

Osnovni pojmovi

2013/14.01

Podatak, informacija i znanje

□ Podatak

- podatak je sirova činjenica koja predstavlja neku istinu iz stvarnog svijeta
- pojedinačni podaci sami za sebe znače malo ili nemaju neko značenje

□ Informacija

- obavijest, predana novost ili znanje
- interpretacija podatka - pročišćen, organiziran i obrađen podatak u smislenom kontekstu
 - informacija je subjektivnog značenja, u kontekstu primatelja

□ Znanje

- gradi se vezivanjem novih informacija na postojeće znanje
- isti podaci mogu biti različito interpretirani ovisno o znanju primatelja
- primjer:

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

□ Informacijska 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)
- **Apstraktни 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

- nerazdvojivi 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 ga se tiče

- npr. MZOŠ i studentske udruge kao okolina visokog učilišta

□ Ulazi i izlazi - sve što ulazi u sustav iz okoline ili ga napušta

- sve ?

□ Sučelja - veze/dirališta između sustava i njegove okoline

- u kom smislu ?

□ Ograničenja – čimbenici koji određuju 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 ...

Poslovni i informacijski sustav

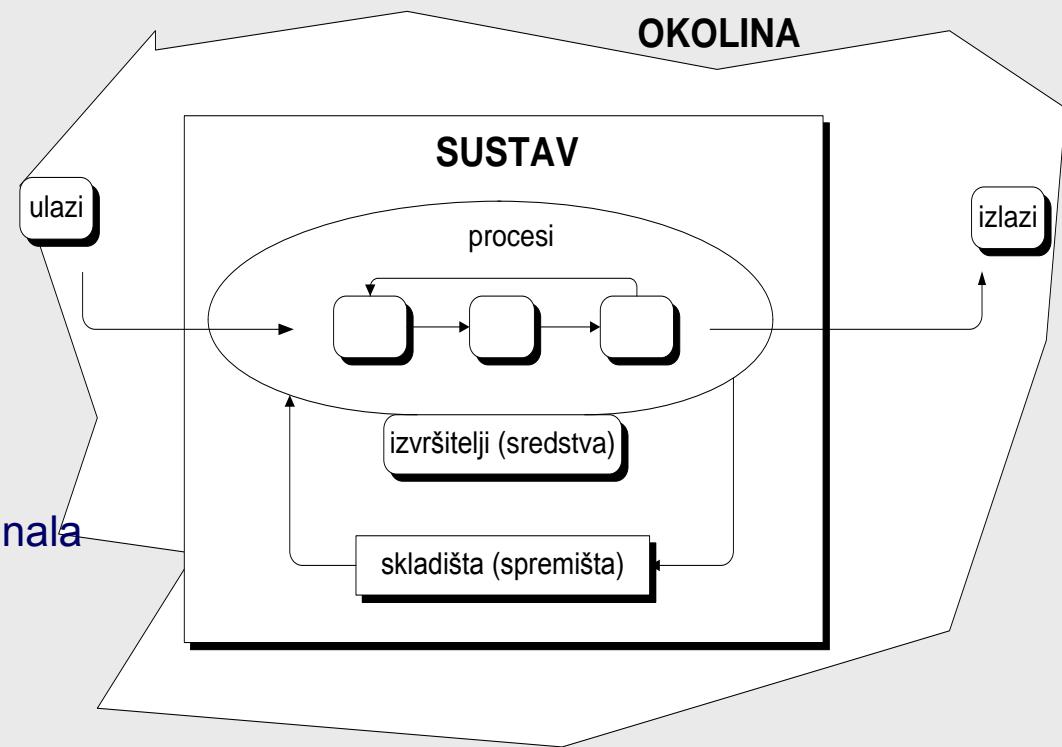
□ Poslovni sustav

- materijalni ulazi i izlazi (sirovine, energija, proizvodi) i informacijski tokovi (poruke, dokumenti, ...)
- procesi: obrada, prerada, proizvodnja
- izvršitelji: osobe, strojevi, alati
- skladišta: spremišta materijala

□ 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)

- 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



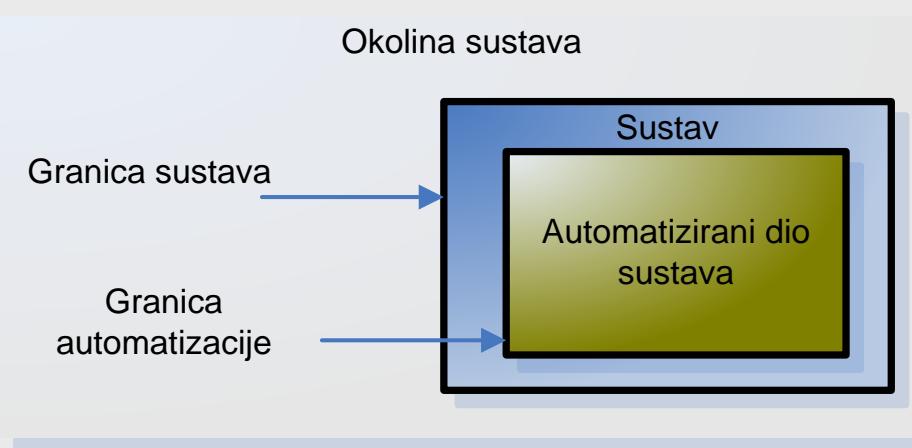
Informacijski sustav podržan računalom

□ 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
- ...



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 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

Sloj IS	Informacije	Učestalost	Korisnici	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 zalihamama.

□ Customer relationship management (CRM)

- upravljanje odnosom s kupcima
- podržava marketing, prodaju i servis, uključujući interakciju s korisnicima

Primjer poslovnog IS

□ 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 nalozima za prodaju.
- Upravljanje ugovorima.
- Fakturiranje.

□ Nabava, skladištenje i logistika

- Zahtjevnice.
- Obrada narudžbenica.
- Evidencija primki materijala.
- Ovjera faktura (likvidacija).
- Upravljanje ugovorima.
- Upravljanje zalihami 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 (nastavak)

□ IS biljnih vrsta

- Taksonomija
- Herbar
- Nalazišta
- Narodna imena
- Sinonimi
- Literatura
- Ekomska 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 (struktурно, 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 su visoko ovisne o stalnoj raspoloživosti (vlasitog) IS!

Projektiranje i izgradnja IS

(ne)Uspješnost informacijskih sustava

- The CHAOS Report , 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%
 - rokova: 222%
- Uspješnost
 - 16.2% uspješnih projekata
 - 52.7% prekoračenje roka/troška ili reducirana funkcionalnost:
 - 31.1% prekinutih projekata
- CHAOS, 2002:
 - 34% uspješnih
 - 49% neuspješnih
 - 17% propalih
- CHAOS, 2009:
 - 32% successful
 - 44% challenged
 - 24% failed
- Čuveni neuspjesi
 - London Ambulance System (1992)
 - Taurus (1993)
 - Denver Airport (1994)

Uzroci neuspjeha

- **Informacije o neuspješnim projektima u RH se rjeđe objavljaju. 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 izvještavanje 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 zloupotreba 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.
 - Aktivna uloga rukovodstva

□ 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 okljevati ako treba revidirati doseg ili otkazati projekt
- Projektirati za rast i promjene

Programsko inženjerstvo

- **Programska potpora, oprema = softver (software)**
 - programi + podaci + dokumentacija
- **Primijenjena programska potpora = aplikacija (application)**
 - namjenski program, primjenska programska potpora
 - računalom podržano rješenje jednog ili više poslovnih problema ili potreba
- **Programsko inženjerstvo (software engineering)**
 - sistematičan, discipliniran i mjerljiv pristup razvoju, primjeni i održavanju softvera = primjena inženjerskog pristupa na softver [IEEE Std 610.12-1990, 1994]
- **Informacijski sustav = sustav aplikacija za upravljanje ljudskim aktivnostima**

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

□ **Metoda (method), metodologija (methodology)**

- smislen i organiziran skup procedura izrade (modela, softvera), uz korištenje dobro definirane notacije
- definira primjenu tehnika i njihovo povezivanje (pr. ERD-DFD)

□ **Tehnika (technique)**

- definira način provođenja postupka i dokumentiranja rezultata tog postupka
- npr. tehnika intervjuiranja, CSF, Delphi, ...

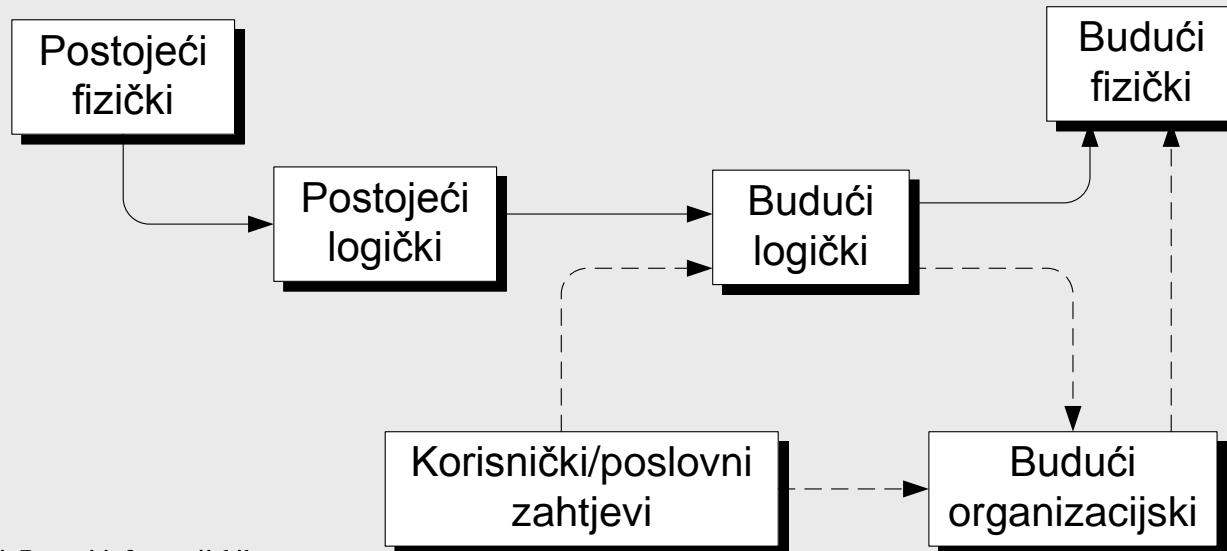
□ **Pomagalo, alat (tool)**

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

Projektiranje IS

❑ slijed izrade fizičkog i logičkog modela postojećeg i budućeg IS

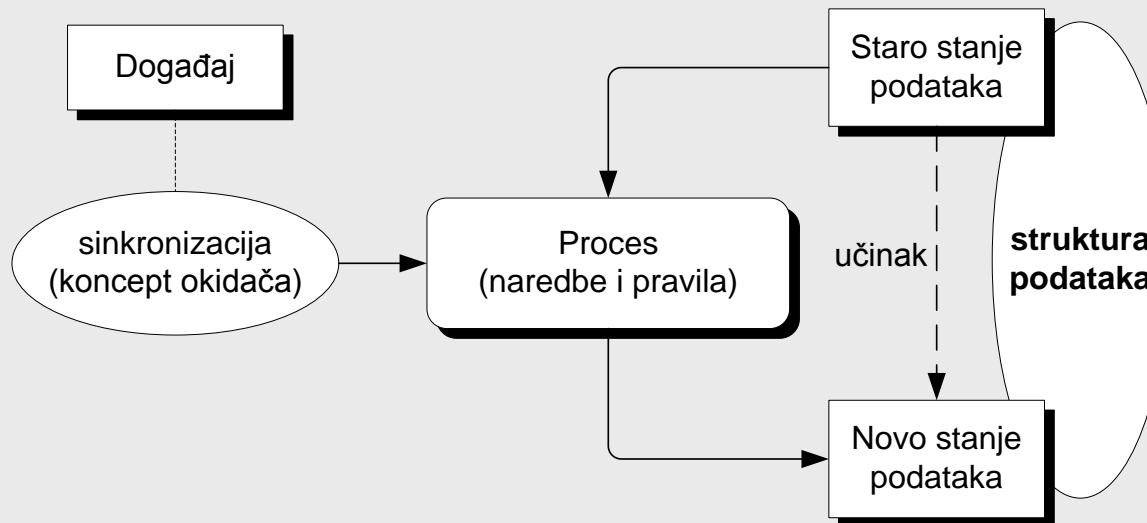
- fizički model (ugradbeni, implementacijski, tehnički)
 - kako je sustav fizički i tehnički ugrađen, te tko, gdje i kada nešto radi
- logički model (esencijalni, konceptualni, poslovni)
 - što je sustav, što radi, što su podaci
- operativni (budući fizički) sustav
 - što, tko i kada, ali ne i gdje
- opcionalno, može se razmatrati organizacijska razina
 - međuzavisnost razvoja organizacije i razvoja sustava



Modeliranje IS

□ Tehnike oblikovanja dijagramima

- podaci (što) – konceptualni (ERD=DEV), logički (relacijski)
- funkcije, procesi (kako) – funkcionalna dekompozicija, dijagram toka podataka
- događaji (kada) – dijagram (promjene) stanja
- objekti, programi, resursi, ... – UML, strukturne karte, mape resursa



□ Izrada modela koji odgovaraju dijelovima stvarnog sustava

- što kada sustav otprije ne postoji ?
- "surogat", preskakanje fizičkog oblikovanja

Ključne aktivnosti

- **Sistemska analiza, Analiza sustava (Systems analysis)**
 - proučavanje poslovanja s ciljem preporuke poboljšanja i specificiranja zahtjeva na rješenje
- **Sistemski dizajn, Dizajn sustava (Systems design)**
 - projektiranje sustava !
 - specifikacija ili konstrukcija računalom podržanog rješenja identificiranih poslovnih zahtjeva
- **SA + SD = informacijsko inženjerstvo**

Ključni sudionici

- **Dionici, interesni sudionici (stakeholders)**
- **Korisnik, naručitelj, klijent (user, customer, client) – osoba ili grupa**
 - Korisnik sustava – krajnji korisnik
 - Vlasnik sustava – naručitelj (stvarni vlasnik ili upravitelj organizacije)
- **Projektant, dizajner sustava (designer)**
 - prevodi poslovne zahtjeve i ograničenja u tehničko rješenje, modeliranjem
- **Graditelj sustava (builder, developer) - razvojnik**
 - izgrađuje, provjerava ispravnost, isporučuje

Ključni sudionici (nastavak)

□ Sistem analitičar (system analyst) [Whitten et. al, 2000]

- proučava problem i poslovne potrebe radi određivanja kako ljudi, podaci, procesi i informacijska tehnologija mogu unaprijediti poslovanje
- "moderni rješavatelj poslovnih problema"
- razumije i poslovanje i tehnologiju
- analitičar/programer, programer/analitičar

□ Poslovni analitičar (business analyst)

- analiza poslovanja, identifikacija koristi, oblikovanje procesa i procedura

[drugi analitičari, npr. infrastrukturni, upravljanja promjenama, ...]

