izrada početnog plana

❑ Izrada početnog plana razvoja IS

- početni glavni plan projekta (master plan, baseline plan)
 - potprojekti, prioriteta, ...
- mogući načini podjele posla na cjeline tako da:
 - cjelinu može obaviti jedna osoba ili ekipa
 - cjelina se može obaviti jednom metodom
 - posao završi jednim "proizvodom" (dokumentom, aplikacijom ili podsustavom)
- okvirni vremenski plan po fazama
- dorađuje se i ažurira sukladno napretku projekta

❑ Odobrenje (nastavka) projekta

- konsolidirani prijedlog projekta, povelja projekta (project charter) može poslužiti kao interni ugovor projekta



Primjer: Početni plan informacijskog sustava

❑ Primjeri: \Planiranje\PlanStandardni, RIS-OkvirniPlan

❑ Primjer početnog (preliminarnog) plana za Ministarstvo

1. Nabavka sustava za upravljanje bazama podataka
2. Tečaj opće informatičke pismenosti za rukovoditelje odjela
3. Tečaj za programere u jeziku za upravljanje bazama podataka
4. Tečaj za administratore baze podataka
5. Izrada prototipa programske podrške za i-tu fazu realizacije
6. Izrada α - verzije aplikacije
7. Testiranje α - verzije u Informacijskim sustavima
8. Uklanjanje uočenih nedostataka i izrada β - verzije programa
9. Tečaj za odabrane korisnike na β - verziji
10. Testiranje kod korisnika u paralelnom radu s dosadašnjim programom
11. Uklanjanje nedostataka i izrada stabilne verzije
12. Zamjena dotadašnjeg programa novim programom, uz preuzimanje podataka
13. Tečaj za ostale korisnike
14. Uvođenje korištenja programa kod ostalih korisnika
15. Tečaj za upoznavanje s mogućnostima programa za odabrano posloводство
16. Prikupljanje primjedbi korisnika i novih zahtjeva
17. Izrada sljedeće faze/verzije programa. Povratak na točku 5).



Povelja projekta

❑ Primjeri: \Planiranje\

- PrijavaProjektaPrimjenelT
- PrijedlogProjektaPIS



Analiza izvodljivosti

Analiza problema

- ❑ **Produbljivanje analize za projekte koji prođu početnu selekciju**
 - "Da li su problemi vrijedni rješavanja?", "Da li je gradnja isplativa?"
 - Detaljnija analiza problema, njihovih uzroka i posljedica
 - "Ne pokušavaj popraviti prije nego shvatiš kako radi!"
 - Analiza poslovnih procesa
 - Koji su najveći problemi?
 - Koja su moguća rješenja problema?
 - Kako informatizacija može pomoći?
 - Grubo modeliranje postojećeg sustava
- ❑ **Mogu se koristiti različite formalne metode**
 - Analiza kritičnih faktora uspjeha (CSF - Critical Success Factors)
 - čimbenici kojima poslovodstvo posvećuje posebnu pažnju
 - čimbenici koji razmjerno brzo i lako doprinose ostvarivanju ciljeva (npr. brzi odgovor na zahtjeve klijenata, odnos cijene i kvalitete proizvoda)
 - Analiza izvodljivosti i procjena troškova-koristi



Primjer: uzroci i posljedice, ciljevi i ograničenja

Istraživanje problema, uzroka i posljedica

ANALIZA UZROKA I POSLJEDICA		CILJEVI I OGRANIČENJA SUSTAVA	
Problem ili mogućnost	Uzroci i posljedice	Ciljevi sustava	Ograničenja sustava
1. Vrijeme odgovora na narudžbu je neprihvatljivo dugo.	1. Promet je povećan, a broj službenika smanjen. Vrijeme obrade narudžbi je isto. 2. Sustav previše ovisi o ručnom unosu. Pojedine vrijednosti se unose više puta. Posljedica je da se narudžbe obrađuju dulje nego je potrebno. 3. Središnje računalo radi na maksimumu svojih kapaciteta, što se očituje u sporijem radu aplikacija za unos narudžbi. Budući da službenici pokušavaju brže raditi, povećao se broj pogrešaka pri unosu. 4. Obrasci za prikupljanje narudžbi iz skladišta nisu oblikovani za učinkovito popunjavanje, što dodatno usporava unos narudžbi.	1. Umanjiti vrijeme unosa pojedine narudžbe za 30%. 2. Ručni unos narudžbi svesti na 50% ukupnog broja. 3. Na zaslону za ručni unos smanjiti broj potrebnih pritisaka na tipkovnicu. 4. Prenijeti unos podataka sa središnjeg na osobna računala. 5. Zamijeniti postojeće obrasce za prikupljanje narudžbi mrežnim sustavom između skladišta i prodaje tako da se eliminiira uporaba papirnatih dokumentacije.	1. Broj zaposlenih u obradi narudžbi se ne može uvećati. 2. Novi sustav mora biti kompatibilan s postojećim Windows XX standardom. 3. Novi sustav mora biti kompatibilan s već odobrenim sustavom za automatsku identifikaciju štapičastim kodom.

- ❑ **Primjer: Istraživanje uzroka**
 - Problem: pad prodaje
 - Vidljivi znak: povećani opoziv (storno) narudžbi
 - Razlog: nezadovoljstvo kupaca
 - Uzrok: spor sustav za naručivanje
- ❑ **Zadatak analitičara je razdvojiti uzroke i posljedice problema.**
 - Na primjer, sporost sustava nije izoliran problem nego može biti posljedica pomanjkanja osoblja, nezainteresiranosti osoblja za posao ili posljedica ručne obrade podataka.



Postavljanje ciljeva

❑ Primjeri poslovnih ciljeva

- Potpomoći reorganizaciju u tržišno orijentirano poduzeće prema EU normama.
- Osigurati informacije o izvorima, razlozima i mjestu nastanka svakog troška u sustavu.
- Uskladiti hijerarhiju odlučivanja s hijerarhijom u poduzeću.
- Racionalizirati utrošak novca za ...

❑ Primjer ciljeva IS

- Problem: Predugo vrijeme unosa narudžbi
- Cilj: Ubrzanje unosa
 - Kriterij: performance, izvedba
- Loš primjer: Povećanje ili smanjenje
 - "Povećati broj unijetih narudžbi u 24 sata za 5%"
- Bolji primjer: Učinkovitost provedbe
 - "Unijeti 98% zaprimljenih narudžbi unutar 24 sata"
- Kriterij mora biti mjerljiv
 - Apsolutni iznos ili relativni iznos konkretne vrijednosti
 - Može biti binaran – postoji ili ne postoji, npr. Omogućiti *online* pretraživanje



Ograničenja

❑ Osoblje

- Odjel informatike smije zaposliti najviše tri stalna zaposlenika

❑ Materijalni trošak

- Nabavka uredskog i potrošnog materijala ne smije premašiti XXX HRK

❑ Računalna oprema

- Projekt se mora obaviti bez nabavke novog hardvera (gospodarstvo?)
- Trošak opreme poželjno predstavlja barem 33% budžeta (znanost!)

❑ Financijska sredstva

- Ukupni budžet projekta je XXXXX HRK (uvijek manji od traženog!)
- Naknade izvođačima ne smiju premašiti XX% ukupnog iznosa



Ključni čimbenici uspjeha

❑ Primjer: Urudžbeni zapisnik

- Izrada adekvatne programske podrške
- Unos svih podataka iz knjiga urudžbenog zapisnika u bazu podataka
- Povezivanje sa službama na fakultetu koji koriste iste dokumente (napraviti dijeljene dokumente)
- Osposobiti djelatnike za rad na novom sustavu

❑ Primjer: CIP

- odobrenje samostalnog budžeta, te dodatno investiranje u softver, hardver i sustav rezervnog napajanja
- integracija sustava
- investicije u fizičku zaštitu (protupožarni i protuprovalni sustav)
- zapošljavanje dodatnih djelatnika sa SSS



Analiza izvodljivosti

❑ Analiza, Studija izvodljivosti (feasibility analysis, feasibility study)

- mjerenje korisnosti, praktičnosti i isplativosti projekta
- procjenjuje se da li je projekt izvodljiv s obzirom na raspoloživa sredstva
- procjenjuje se da li projekt omogućuje poboljšanja
- radi se izvješće o izvodljivosti (feasibility report) i prezentira se relevantnim dionicima ili dostavlja odboru za odabir
 - eventualni povratak u studiju izvodljivosti → revidirano izvješće
- trebala bi se raditi tijekom planiranja, ali i kasnije (npr. nakon faze analize)
 - nakon odluke o pokretanju projekta složenost i opseg projekta mogu se promijeniti pa početno izvodljiv projekt može postati neizvodljiv

❑ Svaki projekt je u početku neodređen i "mutan"

- "bistri" se prikupljanjem informacija i inkrementalnom doradom
- točnost procjene izvodljivosti raste s dubinom analize
- procjena izvodljivosti može biti ponovljena u kontrolnim točkama između ključnih faza projekta



Organizacijska izvodljivost

» Ima li smisla graditi? Hoće li se koristiti?

☐ Operativna izvodljivost

- procjena hitnosti rješavanja problema (planiranje)
- procjena prihvatljivosti rješenja (kasnije faze)

☐ Vrijedi li rješavati problem? Da li predloženo rješenje rješava problem?

» PIECES koncept

- P [performance]
 - potreba za poboljšanjem performansi, npr. protočnost i odziv sustava
- I [information]
 - potreba za poboljšanjem informacija (i podataka), pravodobne, prikladne, ažurne, korisne informacije
- E [economics]
 - potreba za poboljšanjem ekonomije, kontrole troškova ili povećanje profita, npr. usluge i kapaciteti, smanjenje troškova
- C [control]
 - potreba za poboljšanjem kontrole ili osiguranja, npr. točnost, sigurnost i zaštita podataka
- E [efficiency]
 - potreba za poboljšanjem efikasnosti ljudi i procesa, npr. maksimalna uporaba raspoloživih resursa (ljudi, vrijeme)
- S [service]
 - potreba za poboljšanjem usluge kupcima, dobavljačima, partnerima, zaposlenicima, npr. poželjni i pouzdani servisi, elastičnost i mogućnost prilagodbe



Organizacijska izvodljivost

☐ Socio-psihološka izvodljivost

- Da li će rješenje zadovoljiti korisnikove potrebe?
- Do kojeg stupnja?
- Kako će rješenje promijeniti korisnikovu radnu okolinu?

☐ Koji je stav korisnika prema rješenju? Da li će se sustav koristiti?

- Podrška uprave
- Prihvatanje od krajnjih korisnika – otpori njihovoj budućoj ulozi ili tehničkom rješenju i kako ih prevladati
- Promjena radnog okruženja, procedura – prilagodba promjenama
- Rizici slabe informatičke pismenosti, straha od neznanja, novog načina rada ili gubitka posla, otpora radi gubitka kontrole i utjecaja

☐ Procjena upotrebljivosti (najčešće prototipom u kasnijim fazama)

- Krivulja učenja – vrijeme osposobljavanja potrebno za postizanje pune primjene
- Lakoća korištenja – jednostavno sučelje za početnike i povremene korisnike, složenije operacije za iskusne korisnike
- Zadovoljstvo – ponuđenom rješenju korisnik daje prednost pred postojećim



Tehničko-tehnološka izvodljivost

» Može li se sagraditi? Može li se kupiti?

☐ Procjena mogućih rješenja i alternativa

- procjena postojećih rješenja na tržištu ili u drugim organizacijama
- procjena primjenjivosti različitih tehnologija

☐ Primjenjivost rješenja, tehnologije

- Može li se tehnologija jednostavno primijeniti?
- Najsuvremenija tehnologija - naprednost, zanimljivost, rizik
- Zrela i dokazana tehnologija - sigurnost, bolja podrška

☐ Raspoloživost tehnologije (podrazumijeva se da je primjenjiva)

- Može li se nabaviti?
- Ako je riječ o gotovom rješenju (IS iz dućana), ima li to rješenje potrebne karakteristike? Treba li ga i u kojoj mjeri prilagoditi ili doraditi?

☐ Stručnost

- Postoji li potrebna stručnost za primjenu tehnologije?
- Postoji li potrebna vještina za očekivani završetak prema rasporedu?

☐ Primjer: Uprava-IzborAlata



Vremenska izvodljivost

» Kada će biti gotovo? Može li biti gotovo na vrijeme?

☐ Prihvatljivost vremenskog rasporeda

- Razmatra se opravdanost rokova s obzirom na raspoloživu stručnost

☐ Poželjni rokovi

– Bolje je isporučiti ispravan sustav nešto kasnije, nego neispravan ili beskoristan na vrijeme!!!

- kašnjenje pojedinih zadataka ili dodavanje novih zadataka (uslijed upravljanja promjenama) posmiče završetak projekta
- iterativno se ažurira ili predlaže alternativni vremenski plan

☐ Čvrsti rokovi (timeboxing)

1. postavlja se čvrsti rok isporuke
2. određuju prioriteta funkcionalnosti koju treba ugraditi
3. gradi se jezgra sustava (nužne funkcionalnosti)
4. odgađa se ugradnja funkcionalnosti koja ne može biti dovršena na vrijeme
5. u roku se isporučuje sustav s dosegnutom funkcionalnošću
6. ponavljaju se koraci 3. do 5. s nadodanom funkcionalnošću

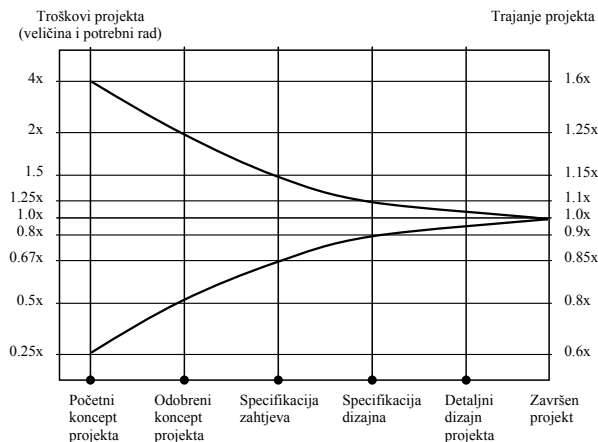
☐ Koliko je razuman predloženi plan?

- iskustvo na sličnim projektima (najbolje)
- modeli koji su predviđeni za tip industrije i tip razvoja (npr. COCOMO)
- opći modeli razvojnog procesa (npr. Rapid Development)



Preciznost procjene

Faza projekta	Potrebni rad i veličina projekta		Trajanje projekta	
	Optimistično	Pesimistično	Optimistično	Pesimistično
Početni koncept projekta	0.25	4.0	0.60	1.6
Odobreni koncept projekta	0.50	2.0	0.80	1.25
Specifikacija zahtjeva	0.67	1.5	0.85	1.15
Specifikacija dizajna	0.80	1.25	0.90	1.10
Detaljni dizajn projekta	0.90	1.10	0.95	1.05



FER \ Fertalj: I

121

Postupak procjenjivanja

□ Osnovni koraci

- Procjena veličine projekta (broj programskih redaka ili funkcijskih točaka).
 - Ovaj korak je intelektualno najzahtjevniji i to je jedan od razloga zašto ga ljudi često preskaču.
 - Alternativno, veličina se može odrediti usporedbom u odnosu prema odgovarajućim dijelovima starog projekta
- Procjena potrebnog uloženog rada u čovjek-mjesecima (čm).
 - Može se izračunati iz procjene veličine projekta i povijesnih podataka rada na sličnim projektima.
- Procjena trajanja (vremenski plan) projekta
 - npr. u kalendarskim mjesecima.
 - Trivijalno se računa iz procjena veličine i potrebnog rada.

□ Meta korak

- Procjene treba izražavati u rasponima i, kako projekt napreduje, postupno smanjivati raspon i povećavati preciznost procjene.



FER \ Fertalj: Razvoj informacijskih sustava

122

Procjena veličine postupkom funkcijskih točaka

- **Funkcijska točka je umjetna mjera veličine programa (Albrecht 1979).**
 - Iz specifikacije zahtjeva lakše je odrediti funkcijske točke nego broj programskih redaka, i one daju točniju mjeru veličine programa u ranim fazama projekta.
- **Broj točaka određuje se iz broja i složenosti elemenata programa:**
- **Ulazi u program**
 - Ekran, ekranski obrasci, dialog prozori, kontrole i poruke kroz koje krajnji korisnik ili drugi programi dodaju, brišu ili mijenjaju podatke u programu.
 - Ovo uključuje sve ulaze koji imaju jedinstveni oblik ili jedinstvenu logiku obrade podataka.
- **Izlazi iz programa**
 - Ekran, izvještaji, grafovi ili poruke koje program stvara za krajnjeg korisnika ili druge programe.
 - Ovo uključuje sve izlaze koji imaju jedinstveni oblik ili traže različitu logiku obrade podataka od ostalih izlaza.
- **Upiti**
 - Kombinacije ulaza i izlaza gdje ulaz rezultira trenutnim i jednostavnim izlazom.
 - Izraz dolazi iz baza podataka i odnosi se na izravnu potragu za podacima koristeći jedinstveni ključ.
 - U suvremenim aplikacijama s grafičkim korisničkim sučeljem granica između upita i izlaza je nejasna jer su upiti nad bazom jednostavni, a izlazi mogu obrađivati podatke i mogu sadržavati složene formate.
- **Unutarnje logičke datoteke**
 - Glavne logičke skupine podataka koje program u potpunosti kontrolira.
 - Logička datoteka se može odnositi na običnu datoteku ili na tablicu u bazi podataka.
- **Vanjske datoteke sučelja**
 - Datoteke koje kontroliraju drugi programi s kojima naš program komunicira.



FER \ Fertalj: Razvoj informacijskih sustava

123

Računanje funkcijskih točki

□ Nemodificirani zbroj funkcijskih točaka.

□ Potrebno je

- odrediti elemente programa (ulaz, izlaze, upite i datoteke),
- odrediti njihovu složenost,
- te ih zbrojiti pomnožene određenim koeficijentom (tablica).

□ Nemodificirani zbroj može se pomnožiti faktorom utjecaja

- Računa se koliki utjecaj na produktivnost imaju čimbenici poput protoka podataka, složenosti programske logike, jednostavnosti instalacije i sl.
- Faktori mogu imati vrijednosti između 0.65 i 1.35.

Karakteristike Programa	Funkcijske točke		
	<i>mala složenost</i>	<i>srednja složenost</i>	<i>velika složenost</i>
Broj ulaza	x3	x4	x6
Broj izlaza	x4	x5	x7
Upiti	x3	x4	x6
Unutarnje datoteke	x7	x10	x15
Vanjske datoteke	x5	x7	x10



FER \ Fertalj: Razvoj informacijskih sustava

124

Primjer zbroja funkcijskih točki

- Pretpostavimo da imamo unutarnju datoteku koja sadrži podatke o djelatnicima nekog poduzeća.
- Podacima manipuliramo preko ulaza kojima unosimo, brišemo i mijenjamo podatke o djelatnicima.
- Također imamo i upit koji prikazuje trenutne podatke o djelatnicima.
- Jednom mjesečno ispisuje se telefonski imenik djelatnika u kojem se nalazi i raspored djelatnika po uredima. Ovo računamo kao izlaz.
- Telefonski imenik uključuje podatke iz datoteke o djelatnicima kojom upravlja druga aplikacija. Ovo računamo kao vanjsku datoteku.
- Pretpostavimo da je složenost svih karakteristika programa osim telefonskog imenika (izlaz) mala.

Karakteristike Programa	Funkcijske točke		
	mala složenost	srednja složenost	velika složenost
Broj ulaza	3x3 = 9	0x4 = 0	0x6 = 0
Broj izlaza	0x4 = 0	1x5 = 5	0x7 = 0
Upiti	1x3 = 3	0x4 = 0	0x6 = 0
Unutarnje datoteke	1x7 = 7	0x10 = 0	0x15 = 0
Vanjske datoteke	1x5 = 5	0x7 = 0	0x10 = 0
Nemodificirani zbroj funkcijskih točaka:			29



FER \ Fertilj: Razvoj informacijskih sustava

125

Određivanje faktora utjecaja

□ Zbroj funkcijskih točki množi se faktorom dobivenim temeljem 14 općih karakteristika sustava

- Stupanj utjecaja - Degree of Influence (DI)
- Faktor utjecaja - Value Adjust Factor (VAF) = $(TDI \cdot 0.01) + 0.65$

Opće karakteristike (GSCs)	Stupanj utjecaja (DI)
1 Prijenos/Komunikacija podataka, npr. Web	1
2 Raspodijeljena obrada, npr. klijent poslužitelj	0
3 Performanse npr. min. Vrijeme odziva	0
4 Konfiguriranje, npr. često postavljanje (setup)	3
5 Učestalost transakcija	0
6 Interaktivni unos podataka (On line data entry)	3
7 Dizajn obzirom na učinkovitost korisnika	5
8 Dinamičko ažuriranje (Online updates)	4
9 Složena obrada, npr. izračuni, pogledi	0
10. Korisnost u drugim aplikacijama	0
11. Lakoća uvođenja (instalacije)	2
12. Lakoća primjene (Operational Ease)	4
13. Raširenost (Multiple sites)	0
14. Mogućnost promjena	3
Ukupni stupanj utjecaja (TDI)	25
Value Adjust Factor (VAF) = $(TDI \cdot 0.01) + 0.65$	$(25 \cdot 0.01) + 0.65 = 0.9$



FER \ Fertilj: Razvoj informacijskih sustava

126

Procjena objektnim točkama i slučajevima korištenja

□ Objektne točke (aplikacijske točke, ne razredi!)

- Pojednostavljena alternativa procjeni funkcijskim točkama
- Procjenjuje se broj zaslonskih maski, izvješća i programskih modula za pristup podacima
- Može se koristiti u ranim fazama

□ Funkcijske i objektne točke

- mogu se odrediti temeljem dijagrama dekompozicije funkcija, dijagrama toka podataka i ER dijagrama, odnosno UC dijagrama i dijagrama razreda



FER \ Fertilj: Razvoj informacijskih sustava

127

Procjena analogijom

□ Skaliranje poznavanjem ciklusa i provedene aktivnosti

- npr. poznata je razdioba opterećenja za poslovne projekte: Planiranje (15%), Analiza (20%), Dizajn (35%), Izrada (30%)
- ako je na planiranje utrošeno 4čm, slijedi Analiza (5.33čm), Dizajn (9.33čm), Izrada (8čm)

□ Skaliranje analizom povijesnih podataka

- struktura proizvoda, vrste projekata, evidencija utrošenog vremena, ...

Project	DBMS	Entities	Relationships	Generated	Manually coded	Coding
EOM	Zim	56	27	75%	sophisticated functions	84%
ISPAP*	Zim	106	310	55%	sophisticated functions	31%
ISSA**	Informix	78	90	20%	sophisticated functions	73%
ISTM**	Informix	85	116	10%	sophisticated functions	64%
ROS	Informix	39	44	90%	sophisticated reports	58%
SEE	Zim	47	83	80%	sophisticated functions	71%

Model	Domains	Attributes	Entities	Primary Keys	Foreign Keys
Initial Model of the central system	285	285	103	99*	0
The model at the end of integration	553	636	146	151*	243
The model at the end of design	338	459	131	135	228

Program element	No. of elements	No. of source code lines	Source code size (KB)
Program module	400	250.000	8.200
Form	170	10.000	300
TOTAL	570	260.000	8.500



FER \ Fertilj: Razvoj informacijskih sustava

128

Određivanje produktivnosti

Standardne vrijednosti za LOC

- Real-time embedded systems, 40-160 LOC/P-month.
- Systems programs, 150-400 LOC/P-month.
- Commercial applications, 200-900 LOC/P-month.

Produktivnost za objektivne točke

- Između 4 i 50 OT / mjesec, ovisno o alatu i vještini

Language	SLOC/UFP
Ada	71
AI Shell	49
APL	32
Assembly	320
Assembly (Macro)	213
ANSI/Quick/Turbo Basic	64
Basic-Compiled	91
Basic-Interpreted	128
C	128
C++	29
ANSI COBOL 85	91
Fortran 77	105
Forth	64
Jovial	105
Lisp	64
Modula 2	80
Pascal	91



Procjena produktivnog vremena IT osoblja

Produktivnost pada prema broju osoba u ekipi.

- Ukupno vrijeme je izraženo s:

$$TPT = nH \left[p - 0.0001 \left\{ \frac{n(n-1)}{2} \right\} \right]$$

gdje je:

TPT	vrijeme produktivnosti tima u satima po tjednu
p	postotak radnog tjedna koji se smatra produktivan
H	broj radnih sati jedne osobe po radnom tjednu
n	broj ljudi zaposlenih u timu

- Ukupna produktivnost raste s brojem članova i dostiže maksimum na $n=60$
- Ukupna produktivnost za $n=20$ i $n=94$ člana približno je jednaka

Broj članova tima (n)	Ukupno vrijeme produktivnosti (TPT) [sati/tjedan]
20	424,8
40	755,2
60	895,2
80	748,8
94	424,5
100	220



Procjena trajanja projekta

- Iskustveno pravilo za računanje optimalne procjene trajanja (procjena trajanja koja daje najmanje troškove) iz procjene rada je:

$$\text{trajanje u mjesecima} = 3.0 \cdot (\text{čovjek-mjeseći})^{1/3}$$

- Ako je procijenjeno da će projekt zahtijevati 65 čovjek-mjeseći, optimalno trajanje projekta je 12 mjeseci ($3.0 \cdot 65^{1/3}$).
- To dalje znači da je optimalna veličina tima 5-6 članova.
- Što se događa kada projekt treba dovršiti brže?
 - Prema različitim izvorima faktor 3.0 može varirati od 4.0 do 2.5.
- Gornja formula je razlog zašto su rasponi procjene širi za procjenu rada nego za procjenu trajanja.
 - Veći projekti traju duže, ali i imaju veće razvojne timove, pa se potrebni rad povećava neproporcionalno brže od trajanja projekta.
 - Ovo podrazumijeva da se veličina razvojnog tima mijenja s veličinom projekta.
 - Ako veličina tima ostaje konstantna, onda su rasponi procjene za potreban rad i trajanje projekta jednaki.



Ekonomska izvodljivost

» Isplati li se graditi? Kada će se isplatiti?

Analiza troškova-koristi (cost-benefit analysis - CBA)

Trošak razvoja sustava (fiksni trošak)

- apsolutni iznos, početna procjena, ažuriranje tijekom napretka projekta
- primjeri:
 - osoblje: plaće, honorari
 - oprema – procjenjuje se nakon odabira tehničkog rješenja
 - izobrazba: tečajevi

Trošak primjene sustava (varijabilni trošak)

- relativan iznos, ovisan o uporabi
- primjeri:
 - režije (struja, telefon)
 - održavanje (ljudski rad)
 - obnova licenci



Kategorije troškova i koristi

- ☐ **Mjerljivi (tangible – opipljiv, određen, shvatljiv)**
 - zna se točan iznos ili iznos može biti procijenjen
- ☐ **Nemjerljivi (intangible)**
 - Pretpostavlja se ili se zna da postoje, ali im se postojanje ili "vrijednost" ne može egzaktno dokazati
- ☐ **Mjerljive koristi**
 - Najčešće izražene kao godišnja ušteda ili ušteda po proizvedenom predmetu
- ☐ **Nemjerljivi troškovi**
 - pad morala, pad produktivnosti (nemjerljivi) ili gubitak tržišta
- ☐ **Nemjerljive koristi**
 - Neopipljive koristi mogu pomoći ili odmoći korisnosti sustava, npr. poboljšano zadovoljstvo kupca, zadovoljstvo zaposlenika, ...



Primjeri troškova i koristi

- ☐ **Vrijednost novog kupca**
 - Vrijednost 300 novih kupaca godišnje koji u prosjeku potroše \$500 po proizvodu koji nakon troškova donosi 12% profita
 - godišnji profit iznosi $300 * \$500 * 12\% = \18.000
- ☐ **Vrijednost postojećih kupaca**
 - Ako izgubimo 100 kupaca od kojih svaki troši \$2500 godišnje, a za njihovo nadomještanje potrebno je potrošiti \$50.000 za reklamu, kratkotrajni gubitak tih kupaca iznosi (pod pretpostavkom da je razina dobiti 12%):
 - $100 * \$2500 * .12 + \$50.000 = \$80.000$
- ☐ **Smanjenje cijene rada ili ušteda smanjenjem posla**
 - Ako smanjimo rad za neki zadatak s 5 minuta na 30 sekundi, a zadatak radi osoba plaćena \$50 na sat
 - ušteda je $(5 - 30/60)/60$ sati po zadatku * \$50 na sat = \$3,75 po zadatku
- ☐ **Nemjerljive koristi se ponekad mogu kvantificirati postavljanjem nekoliko pretpostavki o njihovom učinku:**
 - nezadovoljni kupci naručuju manje i rjeđe
 - ako se to kvantificira možemo dobiti postotak u gubitku posla



Primjer: troškovi razvoja

Osoblje:		
Vrsta	Količina	Cijena
Analitičar sustava	900h * 45kn/h	40,500 kn
Programer	1375h * 36kn/h	49,500 kn
Stručnjak za komunikacije	60h * 40kn/h	2,400 kn
Administrator baza podataka	30h * 42kn/h	1,260 kn
Pisac dokumentacije	240h * 25kn/h	6000 kn
Tajnica	160h * 15kn/h	2,400 kn
Unos podataka	80h * 12kn/h	960 kn

Edukacija:		
Vrsta	Količina	Cijena
„in-house“ poduke za programere	3 dana	7,000 kn
„in-house“ poduka za korisnike	3 dana	10,000 kn

Materijal:		
Vrsta	Količina	Cijena
Kopiranje		500 kn
Diskovi, trake, papir		650 kn

Sklopovlje i programska podrška:		
Vrsta	Količina	Cijena
Windows licence		1,000 kn
Memorija za 20 klijenata		8,000 kn
Periferni uređaji za 20 klijenata		2,500 kn
Mrežni programi		15,000 kn
Office alati		20,000 kn



Primjer: godišnji troškovi rada

Osoblje:		
Vrsta	Količina	Cijena
Programer održavanja/analitičar	250h/god*42kn/h	10,500 kn
Mrežni administrator	300h/god*50kn/h	15,000 kn

Nadogradnja sklopovlja i programske podrške:		
Vrsta	Količina	Cijena
Sklopovlje		5,000 kn
Programska podrška		6,000 kn
Ostali troškovi		3,500 kn



Sadašnja vrijednost troškova i koristi

□ Sadašnja vrijednost (Present value - PV)

» \$ označava novčanu jedinicu u bilo kojoj valuti

- Današnja vrijednost onoga što će postati \$1.00 nakon 'n' godina u budućnosti, ako uzmemo u obzir kamate 'i' iznosi:

$$PV = 1/(1 + i)^n = (1 + i)^{-n}$$

- Razlika predstavlja kamatu koja se može zaraditi tim novcem

□ Primjeri:

- troškovi razvoja od \$100.000 imaju trenutnu vrijednost od \$100.000
- oročenje tih sredstava uz kamatu od 8% donijelo bi 47.93% dobiti od kamata
- korist projekta u iznosu od \$30.000 za pet godina uz kamatnu stopu od 8% ima sadašnju vrijednost od samo: $\$30.000 / (1 + 0.08)^5 = \20.417
- korist projekta u iznosu od \$300.000 postignuta za tri godine uz kamatnu stopu od 8% ima sadašnju vrijednost od samo:
 $\$300.000 / (1 + 0.08)^3 = \238.140



Neto sadašnja vrijednost

□ Neto sadašnja vrijednost - Net Present Value (NPV)

- budući trošak i korist s obzirom na gubitak vrijednosti sredstava
- razlika između sadašnje vrijednosti budućih priljeva i sadašnje vrijednosti budućih odljeva
- NPV = ukupna korist – ukupni troškovi

□ Primjer: Koji je projekt isplativiji ?

Kamata 10%

Projekt 1	0	1	2	3	4	Ukupno	NPV
Trošak	-5,000	-1,000	-1,000	-1,000	-1,000	-9,000	
Korist	0	2,000	3,000	4,000	5,000	14,000	
Cash flow	-5,000	1,000	2,000	3,000	4,000	5,000	2,316

Projekt 2	0	1	2	3	4	Ukupno	NPV
Trošak	-2,000	-2,000	-2,000	-2,000	-2,000	-10,000	
Korist	1,000	2,000	4,000	4,000	4,000	15,000	
Cash flow	-1,000	0	2,000	2,000	2,000	5,000	3,201



Povrat investicije

□ Povrat investicije

- Ulaganja donose korist koja s vremenom postaje sve veća.
- U jednom trenutku prihod dosegne rashod

□ Indeks profitabilnosti (Cost Benefit ratio)

- omjer sadašnje vrijednosti i početnog ulaganja

□ Vrijeme povrata investicije (payback period)

- Vrijeme povrata ukupnog troška
- Razdoblje potrebno da prihod dosegne rashod

□ Točka povrata investicije (break-even point)

- Trenutak u kojem prihod dosegne rashod



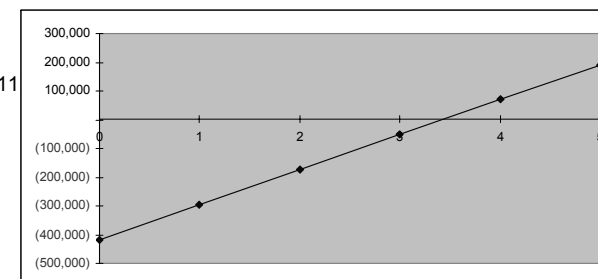
Primjer: Vrijeme povrata investicije

Kamata 12.00%		Godina 0	Godina 1	Godina 2	Godina 3	Godina 4	Godina 5
Trošak / Korist							
Trošak razvoja	(418,040)						
Operativni troškovi		(15,045)	(16,000)	(17,000)	(18,000)	(19,000)	
Faktor za kamatu	1.00	0.893	0.797	0.712	0.636	0.567	
Sadašnja vrijednost	(418,040)	(13,435)	(12,752)	(12,104)	(11,448)	(10,773)	
Kumulativni trošak	(418,040)	(431,475)	(444,227)	(456,331)	(467,779)	(478,552)	
Korist od novog IS		150,000	170,000	190,000	210,000	230,000	
Faktor za kamatu	1.00	0.893	0.797	0.712	0.636	0.567	
Sadašnja vrijednost	0	133,950	135,490	135,280	133,560	130,410	
Kumulativna korist	0	133,950	269,440	404,720	538,280	668,690	
	0	1	2	3	4	5	
Ukupno NPV	(418,040)	(297,525)	(174,787)	(51,611)	70,501	190,138	

□ Vrijeme povrata

$$= 3 + 51611 / (70501 + 51611)$$

$$= 3.42 \text{ godine}$$

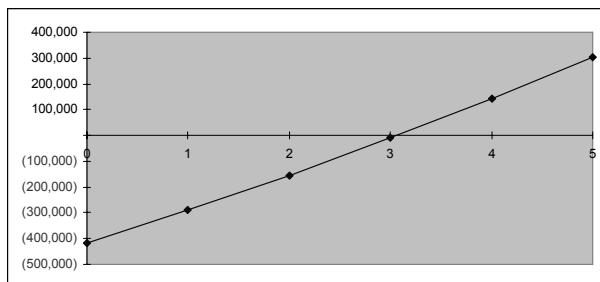


Primjer: Analiza povrata investicije

Kamata 6.00%		Godina 0	Godina 1	Godina 2	Godina 3	Godina 4	Godina 5
Trošak / Korist							
Trošak razvoja		(418,040)					
Operativni troškovi			(15,045)	(16,000)	(17,000)	(18,000)	(19,000)
Faktor za kamatu		1.00	0.943	0.890	0.840	0.792	0.747
Sadašnja vrijednost		(418,040)	(14,187)	(14,240)	(14,280)	(14,256)	(14,193)
Kumulativni trošak		(418,040)	(432,227)	(446,467)	(460,747)	(475,003)	(489,196)
Korist od novog IS			150,000	170,000	190,000	210,000	230,000
Faktor za kamatu		1.00	0.943	0.890	0.840	0.792	0.747
Sadašnja vrijednost		0	141,450	151,300	159,600	166,320	171,810
Kumulativna korist		0	141,450	292,750	452,350	618,670	790,480
Ukupno NPV		(418,040)	(290,777)	(153,717)	(8,397)	143,667	301,284

□ Za kamatu 6%

- NPV = 300k
- VP = 3.05 godina



Povrat investicije

□ Postotak povrata investicije

- ROI - postotak relativne koristi projekta u odnosu na trošak
- $ROI = \frac{\text{ukupna korist} - \text{ukupan trošak}}{\text{ukupan trošak}}$
- preračunato u sadašnju vrijednost, $ROI = NPV / \text{ukupna sadašnja trošak}$
- Za primjer s kamatom 12%, $ROI = 190378 / 478552 = 0.3973 = 39.73\%$
- Za primjer s kamatom 6%, $ROI = 301284 / 489196 = 0.6159 = 61.59\%$

□ Faktor obnavljanja kapitala

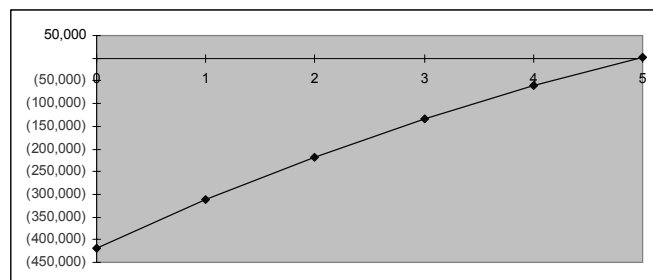
- ROI se obično dijeli sa dužinom projekta kako bi se dobio godišnji ROI → faktor obnavljanja kapitala
- Nizak ROI (~ manji od 10% godišnje) može pokazivati da je korist preniska da bi bila isplativa
- Za primjer s kamatom 12%, $ROI\% = 7.95\%$ godišnje
- Za primjer s kamatom 6%, $ROI\% = 12.32\%$ godišnje



Povrat investicije

□ Interna stopa rentabilnosti - Internal Rate of Return (IRR)

- Interna stopa povrata investicije
- Izražava potrebni postotak povrata potreban da bi se trošak i korist izjednačili u nekom vremenskom razdoblju
- Kamatna stopa pri kojoj je NPV = 0



Modeliranje postojećeg sustava tijekom planiranja

□ Svrha

- Preciziranje dosega projekta
- Verifikacija razumijevanja problema i usaglašavanje percepcije sustava i stavova između sudionika (korisnici, informatičari)

□ Globalni, okvirni, grubi modeli

- Model organizacije i resursa
 - kontekst, organizacijska struktura, prostorni raspored sredstava
- Globalni model procesa
 - funkcionalna dekompozicija
 - tok ključnih poslovnih procesa
 - kolanje dokumenata i protok informacija
- Globalni model entiteti-veze (enterprise data model)
 - kategorije podataka – klase podataka (ne razredi objekata!)