□ Upravitelj projekta

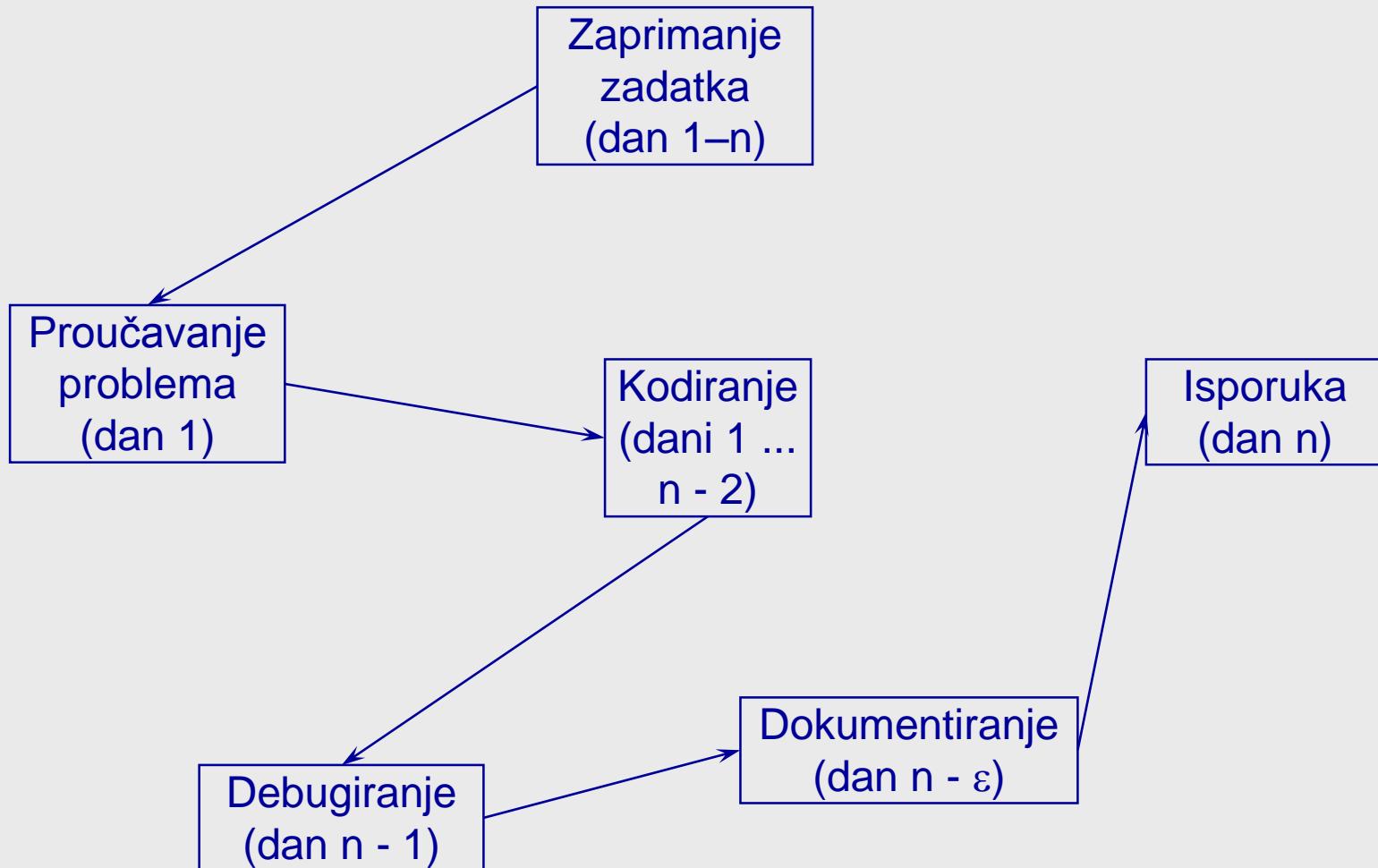
- planiranje, upravljanje ekipom (analitičara, razvojnika, korisnika), nadzor, ...

□ Analitičar ?= konzultant

Životni ciklus razvoja sustava

Systems Development Life-Cycle (SDLC)

Primjer: studentski projekt



Prilagodba prema Modesitt,
LNCS 750, pp. 42

Životni ciklus razvoja

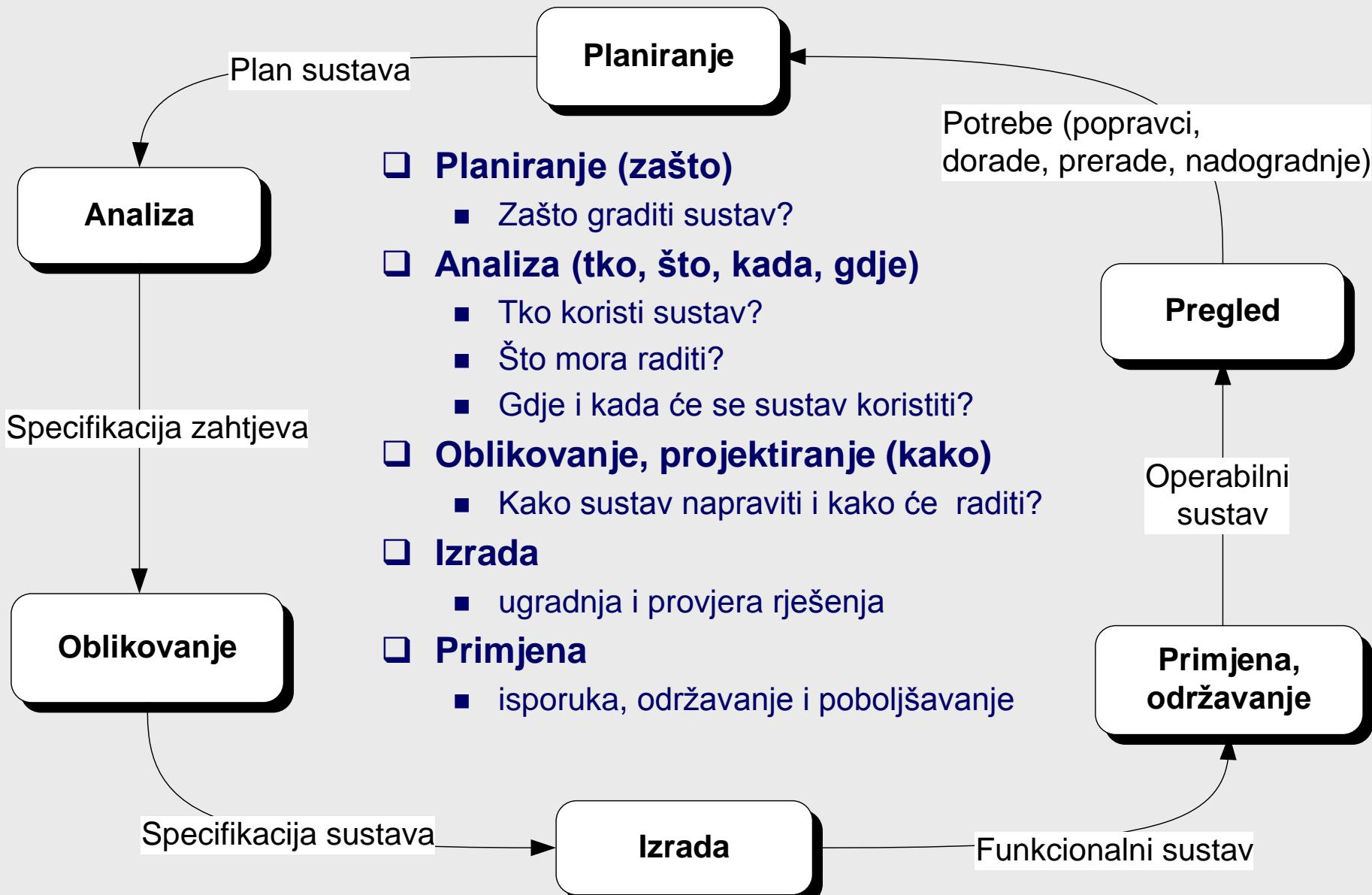
□ Općeniti proces razvoja sustava

- 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 dalnjim koracima

Faze životnog ciklusa



Planiranje sustava

- Zašto graditi sustav ?

□ Inicijacija projekta

- Zahtijevanje sustava (system request), Inicialna strategija (initial strategy)
 - određivanje poslovnih ciljeva, problema i ideja njihovog rješavanja
 - snimka stanja, sažetak poslovnih potreba
- Studija izvedivosti (feasibility study)
 - analiza problemskog područja i određivanje (granica) projekata
 - procjena ključnih aspekata predloženog projekta
 - organizacija, tehnologija, financije, rokovi
- povjerenstvo (steering committee, approval committee) odobrava projekte

□ Upravljanje projektom

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

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

Faza analize

- Tko će koristiti sustav, što će sustav raditi te gdje i kada će biti korišten.

□ Strategija analize (analysis strategy)

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

□ Analiza zahtjeva (requirements analysis)

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

□ Specifikacija zahtjeva (requirements specification)

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

→ Poslovni model sustava, prijedlog sustava

Oblikovanje sustava

- Kako će sustav funkcionirati

□ Strategija projekta (design strategy)

- 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 izvedivosti i plan projekta

Izrada sustava

- 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,
- dokazivanje da sustav radi (da je ispravno napravljen)
- te da radi što je zahtijevano (da je napravljen pravi sustav)

→ Funkcionalni sustav i tehnoški opis sustava

Primjena sustava

□ Uvođenje u primjenu i održavanje

- Uvođenje u primjenu, primjena (operation, production)
 - prijenos (konverzija) podataka, konverzija načina rada
- Održavanje (maintenance)
 - izmjene radi poboljšanja, dorade ili prilagodbe
- Potpora (support)
 - dobavljača opreme, tehničkog osoblja korisnicima

→ Informacijski sustav u primjeni, plan održavanja

□ Pregled (review)

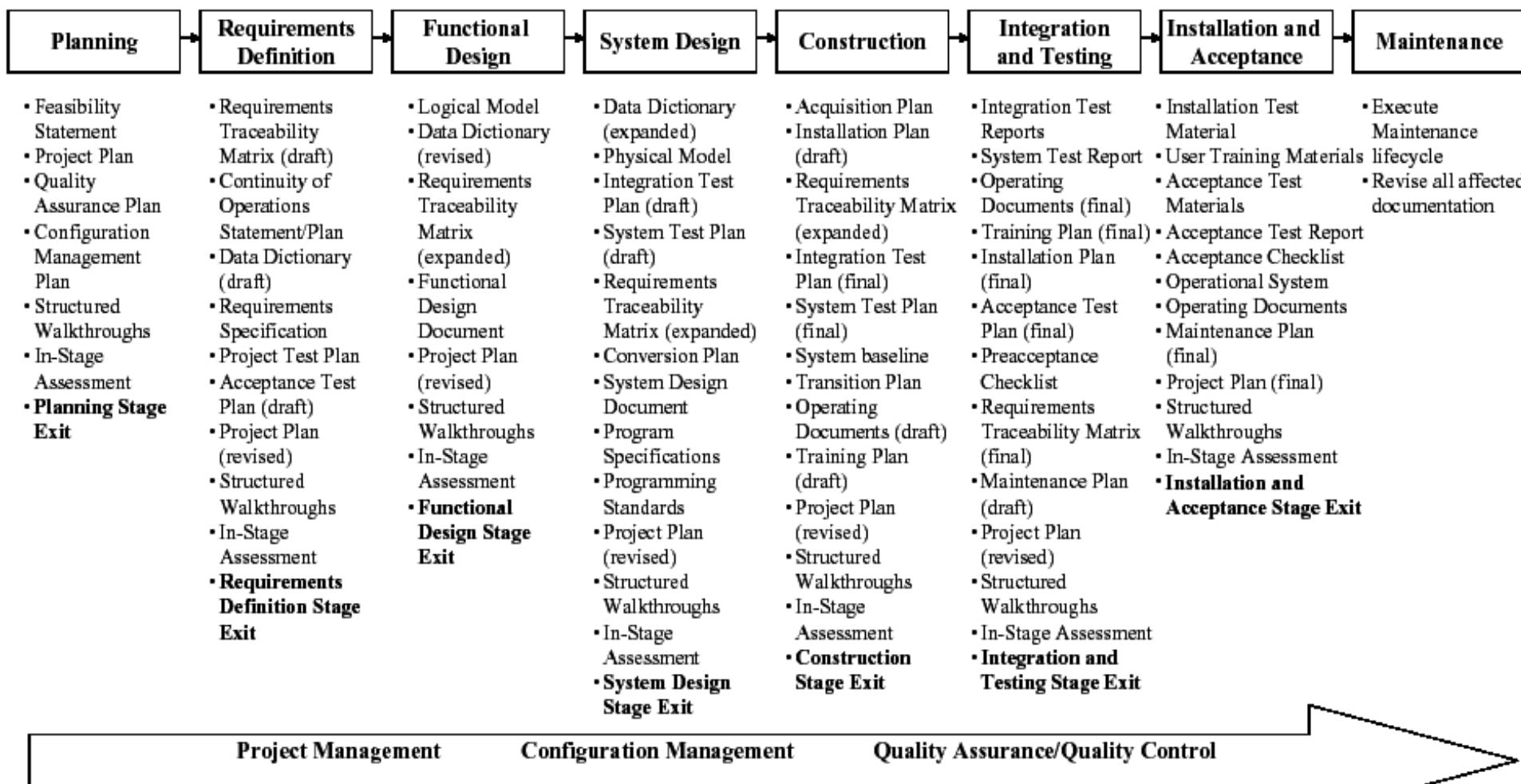
- revizija, preispitivanje čitavog sustava kada su potrebne veće izmjene uslijed promjena u poslovanju ili promjena poslovnih ciljeva

→ 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

Enterprise Information Architecture

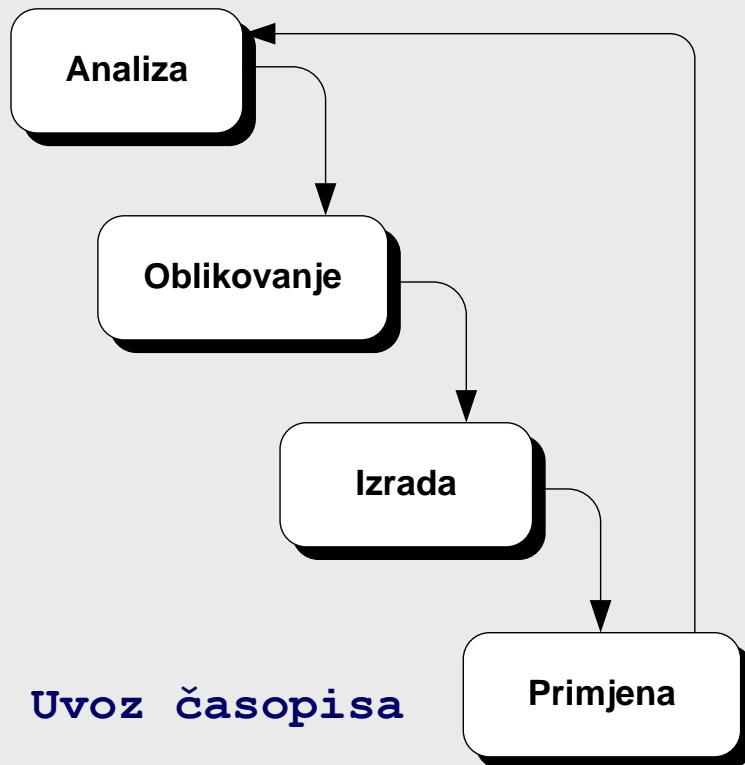


Modeli razvoja

Vodopadni (waterfall) model

□ Klasični vodopadni model

- slijedno napredovanje iz faze u fazu
- nisu dozvoljene naknadne promjene rezultata prethodnih faza
- kvaliteta važnija od troškova i rokova