□ Primjeri: \Planiranje\

- Mesna Industrija-Idejno,
- Ministarstvo-Idejno



Pospješanje održivosti projekta

❑ "Puzeći" doseg projekta (creeping scope)

- Tijekom izvedbe projekta često se događa polagano, ali značajno povećanje obujma uslijed pogrešne procjene, različitog tumačenja zahtjeva između korisnika i izvođača ili promjene zahtjeva
- Granice projekta moraju biti definirane što je moguće preciznije
- Time se kasnije povećanje projekta možda neće ukloniti ali će se barem moći nadzirati

❑ Opcionalno se planira i provodi:

- Izrada prototipa ili oglednog (pilot) projekta i procjena njegove uspješnosti
 - projekt koji se može uspješno i "brzo" realizirati (npr. 3-9 mjeseci)
 - Primjer: Projekt informatizacije zdravstva
 - Primjer: Pilot projekt PIS visokoobrazovnih učilišta
- Procjena održivosti projekta
 - SurvivalTest [McConnell, 1998], <http://www.construx.com/survivalguide/>
 - Primjer: \Planiranje\OdrživostProjekta, SurvivalTest



Modaliteti izgradnje sustava

Dio analize izvodljivosti ili međukorak analize i dizajna

Vlastiti razvoj

❑ Vlastiti razvoj (Insourcing)

- Varijante
 - Razvoj vlastitim informatičkim snagama (in-house), koji može sadržavati osposobljavanje i angažiranje netehničkog osoblja
 - Primjeri: Ministarstvo, Društvo, Poduzeće
 - Razvoj unajmljenim osobljem povremeno ili dugoročno (buy-in, preferred supplier)
 - Primjer: Banka
- Prednosti
 - povećanje stručnosti naručitelja, samostalnost
- Nedostaci
 - trajni trošak kućne informatike
 - problem paralelnog razvoja i održavanja, gomilanje posla

❑ Argumenti za vlastiti razvoj

- informatizacija osnovnog posla (core business), kada ne postoje gotova rješenja na tržištu ili takva da organizacija s pomoću njih postigne komparativnu prednost u odnosu na konkurenciju
- kada postoje dodatni ili posebni razlozi kao što su povećana tajnost podataka i poslovnih procesa ili povećana zaštita IS



Vanjski razvoj

❑ Vanjski razvoj (Outsourcing)

- najam usluge razvoja informacijskog sustava ili njegovih dijelova
 - poduka djelatnika informatičke struke
 - savjetništvo pri provedbi pojedinih faza ciklusa
 - izrada rješenja po specifikaciji naručitelja
- Varijante:
 - Ugovoreni razvoj isporuke gotovog proizvoda (contract out)
 - Dugoročna suradnja s isporučiteljem (strateškim partnerom) ili izdvajanje odjela informatike u preferiranog izvođača (preferred contractor)
- Ključna karakteristika: zavisnost o dobavljaču

❑ Razvoj od strane vanjskih dobavljača

- Prednost: rješenje po mjeri naručitelja
- Nedostatak: visoka cijena unikata
- Primjer: Turistička Agencija

❑ Nabava gotovog rješenja

- aplikacije za standardno poslovanje
 - financijsko poslovanje (accounting)
 - robno-materijalno poslovanje (material management/distribution)
 - upravljanje ljudskim resursima i plaće (HR management, payroll)
 - proizvodnja (manufacturing)
- niža cijena i kraće vrijeme "konfekcije"
- također zahtijeva prilagodbu (parametrizaciju) i konverziju podataka !



Odabir rješenja

❑ Određivanje mogućih rješenja

- Identifikacija rješenja na temelju analize poslovnih zahtjeva
- Ulazi: specifikacija računalne opreme i programske podrške te odabrana tehnološka arhitektura
- Izlazi: moguća rješenja novog sustava i njihove karakteristike

❑ Analiza izvodljivosti alternativnih rješenja

- Procjena alternativa s obzirom na tehničku, operativnu, ekonomsku i vremensku izvodljivost
- Ulazi: karakteristike mogućih rješenja, cijene hardvera i softvera, reference i uvjeti dobavljača
- Izlazi: analiza izvodljivosti za svako moguće rješenje

❑ Prijedlog rješenja sustava koje će se oblikovati i ugraditi

- Odabir onog rješenja koje ima najbolju ukupnu kombinaciju izvodljivosti
- Ulazi: analiza izvodljivosti, plan projekta, procjena veličine projekta
- Izlazi: prijedlog sustava

❑ Kriteriji za odabir

- Primjeri: \Planiranje\IEEE-KriterijiNabave



Primjer: moguća rješenja i njihove karakteristike

Karakteristike	SuperVideo	Video Boss	Video	ZZ Video
Operacijski sustav	Windows	Windows	Dos	Linux
Baza podataka	Access 2	Paradox 8	dBase III	MySQL
Brzina pretraživanja i dohvata podatka	velika	velika	mala	srednja
Programski jezik	Visual Basic	C++	Cliper	Clarion
Raspoloživ izvorni kod	ne	ne	da	ne
Korisničko sučelje	grafičko	grafičko	tekstovno	tekstovno
Integrirani sustav pomoći (on-line help)	da	ne	ne	ne
Dokumentacija (papirnata)	dobra	ne	ne	dobra
Mogućnosti aplikacije	velike	vrlo male	male	velike
Integracija s drugim aplikacijama	dobra	srednja	ne	ne
Brzina ispisa računa	srednja	srednja	velika	velika
Rad s različitim pisacima	da	da	ne	ne
Rad u mreži	da	ne	ne	ne
Krivulja učenja	1-2 dana / 2 tjedna	1 dan / 1 mjesec	1dan / 2 tjedna	1-2 dana / 1 tjedan
Arhiviranje podataka	da	ne	da	da
Upotreba konfiguracije za druge poslove	velika	velika	ne	vrlo mala
Min. potrebno računalo
Preporučeno računalo				
Broj instaliranih paketa	27	10	152	87
Datum prve instalacije	9/95	7/96	4/93	10/94
Cijena paketa	1500 kn	100 kn	500 kn	2000 kn
Cijena min. potrebnog računala (plus monitor i pisac)	3500 kn	4000 kn	2200 kn	3500 kn
Cijena preporučenog računala (plus monitor, pisac i modem)	8000 kn	8000 kn	-	
Cijena operacijskog sustava i licenci				



Vrednovanje mogućih rješenja

❑ Svojstva treba kvantificirati da bi se mogla usporediti

- Koristi se sustav bodovanja da bi se usporedio značaj različitih kriterija.

❑ Model ponderiranog vrednovanja (Weighted Scoring Model)

- Odredi se težinski faktor za svaki kriterij (npr. 0-3).
- Pojedinačnom kriteriju svakog od rješenja dodjeljuje se ocjena iz dogovorenog raspona (npr. 0-5), pomnožena s odgovarajućom težinom.
- Dobiveni pojedinačni rezultati sumiraju se za svako od rješenja.

gdje su

$$S_i = \sum_{j=1}^n s_{ij} w_j$$

S_i = ukupna vrijednost i -tog rješenja

s_{ij} = vrijednost j -tog kriterija za i -to rješenje

w_j = važnost ili težina j -tog kriterija

❑ Što učiniti kada su sustavi (pod)jednako bodovani ?

- Primjer: Uprava-IzboraAlata

❑ Što učiniti ako pojedino svojstvo ima više podsvojstava ?

- <http://www.zpr.fer.hr/zpr/Projekti/IPIS/>



Primjer: analiza izvodljivosti za moguća rješenja

Karakteristike:	Težinski faktor	SuperVideo		Video Boss		Video		ZZ Video	
		Ocjena	Bodovi	Ocjena	Bodovi	Ocjena	Bodovi	Ocjena	bodovi
Operacijski sustav	2	4	8	4	8	1	2	3	6
Baza podataka	1	4	4	4	4	2	2	1	1
Brzina pretraživanja i dohvata podatka	4	5	20	4	16	1	4	4	16
Programski jezik	1	4	4	5	5	2	2	2	2
Raspoloživ izvorni kod	1	0	0	0	0	5	5	0	0
Korisničko sučelje	2	5	10	5	10	3	6	3	6
Integrirani sustav pomoći (on-line help)	2	5	10	0	0	0	0	0	0
Dokumentacija (papirnata)	2	4	8	0	0	0	0	4	8
Mogućnosti aplikacije	4	5	20	1	4	2	8	5	20
Integracija s drugim aplikacijama	3	4	12	3	9	0	0	0	0
Brzina ispisa računa	4	2	8	3	12	5	20	5	20
Rad s različitim pisacima	3	5	15	5	15	0	0	0	0
Rad u mreži	1	5	5	0	0	0	0	0	0
Vrijeme obuke korisnika	1	3	3	5	5	5	5	3	3
Arhiviranje podataka	2	5	10	0	0	5	10	5	10
Upotreba konfiguracije za druge poslove	3	5	15	5	15	0	0	3	9
Broj instaliranih paketa	1	3	3	2	2	5	5	5	5
Datum prve instalacije	1	3	3	3	3	5	5	5	5
Cijena paketa	2	2	4	5	10	4	8	2	4
Cijena računala i sistemskog softvera	3	3	9	2	6	5	15	3	9
Ukupno bodova:			171		124		97		124



Primjer: Planiranje PISFER-Studijalozvodljivosti		Kadrovski	Financijski i biraški	Obratun plaća	Nastava i procedure	Zavodi i projekti	Uruženje i zapisnik	Opća služba	Stadište i materijalno	Komponente poslovno – upravljačkog IS-a
Nadogradnja	Operativna	3	3	3	3	1	1	1	2	0
	Tehnička	3	3	3	3	1	1	1	2	0
	Vremenska	3	3	3	2	1	1	1	1	0
	Ekonomski	3	3	3	3	1	1	1	2	0
Ocjena alternative		3	3	3	2.75	1	1	1	1.75	0
Izrada vlastitog	Operativna	1	1	1	2	3	3	3	2	2
	Tehnička	1	1	1	2	2	2	2	1	2
	Vremenska	1	1	1	1	2	1	1	1	1
	Ekonomski	1	1	1	1	2	1	1	1	1
Ocjena alternative		1	1	1	1.5	2.25	1.75	1.75	1.25	1.5
Nabava gotovog	Operativna	2	2	1	1	2	1	1	2	1
	Tehnička	3	2	1	1	2	1	2	2	2
	Vremenska	3	2	1	1	3	1	2	3	3
	Ekonomski	1	2	1	1	1	1	1	1	2
Ocjena alternative		2.25	2	1	1	2	1	1.5	2	2
Konačan prijedlog alternative	Nadogradnja	x	x	x	x				x	
	Izrada vlastitog					x	x	x		
	Nabava gotovog									x



Prikupljanje informacija i utvrđivanje zahtjeva

Prikupljanje informacija (information gathering)
Prikupljanje podataka (data collection)
Ustanovljavanje činjenica (fact finding)



Postupak intervjuiranja

Intervju

- neusiljen i elastičan razgovor s ispitanikom, slijed pitanja i odgovora
 - ispitanik se pojavljuje u ulozi pasivnog sugovornika (!?)
 - sugovornici: rukovoditelji, krajnji korisnici, ostali sudionici
- standardno uključuje dva sugovornika ali može i više (korisnika, ispitivača):
 - individualni intervju – jedan korisnik ili suradnici koji rade isti posao
 - intervjuiranje grupe – više korisnika iz različitih područja

Organizacija razgovora

- informacije se prikupljaju nizom pojedinačnih razgovora
- (prvi) razgovori trebali bi se voditi prema unaprijed dogovorenom planu i rasporedu → koordinator
 - Primjeri: Analiza\PlanRazgovora
 - postupak je spor i neučinkovit, jer se organizacija razgovora mora obaviti za svaki pojedini razgovor
- nakon dovršetka analize i sinteze dobivenih informacija neke razgovore (ponekad i čitavu seriju) treba ponoviti da bi se upotpunile dobivene informacije ili uskladili proturječni iskazi
- ukupan broj razgovora ne može se unaprijed točno odrediti i treba ga prilagoditi stvarnoj situaciji



Tehnika intervjuiranja

Priprema za razgovor

- utvrđivanje informacija koje treba saznati
- proučavanje postojeće dokumentacije i prethodnih nalaza
- određivanje dokumentacije koju bi trebalo prikupiti
- priprema pitanja koja će biti postavljena tijekom razgovora
 - izrada jezgre tema, to jest standardnih pitanja
 - izrada sveobuhvatnog podsjetnika i izdvajanje prikladnih pitanja

Primjeri: Analiza

- Podsjetnik-teme
- Podsjetnik-priprema
- Podsjetnik-upitnik



Tehnika intervjuiranja

❑ Planiranje i obavljanje razgovora

- okvirno trajanje prvog razgovora je 2 sata (općenito 1,5 – 2,5 sata)
- Početak razgovora
 - predstavljanje sudionika
 - upoznavanje sa svrhom razgovora (prikupljanje informacija o ...)
- Vođenje razgovora
 - postavljanje pitanja i bilježenje odgovora
 - prikupljanje dokumentacije
- Završetak razgovora
 - približno 5-10 minuta prije isteka planiranog vremena
 - prekida se slijed postavljanja pitanja
 - provjerava se da li postoji nešto što je sugovornik htio reći a nije bilo pitano
 - provjerava se da li treba proširiti krug sugovornika
 - dogovara se nastavak razgovora (dopunski razgovor) kada:
 - voditelj razgovora nije postavio sva planirana pitanja ili
 - voditelj smatra da je razgovor nametnuo nova pitanja
 - zahvala na razgovoru



Tehnika intervjuiranja

❑ Preispitivanje i pojašnjenje sadržaja (follow-up)

- provjera zabilježenih navoda radi upotpunjavanja bilješki
- telefonom, elektroničkom poštom, novim sastankom

❑ Dokumentiranje razgovora

- Pisanje zapisnika
 - samostalno pisati zapisnike ("Tko ne razumije, ne može bilježiti.")
 - kada u razgovoru sudjeluje više analitičara, određuje se voditelj razgovora koji je ujedno i zapisničar, a ostali ulažu primjedbe
- Sadržaj zapisnika
 - Vrijeme i mjesto održavanja razgovora
 - Sudionici
 - Sadržaj razgovora (tekst zapisnika)
 - Popis prikupljene dokumentacije
 - Zapisničar
- zapisnik mora sadržavati ono što je rečeno i slijediti tok razgovora
- zapisnik ne smije nametati zaključke, jer su oni rezultat analize

❑ Autorizacija (ovjera) zapisnika

- zapisnik se šalje na uvid sugovorniku, koji šalje potvrdi o vjernosti zapisnika
- po potrebi sugovornik može nadopuniti dijelove za koje smatra da nisu evidentirani ili pojasniti dijelove za koje misli da su krivo protumačeni

❑ Primjeri: Analiza

- Zapisnik-*
- TragRazgovora



Preporuke za vođenje intervjua

❑ Treba pitati o svemu što se smatra važnim

- ništa nije samo po sebi razumljivo i svima jasno
- ne pretpostavljati da se unaprijed zna o čemu se radi

❑ Repertoar i vrste pitanja

- Pitanja zatvorenog tipa:
 - Koliko ... obrađujete (u nekom razdoblju)?
 - Na koji način obrađujete ... ?
- Pitanja otvorenog tipa:
 - Što mislite o ... ?
 - Koji su najveći problemi ... ?
- "Probna" pitanja:
 - Zašto ?
 - Možete li navesti primjer za takvu situaciju?
 - Molim detaljnije objašnjenje za ...



Preporuke za vođenje intervjua

❑ Analiza odgovora

- razlučivanje činjenica od mišljenja
- utvrđivanje da li pojedine činjenice odgovaraju drugima
- analiza činjenica koje se ne poklapaju
- provjera odgovora različitih sugovornika na isto pitanje

❑ Preporučljivo ponašanje

- iskrenost i nepristranost
 - cilj je naći za korisnika najprikladnije rješenje, a ne podupirati određeno programsko rješenje ili računalnu platformu
- pažnja i jezgrovitost
 - "brzo misli, jasno govori"
- izbjegavanje sugestivnih pitanja
- nenametanje zaključaka
- ležernost (+ praćenje reakcija sugovornika)

❑ Grupno intervjuiranje

- izbjegavati i po potrebi nadomjestiti radnim sjednicama
- iznimno provesti, kada se želi utvrditi zajednički interes ili proturječje



Upitnici i ankete

❑ Upitnik (pismeni intervju)

- sadrži pitanja koja se postavljaju tijekom razgovora (okvirno 20)
- može se dostaviti korisniku prije, umjesto ili nakon intervjua
- nedostaci:
 - ispitanik može prilagoditi (kontrolirati) svoje odgovore
 - teško je procijeniti iskrenost (spontanost) odgovora
 - može obeshrabriti ispitanika
- Primjer: Analiza\Upitnik

❑ Anketa (inventar)

- može se obuhvatiti više ispitanika
- pitanja zatvorenog tipa
- odgovori i obrada odgovora mogu se standardizirati
- pogodna za popis resursa
- Primjer: Analiza\Anketa-resursi

❑ Intervjuiranje je elastičnije!

- Pomoću intervjua može se više saznati o stavovima, osjećajima i motivaciji osoblja.
- Tijekom intervjua analitičar može promatrati način na koji ispitanik odgovara i po potrebi proširiti ili usmjeriti pitanja.



Proučavanje dokumenata

❑ Prikuplja se sve do čega se može doći

- Analiza procesa
 - Opis radnih procedura
 - Tipični dokumenti: opis radnih procedura !?, pravilnici i zakoni
- Analiza podataka
 - Tipični dokumenti: obrasci (formulari), izvješća
 - Poželjno je da dokumenti budu reprezentativni, tj. popunjeni na tipičan način. Tako se može ustanoviti koje se informacije i kako unose.
 - Reprezentativni dokumenti najčešće ne ukazuju na izuzetke, to jest podatke koji se rjeđe bilježe ali ipak trebaju.
 - Stalno bilježenje nekih podataka ne mora značiti da su ti podaci stvarno potrebni.
 - Treba prikupiti više uzoraka iste vrste dokumenta! (→ uzorkovanje)

❑ Vrijednost informacija dobivenih analizom (samo) dokumenata je niska.

- Praksa može odudarati od pravilnika i administrativnih obrazaca.
- Treba shvatiti zašto i kada dokumenti nastaju, kako se popunjavaju, koliko su potrebni te što treba promijeniti da bi ih se popravilo (sadržaj, lakoća popunjavanja i čitanja)
- Nužno je modele (podataka) razlučene analizom provjeriti kod korisnika.

❑ Primjeri: Analiza\PrimjeriDokumenata\



Radne sjednice (združeni razvoj)

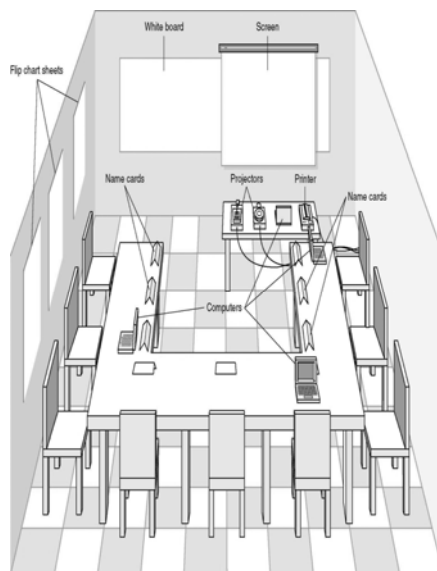
Slika: A.Dennis & B. Haley Wixom,
Systems Analysis and Design,
John Wiley & Sons, 2000

❑ Radni sastanci, sjednice (workshop)

- analitičari i korisnici zajednički provode analizu i oblikovanje
- cilj sjednice je (zajedničko) pronalaženje najboljeg rješenja
- poseban prostor i izolacija
- moderator, dnevni red i zapisnici

❑ Genijalnost grupe (tzv. brainstorming)

- koristi se za prikupljanje ideja i definiranje informacijskih potreba
- korisnike se potiče na aktivno i kreativno sudjelovanje
- izvodi se tako da se svakog ispitanika iz grupe zatraži da definira po njemu idealno rješenje
- svaki sudionik iznosi sve što mu pada na pamet u svezi s problemom koji se rješava
- od predloženih rješenja odabire se najbolje prema realnoj izvodljivosti
- postupak je učinkovit tamo gdje postoje korisnici koji dobro poznaju sustav ali teško prihvaćaju nove ideje



Radne sjednice

❑ Prednosti

- Sjednice su pogodne za projekte kojima se rješavaju problemi važni za čitavu organizaciju ili veći dio poslovanja.
- Izbjegavanje specifičnih (egzotičnih) i nejasnih zahtjeva
- Preciznije ustanovljavanje dosega projekta
- Bolje uočavanje proturječnih zahtjeva

❑ Nedostaci

- sadržaj i dinamika
 - pasivnost sudionika
 - "usitnjavanje" razgovora
 - udaljavanje od tema
- trajanje
 - sjednice bi trebale trajati više dana (5-10) u nekoliko tjedana
 - ovom zahtjevu u praksi je vrlo teško udovoljiti zbog obveza sudionika
- otpor sjednici i pratećoj izolaciji
 - razmjern je razini položaja pojedinog sudionika
 - izraženiji među korisnicima koji smatraju da je to informatički posao

❑ Primjeri: Plastika, VisokoUčilište



Razvoj prototipa

☐ Prototipiranje (prototyping)

- Koristi se kada korisnik ne može točno definirati svoje informacijske potrebe prije nego što se izgradi informacijski sustav.
- Razlog tomu može biti nedostatak postojećeg modela na kojem bi korisnik zasnivao svoje potrebe ili teža vizualizacija budućeg sustava.

☐ Izrada prototipa

- Izgrađuje se sustav koji zadovoljava neke osnovne, inicijalne potrebe.
- U radu s takvim sustavom korisnik otkriva svoje informacijske potrebe te se sustav modificira kako bi se zadovoljile te potrebe.
- Postupak korištenje sustava i modificiranja istog iterativno se ponavlja, a informacijske potrebe korisnika otkrivaju korištenjem sustava.

☐ Izrada prototipa pogodna je u onim okruženjima gdje je teško definirati konkretni model sustava te u okruženjima gdje se informacijske potrebe korisnika mijenjaju ili razvijaju.



Analiza sustava



Analiza sustava

☐ Analiza sustava, sustavska analiza

- Analiza sustava je (1) razmatranje i planiranje sustava i projekta, (2) proučavanje i analiza postojećeg poslovnog i informacijskog sustava te (3) definiranje poslovnih zahtjeva i prioriteta novog ili poboljšanog postojećeg sustava. [Whitten et. al, 2000]

☐ Aktivnosti analize

- detaljna analiza postojećeg sustava te utvrđivanje potreba i zahtjeva
 - Kako radi postojeći sustav?
 - Na koji način ljudi koriste sustav da bi obavili svoj posao?
 - Koji su problemi pri korištenju aplikacija ili uslijed nedostatka aplikacija?
- detaljna specifikacija zahtjeva na IS
 - Koji su podaci potrebni? Tko ih treba? Kada? Od koga? Tko ih stvara?
 - Koji su izlazni podaci? Kakav im je oblik? Koji su izvori podataka?
 - Na koji način se podaci prikupljaju i objedinjuju?
- daljnja razrada granica projekta

☐ Primjeri: Analiza\ ProtokDokumenata, RazmjenaPodataka



Analiza sustava (2)

☐ Pozadinska analiza

- Važno je shvatiti strukturu organizacije, tko u njoj radi, tko je kome podčinjen, kako surađuju različiti odjeli, itd.

☐ Modeliranje sustava

- Za potrebe pozadinske analize može se izraditi shema organizacijske strukture iz koje će biti vidljivo koja osoba ili grupa ljudi obavlja koji dio posla (→ Modeliranje funkcija).
- Za ostale elemente također se rade odgovarajući modeli (→ Modeliranje procesa, modeliranje podataka)

☐ Svrha, cilj i dubina analize

- automatizacija, poboljšanje i preoblikovanje poslovnih procesa



Automatizacija poslovnih procesa

❑ Automatizacija poslovnih procesa - Business Process Automation (BPA)

- Povećanje učinkovitosti korisnika primjenom računalne tehnologije

❑ Tehnike

- Analiza problema (problem analysis)
 - Otkrivanje problema postojećeg sustava i predlaganje rješenja budućeg sustava uz pomoć korisnika
 - Poboljšanja su mala i inkrementalna (npr. otkrivanje da je potreban izvještaj koji ne postoji), ali poboljšavaju učinkovitost i lakoću korištenja sustava
 - Poboljšanja poslovne vrijednosti (business value) su mala
- Analiza uzroka (root cause analysis)
 - Usmjerenost na probleme (simptome), a ne na rješenja
 - Problemima se dodjeljuju prioritete te traže svi mogući uzroci
 - Uzroci se analiziraju sve dok se ne utvrdi pravi uzrok
 - Najbolje je analizirati one uzroke koji se pojavljuju kod više problema



Primjer: automatizacija poslovnog procesa

❑ Primjer: Izrada putnih naloga

- izrađuje se 2700 putnih naloga godišnje
 - nalog za putovanje odobravaju redom voditelj projekta, predstojnik zavoda te dekan ili osoba koja ga zamjenjuje
 - nalozi se urudžbiraju (2 djelatnice) i knjiže te isplaćuje predujam za putovanje (1 djelatnica)
 - po povratku s putovanja putnik popunjava zapisnik o putovanju, a nalog se obračunava (1 djelatnica) te se na blagajni (1 djelatnica) isplaćuje razlika po troškovima ili povrat nerealiziranog ostatka predujma
- u pojedinim razdobljima broj putovanja značajno je povećan
 - nastaje gomilanje zaprimljenih naloga u pisarnici
 - česte su gužve na blagajni

❑ Rješenja:

- kreiranje i ažuriranje naloga na mjestu nastanka (podnositelj)
- automatizacija postupka odobravanja
- isplata predujmova putem tekućeg računa



Poboljšanje poslovnih procesa

❑ Poboljšanje poslovnih procesa - Business Process Improvement (BPI)

- Manje promjene poslovnih procesa radi boljeg iskorištenja tehnologije i povećanja učinkovitosti i djelatnosti

❑ Tehnike

- Analiza trajanja (duration analysis)
 - Analiza vremena po svim koracima procesa postojećeg sustava
 - Procene koje obavlja više ljudi istovremeno treba integrirati i paralelizirati
 - Integracija procesa je mijenjanje glavnog procesa tako da manje ljudi radi na ulazima potrebnim za proces
 - Paralelizacija procesa je mijenjanje procesa tako da se pojedinačni koraci procesa izvode istovremeno
- Koštanje poslovnih procesa (activity-based costing)
 - Analiza troškova po svim koracima procesa
 - Određivanje direktnog troška rada i resursa za svaki ulaz
- Baždarenje i usporedba (Informal benchmarking)
 - Proučavanje poslovanja drugih organizacija
 - Predstavljanje novih ideja koje mogu imati dodatnu vrijednost
 - Uobičajeno za procese kod kojih se obavlja interakcija s korisnicima



Primjer: paralelizacija poslovnog procesa

❑ Primjer: poboljšanje obrade putnih naloga

- izvješće s putovanja temeljem predloška putnog naloga s funkcijom za urudžbiranje – urudžbiranje se obavlja web servisom koji uz attribute pohranjuje i dokument (BLOB)
- potpisani primjerak dostavlja se u pisarnicu na skeniranje
- nalog se može paralelno knjižiti jer je u međuvremenu urudžbiran



Primjer: integracija poslovnog procesa

□ Primjer: obračun nastave na poslijediplomskom studiju

- Stanje: (velik broj sitnih transakcija ugovaranja i isplata, uz prijepis)
 - po obavljenoj nastavi Studentska služba dostavlja evidenciju o angažmanu na predavanjima
 - izvođači zasebno potpisuju autorski ugovor u Pravnoj službi
 - nezavisno se za vođenje (mentorstvo) i sudjelovanja u povjerenstvima za obranu disertacija sastavlja pojedinačni ugovor
 - sve isplate temeljem ugovora provodi Računovodstvo
- Poboljšanje:
 - jednom godišnje (za nastupajuću ak. godinu) izrađuje se autorski ugovor s definiranim općim uvjetima, na kojem su kao izvoditelji uključeni svi predvidivi nastavnici i vanjski predavači
 - jedan primjerak potpisanog ugovora dostavlja se u Studentsku službu
 - tijekom godine naknadno angažirani izvođači potpisuju dodatak ugovoru
 - Studentska služba po obavljenom poslu evidentira obavljeno te generira nalog za isplatu izvoditeljima s pozivom na broj autorskog ugovora



Preustroj poslovnih procesa

□ Preustroj poslovnih procesa

- Reinženjerstvo poslovnih procesa - Business Process Reengineering (BPR)
- Preoblikovanje poslovnih procesa - Business Process Redesign (BPR)
- radikalni redizajn poslovnih procesa, analizom mogućih posljedica, procjenom alternativnih tehnologija, ukidanjem ili zamjenom pojedinih aktivnosti, analizom troškova-koristi i analizom rizika

□ Tehnike

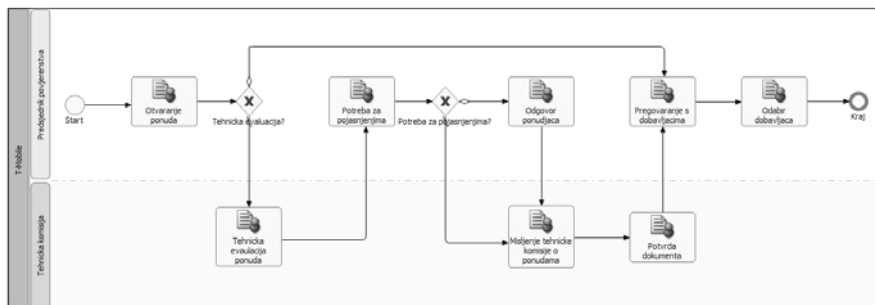
- Analiza posljedica (outcome analysis)
 - Razumijevanje osnovnih rezultata koji osiguravaju vrijednost za korisnika, procjena što bi se moglo ponuditi krajnjim korisnicima
- Analiza tehnologije (technology analysis)
 - Analitičari i voditelji razvijaju listu važnih i zanimljivih tehnologija te donose odluke o koristi i primjeni u poslovnim procesima
- Uklanjanje aktivnosti (activity elimination)
 - Analitičari i voditelji donose odluke za svaku aktivnost unutar poslovnog procesa o mogućnosti njenog uklanjanja i posljedicama



Primjer: heuristički preustroj procesa

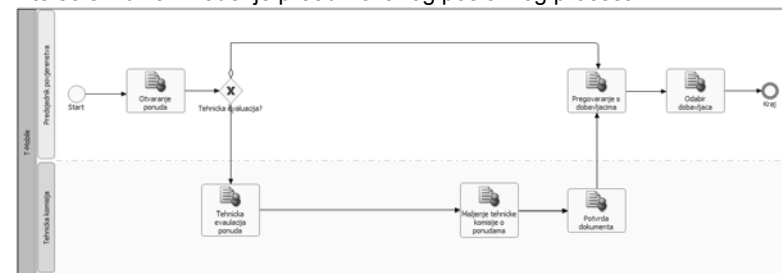
□ Kreće se od postojećeg stanja poslovnog procesa

- npr. potproces unutar poslovnog procesa nabave
- Poslovni proces sastoji se od otvaranja ponuda pristiglih od ponuđača, tehničke evaluacije tih ponuda (u slučaju da je potrebna), slanja dodatnih pitanja ponuđačima (ukoliko postoje), definiranja mišljenja tehničke komisije o ponudama, potvrđivanja tog dokumenta i na koncu odabira dobavljača. U procesu sudjeluju samo dva resursa, predsjednik povjerenstva nabave (15 ljudi) i tehnička komisija (400 ljudi).



Primjer: heuristički preustroj procesa (2)

- simulira se izvođenje postojećeg procesa,
- vrši se preoblikovanje
 - npr. pretpostavit će se da nije potrebno tražiti dodatna pojašnjenja od ponuđača, jer su zahtjevi dobro definirani u kontaktu s naručiteljem tijekom izrade ponude
- te se simulira izvođenje preoblikovanog poslovnog procesa.



Stanje poslovnog procesa	Ukupno prosječno vrijeme trajanja procesa	Ukupan prosječni broj resursa bez posla
Prije preoblikovanja	213.849,88 min	45,980225
Nakon preoblikovanja	198.256,66 min	46,1718



Izbor prikladne tehnike analize

	BPA	BPI	BPR
Poslovna vrijednost	mala do umjerena	umjerena	velika
Trošak projekta	mali	mali do umjeren	veliki
Doseg analize	ograničen	ograničen do umjeren	vrlo širok
Rizik	mali do umjeren	mali do umjeren	vrlo veliki

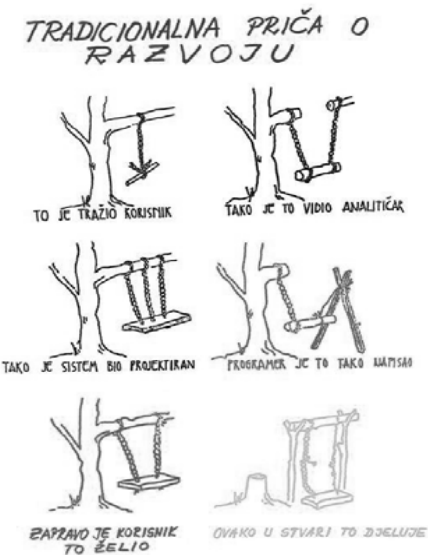
- ❑ **Potencijalna poslovna vrijednosti (potential business value)**
 - BPA ne mijenja poslovne procese, poboljšava učinkovitost
 - BPI poboljšava i učinkovitost i djelotvornost
 - BPR daje velike koristi jer radikalno poboljšava prirodu poslovanja
- ❑ **Trošak projekta (project cost)**
 - BPA zahtjeva najniži trošak jer ima ograničen doseg
 - BPI ovisi o dosegu projekta
 - BPR je obično skup zbog količine utrošenog vremena i zbog količine preoblikovanja procesa
- ❑ **Doseg analize (breadth of analysis)**
 - BPA obično radi s jednim procesom
 - BPI ima ograničen doseg i radi s nekoliko procesa
 - BPR ima vrlo širok doseg i radi s glavnim poslovnim procesima
- ❑ **Rizik pogreške (risk of failure)**
 - BPA ima mali do umjeren rizik jer je budući sustav jasan
 - BPI ima mali do umjeren rizik jer je budući sustav jasan
 - BPR je manje predvidiv i podložan riziku te se provodi uz odobrenje



Preporuke za analizu

Slika: Awad, 1985;
preveo Anonymous
(WWW)

- ❑ **Tehnike i pomagala ne znače puno bez ispravnog pristupa.**
 - Postupak analize mora biti prilagođen korisniku.
 - Ključno je razumjeti suštinu organizacije i način poslovanja
 - Primjer: \Analiza\ PreporukeZaAnalizu
- ❑ **Ono što korisnik želi često nije i ono što korisnik stvarno treba!**
 - Da se priča o klasičnom razvoju ne bi ponavljala, rade se modeli sustava (postojeći fizički i logički, budući logički...).
- ❑ **Paraliza analize**
 - Postupak ponavljanog modeliranja postojećeg sustava i vrednovanja modela može utrošiti značajno vrijeme na modeliranje nečega što će vjerojatno biti izmijenjeno ili zamijenjeno.
 - Ne pretjerivati s modeliranjem postojećeg sustava!



Inženjerstvo zahtjeva

Definiranje zahtjeva

- ❑ **IEEE standard definira zahtjeve kao [IEEE Std 610.12-1990]**
 - (1) Uvjet ili sposobnost koje korisnik treba da bi riješio problem ili ostvario cilj.
 - (2) Uvjet ili sposobnost koji mora posjedovati sustav ili komponenta sustava da bi zadovoljila ugovor, standard, specifikacije ili neki drugi ugovoreni dokument. (3) Dokumentiranu reprezentaciju uvjeta ili mogućnosti definiranih pod (1) ili (2).
- ❑ **Zahtjevi ne sadrže detalje dizajna, detalje implementacije ili informacije vezane uz planiranje projekta.**
 - Pažnja se usmjerava na ono ŠTO se želi izgraditi, a ne ulazi se u detalje kako i kada to napraviti.
- ❑ **Za 40-60% pogrešaka u projektu uzrok leži u pogreškama napravljenim u fazi postavljanja zahtjeva. [Leffingwell, 1997]**
 - Tipična posljedica su "neispunjena očekivanja" (expectation gap): razlika između onog što očekuje naručitelj i onoga što je izvoditelj mislio da treba napraviti.
- ❑ **Naknadne prepravke mogu predstavljati do 40% troškova razvoja, od čega je 70-85% pripisivo pogrešnim zahtjevima [Leffingwell, 1997].**
 - Nepotpuno definirani onemogućuju planiranje projekta i praćenje promjena.

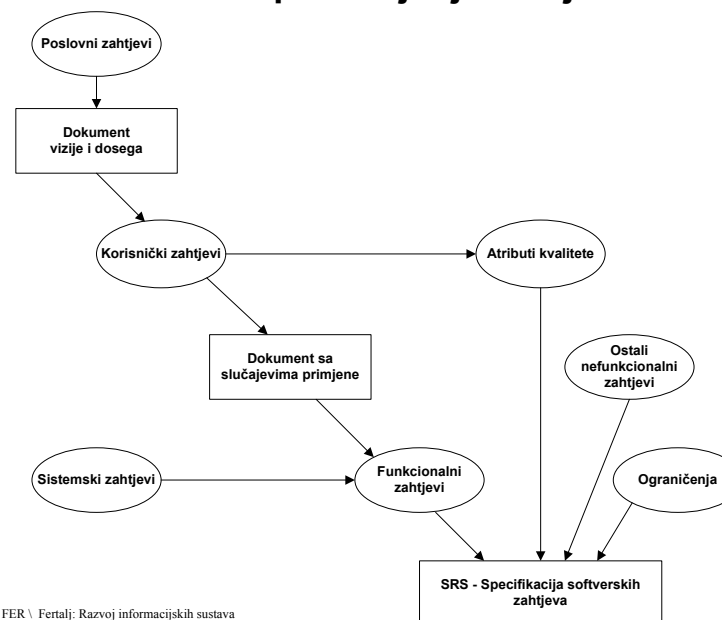


Vrste zahtjeva

- ❑ **Poslovni zahtjevi (zašto)**
 - ciljevi organizacije ili korisnički zahtjevi na višoj razini, opis problema koje treba riješiti
 - primjer: poslovna potreba, "Poboljšanje usluge postojećim klijentima i pridobivanje novih"
 - sadržani u dokumentima u kojima se opisuje vizija i doseg projekta
 - primjer: poslovni zahtjev, "Omogućiti Internet prodaju"
- ❑ **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)**
 - 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
- ❑ **Nefunkcionalni zahtjevi (kako ili kako dobro)**
 - standardi, pravila i ugovori kojih se proizvod mora pridržavati, opisi vanjskih sučelja, zahtjevi na performance, ograničenja na dizajn i implementaciju, te svojstva kvalitete (preciziraju opis proizvoda navodeći karakteristike proizvoda u različitim dimenzijama koja su važne ili korisniku, ili graditelju)
- ❑ **Prioriteti pojedinih elemenata**



Kontekst postavljanja zahtjeva



Primjeri poslovnih zahtjeva

- ❑ **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.
- ❑ **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 naručitelja podudaraju se s poslovnim ciljevima!**



Primjeri zahtjeva krajnjih korisnika

- ❑ **Prehrana kod bilo kojeg pružatelja usluga**
 - 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.
- ❑ **Ukinuti plaćanje unaprijed**
 - Treba izbjeći bilo kakvo plaćanje od strane studenata za potrebe ostvarivanja prava, a posebice unaprijed.
- ❑ **Ukloniti nepotrebne postupke za ostvarivanje prava**
 - Sustav mora biti tako koncipiran da kad se studentu jednom zavedu prava na matičnoj ustanovi nisu potrebna nikakva daljnja dokazivanja za ostvarivanje prava kod pružatelja usluga.
- ❑ **Smanjiti rizik gubitka ostvarenih prava**
 - Sustav mora onemogućiti zlorabu stečenih prava.
- ❑ **Lakše ostvarivanje ostalih prava iz studentskog standarda**
 - Npr. javni prijevoz po povlaštenoj cijeni, kazališta, kina, smještaj u studentskim domovima, student-servis, itd.



Primjer: sistemski zahtjevi

□ Primjer: PISFER-Podprojekt

- Sustav bi trebao biti integriran, tako da se sve informacije unose na mjestu nastanka
- Mogućnost da sve organizacijske jedinice imaju pravo pregleda, ažuriranja i unosa podataka uz određene dozvole
- Omogućiti kontinuirani pristup podacima (ne samo krajem mjeseca)
- Svi podaci bi trebali biti u digitalnom obliku
- Jednostavno i intuitivno korištenje sustava
- Korištenje jedinstvene baze podataka za sve službe



Primjer: funkcionalni zahtjevi

□ Primjer: storno dokumenta

- Radi se storno ulaznih dokumenata, unosom odgovarajućeg dokumenta. Dokument storna „nasljeđuje“ konto dokumenta koji bude storniran.
 - Stavka storna može se obrisati
 - Storno storna nije moguće
 - storno nužno ne poništava cijeli iznos primarnog dokumenta, nego ga korigira za iznos storna, iznos je pozitivan ili negativan, što po volji korisnika može biti i cijeli iznos primarnog dokumenta s negativnim predznakom
- Ne obavlja se storno uplate.
- Uz storno omogućiti i ispravak dokumenta (koji ne utječe na proknjižene stavke).
 - Ako je dokument uplata iznos uplate uvijek mora odgovarati sumi vezanih iznosa po računima pa i nakon ispravka, s tim da je u fazi ispravka moguće izmijeniti specifikaciju računa i pripadajućih iznosa.

□ Primjer: izvješće otvorenih stavaka (zahtjev referencira primjere izvješća)

- IOSi za tekuću poslovnu godinu (IOSI.doc)
- Sumarni IOSi (SumarniIOSI.doc)
 - Kartica dobavljača (KarticaTekuca.doc)
 - Kartica dobavljača s uključenom arhivom i kontinuiranim nastavkom u tekuću poslovnu godinu (KarticaArhiva.doc)



Primjer: nefunkcionalni zahtjevi

□ Viševalutni rad

- Devizni izvod poslovne banke sadrži iznose po tečaju poslovne banke.
- Za devizne uplate evidentirat će se uz kunski iznos i devizni iznos te valuta.
- Sva izvješća osim Kartica moraju sadržavati mogućnost konverzije primarne valute u zadanu valutu po srednjem tečaju NBH na datum knjiženja.

□ Praćenje promjena nad podacima

- svaki zapis sadrži identifikator korisnika i datum unosa/promjene

□ Izvoz podataka

- Omogućiti izvoz izvješća u Excel formatu



Zahtjevi na kvalitetu programske podrške

□ Jedna podjela može biti kao što je prikazano tablicom:

- Odabire se podskup kvalitativnih svojstva važnih za konkretni projekt.

□ Pojedini nefunkcionalni zahtjevi su međusobno proturječni.

- Proturječnost se razrješava postavljanjem prioriteta

Svojstva važnija korisnicima	Svojstva važnija razvojnicima
Dostupnost	Lakoća održavanja
Učinkovitost	Prenosivost
Prilagodljivost	Ponovna upotrebljivost
Integritet	Podložnost testiranju
Interoperabilnost	
Pouzdanost	
Robustnost	
Upotrebljivost	



Zahtjevi na kvalitetu programske podrške

❑ Dostupnost

- Dostupnost predstavlja postotak predviđenog aktivnog vremena sustava tijekom kojeg bi sustav trebao biti u potpunosti funkcionalan.
- Formalno, dostupnost je omjer MTTF / (MTTF+MTTR), to jest srednjeg vremena do kvara i sume srednjeg vremena do kvara i srednjeg vremena do oporavka od kvara
- Primjer: "Sustav će biti barem 99.5% dostupan radnim danima od 6 do 24 i barem 99.95 posto dostupan između 16 i 18 sati."

❑ Učinkovitost

- Učinkovitost je mjera koja govori koliko dobro sustav iskorištava kapacitet procesora, diskovni prostor, memoriju ili komunikacijske strukture.
- Primjer: "10% procesorskog kapaciteta i 15% memorije sustava će biti raspoloživo za vrijeme maksimalnog opterećenja sustava."
- Potrebno je razmotriti minimalne hardverske konfiguracije.

❑ Prilagodljivost

- Prilagodljivost govori koliko je napora potrebno da bi se proizvodu dodale nove mogućnosti.
- Svojstvo je presudno za proizvode koji se rade inkrementalno ili iterativno.



Zahtjevi na kvalitetu programske podrške (2)

❑ Integritet

- Integritet (ili sigurnost) sprječava neovlašteni pristup funkcijama sustava, sprječava gubitak informacija, štiti sustav od virusa i štiti tajnost podatka unesenih u sustav. Integritet je vrlo bitno svojstvo za mrežne aplikacije.
- Primjer: "Pristup transakcijama isplate imaju samo korisnici grupe A."

❑ Interoperabilnost

- Interoperabilnost definira koliko lako sustav može razmjenjivati podatke ili usluge s drugim sustavima.
- Procjenjuje se koje druge aplikacije se namjerava koristiti i koji podaci će se razmjenjivati.
- Npr. "Sustav će moći učitati podatke o transakcijama iz MS Access baze podataka."

❑ Pouzdanost

- Pouzdanost je vjerojatnost da će se softver u nekom razdoblju raditi bez pogreške.
- Mjeri se postotkom ispravno izvedenih operacija, trajanjem rada bez pogrešaka ili učestalosti pogrešaka.
- Treba uspostaviti kvantitativne zahtjeve za pouzdanost s obzirom na veličinu štete u slučaju pogreške i procijeniti da li se isplati maksimizirati pouzdanost.
- Ukoliko proizvod zadovolji zahtjeve za pouzdanost može se smatrati isporučivim (s obzirom na pouzdanost) čak i ako je poznato da još uvijek postoje defekti.
- Primjer: "Najviše 5 od 1000 kemijskih eksperimenata smije propasti zbog programske podrške."



Zahtjevi na kvalitetu programske podrške (3)

❑ Robustnost

- Robustnost je stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pojave pogrešnih podataka, defekata u softverskim ili hardverskim komponentama ili nepredviđenim operativnim uvjetima.
- Primjer: "Kod učitavanja podatka iz neispravne ulazne datoteke sustav će, kad naiđe na redak s pogreškom, prijaviti pogrešku i broj retka te pitati korisnika da li da nastavi učitavati datoteku od sljedećeg retka ili otkáže učitavanje."

❑ Upotrebljivost

- Upotrebljivost mjeri napor koji treba uložiti da bi se podaci pripremili za unos, obradili i protumačili, to jest lakoću korištenja proizvoda (eng. user friendliness)..
- Procjenjuju se krivulja učenja te mogućnost uporabe od strane početnika odnosno iskusnih korisnika.
- Sukladnost standardima ili uzoru: "Sve kratice na u izborniku *File* će biti iste kao i u *File* izborniku programa XXXX."

❑ Lakoća održavanja

- To je mjera koliko lako je popraviti pogrešku ili napraviti izmjenu u softveru. Ovisi o tome koliko lako je razumjeti softver, promijeniti ga i ponovno testirati.
- Lakoća održavanja se može mjeriti prosječnim vremenom potrebnim da se riješi problem i postotkom popravaka koji su ispravno obavljani.
- Primjer: "Izvjешća će biti izmijenjeno u roku od tjedan dana od zaprimanja novih obrazaca za izvješća."



Zahtjevi na kvalitetu programske podrške (4)

❑ Prenosivost

- Napor potreban da bi se dio softvera prenio iz jedne operacijske okoline u drugu jest mjera prenosivosti.
- Dizajnerski principi koji čine softver prenosivim su slični onima koji ga čine ponovno iskoristivim.
- Procjenjuju se dijelovi koji moraju biti prenosivi te okoline na koje softver treba moći prenijeti.

❑ Ponovna upotrebljivost

- Ovo svojstvo govori do koje mjere se softverska komponenta može upotrebljavati u drugim aplikacijama.
- Razvoj ponovno iskoristive softverske komponente znatno je skuplje od razvoja jednokratno korištene komponente.
- Ponovno iskoristiv softver mora biti modularan, dobro dokumentiran, neovisan o specifičnoj aplikaciji i operativne okoline i donekle općenit.

❑ Podložnost testiranju

- Podložnost testiranju govori o tome koliko lako je testirati softversku komponentu kako bi se pronašla pogreška.
- Očekuje se kod proizvoda sa složenim logikom obrade te kod proizvoda sa složenim poveznicama između funkcija.
- Ovo svojstvo je također bitno kod proizvoda koji će se često mijenjati, da bi se olakšalo regresijsko testiranje.



Među zavisnost

	Dostupnost	Učinkovitost	Prilagodljivost	Integritet	Interoperabilnost	Lakoća održavanja	Prenosivost	Pouzdanost	Ponovna isupotrebljivost	Robustnost	Podložnost testiranju	Upotrebljivost
Dostupnost								+		+		
Učinkovitost			-		-	-	-	-		-	-	-
Prilagodljivost		-		-		+	+	+			+	
Integritet		-			-				-		-	-
Interoperabilnost		-	+	-			+					
Lakoća održavanja	+	-	+					+			+	
Prenosivost		-	+		+	-			+		+	-
Pouzdanost	+	-	+			+				+	+	+
Ponovna isupotrebljivost		-	+	-	+	+	+	-			+	
Robustnost	+	-						+				+
Podložnost testiranju	+	-	+			+		+				+
Upotrebljivost		-								+	-	



Karakteristike dobrih izjava o zahtjevima [IEEE]

- ☐ **Potpunost**
 - Svaki zahtjev mora potpuno opisivati funkcionalnost i sadržavati sve informacije potrebne izvođaču pri dizajnu i ugradnji funkcionalnosti.
- ☐ **Točnost**
 - Svaki zahtjev mora točno opisivati funkcionalnost koju treba ugraditi.
 - Referenca za točnost je izvor zahtjeva, npr. korisnik ili specifikacija sistemskih zahtjeva.
 - U slučaju da softverski zahtjev nije u skladu sa sistemskim zahtjevom, smatra se netočnim.
 - Samo predstavnici korisnika mogu odlučiti je li neki korisnički zahtjev ispravno zapisan.
- ☐ **Ostvarivost (izvodljivost)**
 - Zahtjevi moraju biti izvodljivi s obzirom na sredstva i tehnologiju te ograničenja i okolinu sustava.
 - U prikupljanju zahtjeva trebaju sudjelovati razvojnici, radi procjene što se i kako može napraviti.
- ☐ **Nužnost**
 - Pojedini zahtjev treba sadržavati ono što korisnik stvarno treba ili ono što je potrebno radi povezivanja s drugim sustavima ili radi standarda.
- ☐ **Redoslijed po prioritetu**
 - Svakom zahtjevu treba pridijeliti prioritet
 - Kada bi svi zahtjevi bili jednako važni ne bi se moglo mijenjati i replanirati
- ☐ **Nedvosmislenost**
 - Svi se zahtjevi moraju moći na jedinstven način i konzistentno interpretirati.
- ☐ **Mogućnost provjere**
 - Svaki zahtjev treba moći provjeriti – zahtjevi koje nije moguće testirati nisu dobro postavljani.



Primjer nepotpunog zahtjeva

- ☐ **"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

- ☐ **"Program 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

- ❑ "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 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 retka i tekst HTML pogrešaka, te opis pojedine pogreške.
 - Ukoliko nema pogrešaka prilikom analize, neće se generirati izvješće.



Inženjerstvo zahtjeva

- ❑ Inženjerstvo zahtjeva
 - Inženjerstvo zahtjeva dijeli se na: *razvoj zahtjeva* i *upravljanje zahtjevima*.
 - Razvoj zahtjeva dalje se dijeli na *određivanje* (*iznalaženje*) *zahtjeva* (*elicitation*), *analizu*, *specificiranje* i *verifikaciju*. Ove poddiscipline obuhvaćaju sve aktivnosti vezane za prikupljanje, vrednovanje i dokumentiranje zahtjeva.

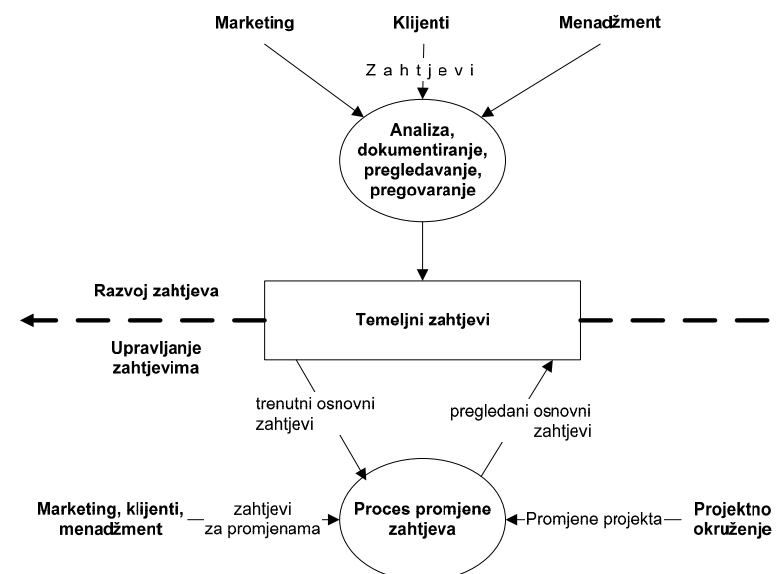


Razvoj zahtjeva i upravljanje zahtjevima

- ❑ Razvoj zahtjeva
 - Identificiranje razreda korisnika i njihovih predstavnika ("najbolji", ključni korisnik).
 - Iznošenje potreba korisnika.
 - Razumijevanje poslovnih potreba, zadataka i ciljeva.
 - Analiza prikupljenih informacija, kategorizacija, odbacivanje nebitnih informacija.
 - Razdioba zahtjeva s razine sustava na podsustave i određivanje udjela tih zahtjeva u komponentama.
 - Razumijevanje relativne važnosti (težine) svojstava kvalitete.
 - Određivanje prioriteta ugradnje.
 - Dokumentiranje i modeliranje zahtjeva
 - Provjera specifikacije zahtjeva
- ❑ Upravljanje zahtjevima
 - Rokovi prihvatanja zahtjeva
 - Pregled promjena zahtjeva i procjena njihovog utjecaja, prije nego što se donese odluka o prihvatanju ili odbijanju promjene
 - Kontrolirana ugradnja prihvaćenih promjena zahtjeva.
 - Ažuriranje plana projekta sukladno promjeni zahtjeva
 - Pregovori o rokovima isporuke ovisno o utjecaju promjena
 - Praćenje i povezivanje pojedinih zahtjeva s odgovarajućim izvornim kodom i primjerima za testiranje.
 - Praćenje statusa aktivnosti i promjena zahtjeva tijekom projekta



Razvoj zahtjeva naspram upravljanja zahtjevima



Zaključivanje zahtjeva

- ☐ **Prikupljanje zahtjeva završava postizanjem (i potpisivanjem) sporazuma o zahtjevima.**
- ☐ **Važno je da obje strane znaju što potpisuju.**
 - U ranoj fazi projekta nije moguće odmah i detaljno znati sve zahtjeve. Nedvojbeno je da će se neki zahtjevi tokom vremena promijeniti.
- ☐ **Razvojna ekipa ne bi smjela imati stav da se potpisani dokument zahtjeva ne može kasnije mijenjati !!!**
 - Specifikacija zahtjeva sadrži osnovicu zahtjeva, a kasniji zahtjevi ili promjene ulaze u proces upravljanja promjenama.
- ☐ **Primjer:** \Analiza\DOO-Specifikacija
 - dvije iteracije teksta, prototip, nekoliko promjena pred isporuku



Tehnike inženjerstva zahtjeva

Određivanje zahtjeva

- ☐ **Poslovni zahtjevi**
 - Sve što opisuje financijski, trgovački ili drugi poslovni probitak koji korisnici, ili organizacija koja razvija sustav, mogu dobiti je najvjerojatnije poslovni zahtjev.
- ☐ **Slučajevi korištenja ili scenariji**
 - Općenite izjave o poslovnim zadacima koje korisnici moraju obavljati najvjerojatnije su slučajevi korištenja, dok specifični opisi zadataka predstavljaju korisničke scenarije.
 - Specifične zadatke treba generalizirati u općenite slučajeve korištenja.
- ☐ **Poslovna pravila**
 - Kada korisnik izjavi da neku aktivnost mogu obavljati samo pojedine osobe ili uloge, pod određenim uvjetima, on možda opisuje poslovno pravilo.
 - Poslovna pravila su operativni principi poslovnih procesa. Oni nisu funkcionalni zahtjevi.
- ☐ **Funkcionalni zahtjevi**
 - Izjava koja počinje s «Korisnik mora moći obaviti neku funkciju», ili «Sustav treba moći demonstrirati određeno ponašanje» je najvjerojatnije funkcionalni zahtjev.
 - Funkcionalni zahtjevi opisuju vidljivo ponašanje sustava i definiraju što će sustav raditi.
 - Treba biti svima jasno zašto sustav «mora» biti u stanju obavljati određene funkcije. Predloženi funkcionalni zahtjevi ponekad predstavljaju zastarjele ili neučinkovite poslovne procese.
- ☐ **Atributi kvalitete**
 - Izjave koje opisuju kako dobro sustav obavlja funkciju su atributi kvalitete, tj. jedna vrsta nefunkcionalnih zahtjeva.
 - Zahtjeve koji opisuju poželjne karakteristike sustava: brzinu, jednostavnost, intuitivnost, robusnost, pouzdanost, sigurnost i efikasnost treba razmotriti s korisnicima da bi se precizno utvrdilo što oni misle pod tim dvosmislenim i subjektivnim terminima.



Određivanje zahtjeva (2)

- ☐ **Zahtjevi vanjskih sučelja**
 - Ovaj razred zahtjeva opisuje veze između sustava i vanjskog svijeta.
 - Specifikacija sistemskih zahtjeva trebala bi uključivati odlomke za ove zahtjeve uključujući mehanizme komunikacije za korisnike, hardver i druge softverske sustave.
- ☐ **Ograničenja**
 - Ograničenja su uvjeti koji ograničavanju izbor rješenja raspoloživih projektantu ili programeru.
 - Spadaju u nefunkcionalne zahtjeve.
 - Neka ograničenja mogu pomoći u zadovoljavanju atributa kvalitete. Primjer je poboljšanje prenosivosti korištenjem samo standardnih naredbi programskog jezika.
- ☐ **Definicije podataka**
 - Definicija podataka je bilo koji opis formata, dozvoljenih vrijednosti, pretpostavljenih vrijednosti ili kompozicija složenih podatkovnih struktura.
 - Sve definicije bi trebalo pohraniti u rječnik podataka.
- ☐ **Ideje o rješenju**
 - Ako korisnik opisuje specifičan način interakcije sa sustavom kojom bi mogao obaviti neku akciju, npr. «Korisnik odabire podatak koji on želi iz padajuće liste», on predlaže rješenje, a ne zahtjev.
 - Predloženo rješenje može odvući pažnju od pravih zahtjeva. Kod postavljanja zahtjeva treba se usmjeriti na ono što je potrebno obaviti, a ne na način realizacije.
 - Treba istražiti zašto korisnik predlaže određenu ugradnju jer to može pomoći u razumijevanju stvarne potrebe i korisnikovih očekivanja o načinu kako bi sustav trebao raditi.



Najčešći problemi pri određivanju zahtjeva

❑ Ponašanja korisnika

- Taktika kuhinjskog sudopera
 - korisnik navodi (preko)brojne potrebe: hrpu nepotrebnih izvještaja, formi, sortiranja, izračuna i sl.
 - ovakav pristup obično je uzrokovan pomanjkanjem iskustva korisnika koji ne zna što bi mu stvarno trebalo zatrebati ili nije u stanju razlučiti ono što je bitno
- Taktika dimne zavjese
 - korisnik traži više mogućnosti a zapravo mu je potrebna samo jedna ili dvije
 - dodatni zahtjevi služe za postizanje bolje nagodbe s analitičarom
 - taktika obično oslikava korisnika s dobrim poznavanjem onoga što želi, a zahtjeve treba reducirati na one realne i izvodljive
- Taktika "Treba mi isto ali u boljem obliku"
 - korisnik koji koristi ovu taktiku nedostaje volja ili znanje, a ponekad oboje
 - šanse za uspješno definiranje problema su male jer samo korisnik može izraziti svoje potrebe i probleme

❑ Korisnik je sklon prešutjeti izuzetke, koji su bitni (nužni !!!) za uspješnu realizaciju, a obično zahtijevaju i najviše napora tijekom ugradnje.

- "To je naša jedina procedura ... (osim kada ...)"

❑ Ne izjednačavati "tako se uvijek radi" s "tako treba raditi"!

- Primjer: Google Search "Company Policy"



Analiza zahtjeva

❑ Analiza zahtjeva

- Analiza zahtjeva obuhvaća pročišćavanje, analizu i pomno ispitivanje prikupljenih zahtjeva kako bi bili sigurni da svi sudionici projekta razumiju što zahtjevi znače, a ujedno služi da se pronađu pogreške, nedostaci i ostali defekti.
- Izrada dijagrama konteksta sustava
 - Jednostavni model koji definira ograde i sučelja sustava te identificira tok informacija i materijala kroz sučelje
- Izrada prototipova korisničkog sučelja
 - model oponašanja kao sredstvo razjašnjenja zahtjeva
- Analiza ostvarivosti zahtjeva
 - procjena troškova, performanci, rizika, konflikata prema ostalim zahtjevima
- Postavljanje prioriteta
 - određuje se relativni prioritet implementacije za pojedini slučaj korištenja, svojstvo proizvoda ili drugi zahtjev
- Modeliranje zahtjeva
 - izrada modela strukture, ponašanja i stanja
- Izrada rječnika podataka
 - središnji repozitorij za definiranje podataka i struktura korištenih u sustavu te zajedničku terminologiju



Analiza i modeliranje sustava

❑ Na temelju izjava korisnika i prikupljene dokumentacije modeliraju se pojedine komponente sustava (proces, podaci, događaji)

- Moguća su preslikavanja uočenih imenica i glagola u konkretne elemente modela kakvi će biti opisani u nastavku.

Tip riječi	Primjer	Komponente analitičkog modela
Imenica	•Ljudi, organizacije, softverski sustavi, podatkovne jedinice, ili postojeći objekti	•Terminatori ili spremnici podataka (DFD) •Entiteti ili njihovi atributi (ERD) •Razredi ili njihovi atributi (dijagram razreda)
Glagol	•Akcije, ono što korisnik može poduzeti, ili događaji koji se mogu dogoditi	•Procesi(DFD) •Odnosi (ERD) •Prijelazi (iz stanja u stanje) (STD) •Metode razreda (dijagram razreda)

"Kemičar ili član osoblja kemijskog laboratorija može *podnijeti* zahtjev za jednom ili više kemikalija. Zahtjev može biti *udovoljen* ili *dostavom pakiranja kemikalije* koja se već nalazila na *zalihi kemijskog laboratorija* ili *upućivanjem narudžbe* za novim *pakiranjem kemikalije* od *vanjskog dobavljača*. Osoba koja upućuje zahtjev mora imati mogućnost *pretraživanja kataloga kemikalija vanjskog dobavljača* dok sastavlja *narudžbu*. Sustav mora *pratiti* status svakog *zahtjeva za kemikalijama* od trenutka kad je ispunjen do trenutka kad je *udovoljen* ili *otkazan*. Također, mora *pratiti povijest* svakog *pakiranja kemikalija* od trena kad *stigne* u kompaniju do trenutka kad je potpuno *upotrijebljen* ili *odbačen*."



Postavljanje prioriteta

❑ Nužno svojstvo (must have) - Da li se 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!
 - Nužni zahtjevi se ne mogu rangirati jer su potrebni za prvu verziju!

❑ Poželjno svojstvo (should have) – Treba ispuniti do kraja projekta

- Ranije verzije sustava mogu pružiti (ne potpunu) funkcionalnost bez tih zahtjeva.
- Poželjni zahtjevi mogu i trebaju biti rangirani.

❑ Neobvezno svojstvo (could have) – Moglo bi se ispuniti

- Svojstva i mogućnosti bez kojih se može.
- Iako bi ih lijepo bilo imati, to nisu pravi zahtjevi.
- Ovi zahtjevi također mogu biti rangirani.

❑ Nepotrebna svojstva (won't have)



Primjeri rangiranja prioriteta

oznaka	značenje	referenca
visok	<i>mission-critical</i> zahtjevi, potrebni za slijedeće izdanje (<i>release</i>)	Wieggers 99
srednji	podrška potrebnim sistemskim operacijama; zahtjeva se, ali može se odgoditi za slijedeće izdanje ukoliko je potrebno	
nizak	poboljšanje funkcionalnosti ili kvalitete	
neophodan	softverski proizvod neprihvatljiv ukoliko ovi zahtjevi nisu ispunjeni na dogovoreni način	IEEE 98
kondicionalan	može poboljšati proizvod, ali proizvod neće biti neprihvatljiv bez toga	
opcionalan	klasa funkcija koje bi mogle biti vrijedne uključivanja, ali možda i ne	
3	mora biti savršeno implementirano	Kovitz 99
2	treba raditi, no ne spektakularno dobro	
1	može sadržavati pogreške (<i>bugove</i>)	



Formalno određivanje prioriteta

Formalni postupci

- *Quality Function Deployment* (QFD) - odnos između osobina predloženog proizvoda i koristi/vrijednosti za naručitelja,
- *Total Quality Management* (TQM) - rangira se svaki zahtjev u odnosu na nekoliko kriterija važnih za uspjeh projekta

Primjer, QFD

- Navesti u tablici sve zahtjeve (*requirements*), osobine (*features*) ili slučajeve korištenja (*use cases*) za koje se žele postaviti prioriteta. Veći broj elemenata po potrebi hijerarhijski grupirati.
- Procijeniti relativnu korist (*benefit*) svakog svojstva za klijenta ili posao u rasponu od 1 do 9, pri čemu 1 predstavlja dvojbenu korist, a 9 enormnu vrijednost.
- Procijeniti relativnu kaznu (*penalty*) ukoliko svojstvo ne bude ugrađeno. Na ljestvici od 1 do 9, 1 predstavlja nepostojanje kazne, a 9 vrlo ozbiljan nedostatak. Pripaziti na zahtjeve koji imaju i nisku korist i nisku kaznu, jer oni pridonose troškovima, a imaju malu vrijednost.
- Stupac koji predstavlja ukupnu vrijednost je zbroj relativne koristi i relativne kazne pomnoženih s odgovarajućim težinama.
- Procijeniti relativni trošak (*cost*) implementacije pojedinog svojstva, od 1 (nisko) do 9 (visoko). Razvojnici mogu rangirati troškove na temelju procjena koje uključuju složenost zahtjeva, količinu rada koje je potrebno uložiti za razvoj korisničkog sučelja, mogućnost ponovne uporabe koda, obujma potrebne dokumentacije, testova itd.
- Na sličan način, procijeniti relativni stupanj rizika vezanog uz svako svojstvo od 1 (izuzetno mali rizik) do 9 (ozbiljna zabrinutost oko izvedivosti, problemi s osobljem ili tehnologijom). Uobičajeno je da su korist, kazna i trošak jednake težine, no svakom od tih faktora može se dodijeliti različita težina.



Primjer formalnog određivanja prioriteta

Proračunati vrijednost prioriteta (*priority value*) za svaku osobinu:

$$\text{prioritet} = \frac{\text{vrijednost}(\%)}{(\text{trošak}(\%)*\text{težina_troška}) + (\text{rizik}(\%)*\text{težina_rizika})}$$

Težina	2	1			1		0.5		
Mogućnost	Rel. korist	Rel. kazna	Vrijednost	Vrijednost%	Rel. trošak	Trošak%	Rel. rizik	Rizik%	Prioritet
M1	5	3	13	15.12%	2	11.11%	1	4.00%	1.152937
M2	9	7	25	29.07%	5	27.78%	3	12.00%	0.860618
M3	5	5	15	17.44%	3	16.67%	2	8.00%	0.843961
M4	2	1	5	5.81%	1	5.56%	4	16.00%	0.428898
M5	4	9	17	19.77%	4	22.22%	8	32.00%	0.517171
Mn	4	3	11	12.79%	3	16.67%	7	28.00%	0.417088
Ukupno	29	28	86	100.00%	18	100.00%	25	100.00%	

Lista se sortira u padajućem redoslijedu prema prioritetu.

- Ono što je na vrhu liste ima najpoželjniji odnos vrijednosti, troška i rizika, te zato treba imati najviši prioritet.



Verifikacija zahtjeva

- Aktivnosti ovjere osiguravaju da su izjave o zahtjevima precizne, potpune i da demonstriraju željene karakteristike.

Provjera dokumentacije sa zahtjevima

- Formalna provjera zahtjeva smatra se jednom od najkorisnijih softverskih tehnika.
- Provodi je mala ekipa (analitičari, dizajneri, ... , korisnici) koja pomno ispituje specifikacije i modele.

Pisanje testova, probnih slučajeva

- Razvoj prototipa, izvođenje specifičnih scenarija korištenja, kako bi se provjerilo očekivano ponašanje proizvoda u određenim uvjetima.
- Probne slučajeve treba povezati unatrag s funkcionalnim zahtjevima kako bi se osiguralo da nijedan zahtjev nije zanemaren.
- Korištenjem test slučajeva verificira se točnost modela i prototipova.

Pisanje korisničkog priručnika

- Skica korisničkog priručnika koja se radi na početku razvoja zahtjeva može poslužiti kao specifikacija zahtjeva ili kao pomoć u analizi zahtjeva.
- Dobar korisnički priručnik opisuje svu vidljivu funkcionalnost. Ostali zahtjevi dokumentiraju se u SRS.

Definiranje kriterija prihvatljivosti

- U suradnji s korisnicima definiraju se uvjeti pod kojima će proizvod zadovoljiti zahtjeve i biti spreman za korištenje.
- Testovi prihvatljivosti rade se na osnovu scenarija korištenja ili opisanih slučajeva korištenja.



Upravljanje zahtjevima

- Upravljanje promjenama podrazumijeva proces za predlaganje promjena i ispitivanje potencijalnih troškova i utjecaja promjena na projekt.

❑ Definiranje postupka za promjenu zahtjeva

- Treba definirati postupak kojim se novi zahtjev ili promjena postojećeg zahtjeva analizira i odlučuje o eventualnom prihvatanju.
- Sve predložene promjene moraju slijediti unaprijed definiranu proceduru.

❑ Uspostava odbora za promjene

- Zahtjevima najčešće upravlja poseban odbor, koji sačinjavaju ključni članovi projekta.
- Odbor odlučuje o usvajanju zahtjeva te postavlja prioritete i rokove njihove ugradnje.

❑ Analiza utjecaja promjena zahtjeva

- Za pojedinu promjenu zahtjeva procjenjuje se utjecaj na organizaciju projekta, vremenski raspored ili druge zahtjeve.
- Analiza utjecaja promjena treba poslužiti donošenju dobrih (ispravnih, jedino mogućih) odluka o (ne)prihvatanju promjena.

❑ Praćenje promjena zahtjeva na svim proizvodima

- Kada je promjena nekih zahtjeva prihvaćena potrebno je kroz matricu praćenja zahtjeva pronaći sve ovisne komponente (izvorni kod, testni slučajevi, neki drugi zahtjev,...).



Upravljanje zahtjevima (2)

❑ Uspostava vremenske osnovice (baseline) i kontrole verzija

- Potrebno je definirati vremensku osnovicu za zahtjeve, tj. sliku dogovorenih zahtjeva u određenom vremenskom trenutku.
- Nakon što je osnovica definirana, zahtjevi prolaze postupak promjena.
- Svaka verzija specifikacije zahtjeva mora biti jednoznačno označena da bi se izbjegle zabune uslijed razlika između prethodnih i trenutne verzije.

❑ Praćenje povijesti promjena zahtjeva

- Zapisi promjena sadrže podatke o vremenu promjene, vrsti i sadržaju promjena, razlozima promjene, oznakom verzije te podatkom o tome tko je ažurirao dokument.