Uvoz časopisa

□ Primjenjivost

- dobro definirano i stabilno okruženje, uhodane ručne obrade ili računalski sustav koji treba unaprijediti
- veliki projekti (investicije)
- zaposlenici neiskusni ili posjeduju ograničeno tehničko znanje

□ Nedostaci

- uvođenje prema gore (bottom up): moduli, podsustavi, sustav
- sustav nepotrebljiv dok ne bude potpuno dovršen
- predodžba o proizvodu na temelju pisane specifikacije
- korisnici prekasno uoče nedostatke
- problem u slučaju pogrešaka ili novih/promijenjenih zahtjeva

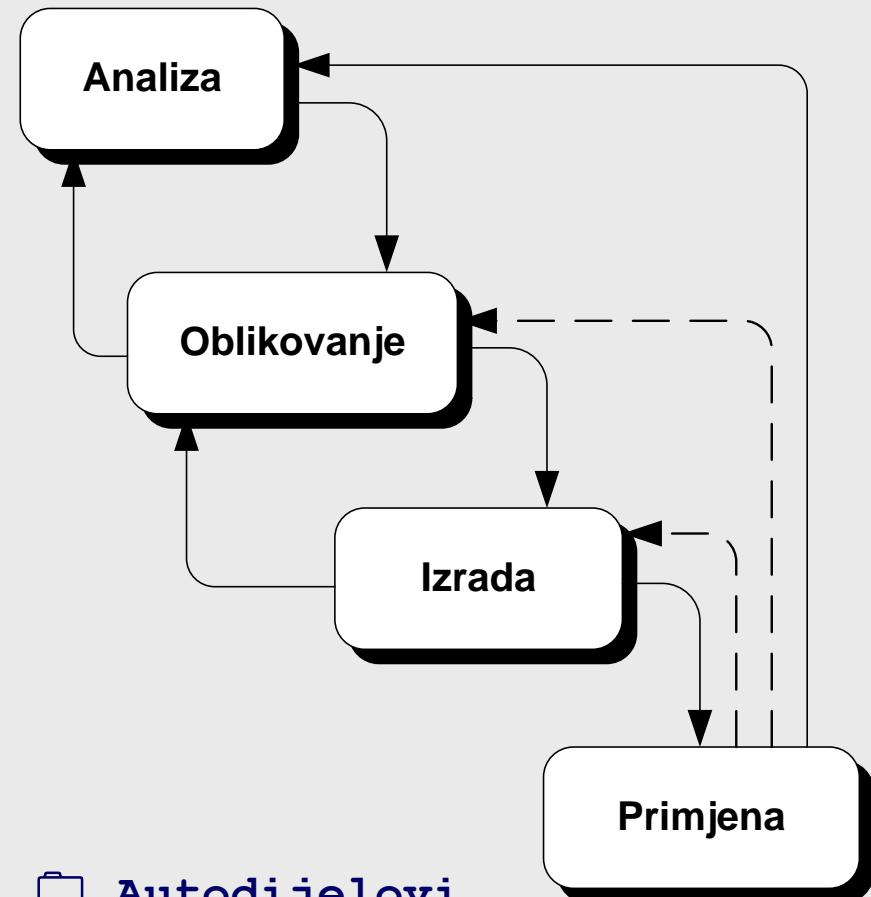
Pseudostruktturni i radikalni vodopadni model

□ Pseudostruktturni model

- povratna veza i mogućnost promjene prethodnih rezultata
- strukturirano programiranje
- uvođenje prema dolje: moduli na višim, pa na nižim razinama

□ Struktturni (radikalni)

- istovremenost različitih faza
- 4GL i generatori aplikacija
- prikladan kada se unaprijed ne zna konačni izgled sustava
- u konačnici mora nastati (papirnati) model sustava

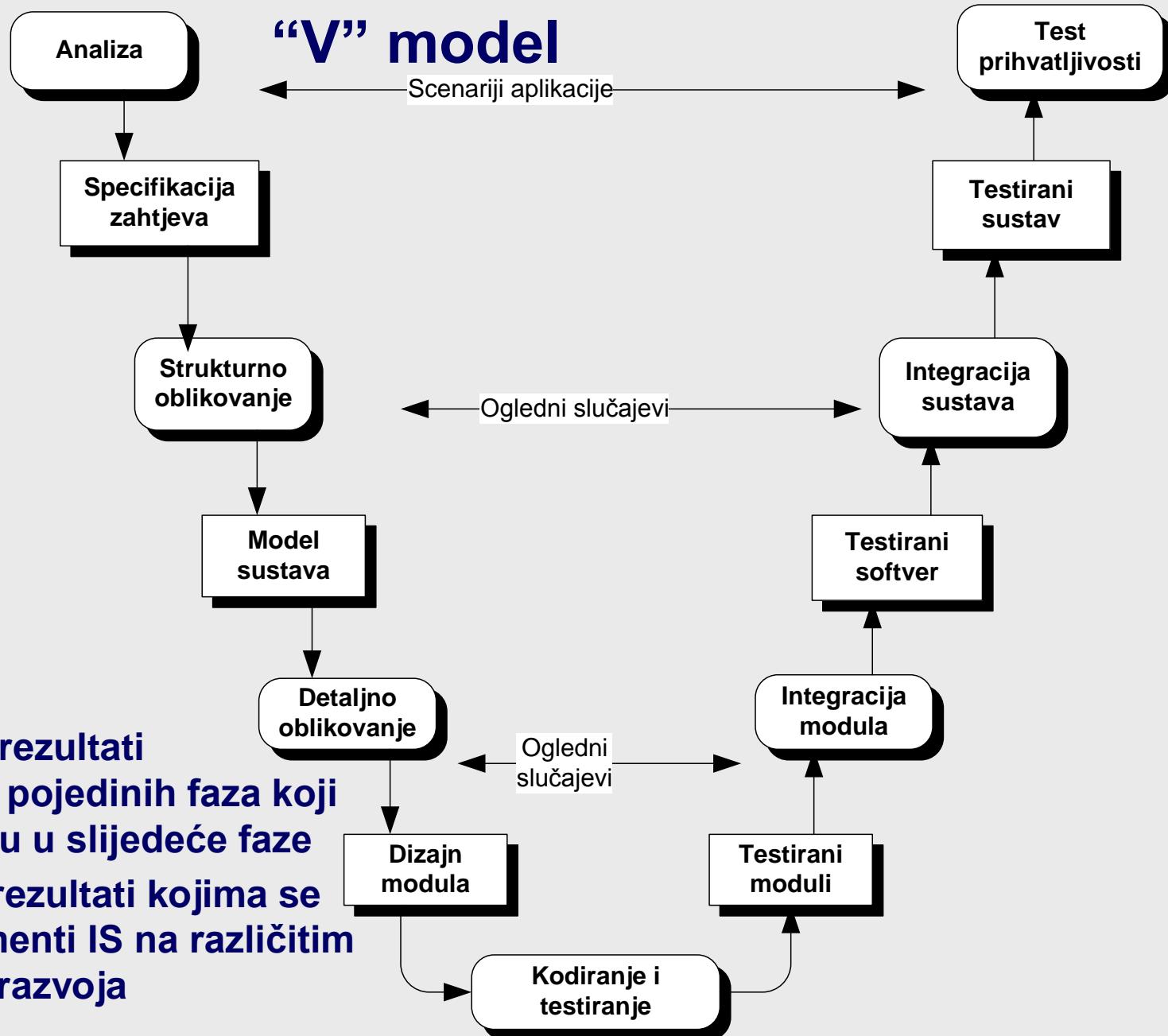


□ Autodijelovi

Validacija

Verifikacija

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



Paralelni razvoj po podprojektima

□ podprojekti koji se oblikuju i izrađuju paralelno

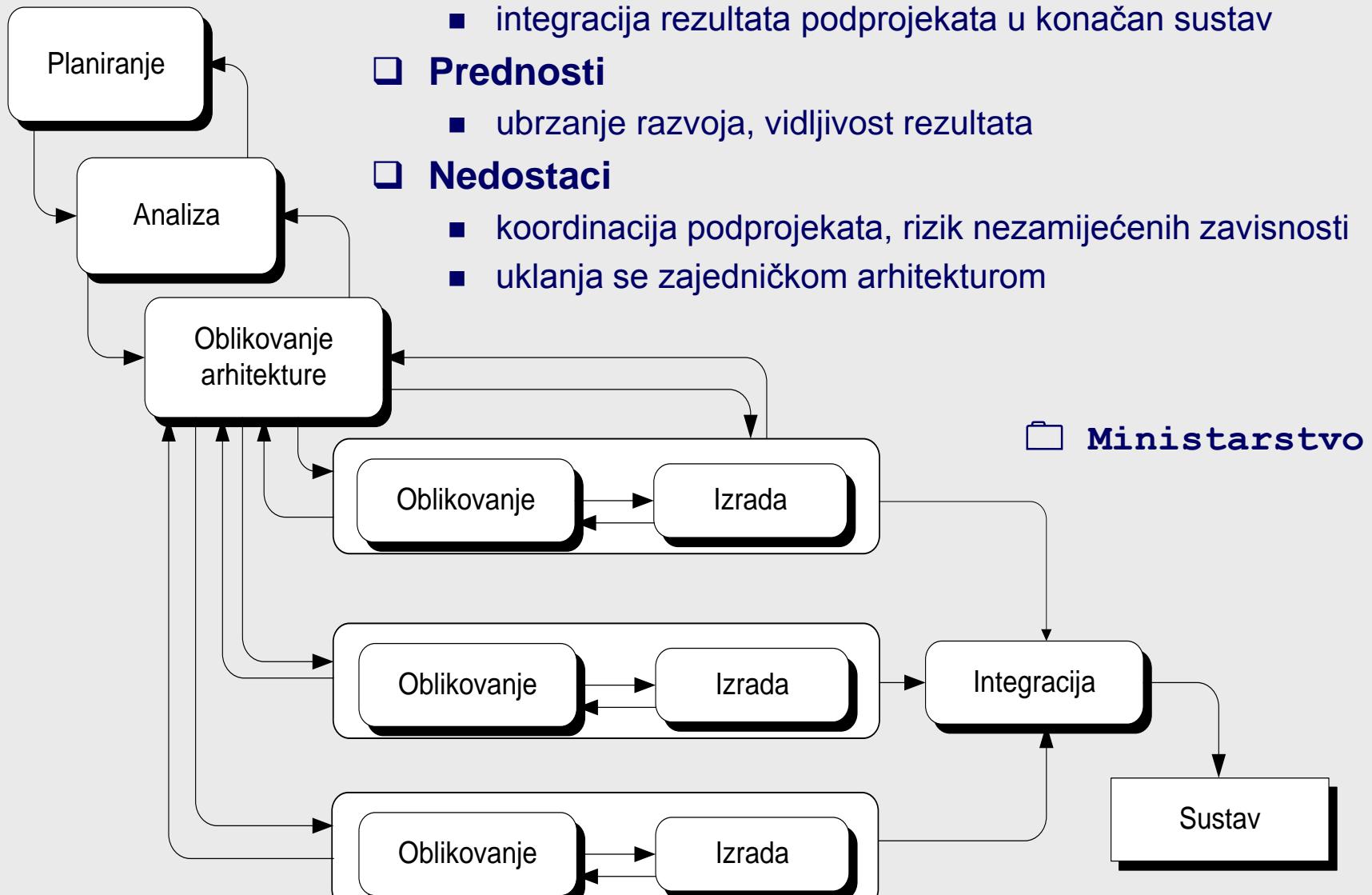
- integracija rezultata podprojekata u konačan sustav

□ Prednosti

- ubrzanje razvoja, vidljivost rezultata

□ Nedostaci

- koordinacija podprojekata, rizik nezamijećenih zavisnosti
- 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 performansi 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)

□ Evolucijsko prototipiranje (sinonim)

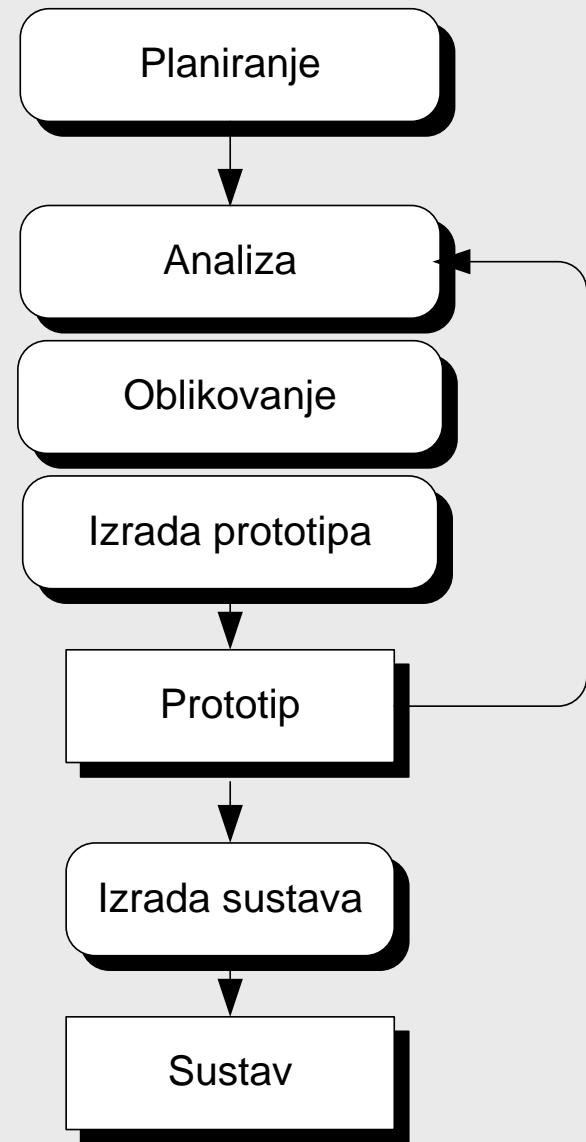
- funkcionalni prototip
- povratna informacija korisnika
- inkrementalna dorada (stepwise refinement)
- na kraju samostalna izrada (bez korisnika)
- mali (*one-man*) projekti s promjenjivim zahtjevima

□ Prednosti

- pokretljivost, prilagodljivost ☺
- bolje određivanje zahtjeva
- vidljivost rezultata

□ Nedostaci

- "zaboravljanje" da prototip nije pravi sustav
- "vječni razvoj"
- izostanak specifikacije, pa druge dokumentacije
- nemogućnost ispravne procjene i planiranja resursa



Ograničeno prototipiranje (constrained prototyping)