❑ Praćenje statusa zahtjeva

- Evidentiraju se ključni atributi funkcionalnih zahtjeva i njihov status (predložen, odobren, ugrađen, provjeren), da bi se u svakom trenutku moglo znati broj zahtjeva u pojedinoj kategoriji.

❑ Mjerenje stabilnosti zahtjeva

- Ako je broj odobrenih promjena tjedno ili mjesečno jako velik može biti da problem nije dobro shvaćen, da opseg projekta nije dobro definiran ili da projekt nije dobro upravljan.

❑ Korištenje alata za upravljanje zahtjevima

- evidencija zahtjeva, međusobne povezanosti i statusa
- verzioniranje zahtjeva i dokumentacije



Dokumentiranje zahtjeva

Dokumentiranje zahtjeva

❑ 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

❑ Specifikacija zahtjeva (Requirements Specification)

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

❑ Primjeri:

- \Analiza\Firma-Zahtjevi, DOO-Specifikacija
- \Predlošci\SpecifikacijaZahtjeva



Predložak dokumenta specifikacije zahtjeva

☐ Predložak nastao na temelju standarda IEEE 830

- Stavke koje nisu primjenjive u konkretnom projektu se izostavljaju.
- Treba uključiti i sadržaj i povijest ispravki dokumenta

<div>1. Uvod</div> <div>1.1 Namjena</div> <div>1.2 Konvencije dokumenta</div> <div>1.3 Tko treba čitati dokument i savjeti za čitanje dokumenta</div> <div>1.4 Opseg proizvoda</div> <div>1.5 Reference</div> <div>2. Sveobuhvatni pregled</div> <div>2.1 Kontekst proizvoda</div> <div>2.2 Funkcije proizvoda</div> <div>2.3 Kategorije korisnika i svojstva</div> <div>2.4 Okruženje u kojem se izvodi proizvod</div> <div>2.5 Ograničenja dizajna i ugradnje</div> <div>2.6 Pretpostavke i ovisnosti</div> <div>3. Zahtjevi za sučeljem</div> <div>3.1 Korisničko sučelje</div> <div>3.2 Hardversko sučelje</div> <div>3.3 Softversko sučelje</div> <div>3.4 Komunikacijsko sučelje</div>	<div>4. Svojstva sustava</div> <div>4.x Svojstvo X</div> <div>4.x.1 Opis i prioriteti</div> <div>4.x.2 Nizovi pobuda/odziv</div> <div>4.x.3 Funkcijski zahtjevi</div> <div>5. Ostali nefunkcionalni zahtjevi</div> <div>5.1 Zahtjevi za performansama sustava</div> <div>5.2 Zahtjevi za sigurnošću korisnika</div> <div>5.3 Zahtjevi za sigurnošću podataka</div> <div>5.4 Kvaliteta programske podrške</div> <div>5.5 Poslovna pravila</div> <div>5.6 Korisnička dokumentacija</div> <div>6. Ostali zahtjevi</div> <div>Dodatak A: Rječnik</div> <div>Dodatak B: Modeli i dijagrami</div> <div>Dodatak C: Lista nedovršenih/neodređenih zahtjeva</div>
--	--



Preporuke za pisanje zahtjeva

☐ Preporuke:

- koristiti kratke rečenice i odlomke
- rečenice trebaju biti kompletne, pravopisno i gramatički ispravne
- pojmove treba upotrebljavati dosljedno
- zahtjeve treba izricati dosljedno: "Sustav će..." ili "Korisnik će..." iza čega slijedi glagol popraćen uočljivim rezultatom, npr.: "Sustav za nadzor skladišta će ispisati sva pakiranja zatraženog artikla u skladištu."
- mora biti jasno što znači da će sustav "procesirati", "podržati", "upravljati".
- izbjegavati nejasne, subjektivne izraze kao što su: lagano, jednostavno, brzo, efikasno, podržati, nekoliko, prihvatljivo ...
- izbjegavati izraze usporedbe kao što su: poboljšati, maksimizirati, minimizirati, umanjiti, optimirati...
- kvantificirati stupanj poboljšanja ili odrediti maksimalne ili minimalne prihvatljive vrijednosti nekog svojstva.



Preporuke za pisanje zahtjeva

☐ Zahtjeve više razine treba razložiti u dovoljan broj zahtjeva niže razine da bi se razjasnio zahtjev i uklonila moguća neodređenost.

- Dobra mjera su zahtjevi koji se mogu nezavisno testirati.
- Zahtjev za koji možete smisliti mali broj test slučajeva najvjerojatnije je opisan dovoljno detaljno.
- Ukoliko su slučajevi za testiranje veliki i složeni, najvjerojatnije se radi o više zahtjeva koje treba razložiti.
- Svi bi zahtjevi trebali biti dosljedno napisani približno jednakom razinom detalja.
- Izbjegavati veznike "i" i "ili" jer je to naznaka da se radi o više zahtjeva u kombinaciji.
- Pojedini zahtjev mora biti naveden samo na jednom mjestu. Za povećanje čitljivosti teksta koristiti reference unutar dokumenta.



Označavanje zahtjeva

☐ Uzastopni brojevi

- Novi zahtjev dobiva raspoloživi serijski broj (npr. Z01, Z02, ...).
- Broj obrisano zahtjeva ne koristi se ponovno.
- Nepregledno, teško za pratiti veći broj zahtjeva

☐ Hijerarhijsko numeriranje

- Zahtjev u poglavlju 3.2. ima oznaku koja počinje s 3.2., npr. 3.2.4.
 - Više znamenaka u oznaci označava detaljniji zahtjev.
- Jednostavna i najčešće korištena tehnika.
 - Uređivač teksta se može podesiti da automatski dodjeljuje oznake.
 - Znamenke ne govore ništa o namjeni zahtjeva.
 - Problem brisanja zahtjeva koje uzrokuje posmak slijednih oznaka.
- Poboljšanje: tekstovne oznake unutar brojčanih hijerarhija
 - Primjer: "Poglavlje 3.2. - Funkcije editora", sa zahtjevima u poglavlju označenim s "ED-01", "ED-02", itd.
 - Ovaj pristup omogućuje hijerarhiju i organizaciju, oznake su kratke i smislene te manje osjetljive na promjene teksta.



Označavanje zahtjeva (nastavak)

❑ Hijerarhijske tekstovne oznake

- Primjer zahtjeva: "Sustav će upozoriti korisnika da potvrdi svaki zahtjev za ispisom više od 10 kopija".
 - Ovaj zahtjev može se označiti s: ISPIS.KOPIJE.POTVRDA.
- Hijerarhijske tekstovne oznake su strukturirane i semantički smislene a zahtjevi se brišu i dodaju bez problema.
- Glavni nedostatak jest da su nezgrapnije u odnosu na hijerarhijsko numeriranje.

❑ Potpunost

- Nijedan zahtjev ili informacija ne smiju nedostajati.
- Zahtjeve koje nedostaju teško je uočiti. Razumijevanje poslovnog procesa može pomoći u sprječavanju nepotpunosti.
- Informacije koje nedostaju treba u specifikaciji označiti posebnom oznakom (npr. TBD - "to be determined"), te ih razriješiti prije početka ugradnje.



Inženjerstvo zahtjeva i upravljanje projektom

❑ Planovi projekta trebaju biti temeljeni na funkcionalnosti koja se ugrađuje i promjenama u zahtjevima koji utječu na te planove.

❑ Odabir odgovarajućeg životnog ciklusa razvoja

- Ako su početni zahtjevi kolebljivi, treba odabrati razvojni ciklus koji će omogućiti ugradnju zahtjeva kad oni budu bolje shvaćeni.
- Klasični "vodopadni" životni ciklus razvoja može se održati samo u slučaju da su zahtjevi potpuno definirani na početku projekta.
- Kada su doseg ili zahtjevi loše definirani prikladniji je inkrementalni pristup uz robusnu arhitekturu.

❑ Zasnivanje plana projekta na zahtjevima

- Plan i raspored moraju se dinamički (iterativno) prilagođavati, sukladno povećanju jasnoće opsega i zahtjeva. Prva procjena truda uobičajeno se zasniva na inicijalnoj viziji i doseg projekta. Rane procjene su nesigurne i treba ih ažurirati nakon što zahtjevi budu bolje definirani i/ili shvaćeni.

❑ Pregovaranje termina isporuke prilikom promjena zahtjeva

- Prilikom usvajanja novih zahtjeva treba procijeniti da li se može održati postojeći raspored i kvaliteta isporuke s postojećim resursima. Ako ne, potrebno je dogovoriti novi rok isporuke. U slučaju neuspjelih pregovora, treba priopćiti vjerojatni ishod te u plan ugraditi stanje rizika i moguće prijetnje uspjehu.



Rizici lošeg planiranja zahtjeva

❑ Nedovoljna uključenost korisnika

- Bez korisnika ne može se točno znati što korisnici žele → neprihvatljivi proizvodi.
- Uz uključivanje krajnjih korisnika, nužna je i potpora rukovodstva!

❑ Čudni korisnički zahtjevi

- Neopravdana promjena mišljenja tijekom izvedbe → prekoračenja rokova i degradacije kvalitete proizvoda
- Definirati viziju, opseg, ciljeve i ograničenja na projekt, kriterije za mjerenje uspješnosti te mehanizam upravljanja promjenama.

❑ Nejasni (dvosmisleni) zahtjevi

- Situacija u kojoj čitatelj(i) zahtjeva zahtjev tumače na više načina → gubljenje vremena i prepravljavanje
- Rješenje može biti analiza/provjera zahtjeva koju provode osobe koje nisu sudjelovale u prikupljanju zahtjeva.



Rizici lošeg planiranja zahtjeva

❑ Pretjerano ukrašavanje

- Želja izvođača da dodaju novu funkcionalnost "koja bi se trebala svidjeti korisnicima" i zahtjev korisnika za dodacima koji dobro izgledaju ali ne pridonose funkcionalnosti → nepotrebni dodaci i gubljenje vremena
- Nužna je procjena izvodljivosti s obzirom na zadane okvire.

❑ Minimalističke specifikacije

- Tendencija postavljanja minimalnih zahtjeva ili skiciranja koncepta, uz želju da ih izvođači nadopune tijekom izvedbe → frustracije izvođača, neispunjena očekivanja korisnika.

❑ Zanimarivanje potreba

- Zanimarivanje potreba određenih (grupa) korisnika → stvaranje 'opozicije' projektu
- Uvažiti ciljne grupe korisnika, uz objašnjenje prioriteta.

❑ Površna analiza zahtjeva - loše planiranje

- "Ovo je moja ideja za Kad bi moglo biti gotovo?"
- Slabo razumijevanje zahtjeva vodi do pretjerano optimističnih procjena. Na prethodno pitanje je teško dati točan odgovor bez detaljnog uvida u zahtjeve.
- U slučajevima nedovoljne informacije ili nedovoljnog razumijevanja problema, preuranjeni konkretni odgovor na pitanje treba izbjegavati.
- Alternativno, može se dati okvirnu procjenu koristeći termine: "u najboljem slučaju", "vjerojatno", "u najgorem slučaju" ili izražavajući postotak sigurnosti procjene.



Reference

- ❑ http://www.jiludwig.com/Requirements_Management_Tools.html



Modeliranje funkcija i procesa

4/13

Strukturirana analiza

- ❑ **Moderna strukturirana analiza = Logički dizajn (česti sinonim)**
 - strukturirana analiza → strukturirani proces i rezultati analize
 - tehnika modeliranja poslovnih zahtjeva na sustav
 - usmjerena procesima, ali se razvila tako da obuhvaća i podatke
 - strukturiranje procesa, ulaza, izlaza te spremišta podataka potrebnih da bi se ogovorilo na poslovne događaje
 - logički modeli – određivanje ŠTO je sustav i ŠTO mora raditi (ne KAKO)
 - koriste se dijagrami toka podataka za logički prikaz poslovnih zahtjeva nezavisno od tehničkih rješenja → logički dizajn
 - izražavaju suštinu sustava → sinonimi: esencijalni, konceptualni, poslovni modeli
 - uključuje određivanje prioriteta zahtjeva



Dekompozicija procesa i funkcija

- ❑ **Logički procesi: funkcije, događaji i elementarni procesi**
 - rad i akcije koji se obavljaju bez obzira na način ugradnje i resurse sustava (ljudi, strojevi, softver)
 - Neke metode poistovjećuju funkcije i procese!
- ❑ **Postupak dekompozicije**
 - stvarni problemi su preveliki i presloženi da bi ih se riješilo odjednom (“u komadu”) → strukturno raščlanjivanje, rastavljanje
 - načelo podijeli pa s/vladaj (lat. divide et impera, eng. divide and conquer)
- ❑ **Sustav se razlaže i opisuje hijerarhijskim modelima**
 - modeli sustava oblikuju se iterativnim razlaganjem s vrha prema dolje
 - razlagati se mogu
 - funkcije i procesi
 - organizacijska struktura
 - struktura podataka
 - struktura programske opreme
- ❑ **Aplikacijski model procesa = logički model procesa sustava ili aplikacije koji se radi u fazi analize**



Logički procesi

❑ Funkcija

- skup logički povezanih trajnih poslovnih aktivnosti i zadataka (djelatnost, posao)
 - funkcija se obavlja stalno (nema određeni početak i kraj)
 - funkciju obavljaju osobe, grupe djelatnika ili organizacijske cjeline
- tipični primjeri: prodaja, proizvodnja, otprema, računovodstvo
- funkcija se može sastojati od desetina pa i stotina diskretnih procesa
 - funkcije se mogu hijerarhijski razložiti do razine diskretnih procesa koji obavljaju određeni zadatak kojim odgovaraju na poslovne događaje

❑ Događaj

- logički dio posla koji se obavlja kao nedjeljiva cjelina → česti naziv *transakcija*
 - pokreće se diskretnim ulazom i završava nakon što proces odgovori odgovarajućim izlazom
- poslovni događaj može se predstaviti jednim procesom kojim sustav reagira na taj događaj
 - logički događaj dalje se razlaže do elementarnih procesa kojima se prikazuje reakcija sustava na taj događaj

❑ Proces (elementarni, primitivni proces)

- postupak, način rada, dosljedna izmjena stanja
- diskretna odluka, aktivnost ili zadatak kojima se obavlja neki posao
 - proces se obavlja uvijek na jednak način (za određeni ulaz daje isti izlaz)
 - trajanje procesa je konačno i određivo (poznati početak, završetak, ponavljanje)
 - za obavljanje se koriste sredstva, npr. ljudska, materijalna (strojevi), financijska



Poslovna pravila i poslovna politika

❑ Poslovno pravilo, poslovna politika

- poslovno pravilo - instrukcije i logika koji određuju proceduru obavljanja procesa
 - ugrađuje se u računalni program (npr. preduvjeti izlaska na ispit, broj polaganja ispita, uvjeti upisa)
- poslovna politika – skup poslovnih pravila
 - u većini organizacija podloga za donošenje odluka

❑ Primjer: subvencioniranje studentske prehrane

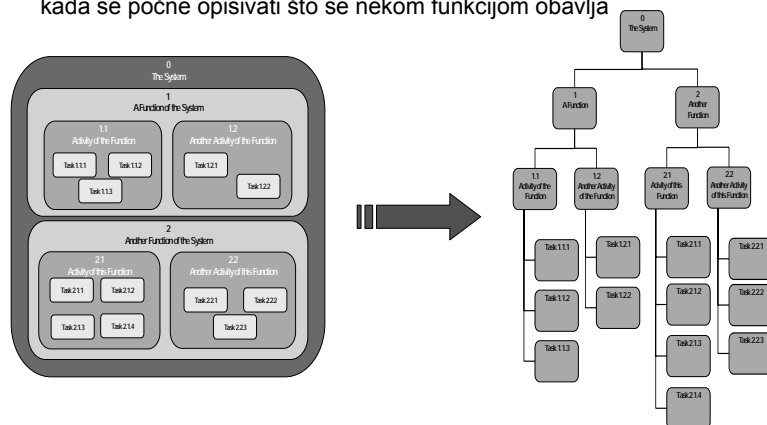
- Ovisno o evidentiranoj razini prava, studenti imaju dnevno dozvoljeni iznos cijene konzumirane hrane koji je subvencioniran. Taj iznos ne predstavlja dnevno ograničenje, već se svakog prvog u mjesecu dodjeljuju prava za čitav mjesec kojima student raspolaže po želji. Mjesečni iznos raspoloživ za subvenciju jednak je umnošku broja dana u mjesecu i dnevnog subvencioniranog iznosa za razinu prava. Npr. za razinu prava 1 (studenti domicilni) dnevno subvencionirani iznos je 16.70 Kn (cijena jednog menija), pa student te razine u mjesecu svibnju ima raspoloživo za subvenciju $16.70 * 31 = 517.7$ Kn. Iznos raspoloživ za subvenciju se akumulira tijekom čitave akademske godine što znači da se neiskorištena prava iz jednog mjeseca prebacuju na slijedeći. Na kraju akademske godine akumulirani iznos se briše.
- Za ostvarivanje bilo koje razine prava veće od 1, potrebno je ...
- Da bi student prvog u mjesecu stekao novi dozvoljeni subvencionirani iznos, moraju biti ispunjeni uvjeti: ...



Modeliranje funkcija

❑ Funkcionalna dekompozicija, dekompozicija funkcija

- koristi se za izradu općeg modela funkcija (modela poslovnih funkcija) promatranog sustava u fazi planiranja → strukturirano planiranje
- hijerarhija funkcija iterativno se razlaže do razine procesa, tj. do trenutka kada se počne opisivati što se nekom funkcijom obavlja



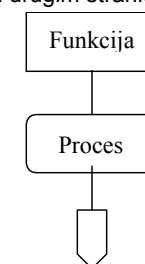
Dijagram dekompozicije funkcija

❑ Dijagram funkcionalne dekompozicije

- eng. Functional Decomposition Diagram (FDD)
- ista notacija koristi se za razlaganje bilo koje hijerarhijske strukture pa se često zove samo Dijagram dekompozicije ili Mapa hijerarhije

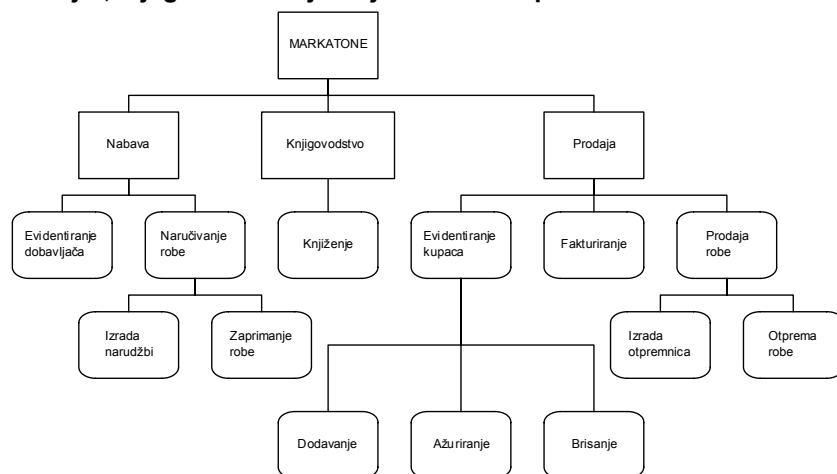
❑ Elementi

- funkcije - označavaju se (glagolskom) imenicom, npr. Prodaja, Proizvodnja
- procesi - označavaju se glagolskim izrazom oblika infinitiv+objekt
- spojnice - spojevi između funkcija i procesa (connector)
- vanjski spojevi - spojevi s dijelovima dijagrama na drugim stranicama (off-page connector)



Dijagram dekompozicije funkcija

Primjer, dijagram funkcija za jedan sustav/podsustav:



Primjeri: \Modeliranje\PutniNaloziBlagajna



Hijerarhijski prikaz funkcija/procesa

Izrada globalnog modela funkcija može započeti izradom hijerarhijske liste funkcija po pojedinim organizacijskim cjelinama.

Primjer:

- NABAVA
 - Evidentiranje dobavljača
 - Nabavka robe
 - Izrada narudžbi
 - Zaprimanje robe
- UPRAVLJANJE OSOBLJEM
 - Evidentiranje službe
 - Zaprimanje u službu
 - Praćenje službe
 - » redovan rad
 - » prekovremeni rad
 - » bolovanje
 - » godišnji odmori
 - Otpuštanje iz službe
 - Obračun plaća



Izrada dijagrama dekompozicije

Postupak

- Korijen = sustav
- Razrada u podsustave i poslovne funkcije
- Daljnja razrada do razine operacionalizacije

Pravila

- svaki proces je roditelj ili dijete
- roditelj mora imati barem dvoje djece
- po većini standarda, dijete smije imati samo jednog roditelja

Preporuke

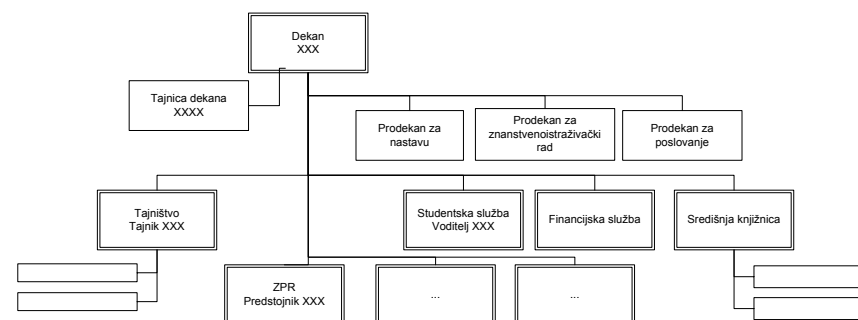
- izostaviti procese koji samo premještaju ili preusmjeravaju podatke, a da ih pri tom ostavljaju nedirnute
- pažnju usmjeriti na procese koji
 - nešto računaju (npr. prosjek ocjena)
 - donose ili potpomažu odluke (npr. određivanje raspoloživosti robe pri naručivanju)
 - filtriraju ili sumiriraju podatke (npr. računi kojima je istekao rok plaćanja)
 - organiziraju podatke u korisne informacije (npr. generiranje izvješća)
 - pokreću druge procese (npr. mijenjaju modalitet rada stroja)
 - rukuju podacima (npr. stvaranje, čitanje, ažuriranje, brisanje) - Ne pretjerivati!



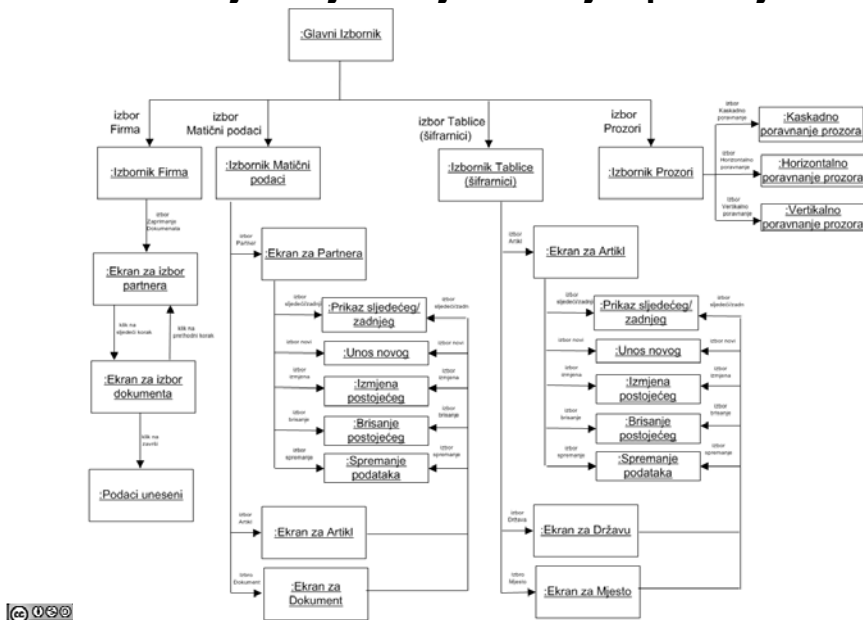
Dijagram organizacije

Schema, mapa, karta organizacije (Organization chart)

- prikaz strukture organizacije hijerarhijom pravokutnika ("kućica")
- svaki pravokutnik reprezentira određenu ulogu ili odgovornost u organizaciji



Primjer: hijerarhija funkcija aplikacije



241

Razrada poslovnih procesa

□ Tehnike modeliranja

- u širinu - svaki dijagram se detalizira prije dekomponiranja (breadth-first)
- u dubinu - identificira se hijerarhija, a zatim se detalizira (depth-first)

□ Razina dekompozicije (Kada stati)

- postupak se provodi do dubine dovoljne za razumijevanje modela (!?)
- napredak do stanja u kojem ulazi i izlazi prevladaju na dijagramu
- nastavak se može provesti
 - oblikovanjem toka rada (npr dijagramom aktivnosti) ili
 - » Napomena: pomagala opće namjene, kao što je Visio, podržavaju neke dijagrame formalnih tehnika ali ih različito zovu ili se notacija razlikuje od izvorne (grupe dijagrama Business Process i Flowchart, npr. IDEF0, Cross Functional FlowChart)
 - oblikovanjem toka podataka (Data Flow Diagram - DFD)

□ Tok rada (workflow) - radna procedura, poslovna procedura

- slijed koraka obrade koji u potpunosti obrađuje jednu poslovnu transakciju
- definira logiku obrade i precizira nositelja
- može imati više varijanti (scenarija)



FER \ Fertalj: Razvoj informacijskih sustava

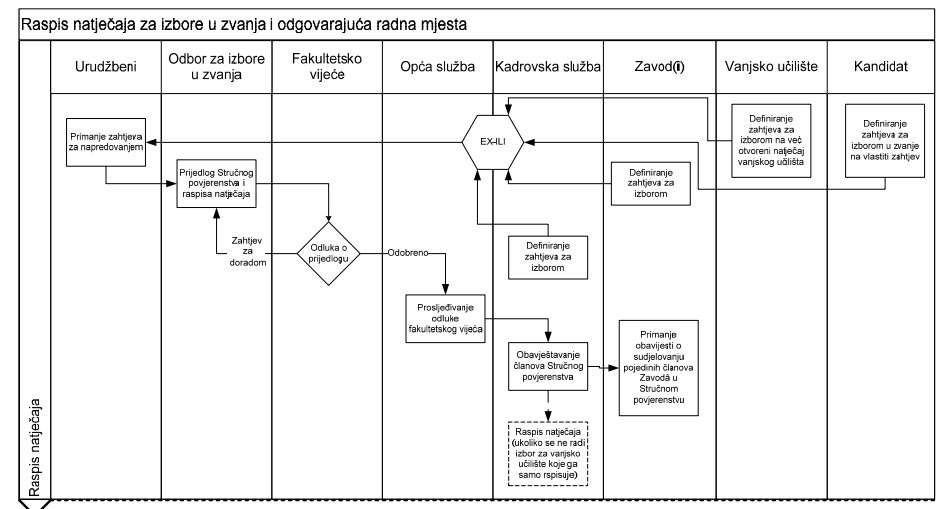
242

Oblikovanje toka rada

□ Primjer: Raspis natječaja za izbor u zvanje ili na radno mjesto

- U urudžbeni zapisnik se predaje zahtjev za raspisom natječaja u zvanje i/ili radno mjesto. Podnesak za izbor u znanstveno zvanje može biti na osobni zahtjev ili na zahtjev organizacije (zavoda). Kako je FER jedna od ustanova s ovlaštenjem za provođenje postupka biranja u zvanja, zahtjev potekao s FER-a provodi sam, ali isto tako provodi postupak i na zahtjev s drugih znanstvenih organizacija koje nisu ovlaštene za provedbu postupka.
- Zahtjev se proslijeđuje Odboru za izbore u zvanja koje predlaže članove Stručnog povjerenstva za provedbu postupka i definira javni natječaj
- Prijedlog razmatra Fakultetsko vijeće te ga ili šalje nazad na doradu Odboru za izbore u zvanja ili samo definira nove članove Stručnog povjerenstva
- Opća služba proslijeđuje odluku Fakultetskog vijeća Kadrovskoj službi
- Kadrovska služba raspisuje natječaj i obavještava članove Stručnog povjerenstva

Primjer: Raspis natječaja



243



FER \ Fertalj: Razvoj informacijskih sustava

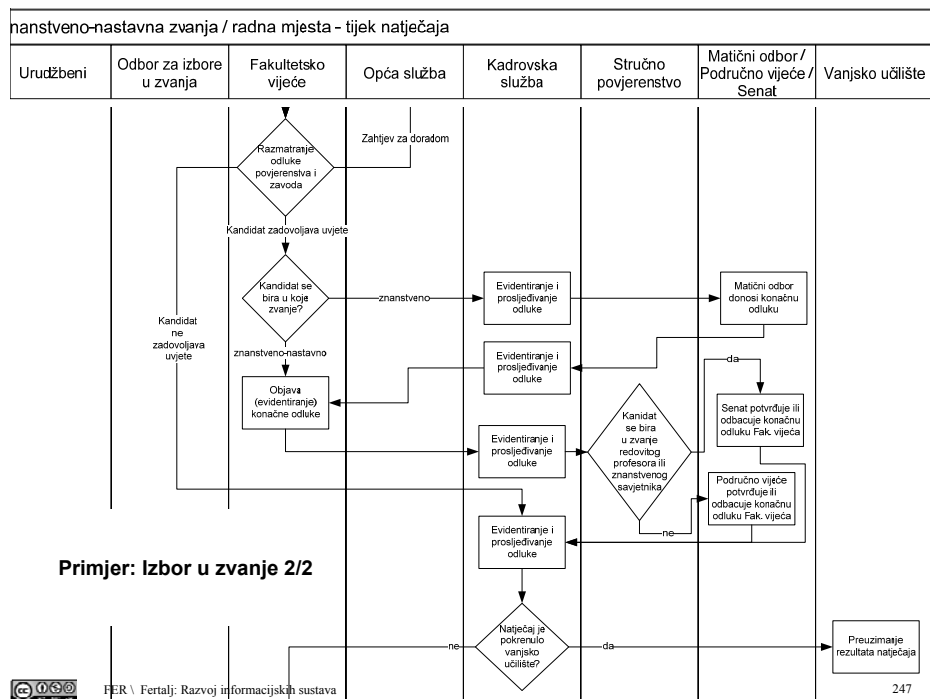
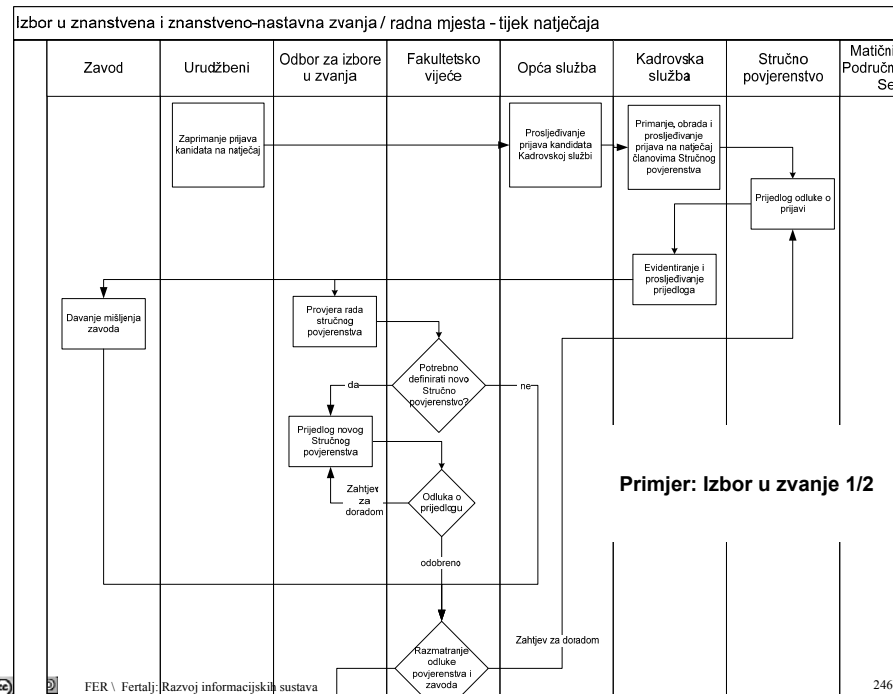


FER \ Fertalj: Razvoj informacijskih sustava

244

Primjer: postupak izbora u zvanje

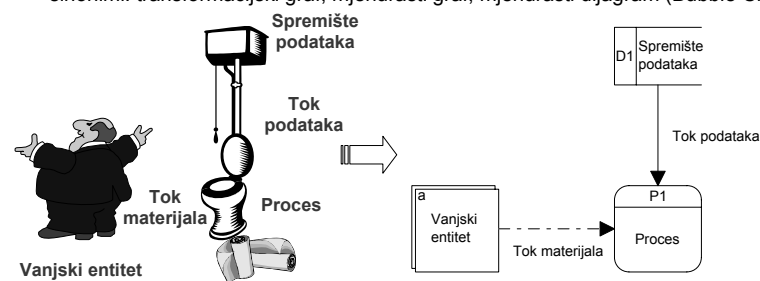
- ❑ Izbor u znanstveno zvanje može se provesti neovisno o radnom mjestu na zahtjev pojedinca ili znanstvene organizacije u kojoj pojedinac radi.
- ❑ Na visokim učilištima izbor u zvanje odvija usporedo s izborom na radno mjesto ukoliko je to potrebno.
 - Zakonom su predviđena sljedeća znanstvena zvanja:
 - znanstveni suradnik
 - viši znanstveni suradnik
 - znanstveni savjetnik.
 - Pristupnik može biti izabran u znanstveno-nastavno zvanje ako ispunjava uvjete za izbor u odgovarajuće znanstveno zvanje:
 - docenta: znanstveni suradnik, viši znanstveni suradnik ili znanstveni savjetnik,
 - izvanrednog profesora: viši znanstveni suradnik ili znanstveni savjetnik,
 - redovitog profesora: znanstveni savjetnik.
- ❑ Postupak je opisan narednim dijagramima (slika vrijedi 1024 riječi)
 - postoje varijante izbora u nastavna zvanja ili suradnička mjesta koje ovdje nisu prikazane



Modeliranje toka podataka

❑ Dijagram toka podataka (DFD - Data Flow Diagram)

- skup dijagrama za dokumentiranje fizičkog i logičkog modela sustava te zahtjeva
 - prikaz protoka, strukture i obrade podataka
 - dokumentiranje logike, poslovnih pravila i procedura
- sinonimi: transformacijski graf, mjehurasti graf, mjehurasti dijagram (Bubble Chart)



❑ Tehnika se primjenjuje pri razvoju aplikacija, otkuda je i potekla

- Ne može se koristiti za opis programske logike, opis promjene stanja, izradu upravljačkih specifikacija ili dizajn korisničkog sučelja!!!