□ Nefunkcionalni prototip

- samo prikaz izgleda

□ Prototipiranje - sredstvo određivanja zahtjeva

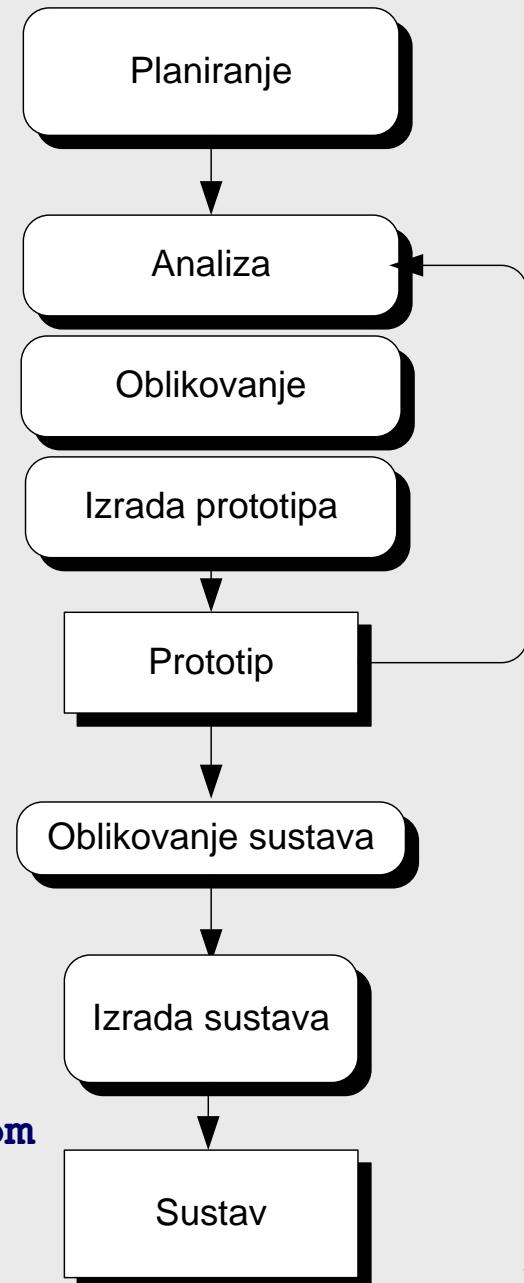
- smanjuje rizik neispunjениh očekivanja
- po određivanju zahtjeva prototip se odbacuje (throwaway prototyping)
- slijedi faza oblikovanja

□ Prednosti

- postoji dokumentacija
- moguće je planiranje

□ Nedostatak

- "bacanje" posla

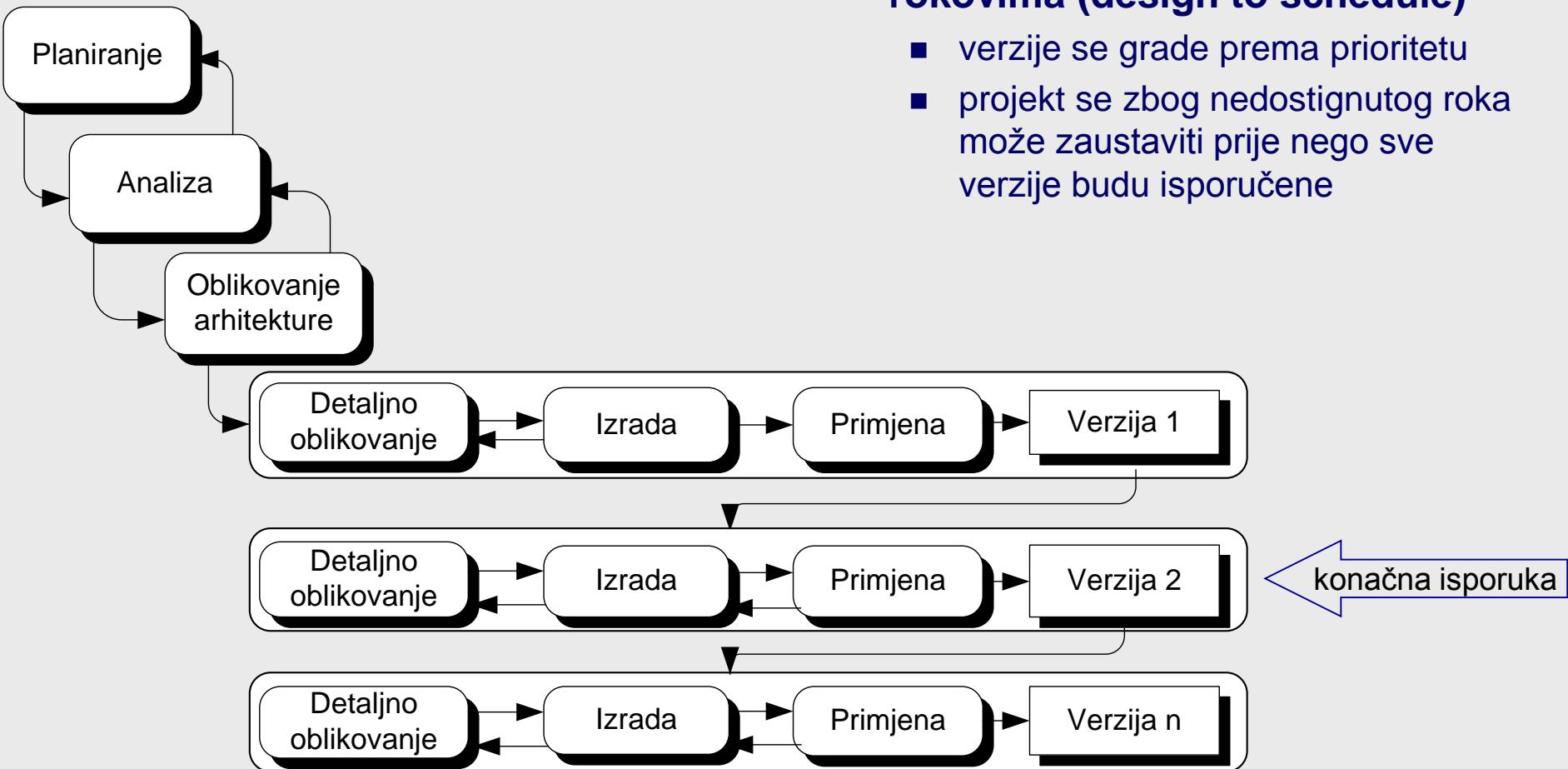


Izdavačka kuća, Telekom

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)
 - slijed verzija koje se isporučuju i generiraju nove zahtjeve
- **Prednosti**
 - isporuka dijelova prije konačnog završetka projekta
 - brža uporaba - ranije vidljiva 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
 - ključne i nužne mogućnosti ugraditi prve!

Evolucijski model, model postupne isporuke

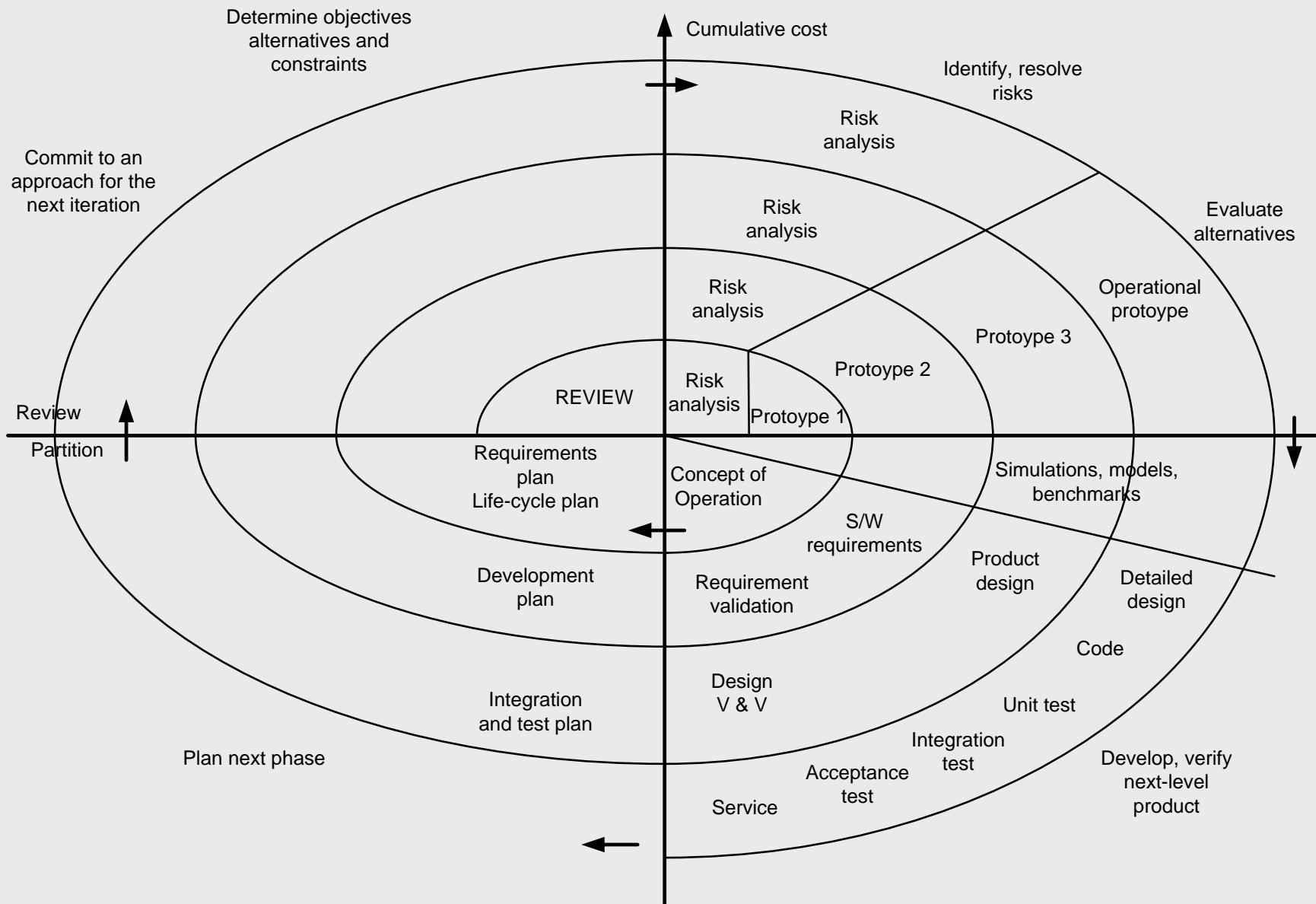


□ **Evidencija biljnih vrsta**

Spiralni model

- **Iterativno se analizira rizik i planira naredna iteracija od 6 koraka:**
 - Određivanje ciljeva, alternativnih rješenja i ograničenja,
 - Određivanje i pronalaženje rješenja za moguće rizike,
 - 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**
 - veliki (skupi) sustavi ili je analiza rizika relativno mali trošak
 - najčešće na interne projekte (naručitelj i izvođač iz iste organizacije)
- **Prednosti**
 - praćenje napretka, sigurnost
- **Nedostaci**
 - složenost, trošak

Spiralni model



Osnovni objektno orijentirani model razvoja

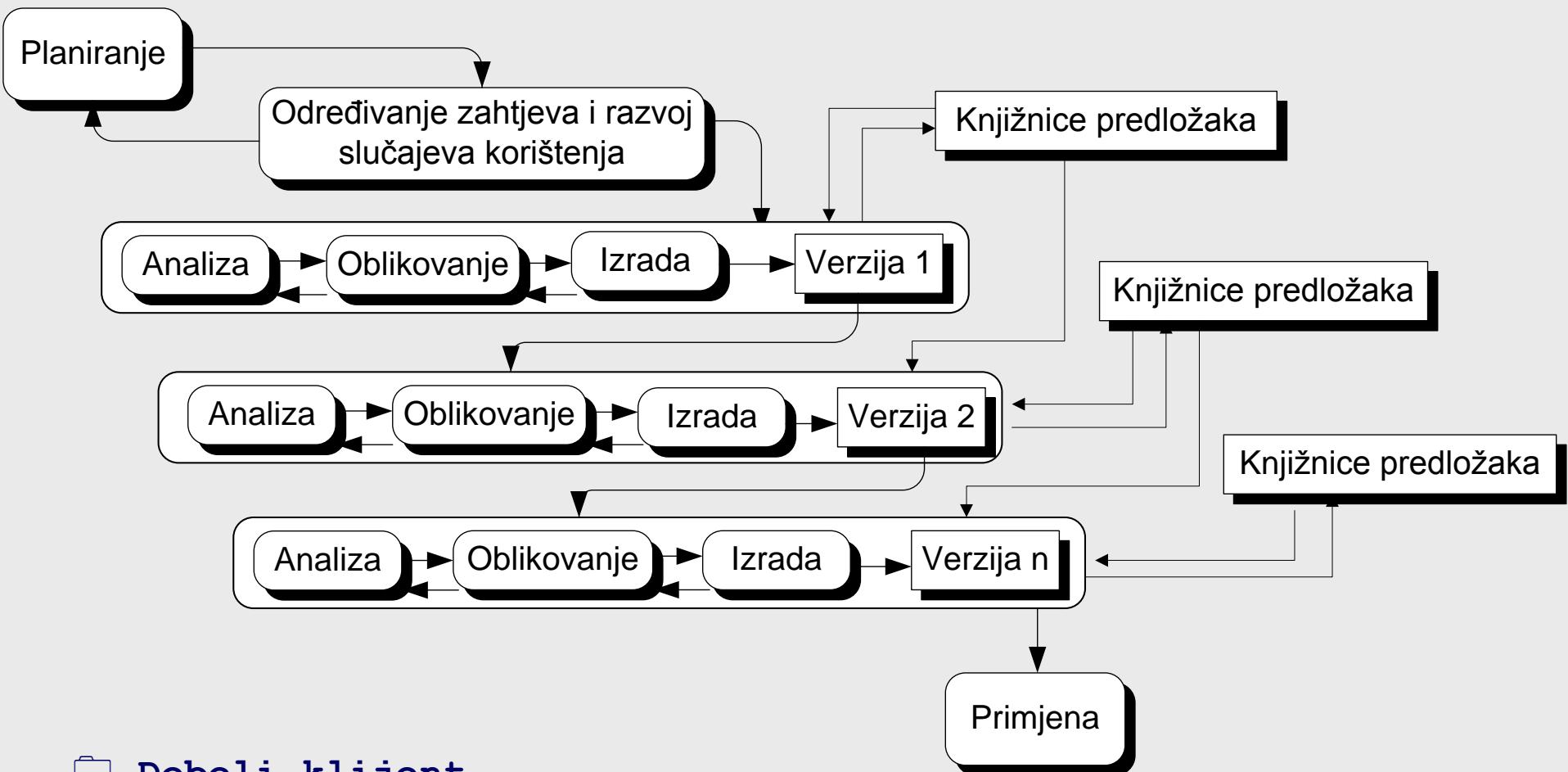
□ Minimalist Object-Oriented Systems Analysis & Design (MOOSAD)

- generički pristup OOSAD, temeljem ujedinjenog procesa (Unified Process) i ekstremnog programiranja (eXtreme Programming)
- po uzoru na fazni razvoj, ključna razlika je u pristupu dekompoziciji
 - tradicionalno, dekompozicija je usmjerena procesima (process-centric) ili podacima (data-centric), OO dekompozicija na objekte (podaci i procesi) + UML

□ U osnovi inkrementalni model razvoja s verzijama-gradnjama

- gradnja (build) – inačica
- gradnje se temelje na slučajevima korištenja po prioritetima
- (svaka) gradnja predstavlja inkrementalni napredak završetku
- gradnja obuhvaća analizu, oblikovanje i izradu
- gradnja isporučuje funkcionalni sustav i predloške rješenja za knjižnice
- gradnja vraća povratnu informaciju o zahtjevima
- gradnje omogućuju upravljanje čvrstim rokovima (timeboxing)
 - u intervalima do dva tjedna ili mjeseca ovisno o veličini i složenosti projekta

MOOSAD pristup



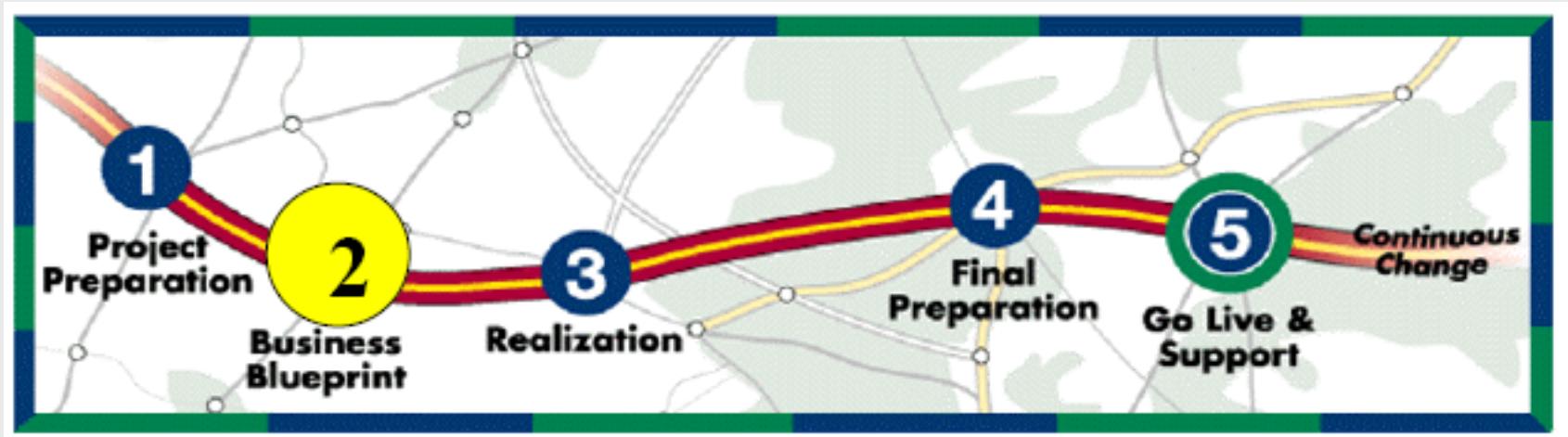
❑ **Debeli klijent**

❑ **Višeslojni poslovni sustav**

Nabava gotovih proizvoda

- **Alternativa razvoju je kupnja postojećeg proizvoda.**
 - Commercial-Off-The-Shelf (COTS), Shrink-wrap – općenito
 - Enterprise Resource Planning (ERP) – sustav za podršku poslovanju
- **Proizvodi rijetko imaju svu potrebnu funkcionalnost, ali imaju i prednosti**
 - Proizvod je vidljiv i može se ocijeniti (!?)
 - Proizvod je "odmah" dostupan po nabavi (!?)
 - U slučaju ERP, treba obaviti prilagodbu koja može potrajati mjesecima
 - SAP: 20k tablica i 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)

Životni ciklus ERP proizvoda





Diskusija, sport i glazba

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 sustav srednje veličine, proizvede se tokom čitavog ciklusa
a) manje od 10 b) 10 - 20 c) 21 - 30 d) više od 30 redaka/osoba/dan

3. Približan broj pogrešaka koje se pronađu u svakih 1000 programskih redaka tijekom razvoja jest:
a) manje od 30 b) 30 - 40 c) 40 - 50 d) 50 - 60

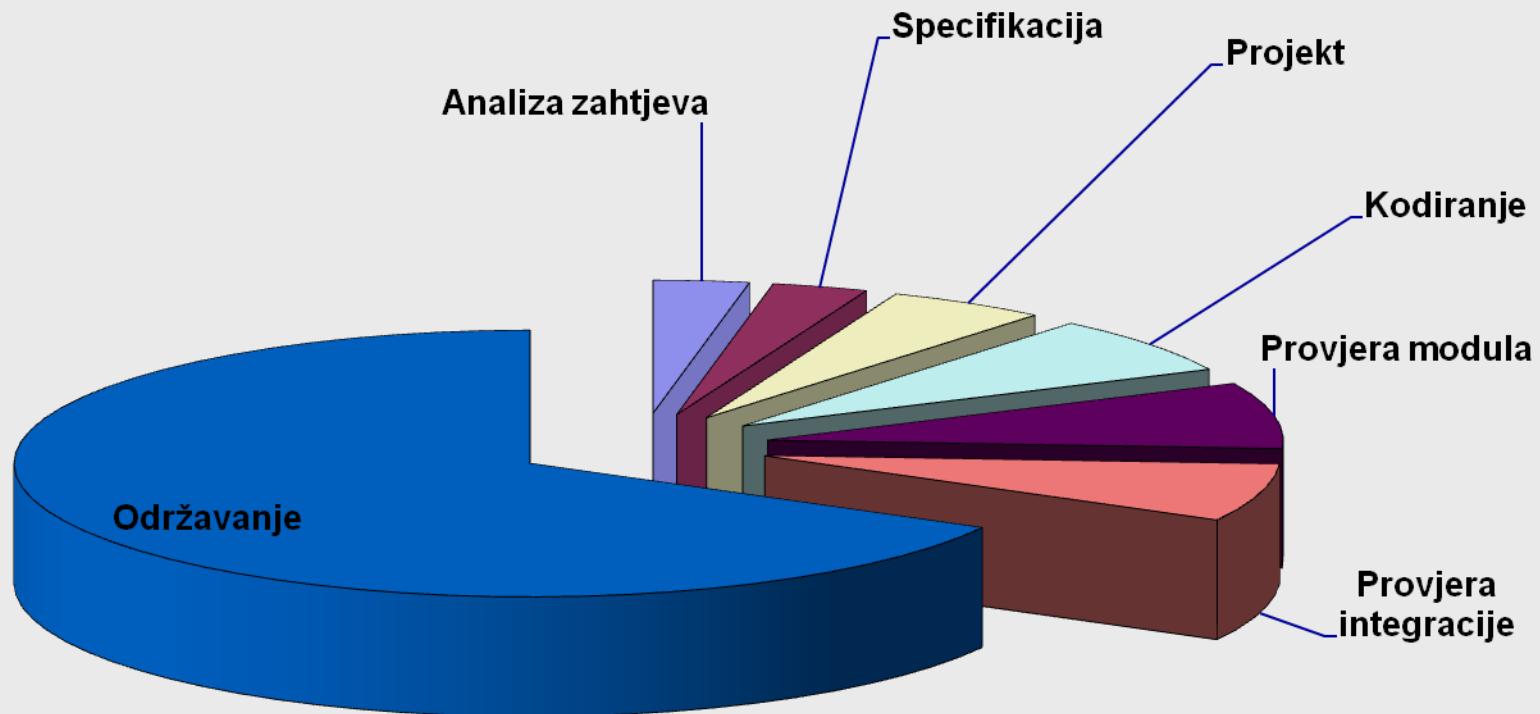
4. Približan broj pogrešaka koje se pronađu u svakih 1000 programskih redaka u isporučenom sustavu jest:
a) manje od 4 b) 4 - 8 c) 8 - 12 d) više od 12

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)

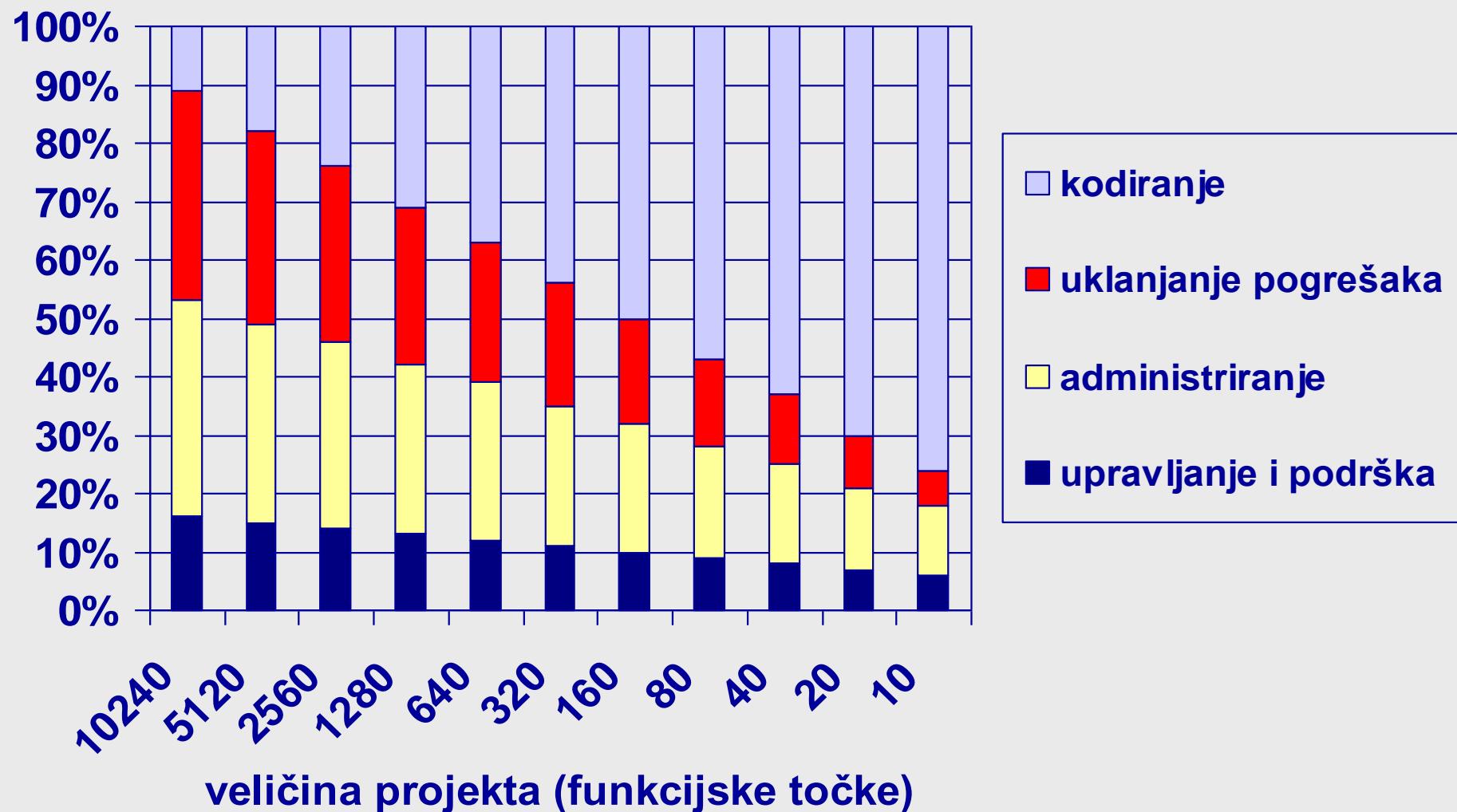
6. Trošak posjedovanja i održavanja softvera je tipično
 - a) za polovicu
 - b) jednako
 - c) dvostrukoputa veći nego što je bio trošak razvoj softvera.
7. Većina pogrešaka koje korisnici pronađu u softveru su rezultat
 - a) programerove pogreške
 - b) problema u oblikovanju zadatka ili njegova razumijevanja
 - 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. Sustav se smatra velikim ako sadrži barem
 - a) 2.000
 - b) 10.000
 - c) 100.000
 - d) 1.000.000redaka izvornog koda

Razdioba troškova u životnom ciklusu softvera



- Analiza:** 3%
- Specifikacija:** 3%
- Projekt:** 5%
- Kodiranje:** 7%
- Provjera modula:** 8%
- Provjera integracije:** 7%
- Održavanje:** 67%

Što se vidi ?



Buzzwords

- ❑ Refaktoriranje je ...
- ❑ TDD ?
- ❑ DDD vs. MDD ?
- ❑ Component Diagram vs. Package Diagram – kojeg kada ?
- ❑ Sequence Diagram ili Collaboration Diagram ili oba ?
- ❑ DSL ?

Reference

□ Resursi

- Association for Information Systems, <http://home.aisnet.org/>
- Kontrolne liste i predlošci projektne dokumentacije <http://www.construx.com>
- Prilagodljivi razvojni proces (Adaptable Process Model) <http://www.rspa.com>

□ Literatura

- Dennis A., Wixom B.H., Tegarden D. Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach, 2nd Edition, Wiley, 2005
- Whitten J.L., L.D. Bentley, Systems Analysis and Design Methods, McGraw Hill, 2007.
- Dennis, Wixom, Roth, Systems Analysis and Design, 4rd edition, Wiley, 2009.

Strategija organizacije

Identifikacija i selekcija projekata

Pokretanje i planiranje projekata

2013/14.02

Strateško planiranje poslovanja (1)

- Izrada poslovne strategije, dugoročno planiranje resursa i akcija
- Uprava definira viziju i misiju organizacije, tj. strateške ciljeve
 - **misija** – svrha (postojanja) organizacije
 - postojeće stanje i ključni procesi te poželjne performanse
 - **vizija** – namjeravano, očekivano stanje u daljoj budućnosti
 - određuje kriterije za donošenje odluka
- Slijede poslovni ciljevi, procesi i zadaci za ispunjenje
 - **što** se želi postići (vrijednosti): prepoznatljivost, kvaliteta, prihodi, ...
 - **kako** to postići: promjenom organizacije, poboljšanjem administracije, ...

Strateško planiranje poslovanja (2)

□ Čimbenici koji utječu na postavljanje ciljeva

- ograničenja (organizacijska, finansijska, zakonska, ...)
- potrebe i želje uprave, poslovodstva, zaposlenika (ugled, utjecaj, ...)

□ Vremenski okviri, razdoblje [Awad, 1985]:

- kratkoročno, obično manje od 2 godine
- srednjeročno, 2-5 godina
- dugoročno, više od 5 godina

□ Primjer:

- <http://www.fer.unizg.hr/strategija>

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?
- Postojeće aplikacije, aplikacije koje se razvijaju, potrebne aplikacije?
- Koji su raspoloživi resursi (osoblje, tehnička sredstva, tehnologija, ...)?