Elementi dijagrama toka podataka

□ Tok podataka (data flow)

- predstavlja skupove podataka koji se kreću kroz sustav
- tokovi ulaze u procese (ulazni), koriste se i mijenjaju tijekom obavljanja procesa (ulazno/izlazni) ili nastaju kao rezultat procesa (izlazni)
- tokovima se pridjeljuju jedinstveni nazivi oblika imenica ili pridjev+imenica, npr. Potvrđena prijavnica, Izlazni račun

□ Proces

- predstavlja aktivnost pretvorbe podataka (ulaznog u izlazni tok podataka)
- procesi se imenuju glagolskim izrazima oblika infinitiv+objekt (npr. Prijaviti ispit) ili glagolskom imenicom (npr. Prodaja, Prijava ispita)
- nazivom treba izraziti što proces obavlja, to jest treba izbjegavati općenite nazive (npr. Obavljanje računovodstvenih poslova)
- opis procesa sadrži opis aktivnosti (algoritam) njegovog djelovanja

□ Spremište podataka (data store)

- predstavlja organizirani i trajni skup podataka
- označava mjesto pohrane podataka, npr. dokument, registrator, datoteka, tablica u bazi podataka (izbjegavati u nazivlju)
- promjena sadržaja spremišta (punjenje, ažuriranje, pražnjenje) i korištenje (čitanje) obavlja se procesima
- spremište se označava imenicom (imenicom u množini), npr. Prijavnica (Prijavnice)

□ Vanjski entitet (external entity, external agent)

- objekt vanjskog svijeta povezan s promatranim sustavom
- određuje granice promatranog sustava
- vanjski entiteti predstavljaju izvorišta i odredišta podataka, to jest izvore i ponore podataka (source, sink)
- vanjski entiteti mogu biti osobe, org. jedinice, ustanove, drugi sustavi ...
- za označavanje entiteta koriste se imenice, npr. Student, Kupac, Dobavljač



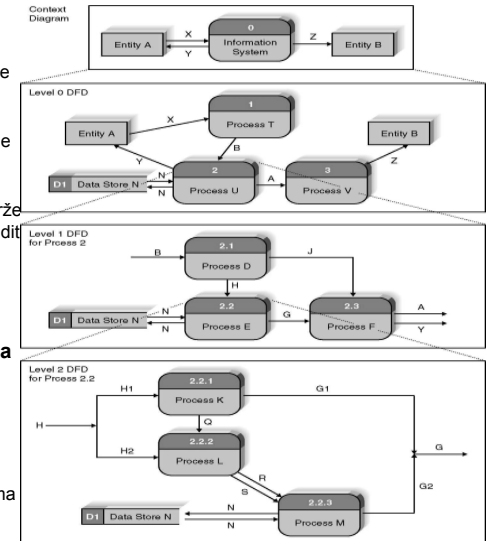
Izrada dijagrama toka podataka

□ Dekompozicija procesa

- polazni dijagram ili dijagram konteksta (context diagram) hijerarhijski se razlaže na poddijagrame do razine osnovnih procesa
- proces na nekoj razini (parent) razrađuje se (explode) dijagramom na nižoj razini (child) – leveling = nivelizacija
- preporuča se izrada dijagrama koji sadrže između 2 i 9 procesa, a poželjno je slijediti "pravilo 7±2"
- postupak se zaustavlja kada postane očigledna ugradnja (implementacija) procesa na najnižoj razini

□ Preporuke za označavanje elemenata

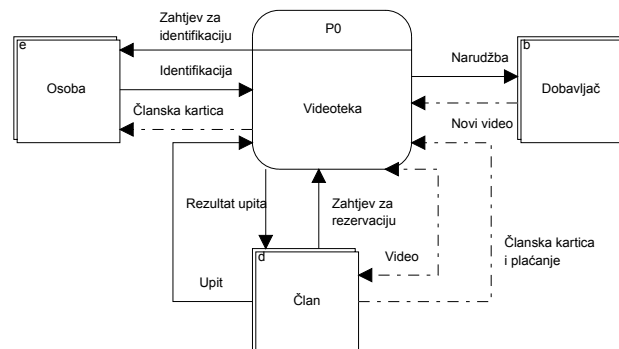
- procesi - hijerarhijske brojčane oznake, razina konteksta = 0
- spremišta, izvori i odredišta – nazivlje velikim slovima, oznake oblika slovo ili slovo+broj
- procesi i tokovi podataka - malim slovima



Izrada dijagrama toka podataka

□ Dijagram konteksta

- prikazuje sustav na najvišoj razini hijerarhije prikaza (top level diagram)
- definira okruženje sustava i područje analize (environmental model)
- prikazuje jedan proces i vanjske entitete
 - započeti s procesom koji prikazuje sustav u cjelini
 - odrediti vanjske entitete i njihovu povezanost sa sustavom



Izrada dijagrama toka podataka

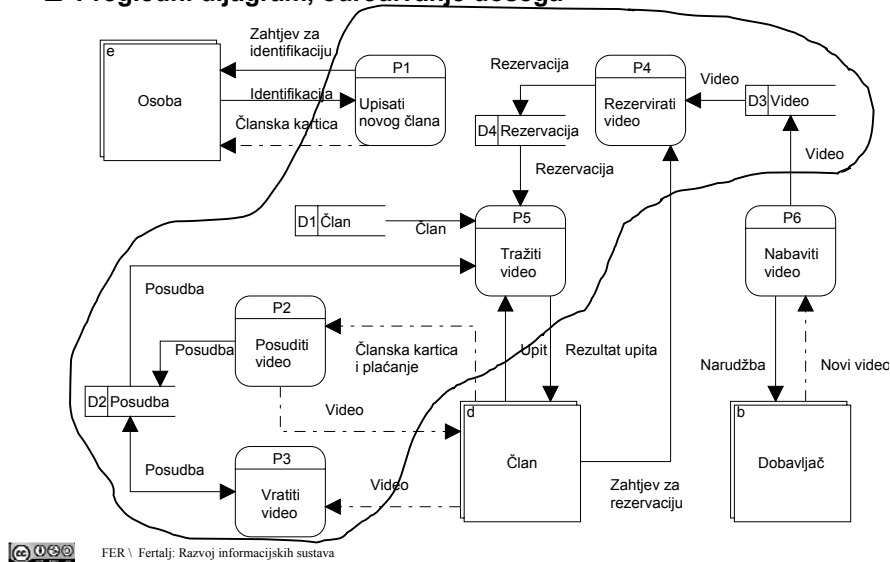
□ Pregledni dijagram (initial diagram)

- uočiti glavne tokove informacija (npr. korišteni dokumenti, potrebni podaci)
- odrediti glavne aktivnosti sustava i prikazati ih odgovarajućim procesima
- uključiti vanjske entitete i tokove podataka s dijagrama konteksta
- složiti se s korisnikom oko granica sustava
- utvrditi procese i spremišta podataka



Izrada dijagrama toka podataka

Pregledni dijagram, određivanje dosega



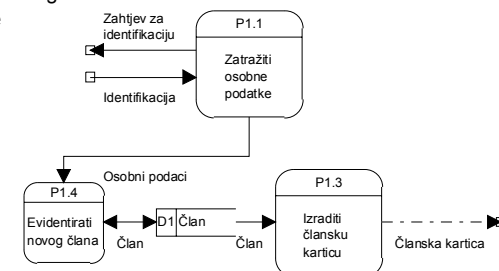
FER \ Fertalj: Razvoj informacijskih sustava

253

Izrada dijagrama toka podataka

Razrada

- za svaki proces s preglednog dijagrama identificirati podaktivnosti
- na primjer, za proces Upisati novog člana:
 - Zatražiti osobne podatke
 - Evidentirati novog člana
 - Izraditi člansku karticu



Ponavljati postupak za svaki od procesa na poddijagramu

- uspostaviti razinu detalja slijedeći "pravilo 7±2"
- provjeriti potpunost i ispravnost modela

Model obrazložiti korisniku a zatim ga ažurirati po potrebi

- Dubinu i uravnoteženost modela teško je odrediti.
- U praksi to može značiti doradu dijagrama u većem broju ponavljanja, čak i kada dijagrame rade iskusni analitičari!!!



FER \ Fertalj: Razvoj informacijskih sustava

254

Pravila i ograničenja prilikom izrade DTP

Pravilo bilance (očuvanja) tokova (level balance rule)

- količina tokova koji ulaze u proces i izlaze iz procesa mora odgovarati količini tokova podprocesa na nižoj razini hijerarhije
- nije dozvoljeno variranje tokova neke razine na nižim razinama (npr. tok T na nižim razinama prikazivati kao T1, T2)

Ograničenja i posebni slučajevi

- Svi objekti modela moraju biti povezani. Nepovezanost pojedinih objekata ukazuje na nepotpunost modela, na primjer:
 - postojanje procesa bez ulaza i/ili izlaza (tzv. čuda i crne rupe)
 - izlaze za koje ne postoji dovoljno ulaza (tzv. sive rupe – najčešće)
 - postojanje nepovezanih spremišta ili vanjskih entiteta
- Ne dozvoljava se neposredna povezanost:
 - vanjskih entiteta
 - spremišta
 - spremišta i vanjskog entiteta
- Nije dozvoljeno:
 - grananje toka u različite tokove, spajanje različitih tokova
 - postojanje "rekurzivnih" procesa



FER \ Fertalj: Razvoj informacijskih sustava

255

Preporuke za izradu DTP

Treba pripaziti na:

- trivijalne tokove – izlazi iz procesa koji ne ulaze u spremišta ili odredišta
 - obično imaju posebno značenje, a mogu se koristiti za prikaz posebnih stanja kao što je dojava poruke sustava (npr. dojava pogreške)
- neposredno povezane procese
 - ako postoje, to znači da jedan od procesa čeka na završetak prethodnog
- procese koji ne obavljaju pretvorbu podataka
 - ako je izlazni tok jednak ulaznom
 - treba preimenovati jedan od tokova ili
 - treba obaviti prespajanje tokova



Procesi se mogu zbivati istovremeno → DTP se ne smije tumačiti kao dijagram toka (flowchart)!



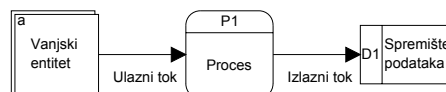
FER \ Fertalj: Razvoj informacijskih sustava

256

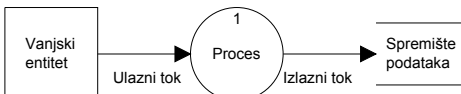
Metode koje koriste DTP

Notacije

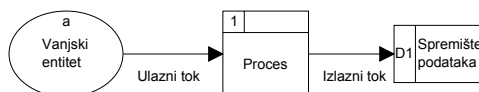
- Gane/Sarson (korištena u primjerima)



- Yourdon/DeMarco



- SSADM



Proširenja modela

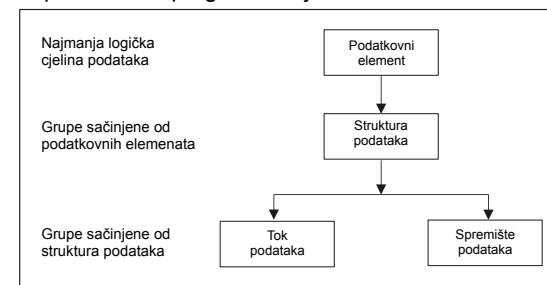
- okidač (trigger) - prikaz učestalosti procesa (npr. tri puta dnevno)
- posebni simboli za prikaz ponavljanja procesa
- razdvajanje i spajanje tokova (alternativni tokovi)
- posebni simboli za tok resursa, tok dokumenata ili tok upravljanja (npr. različite linije ili ikone)



Opisivanje podataka

Rječnik podataka (Data Dictionary)

- mjesto pohrane definicija podatkovnih elemenata i struktura podataka
- strukturirano spremište meta-podataka, to jest podataka o podacima
 - prvotno se pojavio kao proširenje dijagrama toka podataka, za pohranu opisa spremišta podataka i tokova podataka
 - može se koristiti kao alternativna tehnika za prikaz modela podataka
- standardno se upotrebljava BNF notacija (Backus-Naur Form), koja se inače koristi za opis sintakse programskih jezika



Definiranje podataka BNF notacijom

Notacija

=	struktura s lijeva sastoji se od dijelova s desna ("sastavljeno od")
+	agregacija elemenata (logičko I)
()	opcionalnost elemenata u zagradi (0 ili 1)
{ }	ponavljanje (iteracija) elemenata u zagradi, ni jednom do konačni broj puta
[]	alternativa (selekcija) elemenata u zagradi
	odvajanje alternativnih elemenata u [] izrazu
-	početna i završna vrijednost raspona definiranog [] izrazom
**	komentar
@	oznaka ključa

Primjer, račun i stavke računa

```
Racun = @BrRac + DatRac + BrKupca
        + { SifArt, NazArt, Cijena, Kol, Vrijednost }
        + (IznosRac)
```

Može se napisati i kao:

```
Racun = @BrRac + DatRac + BrKupca
        + { StavkaRac }
        + (IznosRac)
StavkaRac = @SifArt, NazArt, Cijena, Kol, Vrijednost
```



Elementarni procesi

Mini-specifikacije (funkcionalne primitive)

- specifikacije za opisivanje osnovnih procesa u dijagramu toka podataka
- mogu poprimiti različite oblike, ali imaju nekoliko zajedničkih elemenata:
 - naziv i broj procesa
 - listu podataka koji ulaze u proces (ulaznih tokova podataka)
 - listu podataka koji iz procesa izlaze (izlaznih tokova podataka)
 - tijelo opisa procesa

Opis procesa

- sadrži algoritam zadatka koji se procesom obavlja, koji može poprimiti različite oblike (→ dizajn programa - opis programske logike)
- na temelju ovog algoritma može se, ovisno o alatu, generirati kôd



Definiranje procesa

□ Primjer definicije procesa (1)

Proces 1:	Provjera raspoloživosti filma
Opis:	Provjera da li u videoteci postoji kopija filma koja se može iznajmiti
Ulazni tokovi:	Upit (Film) Rezervacije Posudbe
Izlazni tokovi:	Rezultat upita (raspoloživ nije raspoloživ ne postoji)
Logika procesa:	<p><u>izračunaj</u> broj kopija traženog filma u videoteci <u>ako</u> je broj kopija veći od nule <u>tada</u> <u>ako</u> je broj kopija veći od (broj posudbi + broj rezervacija) rezultat = "Raspoloživ" <u>inače</u> rezultat = "Nije raspoloživ" <u>inače</u> rezultat = "Ne postoji"</p>



Definiranje procesa

□ Primjer definicije procesa (2)

Proces 1:	Provjera raspoloživosti filma
Opis:	Provjera da li u videoteci postoji kopija filma koja se može iznajmiti
Ulazni tokovi:	Upit (Film), Rezervacije, Posudbe
Izlazni tokovi:	Film nije raspoloživ, Film je raspoloživ
Logika procesa:	<p><u>izračunaj</u> broj kopija traženog filma u videoteci <u>ako</u> je broj kopija veći od nule <u>tada</u> <u>ako</u> je broj kopija veći od (broj posudbi + broj rezervacija) <u>nastavi</u> s P2 (izl. tok Film je raspoloživ) <u>inače</u> <u>ako</u> član želi rezervirati film <u>tada</u> <u>upiši</u> rezervaciju (izl. tok Film nije raspoloživ) <u>inače</u> <u>poruka</u> "Ne postoji"</p>



Primjena modela procesa

□ Strateško planiranje sustava

- definiranje arhitekture sustava u okviru strateškog plana, pri čemu se radi model procesa poduzeća (globalni model procesa)
 - identificira poslovna područja i poslovne funkcije
 - najčešće u formi dijagrama dekompozicije funkcija ili nerazrađenog DTP (npr. dijagramom konteksta određuju se doseg i sučelja sustava)

□ Preoblikovanje poslovnih procesa

- analiza i restrukturiranje poslovnih procesa radi poboljšanja učinkovitosti i uklanjanja birokratizma, prije primjene informacijskih tehnologija
- postojeći procesi se analiziraju i dokumentiraju prikladnim modelima procesa
 - dijagrami toka podataka s fizikalnim primjesama, koje uključuju vremensku dimenziju, protočnost podataka i troškove

□ Analiza sustava

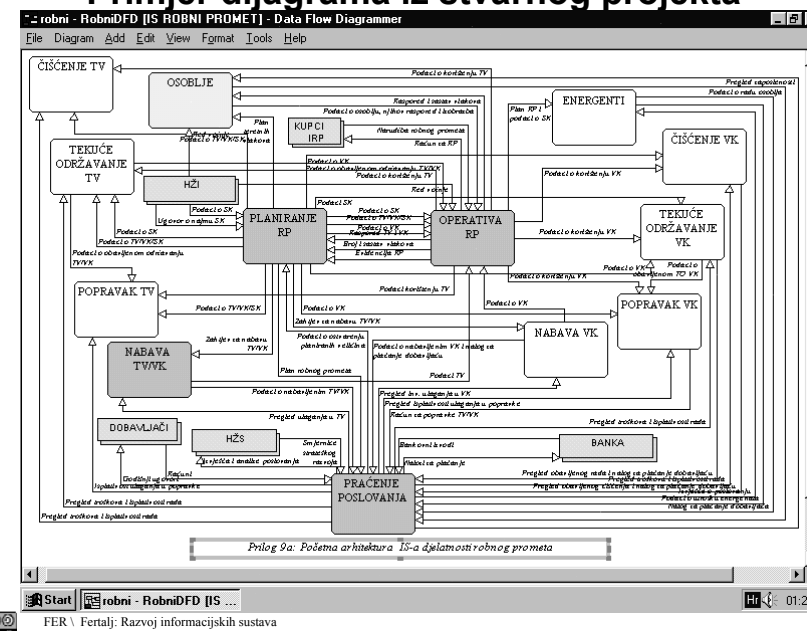
- aplikacijski modeli procesa - logički modeli procesa sustava ili aplikacije
 - teorijski, moguće je proizvesti DTP povratnim inženjerstvom postojećih aplikacija, ali će takav dijagram biti preopterećen fizikalnim primjesama

□ Dizajn sustava

- fizički modeli procesa - dodavanjem upravljačkih komponenti i resursa

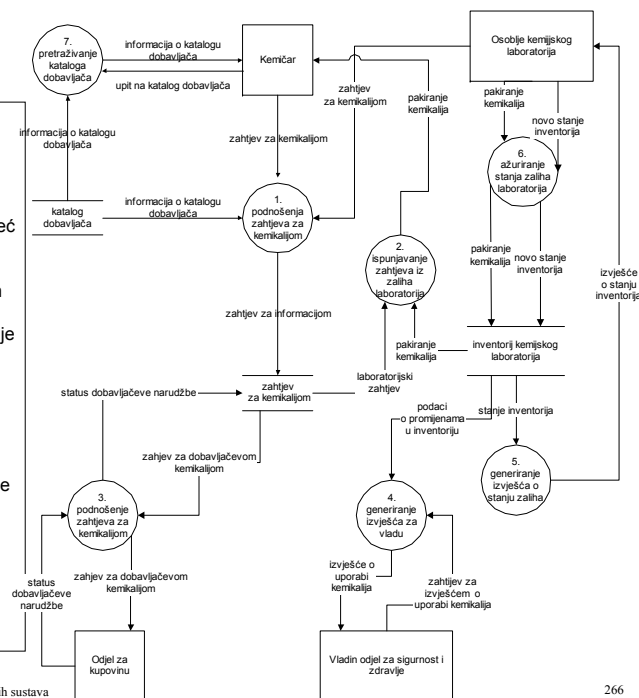


Primjer dijagrama iz stvarnog projekta



Primjeri DTP

"Kemičar ili član osoblja kemijskog laboratorija može podnijeti zahtjev za jednom ili više kemikalija. Zahtjev može biti udovoljen ili dostavom pakiranja kemikalije koja se već nalazila na zalihni kemijskog laboratorija ili upućivanjem narudžbe za novim pakiranjem kemikalije od vanjskog dobavljača. Osoba koja upućuje zahtjev mora imati mogućnost pretraživanja kataloga kemikalija vanjskog dobavljača dok sastavlja narudžbu. Sustav mora pratiti status svakog zahtjeva za kemikalijama od trenutka kad je ispunjen do trenutka kad je udovoljen ili otkazan. Također, mora pratiti povijest svakog pakiranja kemikalija od trena kad stigne u kompaniju do trenutka kad je potpuno upotrijebljen ili odbačen." ...



- ☐ ISSP se može podijeliti na slijedeće funkcionalne cjeline:
- ☐ Unutar svake od cjelina obavlja se više specifičnih procesa koji ...

-
- The diagram illustrates the information system for 'Restoran'. It features several interconnected components:
- Restoran (P3):** The central entity, which interacts with various external systems and internal data.
 - Polovitielj subvencije (b):** A database or system for half-price subsidies, connected to the 'Restoran' via a 'Pregled potrošnje restorana' (Restaurant consumption review) interface.
 - SC i ostali pružatelji usluge (d):** A database or system for service providers and other service providers, also connected to the 'Restoran' via a 'Pregled potrošnje restorana' interface.
 - P2 (CAP):** A central processing unit or database that manages data and interacts with other components.
 - P4 (Tvrka za tiskanje):** A printing unit that receives data from the 'Restoran' and outputs 'Zatrazena kartica' (Requested card) and 'Nova kartica' (New card).
 - P1 (Ustanova):** An institution or organization that interacts with the 'Restoran' and the 'Ustanova' (Ustanova) component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' (Request for card issuance) and 'Zahitjev za promjenu razine prav' (Request for change of rights level) to the 'Ustanova'.
 - Ustanova (Ustanova):** A component that manages the 'Ustanova' and interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Vlasnik (Owner):** A role that interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Ustanova (Ustanova):** A component that manages the 'Ustanova' and interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Restoran (Restoran):** The central entity, which interacts with various external systems and internal data.
 - Polovitielj subvencije (b):** A database or system for half-price subsidies, connected to the 'Restoran' via a 'Pregled potrošnje restorana' (Restaurant consumption review) interface.
 - SC i ostali pružatelji usluge (d):** A database or system for service providers and other service providers, also connected to the 'Restoran' via a 'Pregled potrošnje restorana' interface.
 - P2 (CAP):** A central processing unit or database that manages data and interacts with other components.
 - P4 (Tvrka za tiskanje):** A printing unit that receives data from the 'Restoran' and outputs 'Zatrazena kartica' (Requested card) and 'Nova kartica' (New card).
 - P1 (Ustanova):** An institution or organization that interacts with the 'Restoran' and the 'Ustanova' (Ustanova) component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Ustanova (Ustanova):** A component that manages the 'Ustanova' and interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Vlasnik (Owner):** A role that interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.
 - Ustanova (Ustanova):** A component that manages the 'Ustanova' and interacts with the 'Restoran' and the 'Ustanova' component. It receives 'Zatrazena kartica' and 'Nova kartica' from the 'Ustanova' and sends 'Zahitjev za izdavanje kartice' and 'Zahitjev za promjenu razine prav' to the 'Ustanova'.

☐ **Evidencija matičnih podataka članova**

- [illegible]

Modeliranje događaja

Događaji

□ Događaj

- zgoda, zbivanje u sustavu koja vodi ili pokreće procese sustava
- sâm događaj nije proces, nego okidač procesa koji se njime pokreće
- primjer:
 - kupac dostavi narudžbu, pokreće proces provjere da li se radi o narudžbi postojećeg ili novog kupca, proces stvaranja podataka o narudžbi i stavkama narudžbe, provjeru prethodnih zaduženja kupca, provjeru stanja skladišta itd.



Vrste događaja

□ Vrste događaja

- vanjski događaji – potaknuti od strane vanjskih entiteta, koji zahtijevaju informaciju ili ažuriranje podataka → ulazni tokovi podataka
 - imenuju se tako da naziv sadrži naziv vanjskog entiteta
 - primjer: zahtjev za upis studenta ili zaprimanje narudžbe kupca
- vremenski događaji - vremenski uvjetovani (rok, učestalost) → ulazni upravljački tokovi
 - imenuju se tako da naziv sadrži vremensku oznaku
 - primjer: istek roka plaćanja računa, mjesečni obračun plaća, zaključivanje ispitnog roka
- unutarnji događaji, događaji stanja - posljedica prijelaza sustava iz jednog stanja u drugo, na takav način ta to zahtjeva obradu → ulazni upravljački tokovi
 - primjer: isporuka robe sa skladišta zahtijeva naručivanje nove robe

Modeliranje procesa vođeno događajima

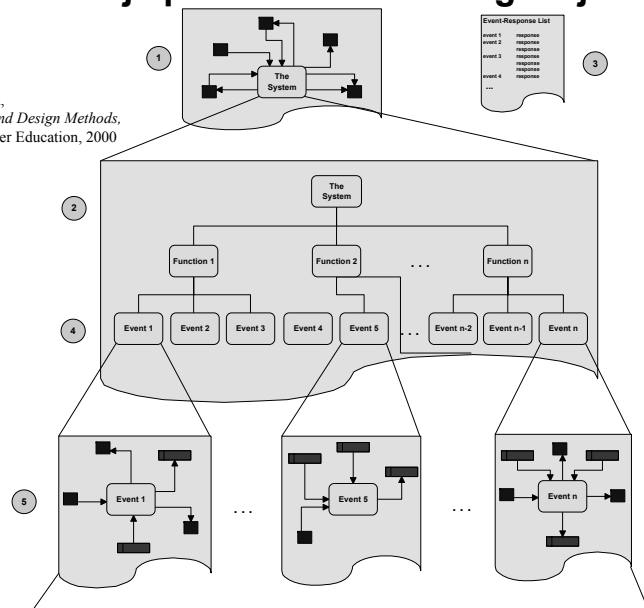
□ Raspodjela događaja (event partitioning)

1. Izrada dijagrama konteksta sustava
 - postavljanje početnog dosega projekta
2. Izrada dijagrama funkcionalne dekompozicije
 - podjela sustava u logičke podsustave i/ili funkcije
3. Izrada popisa događaja i odziva
 - utvrđivanje poslovnih događaja na koje sustav mora odgovoriti
 - element popisa = događaj, ulaz, izlaz
4. Izrada dijagrama dekompozicije događaja
 - dodavanje procesa za rukovanje događajima (event handler)
 - dodaje se po jedan proces za svaki utvrđeni događaj
5. Izrada dijagrama događaja
 - razrada procesa za obradu događaja
 - po jedan dijagram za svaki događaj
6. Izrada dijagrama sustava
 - udruživanjem dijagrama događaja
7. Izrada primitivnih dijagrama
 - razrada dijagramom toka podataka koji sadrži osnovne procese, spremišta i tokove za svaki pojedini događaj



Modeliranje procesa vođeno događajima

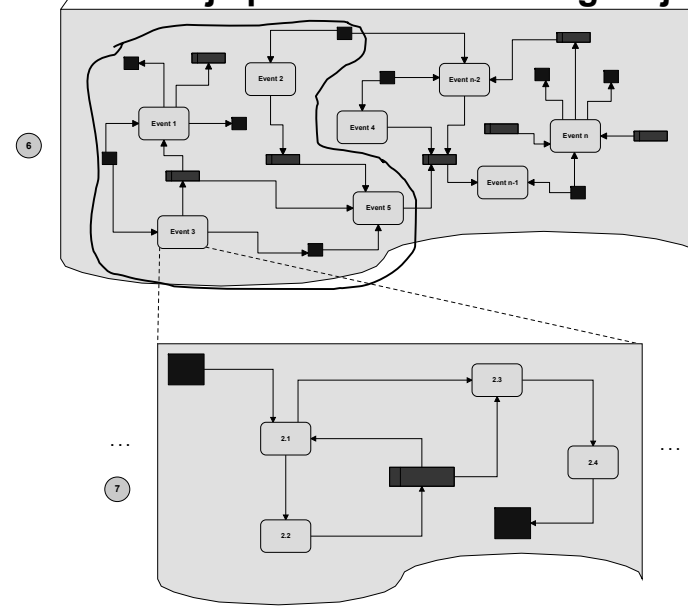
Slika: Whitten et al,
Systems Analysis and Design Methods,
McGraw-Hill Higher Education, 2000



FER \ Fertalj: Razvoj informacijskih sustava

273

Modeliranje procesa vođeno događajima



FER \ Fertalj: Razvoj informacijskih sustava

274

Tablica događaja u sustavu

- ❑ **Događaj**
 - Događaj zbog kojeg sustav obavlja određene aktivnosti
- ❑ **Okidač**
 - Kako sustav zna da se dogodio događaj?
 - Za vanjske događaje, ovo su podaci koji ulaze u sustav.
 - Za privremene događaje, određivanje točke u vremenu koja pokreće rad sustava.
- ❑ **Izvor**
 - Za vanjski događaj, vanjski uzročnik, tj. izvor podataka koji ulaze u sustav
- ❑ **Aktivnosti**
 - Što sustav radi u slučaju događaja?
- ❑ **Odgovor**
 - Koji/kakav izlaz (ako postoji) izlazi iz sustava
- ❑ **Odredište**
 - Kamo ide odgovor?
- ❑ **Primjeri:** \Modeliranje\StrukturiranaAnalizaDizajn



FER \ Fertalj: Razvoj informacijskih sustava

275

Primjer: tablica događaja i procesa

Događaj	Okidač	Izvor	Aktivnosti	Odgovor	Odredište
1. Kupac želi provjeriti raspoloživost artikla	Upit o proizvodu	Kupac	Provjeriti raspoloživost	Podaci o raspoloživosti	Kupac
2. Narudžba	Nova narudžba	Kupac	Stvoriti novu narudžbu	Real-time veza Potvrda narudžbe Podaci o narudžbi Transakcija	Kreditni ured Kupac Otpremništvo Banka
...					
4. Vrijeme za zatražiti izvještaje o poslovanju	Kraj tjedna, mjeseca, godine		Napraviti izvještaje	Izvještaji	Uprava
5. Vrijeme za izraditi izvještaje o transakcijama	Kraj dana		Napraviti izvještaje o transakcijama	Izvještaji o transakcijama	Računovodstvo
6. Kupac, ili uprava, želi provjeriti stanje narudžbe	Upit o stanju narudžbe	Uprava ili Kupac	Ispitati stanje narudžbe	Podaci o narudžbi	Uprava ili Kupac



FER \ Fertalj: Razvoj informacijskih sustava

276

Matrica Entiteti/Događaji

❑ Matrica Entiteti/Događaji

- Entity/Event Matrix, Event/Enquiry, Entity Access Matrix
- pogled na sustav usmjeren događajima (event oriented view)
- matrica sadrži:
 - događaje (retci)
 - entitete (stupci)
 - elemente koji prikazuju učinak događaja na entitete
 - Stvaranje: C (create)
 - Čitanje: R (read) - u nekim metodama se ne bilježi
 - Ažuriranje: U (update) ili M (modify)
 - Brisanje: D (delete)
- provjera dovršenosti:
 - svaki događaj mora imati učinak na barem jedan entitet
 - svaki entitet mora imati događaj koji ga čita, stvara, mijenja ili briše

❑ Primjer: pojednostavnjena rezervacija sobe u hotelu Proljeće

Događaj/Entitet	korisnik	rezervacija sobe	zaduženje	soba	...
privremena rezervacija	C/M	C			
potvrda rezervacije		C/M			
opoziv rezervacije		M			
dolazak gosta	C/M	C/M		M	
povećanje troškova		M	C		
plaćanje usluga		M	M	M	
arhiviranje rezervacije		D	D		



Matrični prikaz modela događaja

❑ U primjeni su različite vrste prikaza, ovisno o metodologiji

- matrica entiteti/događaji (Entity/Event Matrix), npr. SSADM
 - odnos događaja i entiteta: C (create), M (modify), D (delete)
- matrica funkcije/entiteti (Function/Entity Matrix), npr. IEM
 - odnos obrade i podataka: C (create), R (read), U (update), D (delete)
- matrica funkcije/događaji (Function/Event Matrix), npr. SSM
 - odnos obrade i događaja: X (povezanost) ili praznina

❑ Slične matrica može se iskoristiti za definiranje prava

- matrica Korisnik / Proces / X
- matrica Korisnik / Entitet / CRUD
 - Korisnik pri tome predstavlja grupu korisnika ili ulogu korisnika
 - Ovisno o detaljima ugradnje u fizičkom modelu Korisnik može biti i konkretna osoba



Primjer: matrica entiteti/događaji

❑ Primjeri: \Modeliranje\StrukturiranaAnalizaDizajn

	Katalog	Mušterija	Raspoloživ artikl	Narudžba	Naručeni artikl	Transakcija (narudžba)
Provjeriti raspoloživost artikla			R			
Stvoriti novu narudžbu		CRU	RU	C	C	C
Ažurirati narudžbu		RU	RU	RUD	RUD	RUD
Provjeriti status narudžbe		RU		R	R	RUD
Zabilježiti ispunjenje narudžbe					RU	
Zabilježiti istek zaliha					RU	
Stvoriti povrat narudžbe		CRU		RU		C
Dati informacije o katalogu	R		R			
Ažurirati korisnički račun		CRUD				
Distribuirati promo pakete	R	R	R			
Prilagodba naplaćene cijene		RU				CRUD
Ažuriranje kataloga	RU		R			
Kreiranje posebne promocije	R		R			
Kreiranje novog kataloga	C		R			



Određivanje podsustava matricom događaja

❑ Primjer, prvi dio matrice procesi/entiteti za jedan stvarni sustav

	P3	K54	K55	K30	K39	K5	K31	K40	K7	K32	K41	K8	K42	K14	K63	K50	K37	K44	K53	K12	K26	K45	K49	K13	K6	K29	K52	K15	K22
P9	CRUD	CRUD	CRUD	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
P6	CRUD	R	CRUD																										
P10	CRUD	R	R	CRUD	CRUD	CRUD																							
P11	CRUD	R	R	R	CRUD	CRUD	CRUD																						
P12	CRUD	R	R	R	R	CRUD	CRUD	CRUD																					
P8	CRUD	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
P32																													
P44																													
P37																													
P32																													
P29																													
P1																													
P4																													
P3																													
P36																													
P40																													
P45																													
P18																													
P16																													
P7																													
P24																													
P30																													
P35																													
P29																													
P28																													
P23																													
P19																													
P14																													
P20																													
P30																													
P21																													
P47																													
P40																													
P31																													
P45																													
P40																													
P13																													
P26																													
P17																													
P27																													
P21																													
P2																													
P16																													
P36																													
P34																													
P41																													
P42																													
P39																													
P40																													
P43																													
P29																													



Dijagram prijelaza stanja

□ Dijagram prijelaza stanja (State Transition Diagram)

- zasniva se na ideji stroja s konačnim brojem stanja (finite state machine), hipotetičkom mehanizmu koji u nekom trenutku može biti u jednom od konačno mnogo diskretnih stanja
- grafički prikaz promjena stanja, tj vremenski zavisnog ponašanja sustava

□ Elementi prikaza:

- stanje – kumulativni rezultat ponašanja nekog objekta (pravokutnik, krug ili elipsa)
- prijelaz – promjena stanja uzrokovana nekim događajem (usmjerena linija od jednog stanja prema drugom)
- događaj – uvjet promjene stanja i akcija koja se obavlja (opis linije prijelaza oblika događaj/akcija)

□ DPS najčešće opisuje vremenski zavisno ponašanje čitavog sustava

- manji sustavi mogu se prikazati jednim dijagramom
- veći sustavi razlažu se slično dijagramima toka podataka, pri čemu stanje na nekoj razini postaje početno stanje dijagrama na nižoj razini

□ Primjena

- sustavi za rad u stvarnom vremenu (real-time system)
- jezična analiza (parsing)
- dizajn korisničkog sučelja

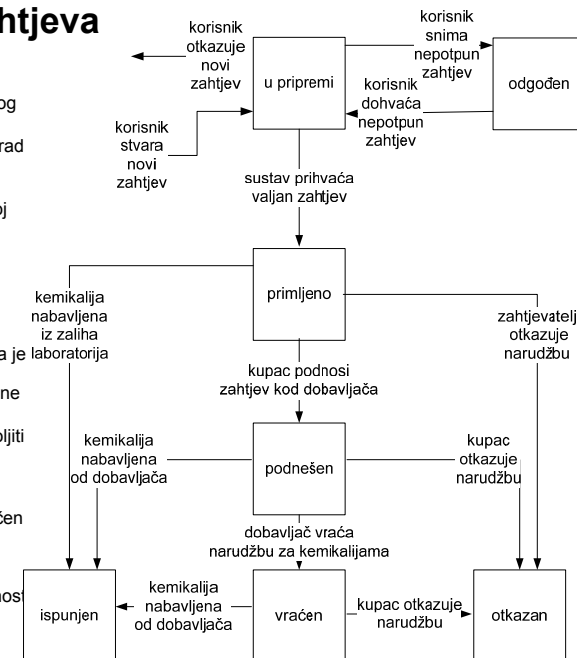


FER \ Fertalj: Razvoj informacijskih sustava

281

Primjer: Stanja zahtjeva

- U pripremi: Zahtjevatelj stvara novi zahtjev, pokrenuvši tu funkciju iz nekog drugog dijela sustava
- Odgođen: Zahtjevatelj je snimio svoj rad (narudžbu) kako bi ju dovršio kasnije bez da ju je predao ili otkazao
- Zaprmljen: Zahtjevatelj je predao svoj zahtjev (narudžbu) koji je sustav prihvatio kao valjan
- Podnesen: vanjski dobavljač je sposoban isporučiti zahtjev i kupac podnosi zahtjev (narudžbu) kod vanjskog dobavljača
- Ispunjen: zahtjev je zadovoljen bilo da je dostavljen iz zaliha kemijskog laboratorija ili da je dostavljen od strane vanjskog dobavljača
- Vraćen: dobavljač nije mogao zadovoljiti narudžbu i vratio je narudžbu uz mogućnost kasnije dostave
- Otkazan: Zahtjevatelj je otkazao primljen zahtjev prije negoli je isporučen ili je kupac otkazao narudžbu kod vanjskog dobavljača prije nego što je dostavio narudžbu ili nakon što je dobavljač vratio narudžbu uz mogućnost kasnije dostave

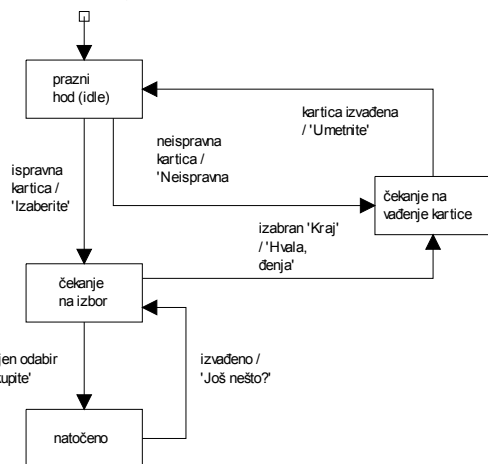


FER \ Fertalj: Razvoj informacijskih sustava

Primjer: sustav za rad u stvarnom vremenu

□ Dijagrami sustava za rad u stvarnom vremenu razlikuju se po tome što sadrže posebno stanje "besposlen"

- Primjer, Sokomat hotela Proljeće ili Bankomat banke Oprosti nam duge naše



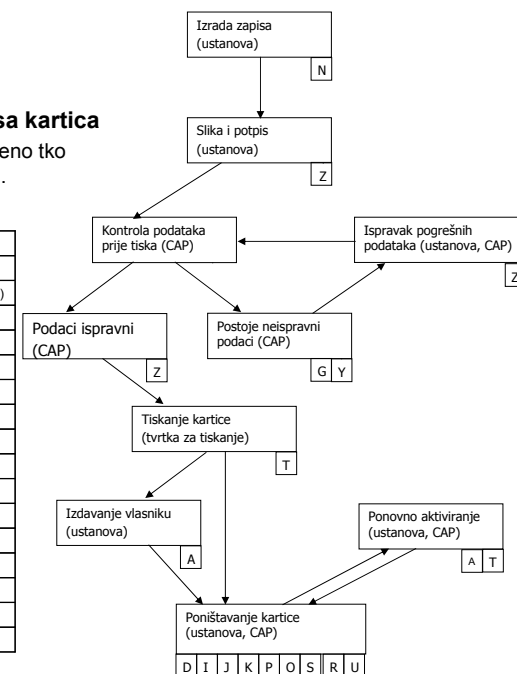
FER \ Fertalj: Razvoj informacijskih sustava

283

□ Primjer: prijelazi statusa kartica

- U zagradama je navedeno tko promjenu smije načiniti.