□ Razlozi zbog kojih treba planirati IS

- otoci informatizacije
- redundancija, nepotpunost, nepovezanost
- tehnološka zastarjelost, različitost

□ Primjer: Sveučilište (bib, mozvag, ..., cop)

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

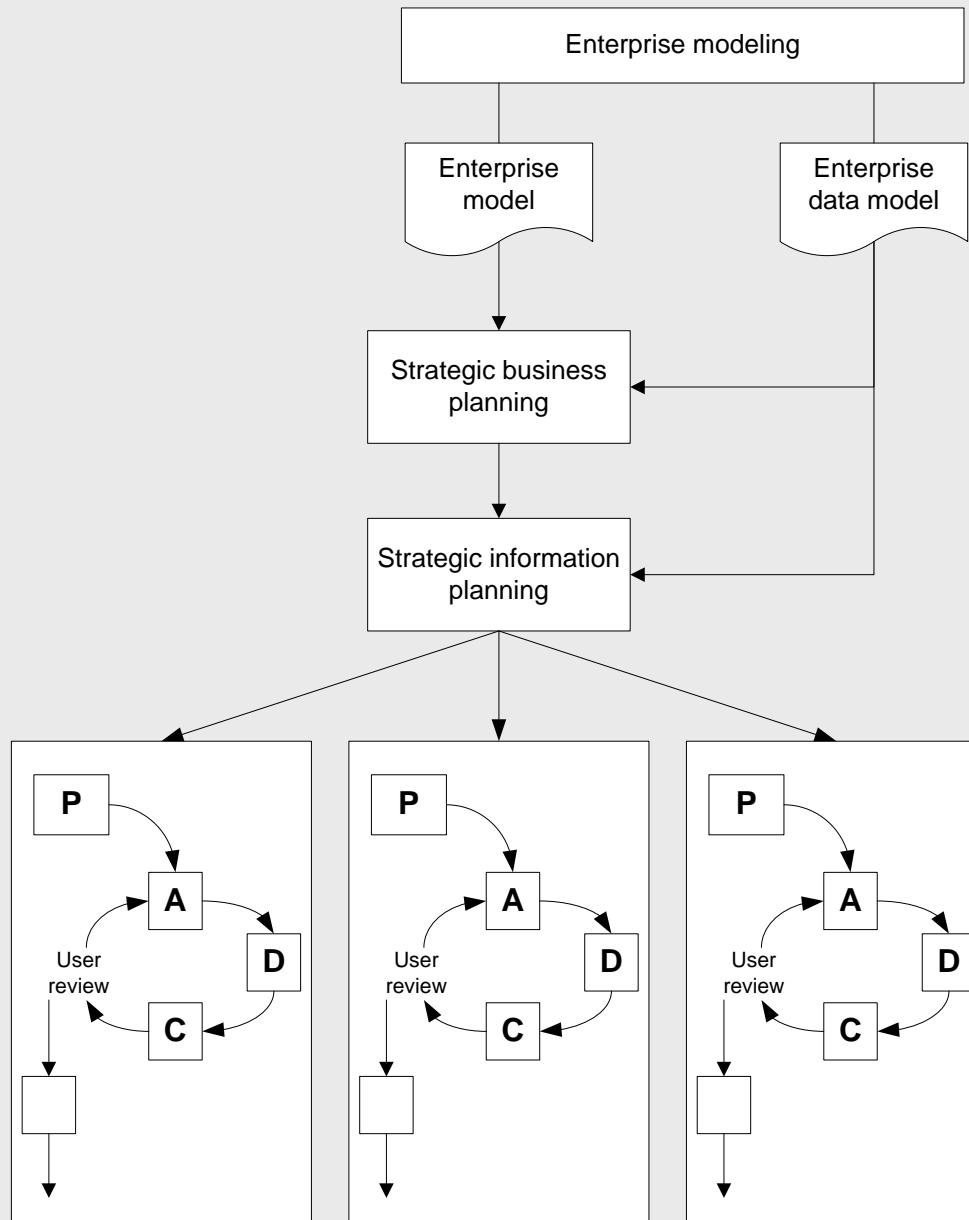
- planiranje IS sukladno strategiji razvoja organizacije
 - informatizacija kao potpora promjeni organizacije i poslovnih procesa
- *istraživanje poslovnog sustava s ciljem definiranja općeg (sveobuhvatnog) plana i arhitekture IS čiji razvoj slijedi*

□ U praksi ...

- organizacija se "dovodi u red" tijekom informatizacije i s pomoću nje
- analiza sustava – evidencija problema i slabosti poslovnih procesa
- dizajn sustava – predlaganje ili nametanje rješenja (uf)

□ 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

□ Poslovna strategija – potpora IT ne može se kupiti

- Ključne vrijednosti: tradicija, ugled, povjerenje, iskustvo, ..., IS
- Orijentacija ka klijentu (CRM) – personalizacija, ..., mobilna prodaja
- Razvoj poslovne inteligencije (BI) – skrivene zakonitosti, ..., konkurentnost, ..
- Usvajanje regulativa (Basel II, Solvency II)

□ Informacijska strategija

- Sigurnost i zaštita IS, ..., kontinuitet poslovanja
- Integriranje aplikacija (SOA, ..., ESB)
- Promjena arhitekture (dvoslojna u višeslojne)
- ...

□ Upravljanje IS – reorganizacija

- Ured za upravljanje projektima
- Centar kompetencije za poslovnu inteligenciju
- Služba za poslovne procese
- ...

Primjer strategije: Poduzeće

□ Prosinac 2005.

- Tridesetak aplikacija, informacijski otoci
- Različite tehnologije, zastarjela tehnologija
- Raspršeno osoblje, zastarjela podjela posla

□ Strategija

- Smjernice razvoja i organizacije informatičke podrške
 - sistematizacija radnih mesta u informatici i reorganizacija službe
 - poduka informatičkih djelatnika
- Smjernice 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)

□ Plan za razdoblje 2006-2009.

ID	Task Name	Duration	Start	Finish	5	2	H2	H
1	Održavanje aplikacija	900 days	Mon 21.11.05	Fri 15.09.09				
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	Sistemske 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 uskladjeni	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 1.12.08	Fri 15.5.09				

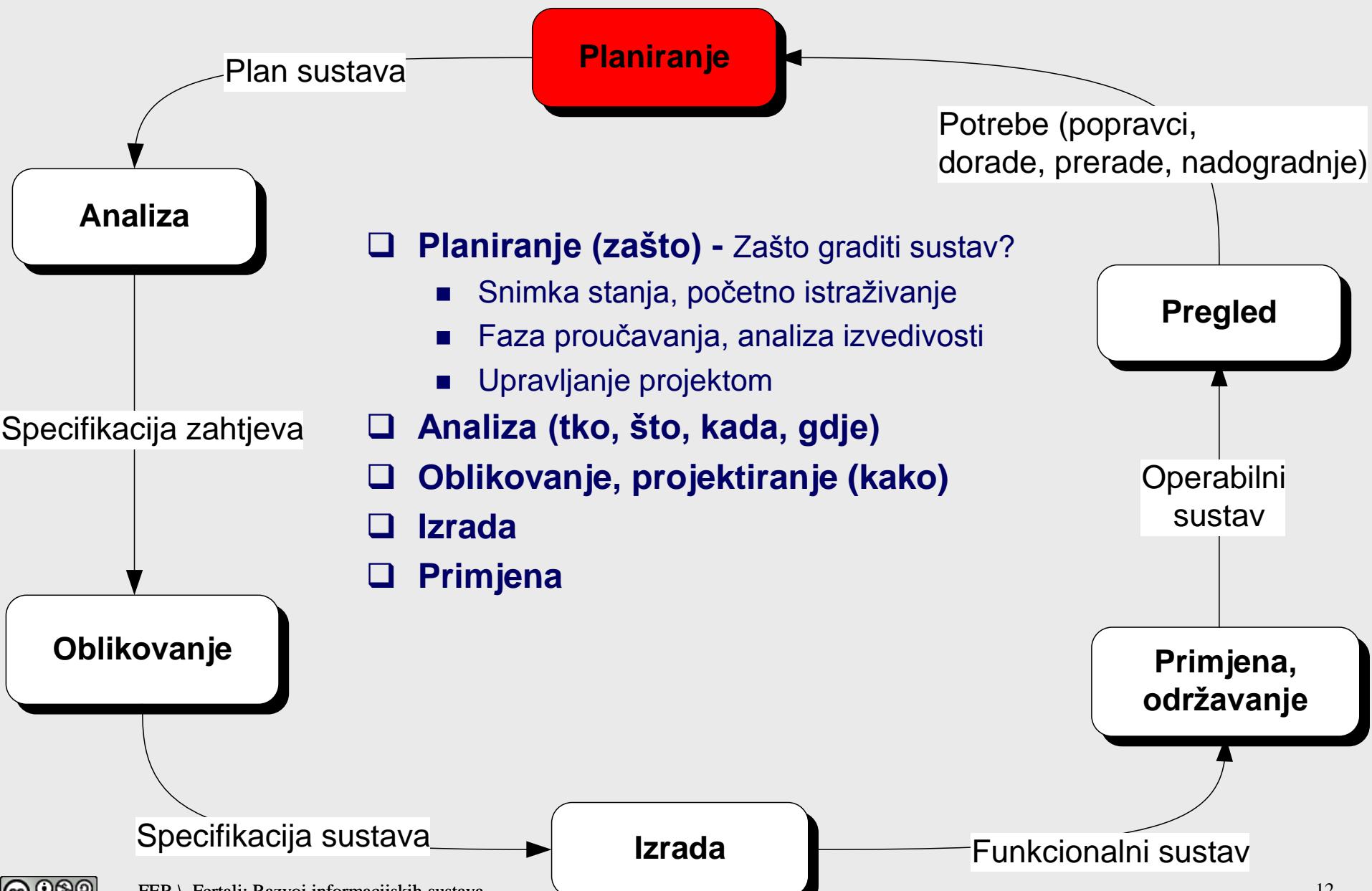
Primjer strategije: Poduzeće (3)

□ Ažurirani plan pripreme 2009.

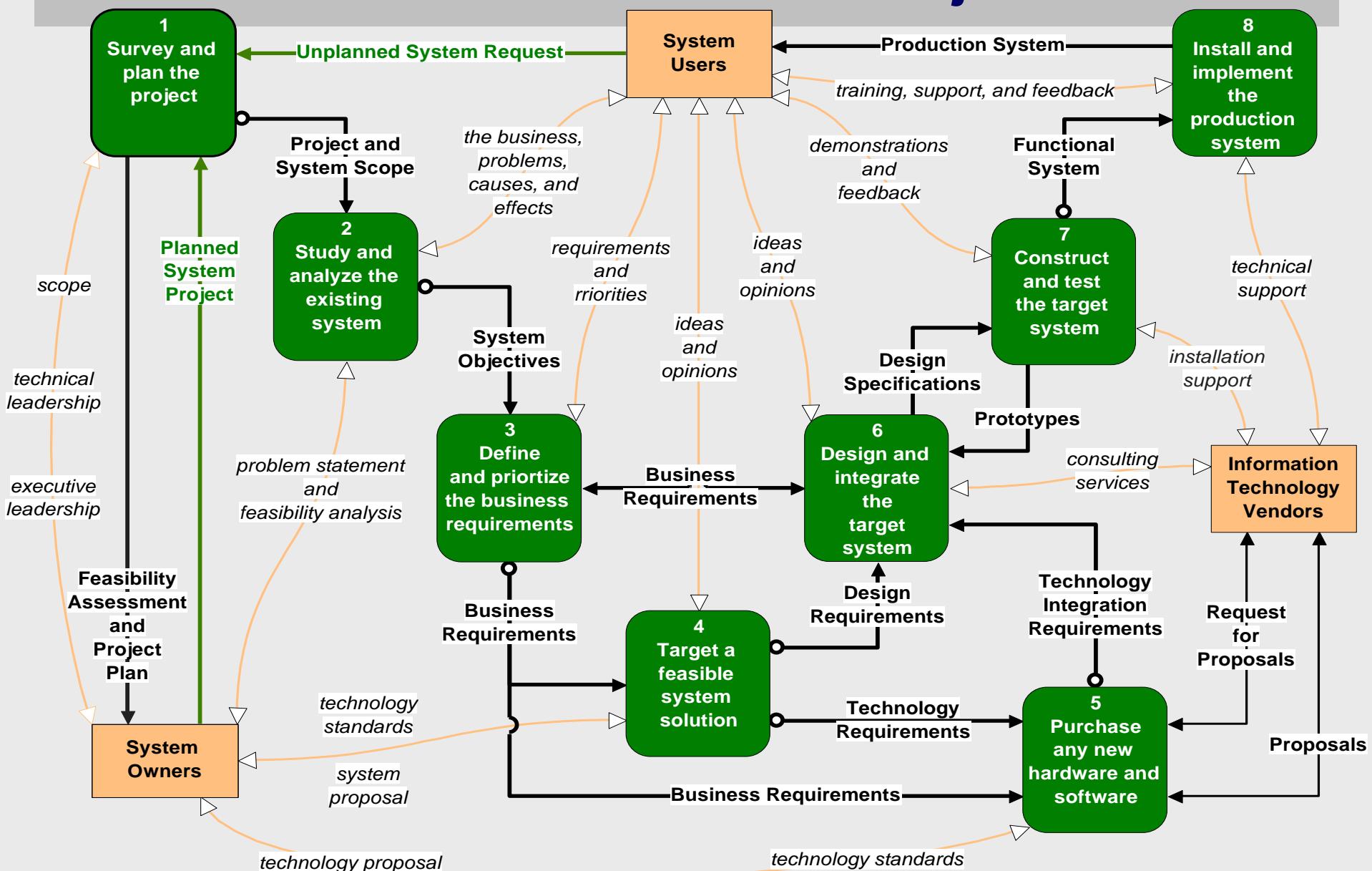
ID		Task Name	Duration	Start	Finish	Qtr 1, Jan
1		Faza pripreme	80 days	Mon 26.1.09	Fri 22.5.09	
2		Us postava infrastrukture projekta (TFS, S	5 days	Mon 26.1.09	Fri 30.1.09	
3		Školovanje za modeliranje podataka	5 days	Mon 2.2.09	Fri 6.2.09	
4		Školovanje za SQL	5 days	Mon 16.2.09	Fri 20.2.09	
5		Školovanje 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

Planiranje i pokretanje projekta



Kontekst informatizacije



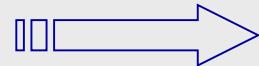
Identifikacija projekta

- **Sponzor projekta = osoba ili grupa koja ustanovi poslovne potrebe**
 - najčešće netko iz poslovanja (Financije, Studentska, Proizvodnja)
- **Pokretači promjena**
 - korisnici - nezadovoljstvo aplikacijama i/ili podacima
 - nepouzdanost, nedostupnost, manjkavost
 - "informacijska glad"
 - reorganizacija, utjecaji okoline
 - promjene organizacijske strukture, poslovnih procesa, ... (pr. Bolonja)
 - pokazatelji poslovanja
 - npr. pad prodaje, uska grla proizvodnje, neplanirano povećanje troškova
 - zastarjela tehnologija
 - npr. razvojni alati, arhitekture, ...

Predlaganje i odabir 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)
 - provodi se snimka stanja i izrada prijedloga - povelje projekta
- temeljem prijava na natječaj organizacije koja traži rješenje
 - javlja se više ponuditelja od kojih se odabire najbolji (najjeftiniji !?)
 - primjer: <https://eojn.nn.hr/Oglasnik/>
- temeljem prijava na natječaj fondova
 - predlagatelj navodi potencijalno tržište te očekivanu korist
 - primjer : <http://www.bicro.hr>



□ Odabir projekta (project selection)

- povjerenstvo/odbor za odabir (steering committee, approval comitee)

Zahtjev na sustav

- **sažetak projekta (sponzor, naziv, cilj, svrha)**
- **poslovne potrebe – poslovni razlozi za pokretanje**
 - npr. povećanje prodaje, povećanje udjela na tržištu, poboljšanje usluge ...
- **poslovni zahtjevi – poboljšane mogućnosti poslovanja**
 - npr. pristup putem interneta, izrada upravljačkih izvješća, ...
- **očekivana korist – poslovna vrijednost (!?)**
 - npr. povećanje prodaje 3%, udjela na tržištu 1%, uštede u održavanju
- **posebnosti i ograničenja - argumenti relevantni za odobrenje**
 - npr. rok, kompatibilnost s politikom sigurnosti i zaštite

Snimka stanja

- Početno istraživanje (survey phase, initial study, preliminary investigation)
 - "Je li projekt vrijedan pažnje?"
 - **problem:** sprječava ispunjenje svrhe, postizanje ciljeva, obavljanje zadaća
 - **prilika:** mogućnost pozitivne promjene, čak i kada ne postoji problem
 - **direktiva:** poslovno ograničenje (pr. pravilnik) ili vanjski utjecaj (pr. zakon)
 - aktivnosti
 - pregled poslovnih planova, inventura aplikacija, intervjuiranje
 - procjena dorade, nadgradnje i izgradnje
- Rezultat:
 - povelja projekta (project charter),
 - početni doseg i plan projekta
- Povjerenstvo odobrava, odbacuje ili odgađa prijedlog
- Primjer:  \Planiranje\PISEF-Snimka-*, jedan krug intervjeta, 30 SA, 1mj
- Primjer: Projekt „M”, 4 mjeseca, 2 SA,
 - 3 faze: prikupljanje informacija, sistem analiza, dokumentiranje snimke

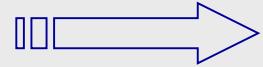
Primjer: postojeći problemi, prijedlozi rješenja

Kratko obrazloženje problema, mogućnosti ili direktive	Hitnost	Vidljivost	Korist	Prioritet	Predloženo rjesenje
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> nametnulo je povećanje zahtjeva za protokol informacija i dokumenata.	6 mjeseci	Srednja	75,000	2	Novi razvoj
3. Trenutno 3 različita sustava za unos narudžbi servisiraju odjeli za audio, video i video igre. Svaki sustav ima vlastito sučelje prema različitom 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 (<i>iz Privatna predstava i Filmsko platno</i>).	12 mjeseci	Niska	15,000	3	Po razvitku 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 (štapićasti kod) koji se razvija za skladište.	3 mjeseca	Visoka	65,000	1	Brza ispravka, a zatim novi razvoj

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

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

Proučavanje problema, analiza izvedivosti



□ Proučavanje problema (study phase)

- "Jesu li problemi vrijedni rješavanja?"
- produbljenje snimke
- uočavanje problema, uzroka i mogućih posljedica
- preciziranje ciljeva, prijedlozi rješenja, procjena izvedivosti
- preciziranje dosega projekta

□ povjerenstvo odobrava, odbacuje ili reducira projekt

□ Analiza izvedivosti (feasibility analysis)

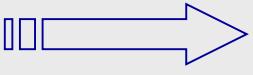
- procjena može li se projekt provesti s obzirom na uvjete ?
- mali projekti - dio pokretanja projekta
- veliki projekti – zasebno, čak i više puta, ili kao zaseban projekt !

Pokretanje projekta i početno planiranje

- Može teći usporedno i temeljem snimke stanja i dublje analize
- Uspostava projekta
 - plan rada, ustroj ekipe, uspostava nadzora i kontrole
 - važno je osigurati predanost dionika zajedničkom cilju (commitment) !
- Određivanje svrhe i ciljeva projekta (goal, objectives)
 - **Svrha** – nespecifična, krajnji cilj djelovanja i krajnji predmet želje
 - **Cilj** – konkretan, može ih biti više, stanje koje projekt nastoji postići
- Doseg (scope)
 - **System boundary** – Granice sustava ? Što će biti napravljeno ?
 - **Constraints** – Ograničenja (teh, org, fin) ? Što neće biti napravljeno ?
 - **Objectives** – Što će se postići ? Čemu će služiti ?
 - **Permissions** – Tko će i pod kojim uvjetima koristiti ?
 - **End products** – Kako se mjeri dovršenost i uspjeh (neuspjeh) ?
- Početno planiranje
 - strukturiranje životnog ciklusa
 - okvirni vremenski plan po fazama

Primjeri strategije, izvedivosti i plana

- Strategija srednjoročnog razvoja IS Croatia osiguranja d.d., 2008.
- Specifikacija sustava Cro-Fauna, Državni zavod za zaštitu prirode, 2007.
- Strategija razvoja informacijskog sustava Hrvatskih šuma d.o.o., 2005.
- ...
- Idejno rješenje informacijskog sustava Personalne uprave MORH, 1996.
- Globalno idejno rješenje informatizacije za Sljeme d.d., Sljeme, Zagreb, 1995.



Proučavanje problema, analiza izvedivosti

(detaljnije)

Analiza problema

□ Produbljivanje analize za projekte koji prođu početnu selekciju

- "Jesu li problemi vrijedni rješavanja?", "Je li 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 sustava

□ Mogu se koristiti različite metode # primjeri slijede

- Analiza kritičnih faktora uspjeha (CSF - Critical Success Factors)
 - čimbenici kojima poslovodstvo posvećuje posebnu pažnju
 - čimbenici koji razmjerno brzo i lako doprinose ostvarivanju ciljeva
- Procjena troškova-koristi (Cost-Benefit Analysis, CBA)

Istraživanje problema, uzroka i posljedica

□ Primjer: Istraživanje uzroka

- Problem: pad prodaje
- Vidljivi znak (manifestacija): povećani opoziv (storno) narudžbi
- Razlog (posljedica): nezadovoljstvo kupaca
- Uzrok: spor sustav za naručivanje

□ Treba razdvojiti uzroke i posljedice problema !

- sporost sustava nije izoliran problem nego može biti posljedica
 - pomanjkanja osoblja, "pretjerane" obrade, ručne obrade, ili ... ?
- primjer: pisarnica FER-a
 - (diskusija: što učiniti)

Postavljanje ciljeva

□ Primjeri poslovnih ciljeva (stvarnih projekata)

- 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 *nečega*
- Cilj: Ubrzanje unosa
- Kriterij: performanse, izvedba – mora biti mjerljiv
 - Apsolutni iznos ili relativni iznos konkretnе vrijednosti
 - Može biti binaran – "postoji ili ne postoji", npr. online pretraživanje
- Loš primjer: "Povećati broj unosa u 24 sata za 5%"
- Bolji primjer: "Unijeti 98% nečega unutar 24 sata"

Ograničenja

□ Osoblje

- Popunjenošć odjela informatike je 60%
- Moguće je zaposliti najviše $X << 100\%$ zaposlenika

□ Materijalni trošak

- Nabavka uredskog i potrošnog materijala ne smije premašiti XX 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 XX HRK (uvijek manji od traženog!)
- Naknade izvođačima ne smiju premašiti XX% ukupnog iznosa (teži 0!)

□ Tehnička ograničenja (OS, IDE, ...) – trivijalna !

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 koje koriste iste dokumente (napraviti dijeljene dokumente)*
- *Ospozobiti 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*

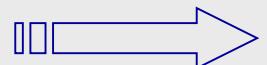
□ Rješenja, kao i problemi, najčešće nisu informatički !!!

Analiza izvedivosti

□ Analiza/Studija izvedivosti (feasibility analysis, feasibility study)

- procjena korisnosti, praktičnosti i isplativosti projekta
- izvješće (feasibility report) - relevantnim dionicima ili odboru za odabir
 - eventualni povratak u studiju izvedivosti → revidirano izvješće

□ Kakva izvedivost ?



- organizacijska izvedivost
- tehničko-tehnološka izvedivost
- vremenska izvedivost
- ekonomska izvedivost - analiza troškova koristi

□ Kada provesti ?

- uobičajeno tijekom planiranja
- po promjeni dosega – početno izvediv projekt može postati neizvediv
- između ključnih faza projekta – uz izvedivost, dokaz održivosti

Organizacijska izvedivost (1)

» Ima li smisla graditi ? Hoće li se koristiti?

□ Operativna izvedivost

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

□ Vrijedi li rješavati problem? Rješava li predloženo rješenje problem?

» PIECES koncept

- **P** [performance]
 - protočnost i odziv sustava
- **I** [information]
 - informacije (podaci): pravodobne, prikladne, ažurne, korisne
- **E** [economics]
 - kontrola troškova ili povećanje profita
- **C** [control]
 - poboljšanje kontrole ili sigurnosti, npr. točnost, sigurnost i zaštita podataka
- **E** [efficiency]
 - efikasnost, maksimizacija uporabe resursa (ljudi, sredstva, novac - vrijeme)
- **S** [service]
 - usluga korisnicima (zaposlenici, partneri, ...), elastičnost i prilagodljivost

Organizacijska izvedivost (2)

□ Socio-psihološka izvedivost

- 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
- Promjena radnog okruženja, procedura – prilagodba promjenama
- Prihvatanje krajnjih korisnika – otpori budućoj ulozi ili tehničkom rješenju
 - Rizici : informatička nepismenost, strah od neznanja, novi način rada, gubitak posla, 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, složenije za iskusne
- Zadovoljstvo – ponuđenom rješenju korisnik daje prednost pred postojećim

Tehničko-tehnološka izvedivost

» Može li se sagraditi? Može li se kupiti?

□ Procjena mogućih tehničkih rješenja i alternativa

- postojeća rješenja na tržištu ili u drugim organizacijama

□ Primjenjivost rješenja, tehnologije

- Najsuvremenija tehnologija - naprednost, zanimljivost, rizik
- Zrela i dokazana tehnologija - sigurnost, bolja podrška

□ Raspoloživost tehnologije

- Može li se nabaviti?
- Ima potrebne karakteristike?
- Treba prilagoditi ili doraditi?

□ Stručnost

- Postoji li potrebna stručnost i vještina za primjenu tehnologije?

□ Primjer: Uprava-IzborAlata

Vremenska izvedivost

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

Prihvatljivost vremenskog rasporeda

- opravdanost rokova s obzirom na raspoloživu stručnost

Poželjni rokovi

- bolje isporučiti potrebno nešto kasnije, nego beskorisno na vrijeme !!!
- uvažavanje promjena posmiče završetak projekta
- iterativno se ažurira i predlaže alternativni vremenski plan

Čvrsti rokovi (timeboxing)

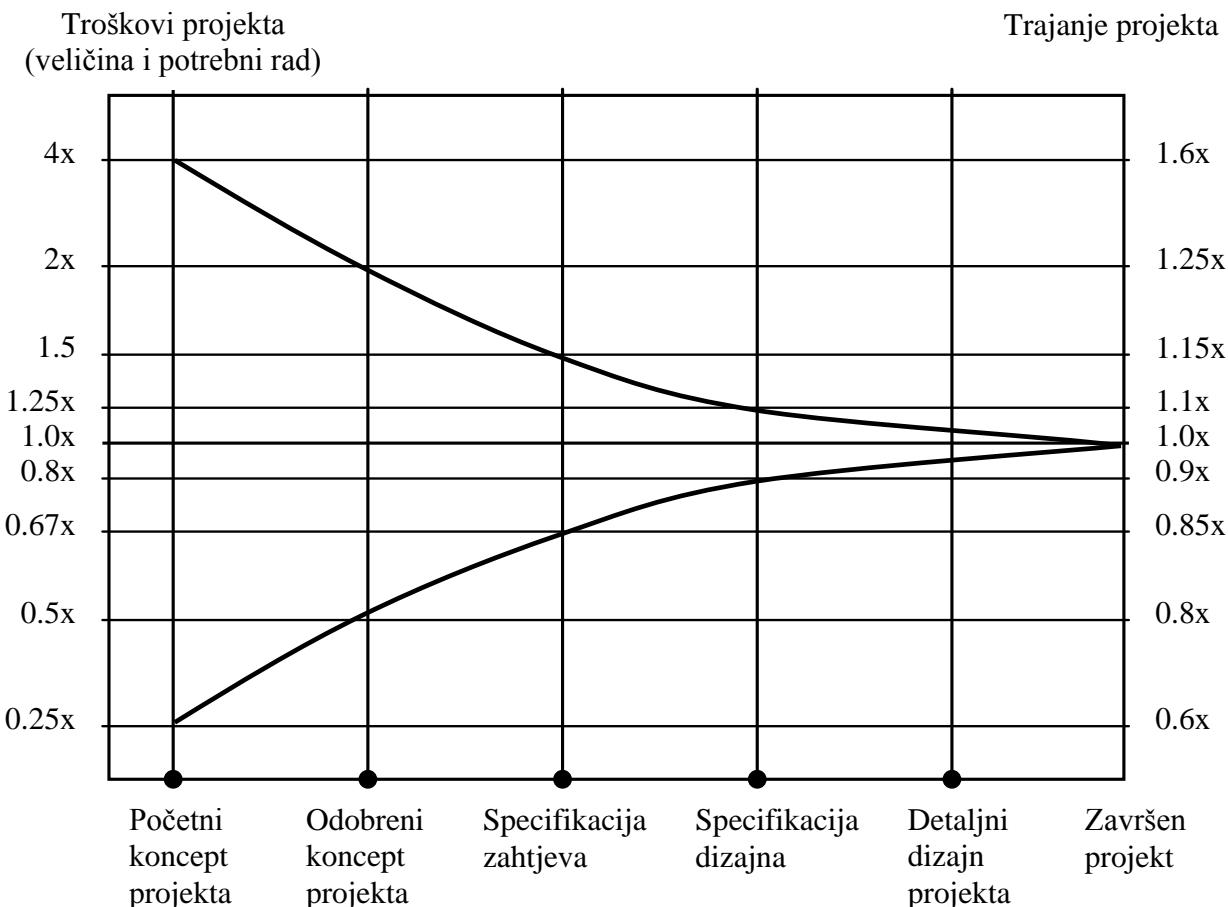
1. postavlja se čvrsti rok isporuke
2. određuju prioriteti funkcionalnosti koju treba ugraditi
3. gradi se jezgra sustava (nužne funkcionalnosti)
4. odgađa se ugradnja funkcionalnosti koja ne može biti dovršena na vrijeme
5. u roku se isporučuje sustav s dosegnutom funkcionalnošću
6. ponavljaju se koraci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

Problem (ne)preciznosti 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



Postupak procjenjivanja

□ Osnovni koraci

1. Procjena veličine projekta (metrika projekta)

- broj funkcijskih točaka (function point – FP) ili programskih redaka (LOC)
- broj objektnih točaka
- analogija s prethodnim projektima

2. Procjena potrebnog napora u čovjek-mjesecima (čm)

- iz procjene veličine projekta i produktivnosti
- tablice produktivnosti, algoritmički modeli, povijesni podaci

3. Procjena trajanja projekta (vremenski plan)

- trivijalno iz napora i veličine ekipe u kalendarskim mjesecima ili godinama
- npr. procijenjeni rad 36 čm / 6 članova ekipe = 6 kalendarskih mjeseci

□ Meta korak kojim se rješava problem nepreciznosti procjene

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

Procjena veličine projekta i napora (1. i 2.)

□ Funkcijske točke (FP)

1. procjena broja ulaza, izlaza i datoteka i njihove složenosti * utjecaj * faktor prilagodbe
 - primjer: 30 LOC / FP za C++
 - procijenjenih 6000 FP zahtjeva izradu 180000 LOC
2. produktivnost za poslovni sustav: 500 LOC/čm → ukupni napor = 360 čm

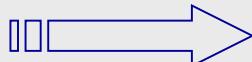
□ Objektne točke (aplikacijske točke, ne programski razredi!)