A	Izdana
D	Student je diplomirao
G	Pogrešni podaci (koristi samo tvrtka za tiskanje)
I	Izgubljena
J	Poništena zbog promjene JMBG-a
K	Istekao apsolutni rok
N	Nezavršena
O	Oduzeta
P	Poništena
R	Student je upisao prekid
S	Student se ispisao
Š	Oštećena
T	Tiskana
U	Ukradena
X	Nestala
Y	Poništena zbog neispravnih podataka
Z	Zatražena



FER \ Fertalj: Razvoj informacijskih sustava

284

Mape dijalogu

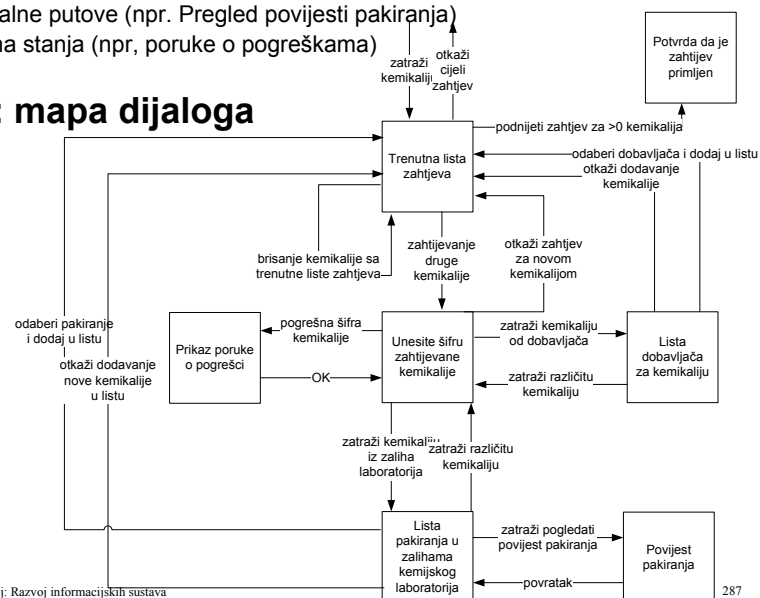
- ❑ **Korisničko sučelje u mnogim aplikacijama se može promatrati kao konačni automat.**
 - Jedan element sučelja (izbornik, radna površina, komandna linija ili prozor dijaloga) može biti aktivan u određenom trenutku. Korisnik može doći do ograničenog broja drugih elemenata ovisno o akcijama koje poduzima.
 - Broj putanji kojima korisnik može mijenjati dijaloge je obično vrlo velik, ali je konačan i mogućnosti su najčešće poznate.
- ❑ **Mape dijaloga prikazuju sustav na visokoj razini apstrakcije.**
 - Prikazuju elemente dijaloga u sustavu i mogućnost navigacije između njih ali ništa ne govore o dizajnu zaslona.
 - Korisnici i razvojni tim mogu zajednički razmatrati dijalog mape kako bi postigli zajedničko gledište o tome kakva treba biti korisnikova interakcija sa sustavom kako bi postigao određeni zadatak.
 - Dijalog mape su također korisne kod modeliranja vizualne arhitekture web sjedišta, pa se ponekad tako i nazivaju (eng. site maps). Navigacijski linkovi na sjedištu pojavljuju se kao tranzicije na dijalog mapi.

Mape dijalogu

- ❑ **Koristi se notacija dijagrama dijagrama stanja**
 - Uvjet koji pokreće korisničku navigaciju prikazan kao tekst na strelicama.
 - Postoji nekoliko vrsta pokretačkih uvjeta:
 - Korisnička akcija, kao što je pritisak tipke ili klik na link ili gumb dijaloga
 - Podatkovna vrijednost, npr. pogrešan unos koji izaziva poruku o pogrešci
 - Sistemski uvjet, kao što je signal da je pisač ostao bez papira
 - Kombinacija navedenih, npr. tipkanje opcije iz menija i pritisak na tipku Enter
- ❑ **Radi pojednostavnjenja mogu se izostaviti neke općenite funkcije, npr.**
 - pritisakanje tipke F1 da bi se dobila pomoć
 - standardni navigacijski linkovi koji se pojavljuju na svakoj stranici.
- ❑ **Pri analizi mapa predstavlja interakciju na konceptualnoj razini**
 - Konkretna ugradnja može biti drugačija.
 - Primjer:
 - Korisnik inicira model korištenja odabirom opcije "zatraži kemikaliju" iz glavnog izbornika a što se ne vidi na dijagramu.
 - Nije nužno prikazivati detalje (npr. Potvrdu otkazivanja čitavog zahtjeva)
 - Ključno je da korisnik i razvojnici jednako "razumiju" opisanu funkcionalnost.

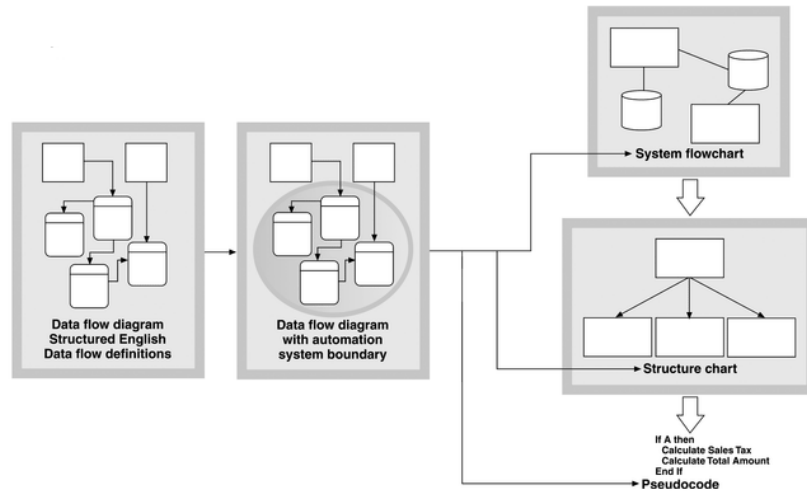
- ❑ **Mape dijaloga mogu prikazati**
 - Alternativne putove (npr. naručiti od dobavljača) koji odstupaju od uobičajenog puta (kemikalija iz zaliha).
 - Opcionalne putove (npr. Pregled povijesti pakiranja)
 - Posebna stanja (npr. poruke o pogreškama)

Primjer: mapa dijaloga



Strukturirani dizajn programa

Strukturirani dizajn

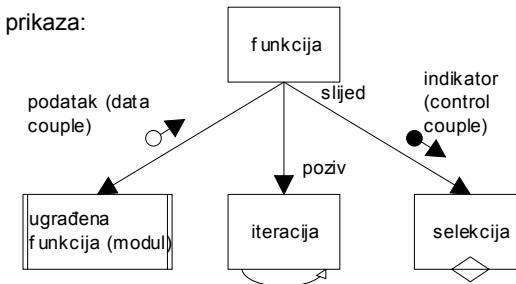


Strukturna karta

Strukturna karta (Structure Chart)

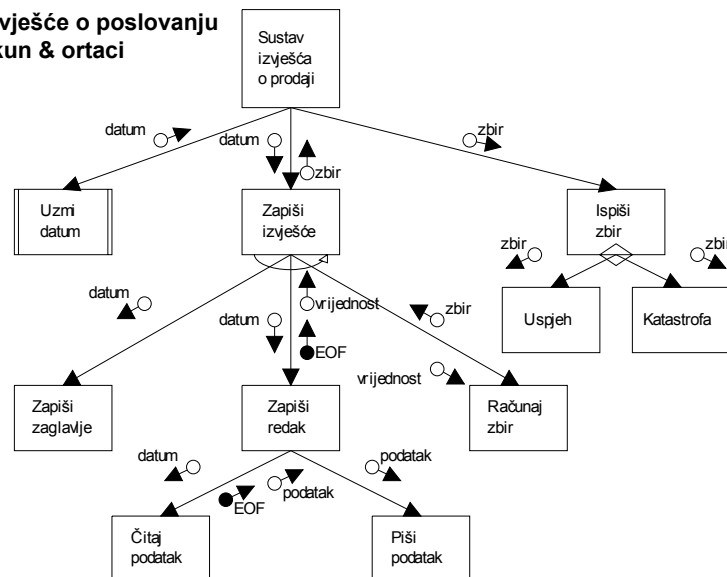
- modeliranje programske podrške na temelju dijagrama toka podataka
 - dijagram toka podataka prikazuje ŠTO treba postići
 - strukturnom kartom izražava se KAKO ostvariti zahtjeve
- prikaz hijerarhije programskih modula koji uključuje
 - prijenos podataka i upravljanja između različitih razina obrade
 - prikaz slijedne, ponavljajuće i uvjetne obrade

elementi prikaza:



Strukturna karta

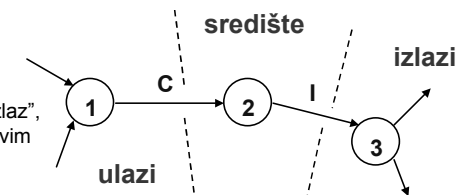
Primjer, izvješće o poslovanju tvrtke Tajkun & ortaci



Transformacijska analiza

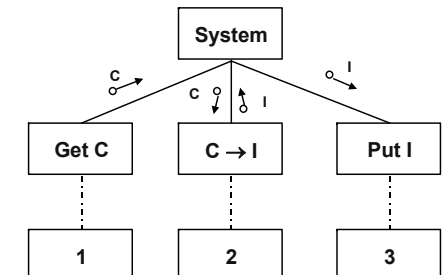
Transformacijska analiza (transform analysis)

- analiza promjene/pretvorbe podataka
- primjenjiva na sustave koji imaju strukturu oblika "ulaz-središte-izlaz", tj. aplikacije s jasno raspoznavljivim ulazima, središnjom obradom i izlazima, koji se daju prikazati linearnim tokom podataka



Struktura dizajna prikladna za ovakve sustave sastoji se od tri odgovarajuća elementa (ulaz, obrada, izlaz), tj. podsustava:

- Get C, koji pribavlja podatak C
- C → I, koji obradom pretvara podatak C u podatak I
- Put I, koji ispisuje rezultat I



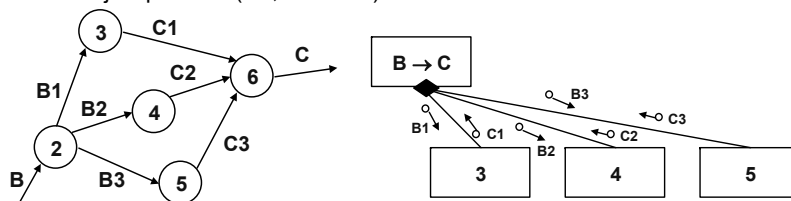
Transakcijska analiza

Transakcijska analiza (transaction analysis)

- analiza izvršenja/obavljanja obrade
- primjenjuje se na sustave sa jasno raspoznavljivim središtima izvršenja (transaction centre), tj. sustave u kojima se donosi odluka o tome koji će se proces koristiti za pretvorbu ulaza u izlaze (npr. interaktivne aplikacije)
- ulaz se usmjerava nekom od modula obrade, a pojedini izlazi se kasnije koriste u daljnjoj obradi

Primjer,

- središte zaprima ulaz B, koji se usmjerava (kao B1, B2 ili B3) u odgovarajući proces (3, 4 ili 5)
- rezultirajući podatak (C1, C2 ili C3) koristi se kao izlazni tok C.



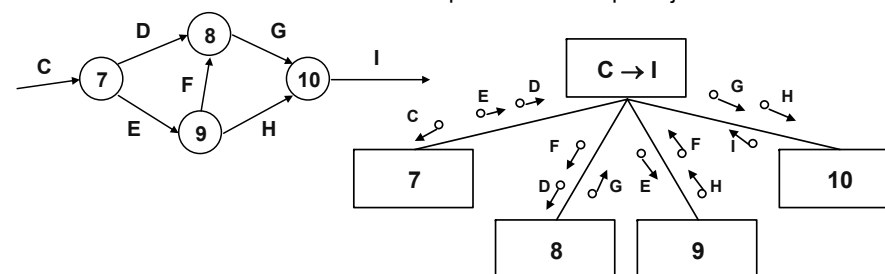
Umreženi procesi

Ostale strukture

- sustavi koji nisu ni transformacijski niti transakcijski obrađuju se na poseban način
- najčešće se oblikuju plošnim razlaganjem glavnog procesa u sastavne procese
- nadređeni proces mora pribaviti sve ulaze potrebne za obavljanje pojedinih podređenih procesa te prikupiti i čuvati proizvedene rezultate obrade

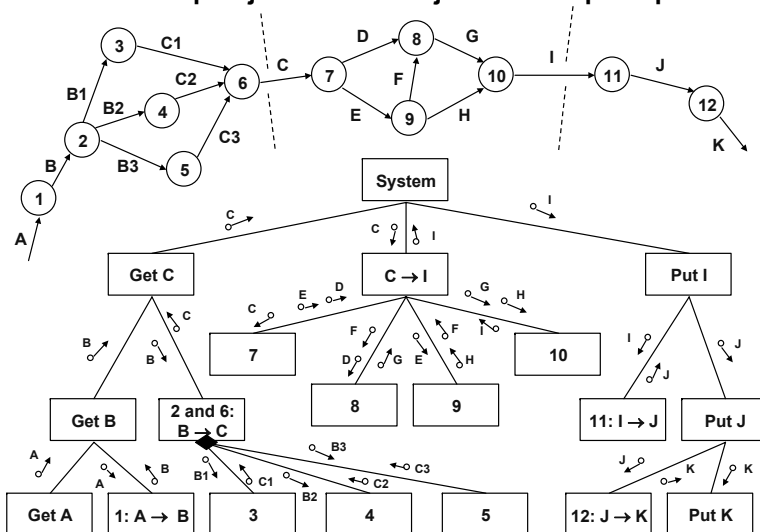
Primjer

- strukturna karta za DTP razložen plošnom dekompozicijom



Složeni sustavi

Stvarni sustavi – primjena kombinacija osnovnih postupaka



Primjeri

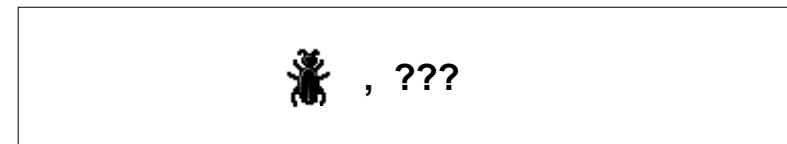
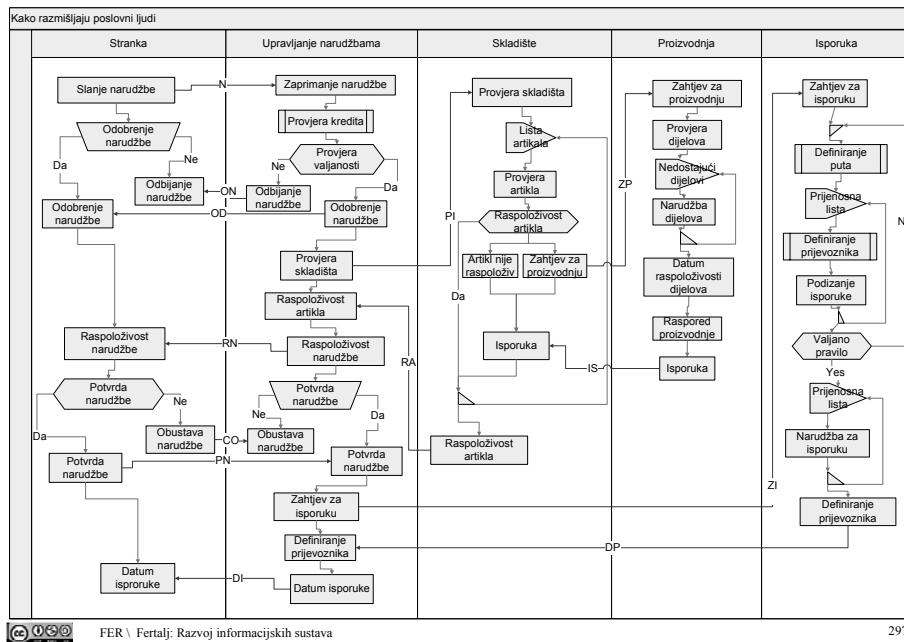
Modeliranje\StrukturiranaAnalizaDizajn

- primjer strukturiranog oblikovanja

Modeliranje\KakoRazmišljajuPoslovnjiLjudi

- Izvor: Howard Smith - The Third Wave of Business Process Management





(Konceptualno) Modeliranje podataka

5/13

Modeliranje podataka

□ Modeliranje podataka (Data modeling)

- tehnika organiziranja i dokumentiranja podataka sustava
- sinonimi:
 - modeliranje baze podataka (podaci se najčešće pohranjuju u BP)
 - modeliranje informacija (općenitije)

□ (po mnogima) najvažnija tehnika oblikovanja

- podaci su resurs koji se dijeli između većeg broja procesa
- strukture podataka i njihova svojstva su trajniji i stabilniji od procesa
- modeliranje podataka završava brže nego modeliranje procesa
 - modeli podataka brže se približavaju rezultatu nego modeli procesa
 - modeli podataka su bitno manji od modela procesa i objektnih modela
- modeli podataka postojećeg i budućeg sustava međusobno su sličniji nego modeli procesa postojećeg i budućeg sustava, ili ih je lakše preoblikovati, pa se manje posla "baca"
- dobra komunikacija s korisnicima, utvrđivanje nazivlja
- lakše određivanje definiranje dosega projekta
- ...

Vrste modela

❏ Konceptualni

- najčešće model entiteti-veze
- poslovni pogled na podatke
- zanemaruje nekritične detalje
- uglavnom sadrži značajne entitete i veze "više prema više"
- može sadržavati važne atribute i ključeve

Logički

- relacijski (najčešće), postrelacijski, objektno-relacijski
- uobičajeno podliježe relacijskoj teoriji
- podrazumijeva potpuno normalizirane entitete (samo veze jedan prema više)
- potpuno definirani atributi i njihove domene (logičke tipove vrijednosti)
- potpuno definirani ključevi

Fizički

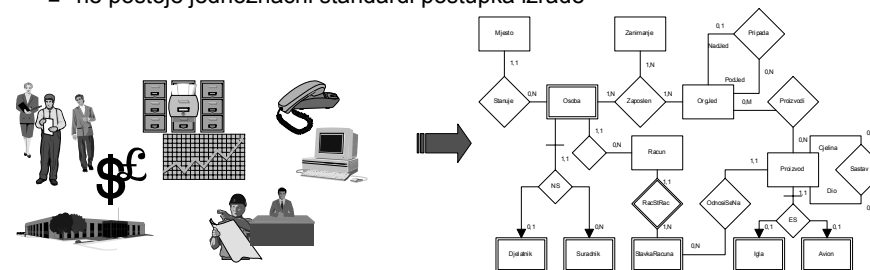
- logički model preveden u model podataka za odabrani SUBP
- uključuje stvarne tipove, indekse, poglede, ugradnju referencijskog integriteta te fizičke parametre (podatkovni prostor, volumetrija)



Model entiteti-veze

Model entiteti-veze

- model baze podataka koji opisuje entitete (pojavnosti, instance), njihove attribute i veze.
 - "The entity-relationship model: toward a unified view of data", P.P. Chen, ACM Transactions on Database Systems 1:1 pp 9-36, 1976.
- za prikaz se koristi dijagram entiteti-veze (Entity-Relationship Diagram, ERD)
 - naziva se još i dijagram objekti-veze (izbjegavati!)
 - postoje različite notacije, npr. Chen, Martin, ...
- ne postoje jednoznačni standardi postupka izrade



Entiteti

- ❑ **Entitet (entity)**

- Stvar koja se može zasebno identificirati [P. Chen, 1976]
- Logička reprezentacija podatka [C.Finkelstein, 1989]
- Bilo što o čemu pohranjujemo informaciju [J.Martin, 1989]

❑ **Primjeri entiteta:**

- osoba, npr. Kit Karson
- objekt, npr. video zapis Prohujalo s vihorom
- apstraktni pojam, npr. hrvatski jezik ili iskustvo (poznavanje jezika)
- ustanova (FER), organizacija (Hotel Proljeće) ili organizacijska cjelina (Ured za istraživanje ruda i gubljenje vremena)
- događaj (situacija, stanje) - npr. rođenje, školovanje, zaposlenje, smrt
- povezanost različitih objekata stvarnog svijeta, npr. rodstvo

☐ **Grupiranje pojedinačnih pojava (instanci)**

- ovisno o metodi
 - skup entiteta (entity set),
 - tip entiteta (entity type),
 - razred entiteta (entity class)
- možemo poistovjetiti pojam entitet sa skupom entiteta, kada ne razmatramo podatke
- označava se imenicom (u jednini) → Osoba. Fakultet



Atributi

☐ **Atribut** - predstavlja neko obilježje, značajku entiteta

- sinonimi: svojstvo (property), element, polje (field)
- po vrijednostima koje predstavljaju, atributi mogu biti:
 - jednostavni atributi (simple attribute) - vrijednost atributa je pojedinačni podatak,
 - primjer: Prezime, Ime
 - složeni, sastavljeni atributi (compound/composite/concatenated attribute, data structure) - vrijednost je uređena n-torka jednostavnih atributa
 - primjer: datum = (dan, mjesec, godina)
 - višeznačni atributi (multivalued attribute) - atributi s više vrijednosti istog tipa
 - primjer: Osoba.Telefon = (TelefonNaPoslu, TelefonKodKuce, MobilniTelefon)
- s obzirom na pohranu vrijednosti, atributi mogu biti:
 - atributi pohrane (stored attribute)
 - izvedeni atributi (derived attribute) - vrijednost temeljem vrijednosti drugih atributa
 - primjer: starost = (DanašnjiDatum - DatumRođenja)

❑ Tipovi podataka atributa

- netehnički (logički) - opći tipovi koji se koriste u sistem analizi
 - primjeri: *broj, datum-vrijeme, znakovni niz, tekst, BLOB*
- tehnički - generički, koji se mogu preslikati u konkretne tipove, npr. *integer, character*
 - konkretni tipovi SUPB, npr. *char, int, byte (SQLserver)*



Ključevi

❑ Ključ (key) ili identifikator

- atribut ili skup atributa koji (svojim vrijednostima) jednoznačno identificira svaki od entiteta u nekom skupu entiteta
- mora se sastojati od bar jednog atributa → jednostavan ključ
 - primjer: OSOBA = @JMBG + Prezime + Ime ...
- može se sastojati od više atributa → složeni, sastavljeni, ulančani ključ
 - primjer: MJESTO = @ŠifraDržave+@ŠifraMjesta +...
- može postojati više kandidata za ključ
 - primjer: Valuta.ŠifraValute (385), Valuta.KratikaValute (HRK)
- nakon odabira (primarnog) ključa, ostali mogući ključevi postaju alternativni ključevi (alternate key)

❑ Strani ključ (foreign key)

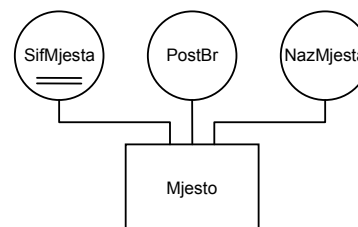
- skup atributa čije vrijednosti odgovaraju vrijednostima ključa drugog entiteta
 - primjer: Osoba.SifMjesta odnosi se na Mjesto.SifMjesta



Elementi dijagrama Entiteti-Veze

❑ Primjer,

- entiteti Mjesto i Osoba
- primarni ključ Mjesto.SifMjesta i strani ključ Mjesto.SifMjesta
- izvedeni atribut Staz
- višeznačni atribut Telefon

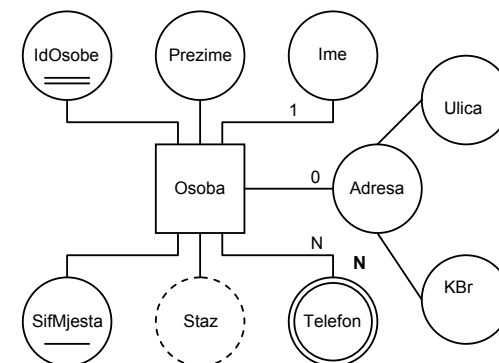


❑ Atributi mogu imati kardinalnost

- 0, pozitivni cijeli broj ili znak (npr. N)

❑ Sastavljeni atributi imaju attribute

- primjer, Adresa



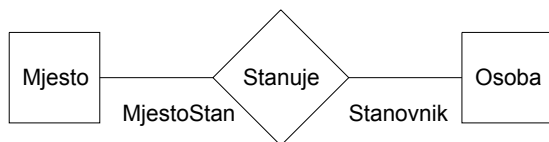
Veze

❑ Veza (relationship)

- pokazuje odnos između entiteta
 - analogno entitetima, pojedinačna veza uspostavlja se na razini instanci entiteta, a veze se grupiraju u skupove veza (relationship sets)
 - kada se ne razmatraju instance, pojam veza podrazumijeva skup veza
- može izražavati ulogu entiteta koje povezuje
- imenuje se glagolom ili glagolskom imenicom

❑ Primjer (veza i uloge)

- Osoba STANUJE u Mjestu (Osoba je STANOVNIK Mjesta)
- u Mjestu STANUJE Osoba (Mjesto je MJESTO STANOVANJA)



Stupanj, tip i kardinalnost veze

❑ Stupanj veze (degree of relationship)

- broj entiteta koji sudjeluju u vezi (1, 2, 3, ... N) - unarna veza (refleksivna, rekurzivna, involucijska), binarna, ternarna, ..., n-arna veza

❑ Tip, klasifikacija veze (type of relationship) - odnos entiteta u vezi

- jedan-prema-jedan (1:1),
- jedan-prema-više (1:N), pri čemu mogu postojati paralelne veze
- više-prema-više (M:N)

❑ Modalitet veze (modality) i kardinalnost veze (cardinality)

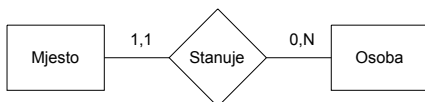
- minimalni i maksimalni broj drugih instanci s kojima se pojava nekog entiteta može povezati → pojednostavnjeno, donja i gornja granica kardinalnosti
 - donja granica (modalitet) može biti 0, pozitivni cijeli broj ili znak (npr. M)
 - donja granica = 0 → djelomično, neobavezno (optional) pridruživanje
 - donja granica ≠ 0 → potpuno, obavezno (mandatory) pridruživanje
 - gornja granica (kardinalnost) može biti pozitivni cijeli broj ili znak (npr. N)
- veze su dvosmjerne pa se kardinalnost definira za svaku stranu veze
- stil označavanja: donja,gornja kao donja..gornja, pr. 0,N kao 0..N



Binarne i unarne veze

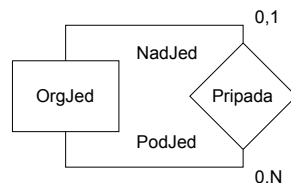
Primjer: binarna veza 1:N

- što kada bi kardinalnost mjesta bila 0,1 ?



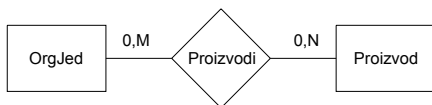
Primjer: unarna veza 1:N

- što kada bi modalitet bio 1 ?

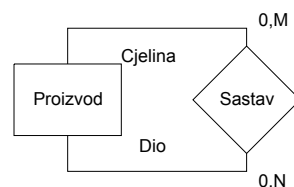


Primjer, binarna veza M:N

- što kada bi kardinalnost s bilo koje strane bila 1,N, odnosno 1,M ?



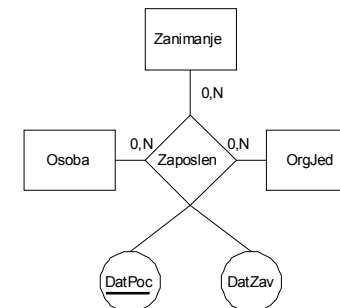
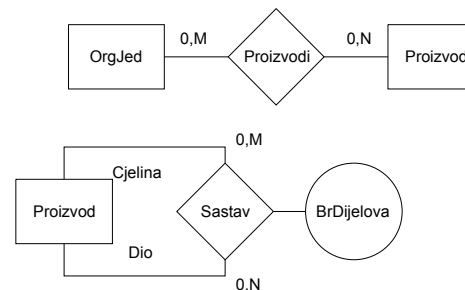
Primjer, unarna veza M:N



Spojni entiteti, veze višeg stupnja, veze s podacima

Asocijativni entitet (Gerund)

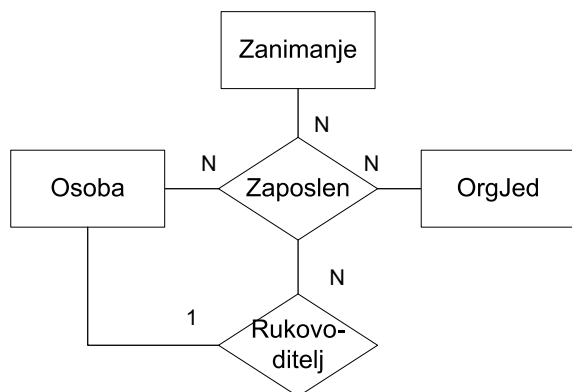
- konkatenirani entitet (concatenated entity),
- kompozitni entitet (composite entity)
- binarna veza M:N, npr. Proizvodi
- ternarna veza ili veza stupnja >3, npr. Zaposlen
- veza o kojoj se želi pohraniti podatke (veza s opisnim atributima), pr. Sastav



Agregacija

Agregacija (aggregation) - agregacijska veza

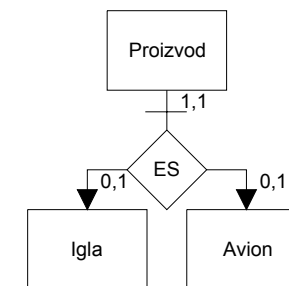
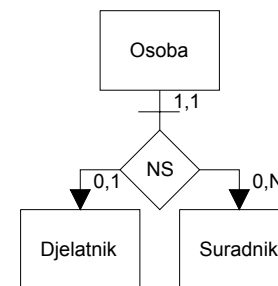
- sinonim: agregacijski entitet (aggregate entity)
- veza koja sudjeluje u vezama s drugim entitetima
- poimlje se kao apstraktni entitet
- u nekim metodama označava odnos cjelina-dio



Specijalizacija/Generalizacija

Generalizacija/Specijalizacija - takozvana "jest" veza ("is a")

- odnos nekog entiteta (nadtip) i njegovih posebnosti (podtip)
 - nadtip (supertype) - generalizacija podređenih, sadrži zajedničke attribute
 - podtip (subtype) - pecijalizacija nadređenog, sadrži svojstvene attribute
- potpunost generalizacije:
 - djelomična – ne mora se specijalizirati
 - potpuna – mora se specijalizirati
- specijalizacija može biti:
 - neekskluzivna - Osoba u isto vrijeme može biti i Djelatnik i Suradnik
 - ekskluzivna - Proizvod je Igla ili Avion, ali ne istovremeno
 - uvjetovana – umirovljenici su samo osobe preko N godina



Jaki i slabi entiteti, Identifikacijska veza

Jaki entitet

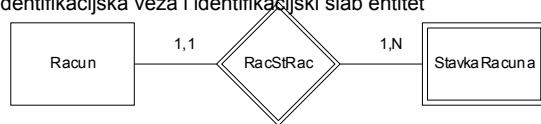
- postoji samostalno, nezavisan/dominantan entitet, pr. Mjesto
- zanimljivo, to su uglavnom šifarnici ☺

Slabi entitet (weak entity)

- postoji samo ako postoji jaki entitet u vezi
- većina baze je slaba !
- entitet može biti slab uslijed više veza – slabost/jakost razmatrati lokalno

Identifikacijska veza

- definira identifikacijski slabi entitet - entitet koji se ne identificira samostalno, tj. entitet koji nema vlastiti ključ, nego se njegov ključ tvori od ključa jakog entiteta i (opcionalno) vlastitog atributa (deskriptora), pr. StavkaRacuna
- Primjer, identifikacijska veza i identifikacijski slab entitet



Specijalizacije su u pravilu identifikacijski slabe

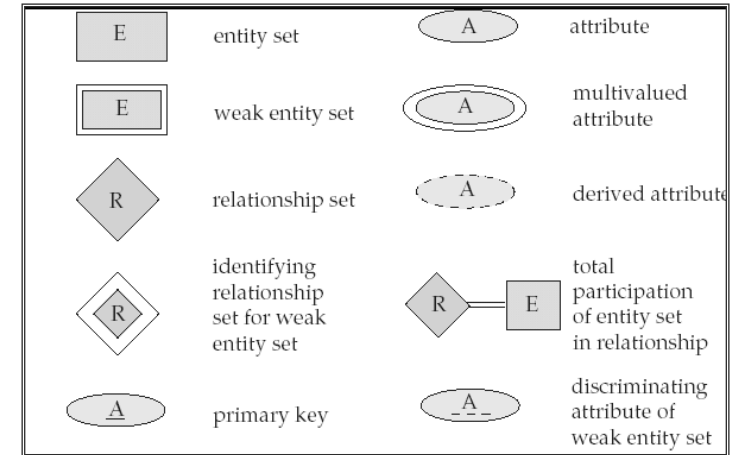
- Primjer, podtipovi: Djelatnik, Suradnik, Igla, Avion

Komentirati kardinalnost primjera



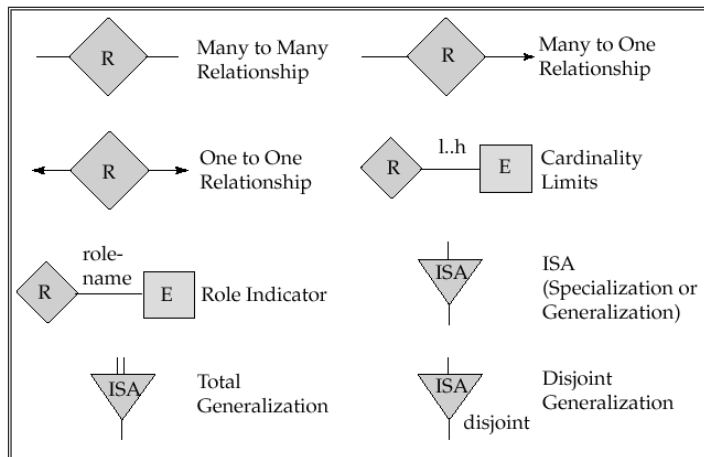
Proširena Chenova notacija (1)

- Potpuna participacija – svaki entitet mora sudjelovati u barem jednoj vezi



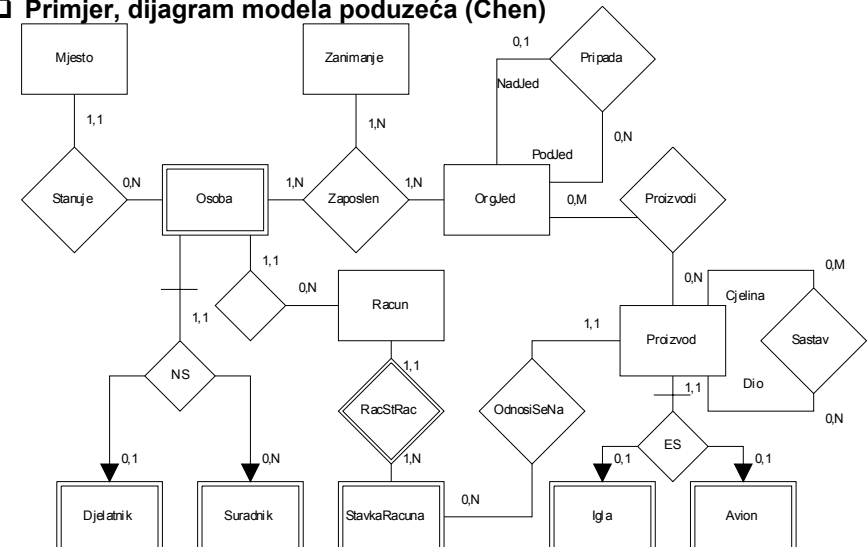
Proširena Chenova notacija (2)

- Potpuna generalizacija – entitet se mora specijalizirati



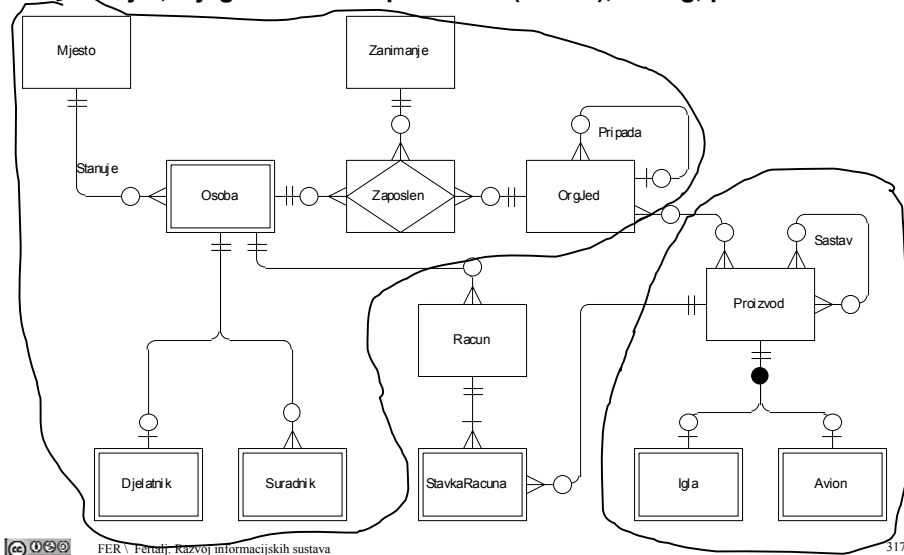
ERD notacija - Chen

Primjer, dijagram modela poduzeća (Chen)



ERD notacija - Martin

Primjer, dijagram modela poduzeća (Martin), doseg, podsustavi

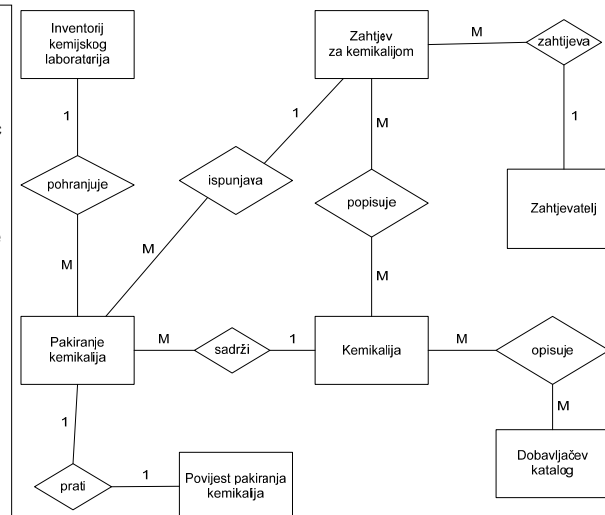


FER \ Fertilj: Razvoj informacijskih sustava

317

Izrada ERD analizom izjava korisnika

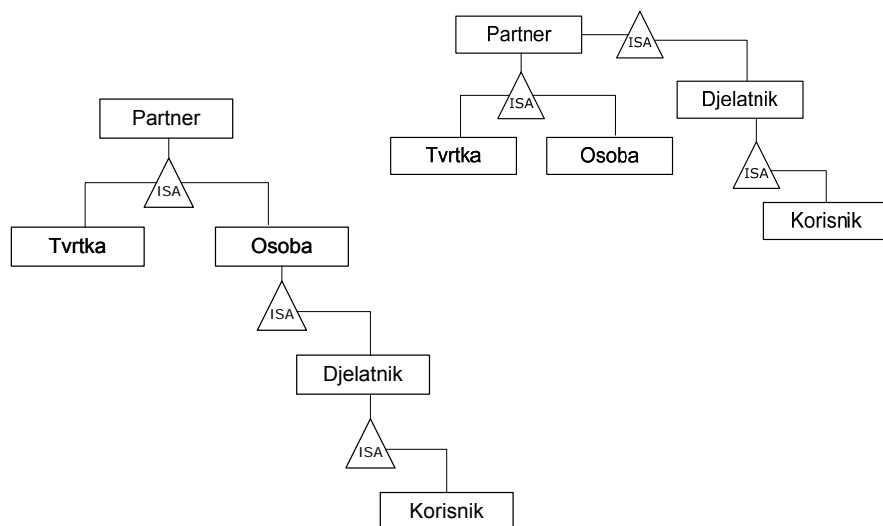
"Kemičar ili član osoblja kemijskog laboratorija može podnijeti zahtjev za jednom ili više kemikalija. Zahtjev može biti udovoljen ili dostavom pakiranja kemikalije koja se već nalazila na zalih kemijskog laboratorija ili upućivanjem narudžbe za novim pakiranjem dobavljača. Osoba koja upućuje zahtjev mora imati mogućnost pretraživanja kataloga kemikalija vanjskog dobavljača dok sastavlja narudžbu. Sustav mora pratiti status svakog zahtjeva za kemikalijama od trenutka kad je ispunjen do trenutka kad je udovoljen ili otkazan. Također, mora pratiti povijest svakog pakiranja kemikalija od trena kad stigne u kompaniju do trenutka kad je potpuno upotrijebljen ili odbačen."



FER \ Fertilj: Razvoj informacijskih sustava

318

Primjer: hijerarhija specijalizacija



FER \ Fertilj: Razvoj informacijskih sustava

319

Razvoj modela podataka

Razvoj modela podataka

❑ Razvoj modela podataka

- Globalni model podataka (enterprise data model)
- Model konteksta podataka (context data model)
- Model s definiranim ključevima (key-based data model)
- Model s potpuno određenim atributima (fully attributed data model)
- Potpuno opisani model (fully described data model)

❑ Konceptualni model postane logički u trenutku kada sadrži samo veze 1:N s definiranim stranim ključevima ili se takve mogu jednoznačno generirati

- Odluke o konačnom dizajnu donose se tijekom pretvorbe konceptualnog modela u relacijski → Logičko modeliranje podataka



Konceptualni model podataka

❑ Globalni model podataka

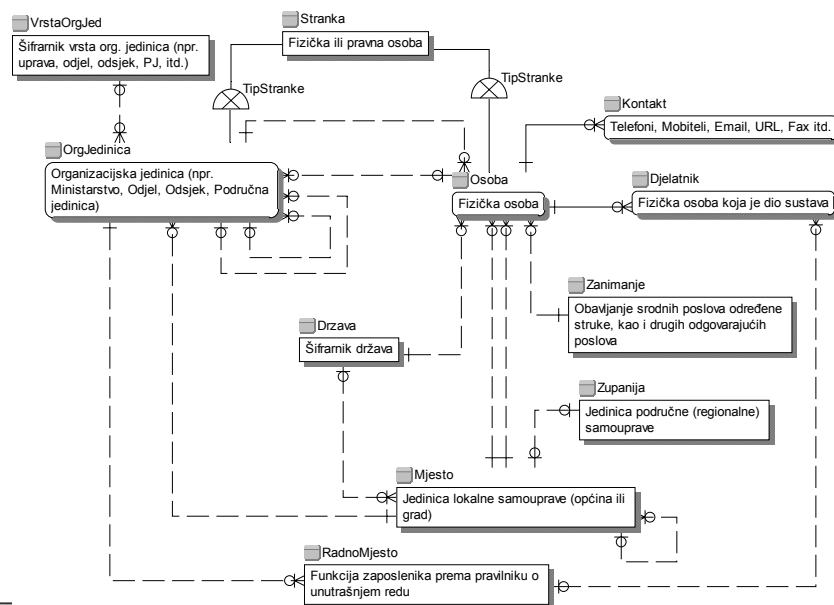
- podatkovni model poduzeća (enterprise data model) u fazi planiranja
- konceptualni model entiteti-veze koji najčešće sadrži neodređene veze i nerazrađene kategorije podataka, a pojedine veze mogu i nedostajati
- definira mogući doseg sustava i ukupne informacijske potrebe
- Primjer: Analiza\IdejnoSljeme

❑ Model konteksta podataka

- na početku analize
- konceptualni model koji sadrži samo one entitete koji će biti obuhvaćeni tehničkim rješenjem → aplikacijski model podataka
- rafinira doseg bez detalja o entitetima ili detalja o poslovnim pravilima
- ERD s entitetima i vezama, bez atributa ili samo s osnovnim atributima
- uz obične veze tipa 1:N sadrži veze višeg stupnja i specijalizacije



Primjer: konceptualni model konteksta



Prijelaz prema logičkim modelima

❑ Model s definiranim ključevima

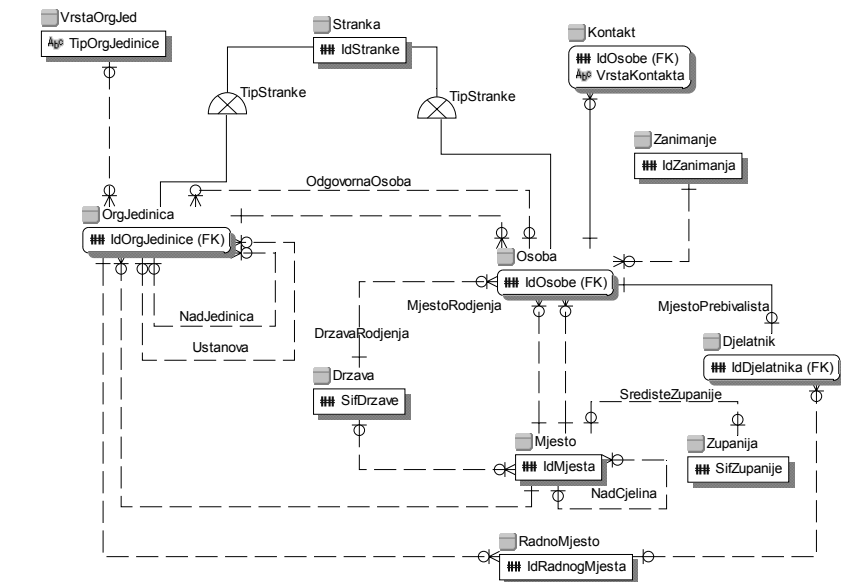
- eliminacija neodređenih veza (veze višeg stupnja, veze s atributima, agregacije) i njihovo nadomještanje asocijativnim entitetima
- određivanje ključeva (primarnih, alternativnih, stranih)
 - ako se PK ne može odrediti, možda se ne radi o skupu entiteta
- preciziranje kardinalnosti veza
- definiranje generalizacijskih hijerarhija
 - određivanje specijalizacija, tj. podtipova entiteta, npr. Igla, Avion
 - definiranje klasifikacijskog atributa nadtipa (diskriminator podtipa)
 - npr. $\text{Proizvod.TipProizvoda} \in \{ \text{"Igla"}, \text{"Avion"} \}$

❑ Model s definiranim atributima

- dodavanje preostalih opisnih atributa
- određivanje podskupova podataka
- definiranje domena, logičkih tipova podataka i standardnih vrijednosti atributa



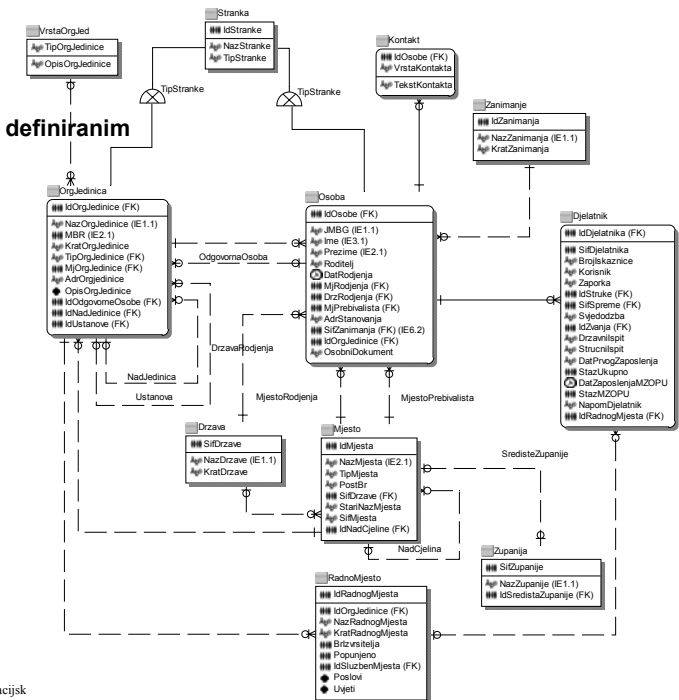
Primjer: Model s definiranim ključevima



FER \ Fertalj: Razvoj informacijskih sustava

325

Primjer: Model s definiranim atributima



FER \ Fertalj: Razvoj informacijskih sustava

Dokumentiranje i konverzija modela entiteti-veze

Potpuno opisani model

- potpuni opis atributa, logičkih tipova podataka i standardnih vrijednosti
- dodatni opisi: prava pristupa podacima, ...
- vremenski najzahtjevniji zadatak, uobičajeno se provodi na kraju, ali može započeti usporedno s izradom modela zasnovanog na ključevima ili definiranjem opisnih atributa

Daljnja konverzija modela

- Pretvorba modela entiteti-veze u relacijski model podataka
- Fizičko oblikovanje podataka → faza dizajna
 - konverzija logičkog u fizički model (schema baze podataka)
 - normalizacija i prilagodba uslijed tehničkih ograničenja i performanci

Logičko modeliranje podataka



FER \ Fertalj: Razvoj informacijskih sustava

327

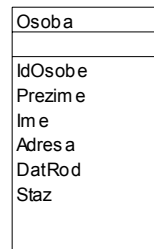
Pretvorba modela E-V u relacijski model

Entiteti

- entitet (skup entiteta) → relacija, npr. Mjesto, Osoba

Atributi

- kardinalnost 0 → opcionalna vrijednost (null)
- kardinalnost 1 → zahtijevana vrijednost (not null)
- kardinalnost N → višeznačni atribut (npr. Osoba.Telefon)
- atribut → atribut, npr. Osoba.Prezime
- izvedeni atribut → atribut pohrane ili se izostavlja, npr. Osoba.Staz
- složeni atribut → atribut (grupiranjem), npr. (dan, mjesec, godina) → datum



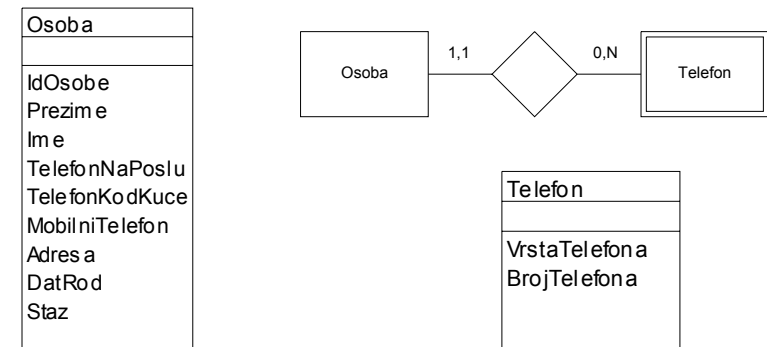
Pretvorba višeznačnih atributa

višeznačni atribut →

- skup odgovarajućih atributa, npr. Osoba.Telefon → Osoba.TelefonNaPoslu, Osoba.TelefonKodKuce, Osoba.MobilniTelefon

ili

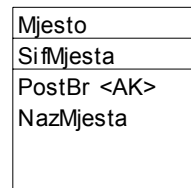
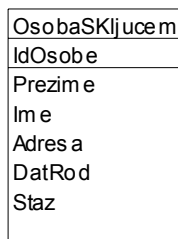
- slabi entitet, npr. Osoba.Telefon → Telefon (IdOsobe, VrstaTelefona, BrojTelefona)



Pretvorba ključeva

Ključevi

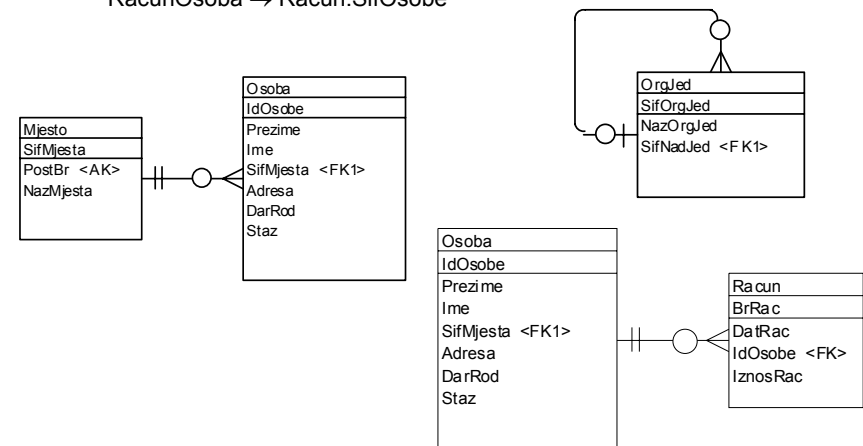
- ključ → primarni ključ, npr. Osoba.IdOsobe, Mjesto.SifMjesta
- alternativni ključ → indeks nad jedinstvenim vrijednostima (unique index) + oznaka zahtijevane vrijednosti (not null), npr. Mjesto.PostBr



Pretvorba binarnih veza

Binarna veza 1:N → strani ključ

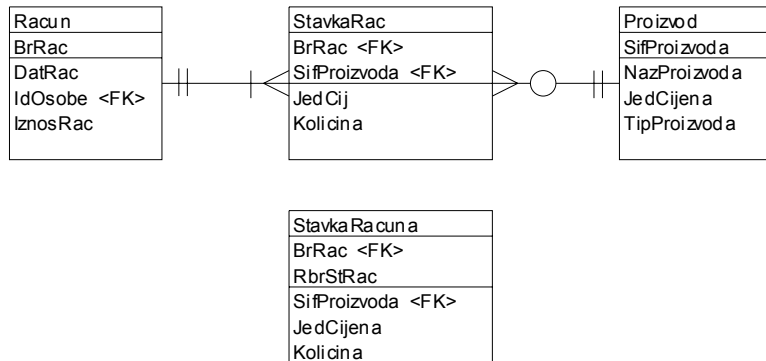
- egzistencijalni slabi entitet → obični strani ključ
- npr. Stanuje → Osoba.SifMjesta, Pripada → OrgJed.SifNadJed, RacunOsoba → Racun.SifOsobe



Pretvorba identifikacijskih veza

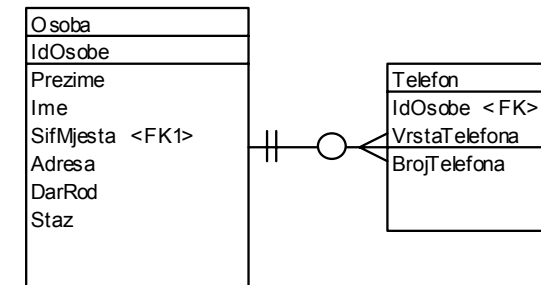


- identifikacijski slabi entitet → nasljeđuje ključ jakog entiteta
 - spojni ključ (compound key), npr. StavkaRacuna (BrRacuna, SifProizvoda, JedCijena, Kolicina)
 - » ili
 - kompozitni ključ (composite key), npr. StavkaRacuna (BrRacuna, RbrStRac, SifProizvoda, JedCijena, Kolicina)



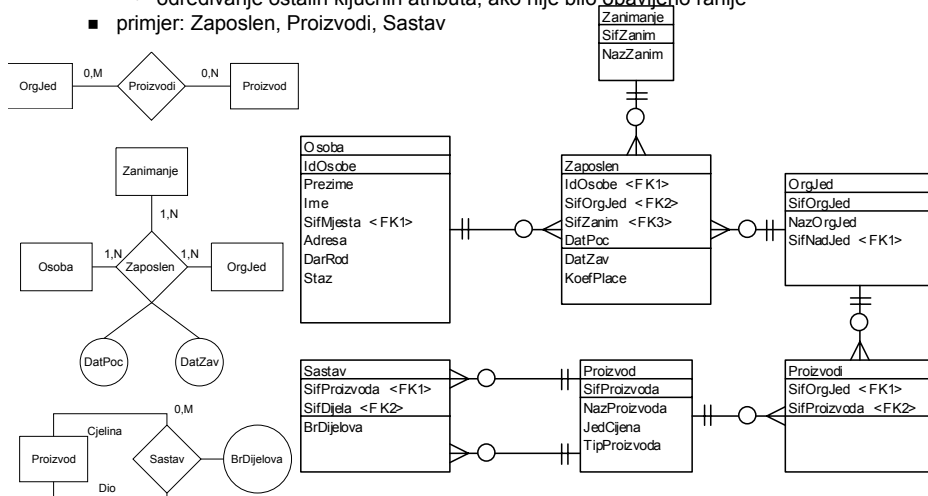
Pretvorba identifikacijskih veza (2)

- Spoj prema vrsti telefona ne treba postojati



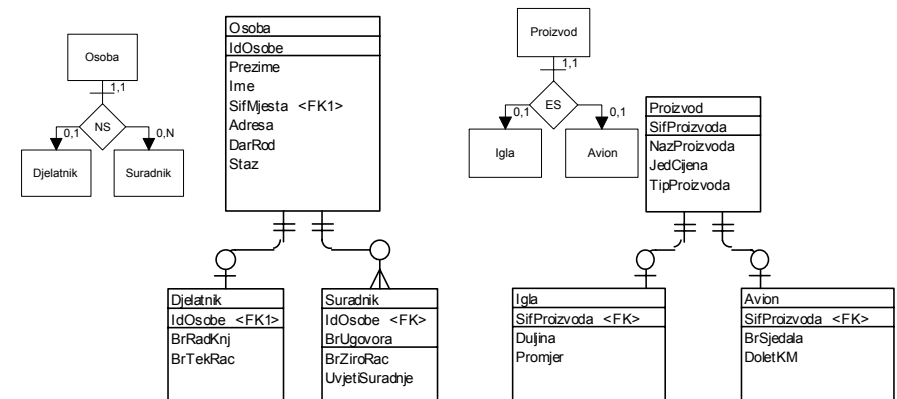
Pretvorba nespecifičnih veza

- Nespecifične veze → asocijativni entitet + n binarnih veza 1:N
 - ključ asocijativnog entiteta = unija ključeva entiteta spojenih vezom
 - određivanje ostalih ključnih atributa, ako nije bilo obavljeno ranije
 - primjer: Zaposlen, Proizvodi, Sastav



Pretvorba specijalizacija

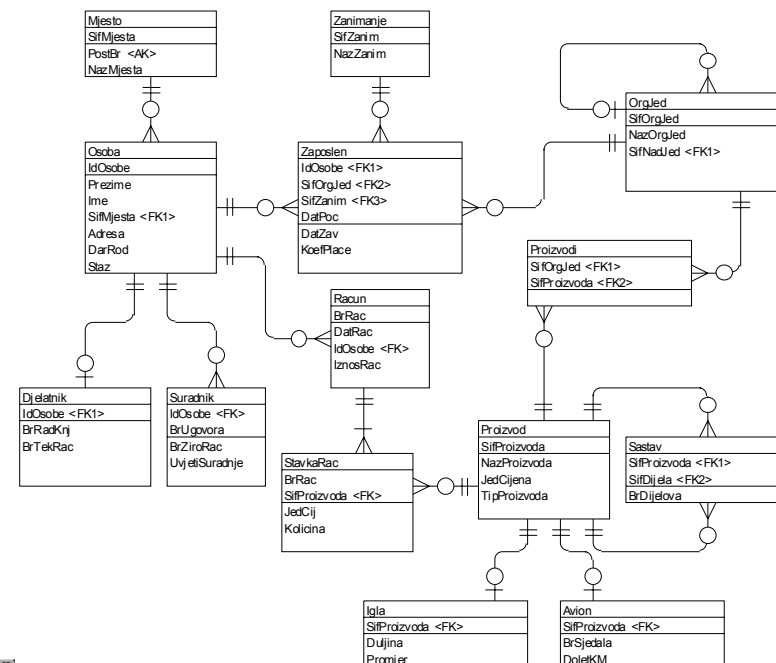
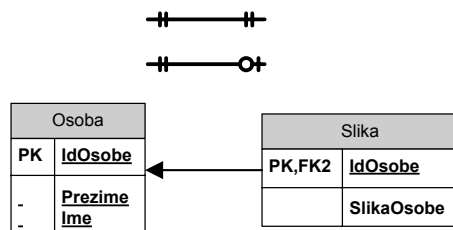
- Specijalizacija nadtipa u n podtipova → n binarnih veza
 - nadtip → (jaki) entitet, kojemu se po potrebi određuje klasifikacijski atribut, npr. Proizvod.TipProizvoda
 - podtip → (identifikacijski) slabi entitet, npr. Igla, Avion, Djelatnik, Suradnik
- Da li i kada koristiti klasifikacijski atribut ?



Pretvorba ostalih binarnih veza

Binarna veza 1,1:0,1 i binarna veza 1,1:1,1 → strani ključ

- identifikacijski slabi entitet bez deskriptora



Preporuke za izradu modela podataka

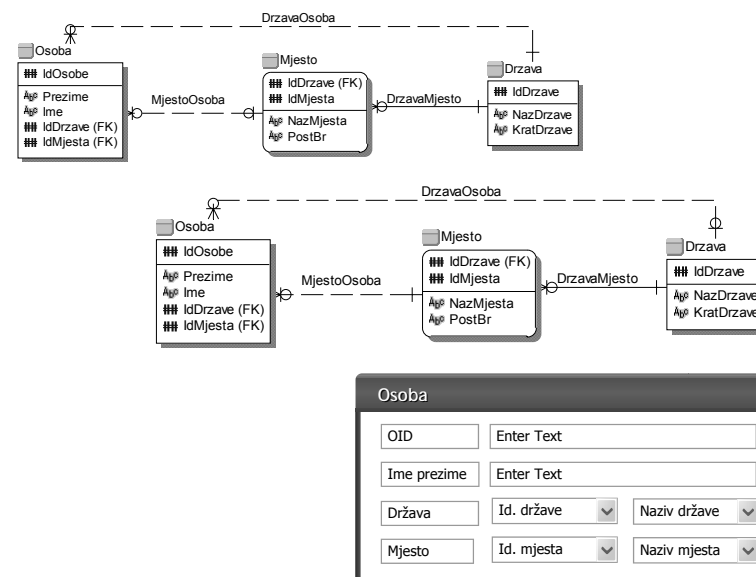
Sinkronizacija s modelom procesa

- spremišta podataka u modelu procesa su entiteti u modelu podataka

Provjera modela

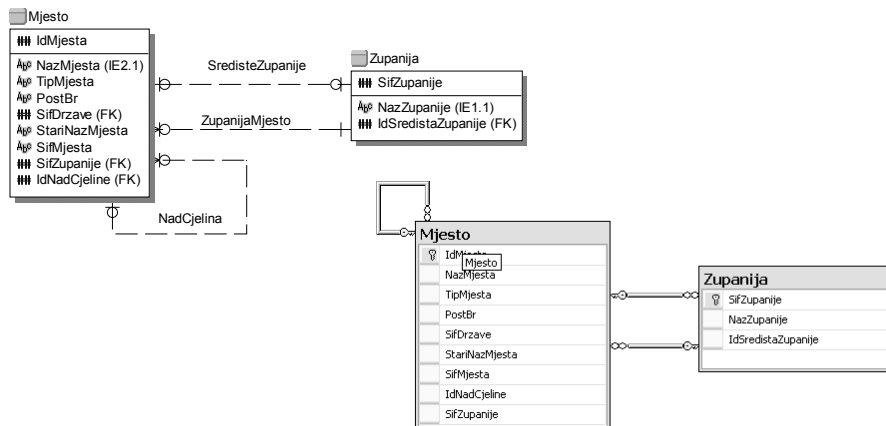
- pripaziti na sinonime (različite nazive istih objekata)
- pripaziti na homonime (jednake nazive različitih objekata)
- provjeriti sumnjive i redundantne veze i po potrebi ih ukloniti (oprez!)
 - uklanjanje neke od paralelnih veza
 - uklanjanje veza koje se daju izvesti iz drugih (tzv. trijade) - pripaziti na kardinalnost veza u lancu
 - ukloniti balansirane veze 1,1:1,1 (ukoliko je moguće!)
 - pripaziti na nebalansirane veze 0,1:1,1 (zavisnost entiteta)
 - cirkularne reference
- izbaciti izvedene attribute
 - moгуći gubitak informacije o potrebnim izvedenim/izračunatim poljima

Uklanjanje veza koje se daju izvesti iz drugih ?



Problem paralelnih i cirkularnih veza - ukloniti?

- primjer: Zupanija 1:N Mjesto i Zupanija 1:1 MjestoSrediste
- slično: OrgJedinica 1:N Djelatnik i OrgJedinica 1:1 DjelatnikUpravitelj
- problem: zahtijevanost vrijednosti stranih ključeva
- problem: realizacija veze 1:1, redundantna središta županija

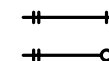


FER \ Fertalj: Razvoj informacijskih sustava

Balansirane i nebalansirane binarne veze

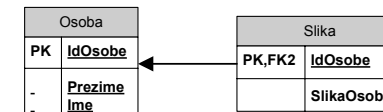
Udruživanje u zajedničku tablicu

- Binarna veza 1,1:1,1 – slika NOT NULL
- Binarna veza 1,1:0,1 – slika NULL



Razlozi za udruživanje

- jednostavniji dohvat podataka
- ubrzanje upita



Razlozi za razdvajanje

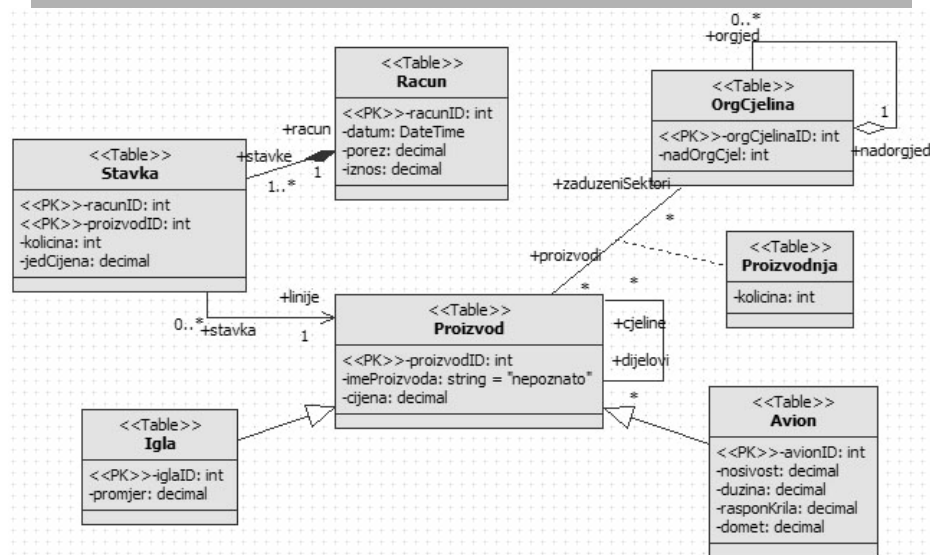
- ušteda diskovnog prostora
- utjecaj na fizički raspored podatka
- tablice u vezi 1:1 s prevelikim brojem stranih ključeva



FER \ Fertalj: Razvoj informacijskih sustava

342

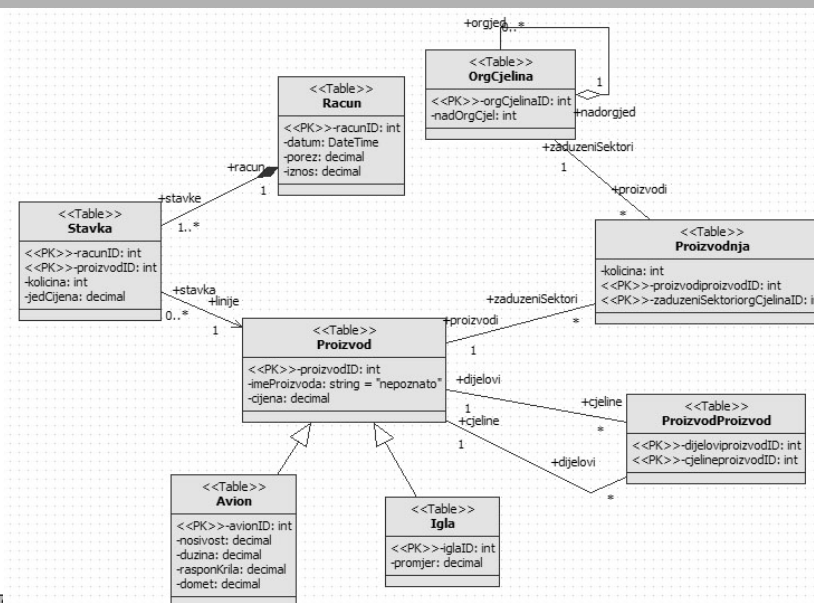
Za sladokusce: konceptualni UML model ...



FER \ Fertalj: Razvoj informacijskih sustava

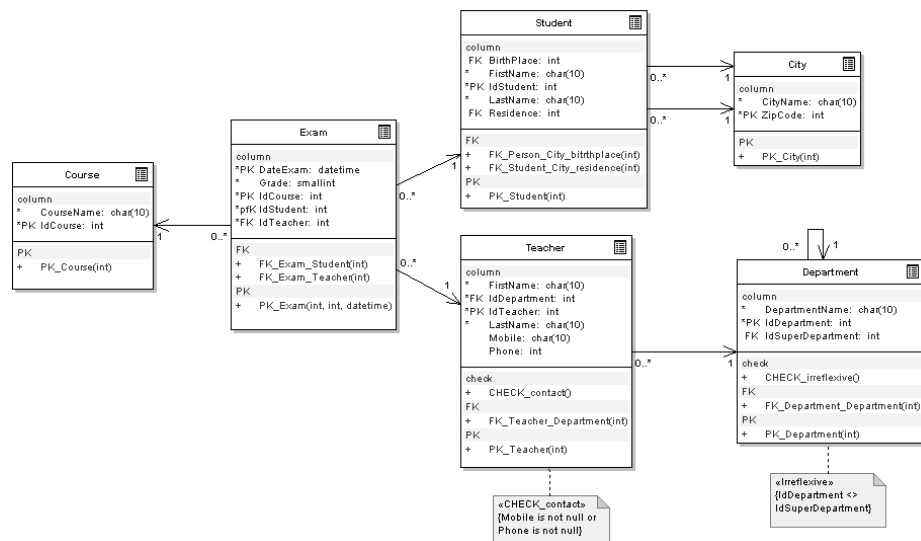
343

... i model nakon transformacije

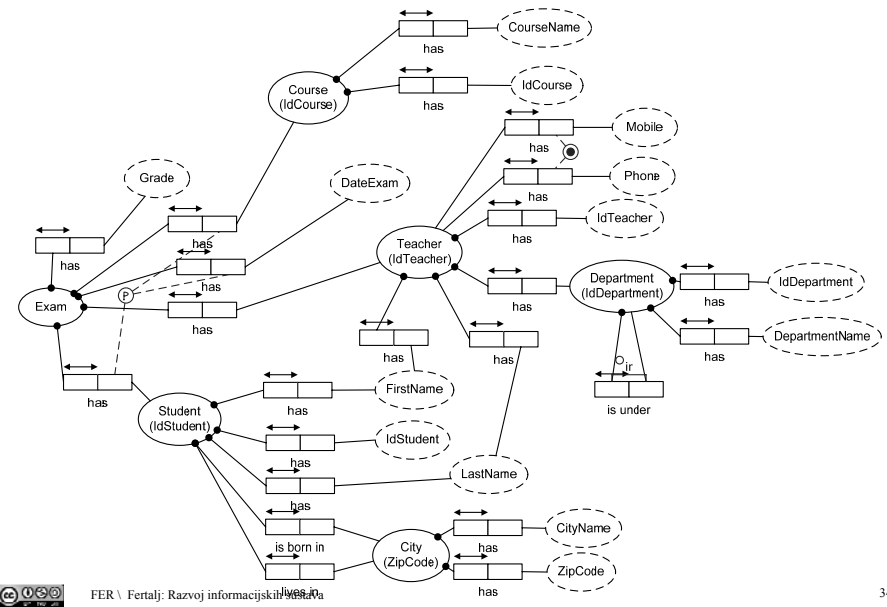


344

UML ...



... i Object Role Modeling (ORM)



Šifarski sustav

Određivanje ključeva

Šifarski sustav

□ Serijske šifre

- brojevi koji se slijedno pridjeljuju svakoj novo dodanoj instanci entiteta
- u modernim SUBP mogu se generirati uz opcionalna ograničenja
- primjer: SQL Server IDENTITY [(seed , increment)]

□ Blok šifre

- slično serijskim šiframa, s tim da su serijski brojevi grupirani prema značenju
- primjer satelitskih TV kanala: 100-199 PAY PER VIEW, 200-299 CABLE CHANNELS, 300-399 SPORT, 400-499 ADULT, 500-599 MUSIC-ONLY, ...

□ Alfanumeričke oznake

- ograničeni skup znakovnih oznaka, često kombiniranih s brojevima
- primjer, oznake država: HR, DE, IT, SI

□ Samgovoreće šifre (significant position codes)

- svaka znamenka ili grupa znamenki opisuje neko svojstvo instance
- primjer: JMBG, a često se koriste i u skladišnoj evidenciji (dimenzije automobilske gume, električne žarulje)

□ Hijerarhijski kodovi

- podjela u grupe, podgrupe itd.



Primjeri šifrnika

□ Primjer, studentska prehrana

- Pravilnikom o studentskoj prehrani propisan je status upisa potreban za ostvarivanje prava na subvencioniranu prehranu.
- Npr. mora biti upisan u tekuću akademsku godinu na redovnom studiju ili studiju za osobne potrebe.

Šifra statusa	Opis statusa	Pravo prehrane
R	redovan studij	DA
P	paralelni studij	NE
U	studij uz rad	NE
V	vanredni studij	NE
L	paralelni studij s pravom prehrane	DA
O	studij za osobne potrebe	DA

□ Primjer, IUCN kriteriji ugroženosti

IdIUCN	NazIUCN	NazIUCNENG
A	Redukcija populacija (smanjivanje broja jedinki)	Reduction in population size
A1	reverzibilna, razumljiva, obustavljena	reversible AND understood AND ceased
A1a	a-direktno opažanje	a-direct observation
A1b	b-indeks učestalosti	b-an index of abundance
...		
B1biii	iii- područja, obima i/ili kvalitete staništa	iii-area, extent and/or quality of habitat
B1biv	iv- broja lokaliteta ili subpopulacija	iv-number of locations or populations
B1bv	v- broja zrelih individuuma	v-number of mature individuals
B1c	kolebanje	fluctuations
...		



Primjer jedinstvene šifre

□ JMBAG je jedinstveni broj u sustavu kojeg student dobiva prilikom upisa na neko visoko učilište i zadržava sve do kraja svog studija.

- Ako se isti student upiše na dvije ili više ustanova uvijek zadržava JMBAG koji je dobio na prvoj ustanovi.
- Djelatnici u ustanovama ne unose taj podatak već se on automatski generira.
- JMBAG ima deset znamenki podijeljenih u dvije grupe te kontrolnu znamenku:
 - Prve četiri znamenke označavaju matičnu ustanovu vlasnika
 - Sljedećih 5 znamenki su oznaka vlasnika u ustanovi (matični broj vlasnika ili se generira slijedno)
 - Unutar ustanove JMBG i matični broj (broj indeksa) su također jedinstveni.

□ Broj kartice generira se automatski, a u sebi sadrži 6 grupa znamenki:

- jedinstveni broj u međunarodnom kartičnom poslovanju (IIN)
 - uvijek **601983** i prema međunarodnom standardu ISO/IEC 7812 na jedinstven način identificiraju Studentsku karticu u međunarodnom sustavu kartičnog poslovanja
- oznaka vrste kartice (1 znamenka)
 - npr: 1 - student i 4 - privremena kartica
- redni broj kartice koju je student dobio (1 znamenka)
- JMBAG (10 znamenki)
- Kontrolna znamenka (1 znamenka),

601983 11 0036324986 0
A BC D E



Izrada šifarskog sustava

□ Izrada šifarskog sustava

- Sustav šifriranja bi trebao biti smislen i prikladan da dodavanje novih šifara bude jednostavno
- Tamo gdje je to moguće treba preuzeti postojeće šifrnike, od drugih ustanova ili iz postojećih sustava
- Oznake definirane zakonom ili drugim propisima treba preuzeti i prilagoditi
- Ostale šifrnike definirati tako da se naknadno mogu nadograđivati
- Izbjegavati samogovoreće šifre

□ Primjer: IPISVU

- šifre djelatnika kao redni brojevi, honorarci oblika 9999*
- dobavljači iz državne riznice, nadopunjeni vlastitim



Dizajn baze podataka

Ugađanje i (de)normalizacija

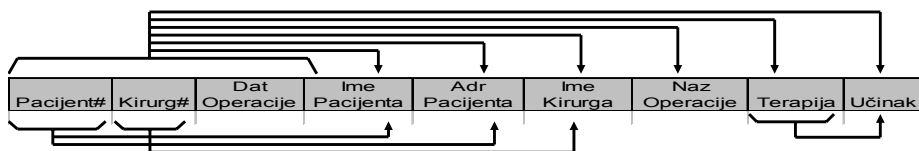
Normalizacija

□ Normalizacija

- postupak strukturiranja sheme relacijske baze podataka tako da se ukloni što više neodređenosti (zalihosti)
- stupanj normalizacije povećava se od 1NF do 5NF
 - većina dizajnera zaustavlja se na 3NF ili na BCNF (Boyce-Codd NF)

□ svodi se na ispunjenje tzv "relacijske zakletve" [Finkelstein, 1989]:

- Ključ
 - 1NF: nema ponavljajućih grupa, definiran primarni ključ
- Cijeli ključ
 - 2NF: svi neključni atributi u potpunosti zavisni o čitavom PK
- Ništa drugo nego ključ
 - 3NF: svaki neključni atribut je neposredno zavisn samo o PK
- Tako mi Codd pomogao! (Dr. E. F. Codd - otac relacijske teorije BP)
 - BCNF: ne postoje kompozitni i/ili preklapajući kandidati ključeva



Denormalizacija

□ Denormalizacija - postupak dovođenja (redukcije) baze podataka u nižu normalnu formu

- Ne smije se miješati s pogreškom nedovođenja u jaču normalnu formu, tj. lošim dizajnom (bad design)

□ Razlozi za denormalizaciju

- poboljšanje performanci, npr. ubrzanje upita eliminacijom vanjskih spajanja
 - pad performanci međutim može biti uzrokovan lošim dizajnom, pa bi u tom slučaju denormalizacija mogla dovesti do još većih problema !
- olakšanje pristupa nekim podacima koje je komplicirano dohvatiti

□ Denormalizaciju treba obaviti samo tamo gdje je to stvarno nužno i na takav način da ne ugrožava integritet podataka

- održavanje redundancije podataka zahtijeva aplikacijsko upravljanje integritetom podataka koje je "skupo"



Preporuke za denormalizaciju

□ Poznavati vlastitu bazu podataka

- kako je logički strukturirana te kako ju aplikacije koriste
- poznavanje frekvencije obrade podataka

□ Denormalizirati mjestimično i oprezno

- Prije denormalizacije ili za usporedbu probati druge tehnike, primjerice mehanizme izračunatih stupaca koje moderne baze podataka podržavaju

□ Analizirati postupke pristupa podacima

- Performance se mogu poboljšati pažljivim postavljanjem indeksa i optimizacijom upita te forsiranjem odgovarajućeg plana njihovog izvršenja

□ Procijeniti fizičke resurse

- povećanje ili bolja raspodjela radne memorije
- dodavanje ili zamjena procesora
- fizički raspored smještaja podataka
- kvaliteta i optimizacija diskovnog prostora



(umjesto denormalizacije)

□ Postavljanje indeksa

- primarni i strani ključevi implicitno indeksirani
- postaviti indekse nad najčešće korištenim poljima, tj poljima koja se koriste za grupiranje, sortiranje ili selekciju uz uvjet (ubrzanje pretraživanja)
- ako su ključevi složeni pretraga će raditi brzo uglavnom po prvom polju
 - po potrebi postaviti dodatne indekse nad preostalim poljima
- ukloniti indekse prigodom masovnih obrada podataka, pa ih kasnije vratiti

□ Ubrzanje i optimizacija upita

- forsira redoslijed spajanja naveden u upitu
 - primjer: SELECT ... OPTION (FORCE ORDER)
- primoravanje nekorištenja indeksa primjenom neškodljive funkcije nad poljem nad kojim se uobičajeno koristi indeks
 - primjer: WHERE NULLIF (polje , "") = uvjet
- optimizacija za određeni broj prvih zapisa
 - primjer: SELECT ... OPTION (FAST n_rows)
- izbjegavanje "ILI" uvjeta



Uklanjanje nul-vrijednosti

Indeksi ne pomažu u slučaju nul-vrijednosti

- opisna polja - zahtijevana vrijednost + pretpostavljena vrijednost

NULL strani ključevi - vanjska spajanja koja ne koriste indekse

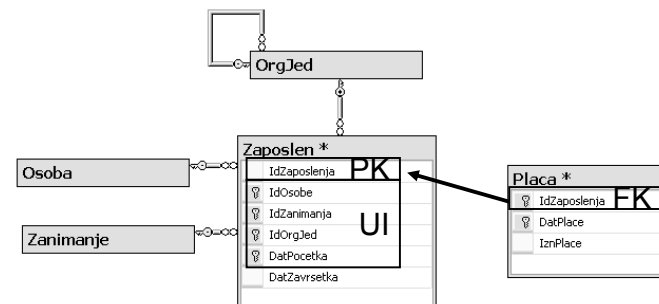
- strane ključeve postaviti na NOT NULL
- definirati posebne vrijednosti u šifranicima
- primjer: 0-nepoznata vrijednost, 999-nepostojeća vrijednost



Nadomjesni ključevi (pojednostavnjenje veza)

Nadomjesni (surogatni) ključevi

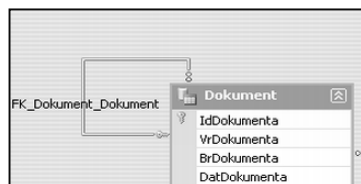
- ispred ključa složenog od većeg broja atributa (npr. ≥ 3) umeće ključ sa samopovećavajućim vrijednostima (serial), a na originalni ključ postavlja se jednoznačan (unique) kompozitni indeks
- teorijski se ne preporuča za asocijativne entitete, koji nasljeđuju ključeve svojih roditelja, jer se time gubi smisao identifikacijske veze
- praktično, nadomjestak treba ugraditi kada je tablica u koju se ugrađuje referencirana iz drugih tablica (npr. Placa referencira Zaposlen)



Nadomjesni ključevi (aplikacijski razlozi)

Problem odabira vrijednosti složenog stranog ključa u padajućim listama koje evidentiraju samo jednu vrijednost odabranog retka

Primjer: Prethodni dokument (RPPP)




"Čisti" dizajn

Takozvani "čisti" dizajn – dosljedna ugradnja nadomjesnog ključa sa samopovećavajućim vrijednostima u sve tablice baze podataka

- Prednost:
 - pojednostavnjenje ugradnje
- Nedostatci:
 - umnožavanje ključeva (nadomjesni primarni, originalni alternativni)
 - nepreglednost izvornog stranog ključa
 - primjer: umetanje Valuta.IdValute (serial 876) ispred OznValute (string-HRK)
 - za posljedicu ima, npr. Dokument.IdValute umjesto Dokument.OznValute



Uvođenje zalihosti

- ❑ **Dodavanje atributa za vrijednosti koje se daju izračunati iz drugih**
 - atribut pohrane za izvedenu vrijednost koja se može izračunati agregatnom funkcijom, na primjer:
 - iznos dokumenta kao suma iznosa stavki
 - oznaka zbirnog stanja kada se vrijednosti pojedinih stanja nalaze u tablici s velikim brojem zapisa (stanje skladišta, saldo na računu)

- ❑ **Primjer: iznos dokumenta**

- aplikacijski, pisanjem programskog koda za izračun (pogledati RPPP)
- pohranjenom procedurom, po potrebi uz IdDokumenta kao argument uz:
 - WHERE Dokument.IdDokumenta = @IdDokumenta
- okidačem tablice Stavka uz:
 - WHERE Dokument.IdDokumenta IN Select (IdDokumenta FROM inserted)

```
UPDATE Dokument SET IznosDokumenta =  
(SELECT SUM(KolArtikla*JedCijArtikla*(100-PostoRabat)/100)  
* (1+Dokument.PostoPorez)  
FROM Stavka WHERE Stavka.IdDokumenta=Dokument.IdDokumenta)
```

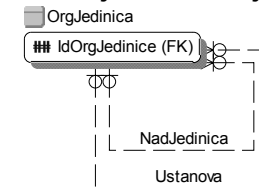


Uvođenje zalihosti (2)

- ❑ **Redundantna vrijednost koja se inače dohvaća složenim i/ili sporim upitima**
- ❑ **Primjer, osoba s identifikatorom zadnjeg zaposlenja**
 - može naravno i drugim operatorima, npr. >= ALL

```
SELECT Osoba.IdOsobe,  
       IdZaposlen = (SELECT IdZaposlen FROM Zaposlen  
                     WHERE Zaposlen.IdOsobe=Osoba.IdOsobe  
                     AND DatPocetka = (SELECT Max(DatPocetka)  
                                         FROM Zaposlen AS Z  
                                         WHERE Z.IdOsobe=Osoba.IdOsobe) )  
FROM Osoba
```

- ❑ **Primjer: redundantni strani ključ na vrh hijerarhije**



Udruživanje i razdvajanje tablica

- ❑ **Pogledati foliju "Balansirane i nebalansirane binarne veze"**
- ❑ **Udruživanje tablica (Combined tables) – najčešće onih u vezi 1:1**
 - Baza Firma: Partner+Tvrtka+Osoba
 - Model RIS: Proizvod+Igla+Avion

Meta-modeliranje

Rječnik podataka

Meta-baza

Primjena meta-modela u izradi aplikacija



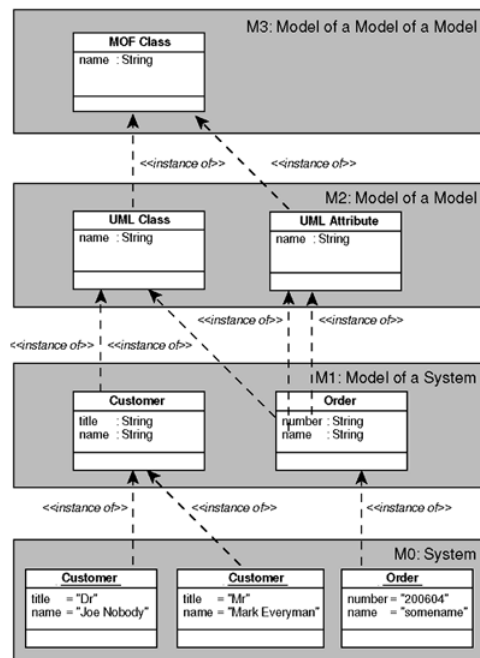
Meta-modeliranje

Meta-modeliranje = modeliranje podataka o podacima

- oznaka "meta" označava višu razinu apstrakcije
- metapodaci su podaci o podacima
- metamodel je model za opisivanje drugih modela

Primjene

- Definiranje i održavanje baze podataka
- Generiranje aplikacija, generiranje podataka, generiranje meta podataka, ...
- Dinamička prilagodba aplikacije podacima, supstitucija sličnih tablica, ...



Pohrana meta-modela

Meta-model – model modela

Meta-baza – baza podataka za pohranu modela

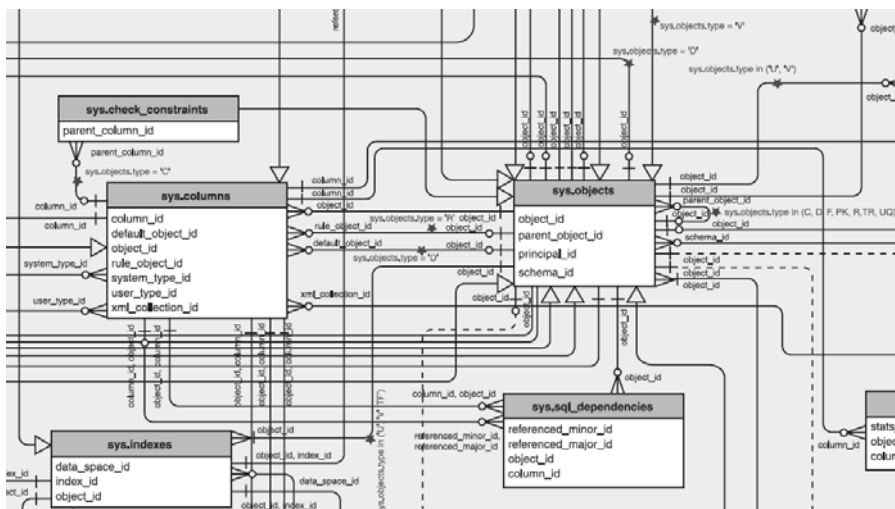
- rječnik podataka, repozitorij, riznica

Rječnik podataka – dio SUBP s podacima potrebnim za upravljanje bazama podataka, kao što su definicija strukture, evidencija dozvola, ...



Schema rječnika SUBP

Primjer: Resursi\SQL2005_Sys_Views.pdf



Sistemske tablice i objekti

Sistemske tablice

- objects
- columns
- indexes
- constraints
- ...

Pohranjene procedure za rad s rječnikom

- sp_Help
- ...

Vrste objekata

```
SELECT name AS 'ObjectName'
, 'ObjectType' = CASE xtype
WHEN 'C' THEN 'CHECK'
WHEN 'D' THEN 'DEFAULT'
WHEN 'F' THEN 'FOREIGNKEY'
WHEN 'L' THEN 'Log'
WHEN 'FN' THEN 'ScalarFunction'
WHEN 'IF' THEN 'InlineTableFunction'
WHEN 'P' THEN 'StoredProcedure'
WHEN 'PK' THEN 'PRIMARYKEY'
WHEN 'RF' THEN 'ReplicationFilterStoredProcedure'
WHEN 'S' THEN 'SystemTable'
WHEN 'TF' THEN 'TableFunction'
WHEN 'TR' THEN 'Trigger'
WHEN 'U' THEN 'UserTable'
WHEN 'UQ' THEN 'UNIQUEConstraint'
WHEN 'V' THEN 'View'
WHEN 'X' THEN 'ExtendedStoredProcedure'
END
FROM sysobjects
ORDER BY xtype, name;
```



Korištenje rječnika

□ Primjer:

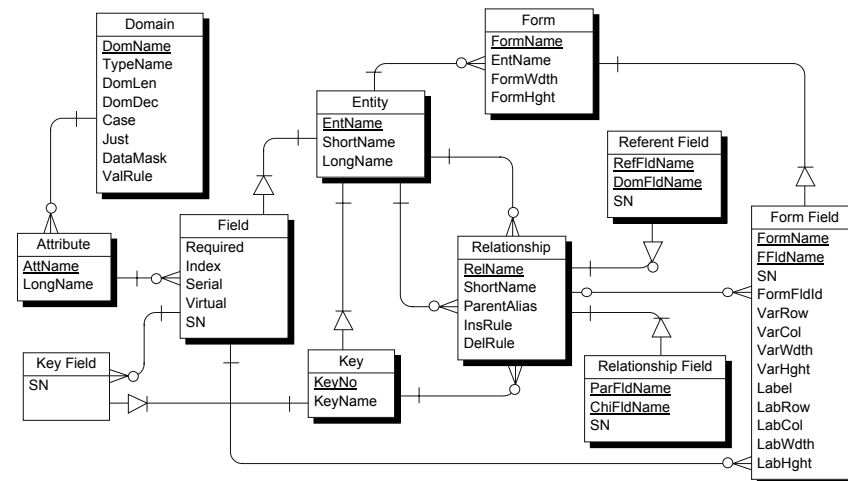
```
CREATE PROCEDURE ap_CountALL
AS

DECLARE @tablename varchar(30)
DECLARE @sql varchar(75)
DECLARE cu_names CURSOR FOR SELECT name FROM sysobjects
WHERE type = 'U'
OPEN cu_names
FETCH NEXT FROM cu_names INTO @tablename
WHILE (@@fetch_status <> -1)
BEGIN
    IF (@@fetch_status <> -2)
    BEGIN
        SELECT @sql = 'SELECT ''' + @tablename
        + ''', COUNT(*) FROM ' + @tablename
        EXEC (@sql)
    END
    FETCH NEXT FROM cu_names INTO @tablename
END
DEALLOCATE cu_names
GO
```

	(No column name)	(No column name)
1	Mjesto	9120
	(No column name)	(No column name)
1	Partner	554
	(No column name)	(No column name)
1	Osoba	63
	(No column name)	(No column name)
1	Tvrtka	491
	(No column name)	(No column name)
1	sysdiagrams	1
	(No column name)	(No column name)
1	Dokument	857
	(No column name)	(No column name)
1	Artikl	1531
	(No column name)	(No column name)
1	Drzava	239



Model meta-baze generatora aplikacija



Primjer generiranja programskog koda

```
ForEach Ent -> sEnt
<<
CREATE TABLE >> Print sEnt.EntName << (>>

ForEach Fld
Where Fld.EntName=sEnt.EntName -> sFld
<<
>> Print sFld.Fldname, " ";
...
```



```
CREATE TABLE Drzava
(
    Ozndrz CHAR(2) NOT NULL,
    NazDrz CHAR(25) NOT NULL,

    PRIMARY KEY (OznDrz) CONSTRAINT pk_drzava
);
...
```



Meta-modeliranje aplikacije

□ Prednosti

- Pojednostavnjivanje modela
- Smanjenje broja objekata

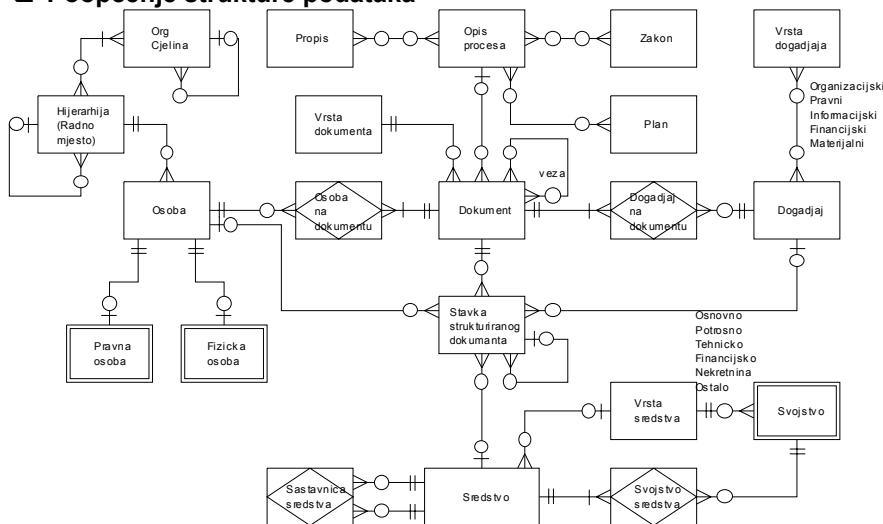
□ Nedostatci

- Otežano razumijevanje
- Problem preslikavanja realnih objekata u objekte metamodela



Ideja aplikacijskog meta-modeliranja (1)

□ Poopćenje strukture podataka



Ideja aplikacijskog meta-modeliranja (2)

□ Primjer, narudžba

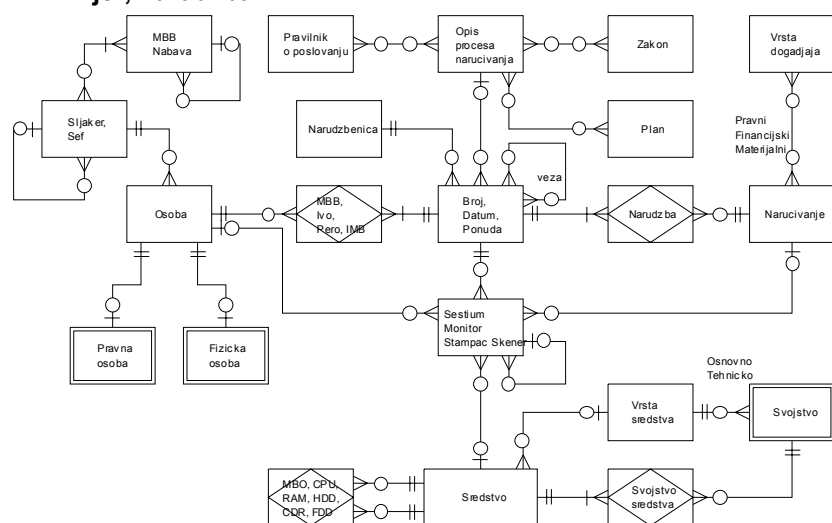
Narudžba MBB-Nabava za osnovno tehničko sredstvo prema poduzeću IMB
 Br. xxxx od xx.xx.200x.
 Veza: Ponuda poduzeća IMB br. xxxx od xx.xx.200x.
 Narudžbu izradio: Pero
 Odobrio njegov šef: Ivo
 Stavke:

Računalo IMB Šestium:	xx kom
MBO, CPU, RAM, HDD, CDR, FDD, CASE	
Monitor 15"	xx kom
Štampač	xx kom
Skener	xx kom



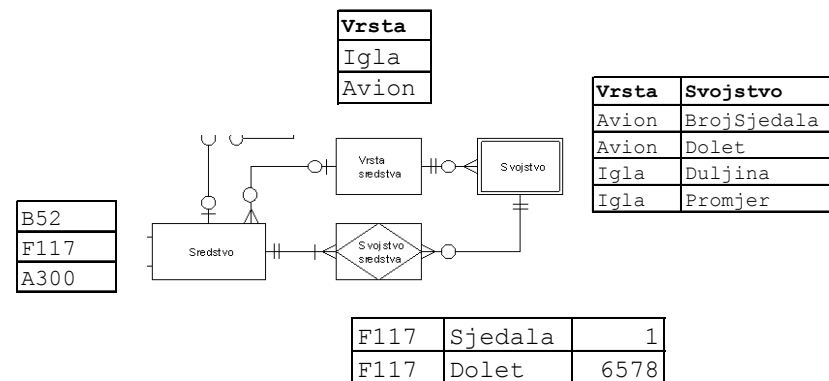
Ideja aplikacijskog meta-modeliranja (3)

□ Primjer, narudžba



Primjena na opisivanje podataka

□ Primjer: opisivanje podataka



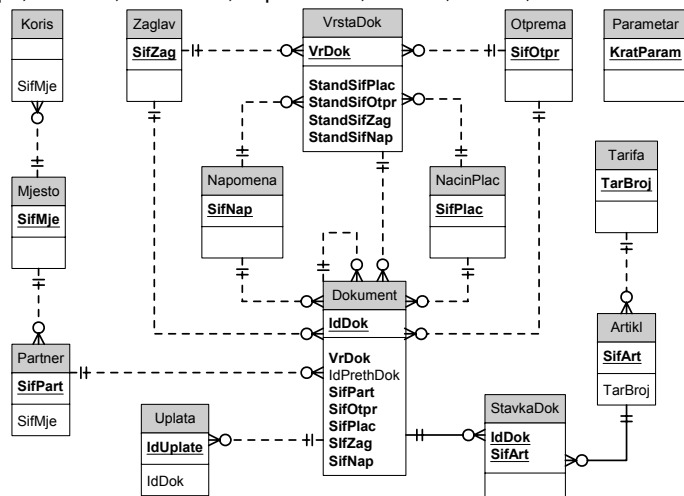
□ Što s tipovima vrijednosti ?



Praktična primjena meta-modeliranja

☐ **Supstitucija većeg broja tablica jednake ili slične strukture**

- Upit, Ponuda, Narudžba, Otpremnica, Primka, Račun, ... sa stavkama



Primjer meta-podataka

❑ Definiranje dokumenata, modeliranje poslovnih pravila

Vrsta	Naziv	Naziv opcije	Mat.pred	Fin.pred	Stand.plac	Stand.otpr	Stand.zag	Stand.nap
S	Korekcija salda	Korekcije &salda	0	1	1	2	13	10
K	Korekcija skladišta	&Korekcije	1	0	1	2	13	10
N	Narudžba	&Narudžbe	0	0	1	2	13	10
I	Otpremnica	O&tpremnice	-1	0	1	2	13	10
O	Otpremnica-Račun	&Otpremnice-Računi	-1	1	1	2	13	10
D	Predračun	Pre&dračuni	0	0	1	2	13	10
P	Primka	&Primke	1	0	1	2	13	10
R	Račun	&lazni računi	-1	1	2	2	13	10
U	Ulazni račun	&Ulazni računi	0	-1	1	2	13	10
		Šifra zag	Tekst zauplavlja					

Sifra zag	Tekst zaglavlja
13	
18	Na temelju Vaše narudžbenice zaračunavamo Vam sljedeće:
20	Na temelju našeg ugovora o kupoprodaji video kasetna naručujemo:
25	Na temelju našeg dogovora o prodaji knjiga, molimo da nam pošaljete:
26	Na temelju Vaše pismene narudžbe zaračunavamo Vam sljedeće:
28	Na temelju našeg dogovora o tisku radnih listova, šaljemo Vam narudžbu:
28	Na temelju Vaše tel. narudžbe zaračunavamo Vam:
29	Na temelju Vaše usmene narudžbe zaračunavamo Vam:
30	Na temelju našeg obračuna prodaje radnih listova molimo Vas da nam fakturirate:
31	KOMISIJONA PRODAJA
32	Na temelju Vašeg izvještaja o komisijonoj prodaji zaračunavamo Vam:
33	Na temelju Vaše telefonske narudžbe šaljemo Vam uzorke:
34	Reklamni uzorci
35	Zamjena knjiga:
36	Na temelju našeg dogovora o kompenzaciji šaljemo Vam račun:
37	Zamjena kasetna:

Primjer aplikacije nad meta-modelom

- ❑ **Primjer, prilagodba funkcionalnosti**

Dokument

Partner: Brijuni Datum: 09.09.96 Pridobilo: Brijuni

Datum valute: 09.09.96 Dana valute: 12 10 % Rabat Račun PP: 432.00 Datum storna:

Otprema: Plovec: 18.00 Datum aktualiziranja:

Datal lokalizacija: u postrojima Cijena: 18.00 Datum aktualiziranja: 09.09.96

Slovin: kupa Cijena: 18.00 Datum ispis: 09.09.96

Zagljebe: Cijena: 0.00 Datum uplate:

ESC

Sifra Naziv	Kolonina	Cijena	Iznos	Pribitak	Pribitak	PPF	Plovec	Ukupno
1 Paketi doli zrel	8.00	20.00	160.00	10	16.00	0	0.00	144.00
2 Paketi mliho novo	8.00	20.00	160.00	10	16.00	0	0.00	144.00
3 Ukupno uplate	8.00	20.00	160.00	10	16.00	0	0.00	144.00

Stavka: od 480.00 48.00 0.00

Dokument: od Brijuni 0 Brijuni Update Zloimo

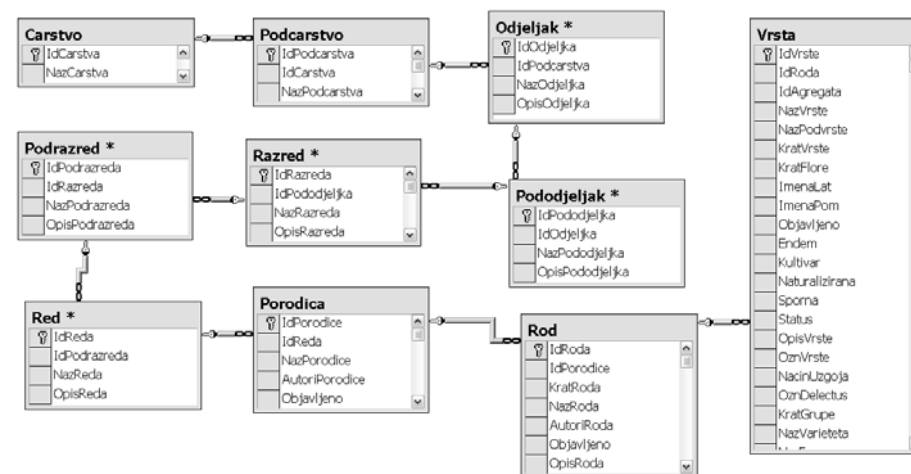
F3-Brisanje F5-Tablica F7-Ispis F8-Ucina F9-Duhvat F10-Ispis ESC-Kraj

□ **Primjer:**
pojednostavnjenje
izračuna

```
SELECT Artikl.SifArt, Artikl.NazArt,
SUM(StavkaDok.Kolicina*VrstaDok.PredMat)
FROM Dokument INNER JOIN VrstaDok ON Dokument.VrDok=VrstaDok.VrDok
INNER JOIN StavkaDok ON StavkaDok.IdDok = Dokument.IdDok
INNER JOIN Artikl ON StavkaDok.SifArt = Artikl.SifArt
WHERE VrstaDok.PredMat <> 0
AND Artikl.IdeSklad <> 0
AND Dokument.DatSklad IS NOT NULL
AND Dokument.DatStorno IS NULL
GROUP BY Artikl.SifArt, Artikl.NazArt
```

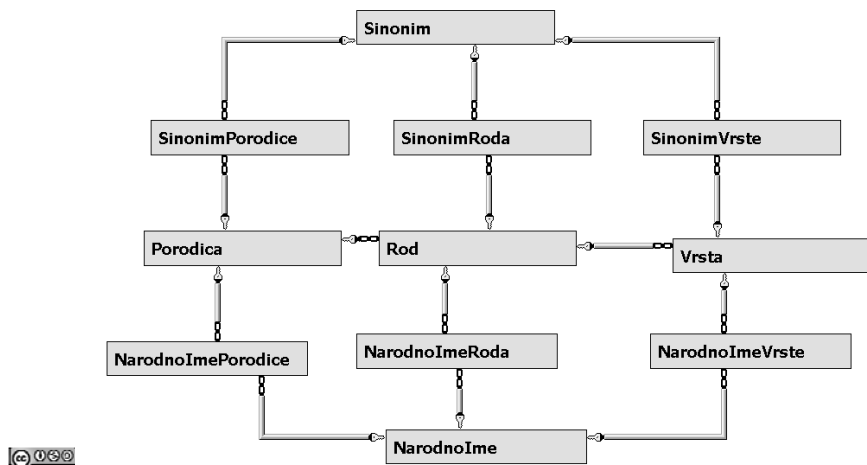
Primjer: kaskada sličnih entiteta

- ☐ Svi imaju identifikator i naziv, neki imaju opis i objavljeno ...
- ☐ Moguća rješenje: generalizacija s 9 specijalizacija



Primjer: kaskada sličnih entiteta (2)

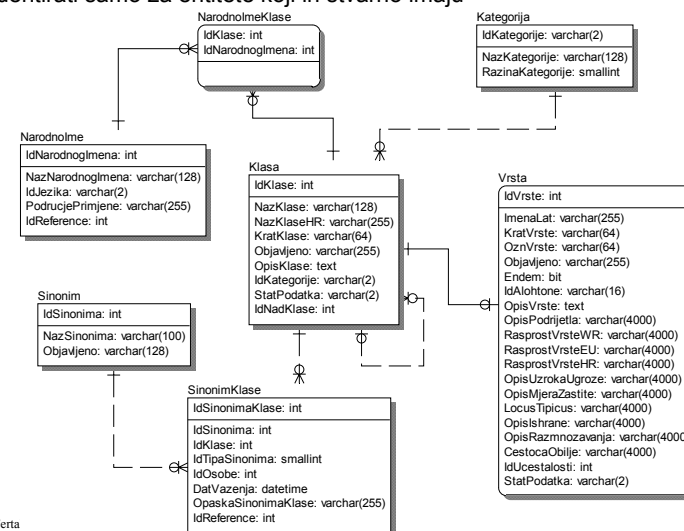
- ☐ Neki entiteti međutim imaju i spojne tablice na narodna imena i sinonime, približno jednake strukture
- ☐ Rješenje: još tri specijalizacije ?



381

Primjer: kaskada nakon modeliranja

- Zadržana rekurzivna hijerarhija, izjednačene strukture, jednom specijalizacijom te spojnim entitetima na narodna imena i sinonime koji će se evidentirati samo za entitete koji ih stvarno imaju




382



Zadaci

- ❑ **Zadaci za razgibavanje**
 - ER model videoteke (Član, Posudba, Rezervacija, Video)
 - Na posudbu se plaća zakasnina
 - ER model cjenika u restoranu brze prehrane
 - Različite cijene samostalnog proizvoda i proizvoda u paketu
- ❑ **Domaća zadaća**
 - Čime crtati ?

[neke] Reference

- ❑ **Great News - The Relational Model is Dead**
 - <http://www.wiscorp.com/SQLStandards.html>
 - <http://www.tdan.com/view-articles/5093>
- ❑ **A UML Profile for Data Modeling**
 - <http://www.agiledata.org/essays/umlDataModelingProfile.html>
- ❑ **Mapping Objects to Relational Databases**
 - <http://www.ambysoft.com/essays/mappingObjects.html>
- ❑ **Mapping Objects to Relational Databases: O/R Mapping In Detail**
 - <http://www.agiledata.org/essays/mappingObjects.html>
- ❑ **Entity Relationship Modeling with UML**
 - http://www.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_u ml.pdf
- ❑ **A Repository Model - Business Rules**
 - <http://www.tdan.com/view-articles/4978/>
- ❑ **Why (not) ORM?**
 -  Modeliranje\IT\2006 UMLORM PID236496.pdf
- ❑ **OMG's MetaObject Facility**
 - <http://www.omg.org/mof>