1. procjena broja zaslonskih maski, izvješća i modula za pristup podacima
 - produktivnost 4 -50 OT / mjesec, ovisno o alatu i vještini (statistike nema)
2. primjer: 500 OT uz produktivnost 25 OT/mj. zahtjeva 20 čm
 - primjer2: 2 h / šifrarnik, 4h / osnovna tablica, 8h / master-detail, 40 h / izuzetak *

□ Funkcijske i objektne točke

- temeljem dijagrama dekompozicije funkcija, dijagrama toka podataka i ER dijagrama, odnosno UC dijagrama i dijagrama razreda
- ... koji slijede kao rezultat dublje analize, a češće oblikovanja !

□ Procjena analogijom



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 sličnom projektu na planiranje utrošeno 4čm, u ovom se može očekivati Analiza ($4/15\% * 20\% = 5.33\text{cm}$), Dizajn (9.33čm), Izrada (8čm)

□ Skaliranje analizom povijesnih podataka (paušalno - rule of thumb)

- 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

Procjena trajanja projekta (3.)

- **Iskustveno pravilo za optimalnu procjenu trajanja temeljem npora:**
 - optimalno trajanje u mjesecima = $3.0 \cdot (\text{čovjek-mjeseci})^{1/3}$
- **Za procjenu projekta koji zahtijeva napor 65 čm,**
 - optimalno trajanje je $3.0 \cdot 65^{1/3}$, tj 12 mjeseci
- **To dalje znači da je optimalna veličina tima**
 - $65/12$, tj. okvirno 5-6 članova
- **Što se događa kada projekt treba dovršiti brže ili sporije?**
 - prema različitim izvorima faktor 3.0 može varirati od 4.0 do 2.5
 - više članova komplicira komunikaciju pa će biti manje produktivni
 - možemo angažirati manje članova, ali će nas dulje koštati
- **I zato ... procjena ekonomske izvedivosti**

Ekonomска извештавост

» Isplati li se graditi? Kada će se isplatiti?

□ Analiza troškova-koristi (Cost-Benefit Analysis - CBA)

□ Fiksni troškovi – neovisni o poslovnim aktivnostima

- absolutni iznos, početna procjena, ažuriranje tijekom projekta
 - osoblje: plaće, izobrazba (tečajevi)
 - oprema – nabava nakon odabira tehničkog rješenja

□ Varijabilni troškovi – proporcionalni poslovnim aktivnostima

- relativan iznos, ovisan o uporabi, npr:
 - režije (struja, telefon, internet)
 - putni troškovi
 - materijalni troškovi i troškovi održavanja (ljudski rad)

Kategorije troškova i koristi

□ Mjerljivi (tangible – opipljiv, određen, shvatljiv)

- zna se točan iznos ili iznos može biti procijenjen
- mjerljivi troškovi -
 - plaće, režije, licence, ...
- mjerljive koristi
 - izražene kao godišnja ušteda ili ušteda po proizvedenom predmetu

□ Nemjerljivi (intangible)

- postojanje ili "vrijednost" ne može se egzaktno mjeriti ili dokazati
- nemjerljivi troškovi
 - pad morala, (nemjerljivi) pad produktivnosti ili gubitak tržišta
- nemjerljive koristi
 - mogu pomoći ili odmoći korisnosti proizvoda,
 - npr. poboljšano zadovoljstvo kupca, zadovoljstvo zaposlenika, ...

Primjeri troškova i koristi

□ Vrijednost novog kupca

- Vrijednost 300 novih kupaca godišnje koji prosječno potroše \$500 po proizvodu koji nakon troškova donosi 12% dobiti
- godišnja dobit iznosi $300 * \$500 * 12\% = \18.000

□ Vrijednost postojećih kupaca

- gubitak 100 kupaca od kojih svaki troši \$2500 godišnje uz dobit 12%, a za čije nadomještanje/reklamiranje treba uložiti \$50.000, gubitak iznosi:
- $100 * \$2500 * .12 + \$50.000 = \$80.000$

□ Smanjenje cijene rada ili ušteda smanjenjem posla

- smanjimo trajanje zadatka s 5 min na 30 sec, a radi osoba za \$50 na sat
- ušteda je $(5 - 30/60)/60$ sati po zadatku * \$50 na sat = \$3,75 po zadatku

□ Nemjerljive koristi nastojimo izraziti paušalnom procjenom iznosa

- nezadovoljni kupci naručuju manje i rjeđe – postotak gubitka u prihodu

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	

Sklopoljje 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 sklopoljia i programske podrške:		
Vrsta	Količina	Cijena
Sklopolje		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 (*interest*) '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
- oričenje tih sredstava na 3 godine uz kamatu od 8% donijelo bi 25.97% dobiti od kamata, tj. $(1 + 0.08)^3 / 100$
- obratno, korist \$100.000 postignuta u 3. godini sada vrijedi \$79.380, tj. $\$100.000 / (1 + 0.08)^3$

□ Primjer: \Planiranje\NPV.xls

- Pitanje: Što je realna "kamata" na budući trošak / prihod projekta ?

Neto sadašnja vrijednost

□ Neto sadašnja vrijednost - Net Present Value (NPV)

- budući trošak i korist s obzirom na gubitak vrijednosti sredstava
- razlika PV budućih priljeva i PV budućih odljeva
- $NPV = (\text{ukupna korist} - \text{ukupni troškovi})$ preračunati na današnji dan

□ Primjer: Koji je projekt isplativiji ?

$$NPV = \sum_{i=1}^n \frac{\text{values}_i}{(1 + rate)^i}$$

Kamata	10%					
Projekt 1	1	2	3	4	5	Ukupno
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
Projekt 2						
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

NPV

2,316

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 koristi i sadašnje vrijednosti troškova (PVB / PVC)
- za prethodni primjer $9743/7427 \sim 1.31$, odnosno $10783/7582 \sim 1.42$
- favorizira brzi povrat investicije a ne dugoročnu dobit

□ 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%					
Trošak / Korist	Godina 0	Godina 1	Godina 2	Godina 3	Godina 4	Godina 5
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

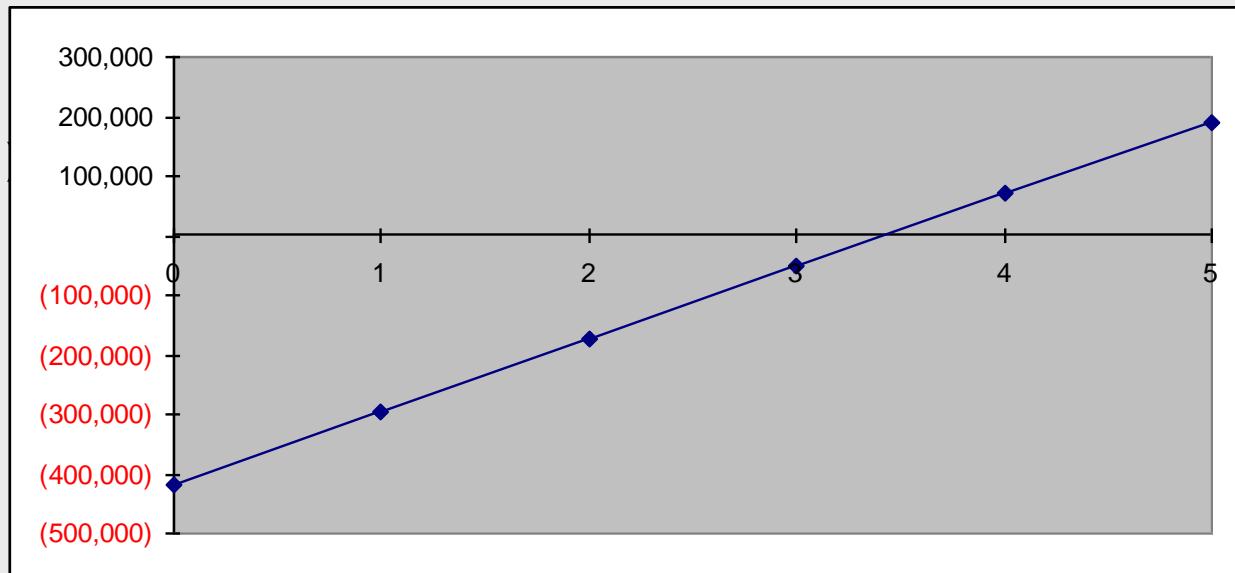
Vrijeme povrata

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

$$= 3.42 \text{ godine}$$

Primjer: [\Planiranje\ PovratInvesticije.xls](#)

Komentirati 0. i 1. godinu



Primjer: Analiza povrata investicije

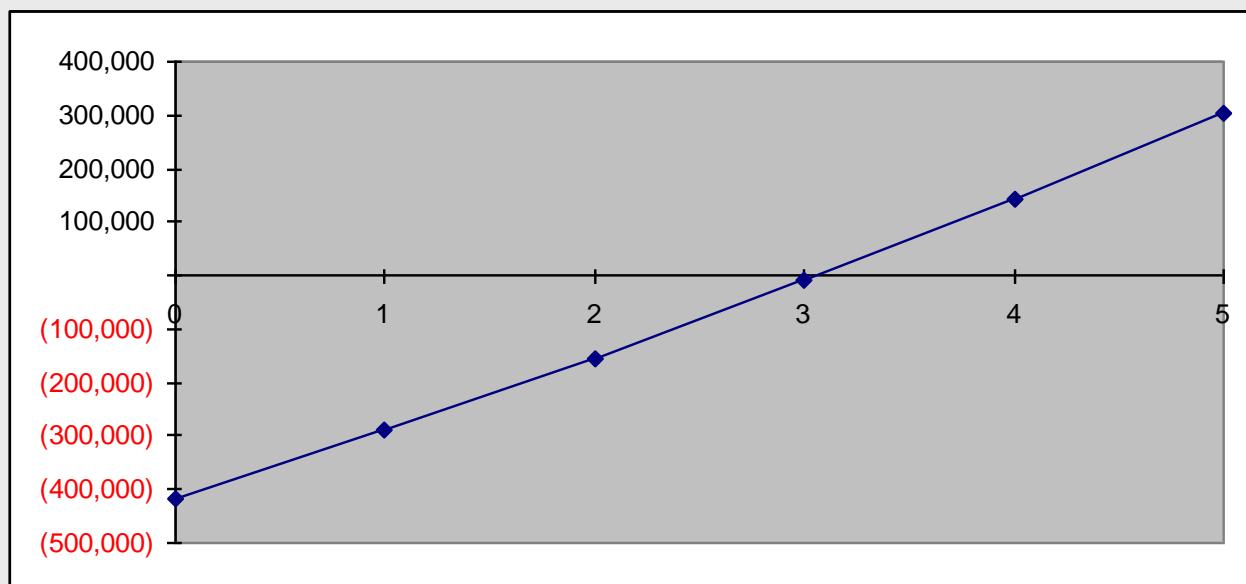
Kamata

6.00%

Trošak / Korist	Godina 0	Godina 1	Godina 2	Godina 3	Godina 4	Godina 5
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
	0	1	2	3	4	5
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
- $ROI = (\text{ukupna korist} - \text{ukupan trošak}) / (\text{ukupan trošak})$
- preračunato u sadašnju vrijednost, $ROI = NPV / (\text{ukupan sadašnji trošak})$
- Za primjer s kamatom 12%, $ROI = 190378 / 478552 = 0.3973 = 39.73\%$
- Za primjer s kamatom 6%, $ROI = 301284 / 489196 = 0.6159 = 61.59\%$

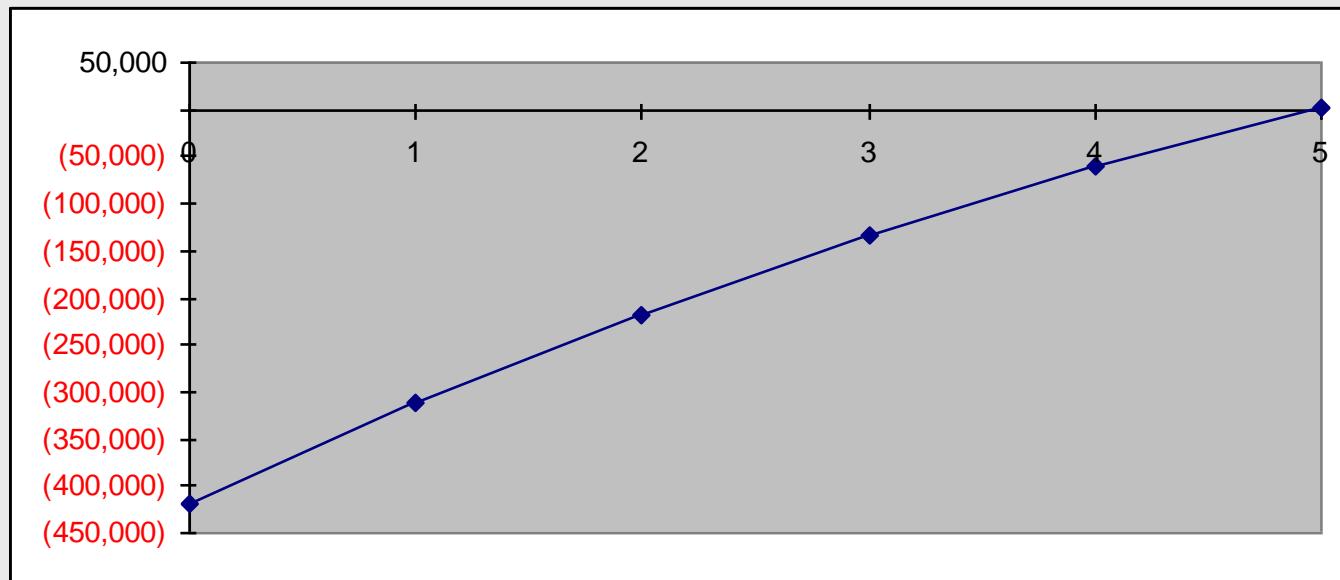
□ Faktor obnavljanja kapitala

- ROI se obično dijeli s 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

Interna stopa rentabilnosti

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

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



Modaliteti izgradnje sustava

Dio analize izvedivosti ili međukorak analize i dizajna

Vlastiti razvoj (insourcing)

□ Varijante

- Razvoj vlastitim snagama (**in-house**), može sadržavati osposobljavanje i angažiranje netehničkog osoblja
 - Primjeri: Ministarstvo, Društvo, Poduzeće
- Najam osoblja povremeno/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 postoji gotova rješenja na tržištu ili takva da se postigne komparativnu prednost
- posebni razlozi - povećana tajnost podataka i procesa ili sigurnost IS

Vanjski razvoj (outsourcing)

□ Najam usluge razvoja IS ili njegovih dijelova

- poduka djelatnika informatičke struke
- savjetništvo pri provedbi pojedinih faza ciklusa
- izrada rješenja po specifikaciji naručitelja
- ...

□ Varijante

- Ugovoren razvoj isporuke gotovog proizvoda (**contract out**)
- Dugoročna suradnja s isporučiteljem (strateškim partnerom) ili izdvajanje odjela informatike u preferiranog izvođača (**preferred contractor**)
- **Multisourcing** – optimalni skup različitih dobavljača, uz vlastiti razvoj

□ Ključna karakteristika

- zavisnost o dobavljaču

Crowdsourcing

□ **Crowdsourcing – okupljanje (velikih) grupa kreativnih pojedinaca**

- koji u slobodno vrijeme nastoje stvoriti novu vrijednost ili
- postići revolucionarno poboljšanje

□ **Primjeri**

- Google – najveći, npr.
 - *Using the Photosphere feature on Android handsets or a DSLR camera, users can now take 360 degree panoramas of places of interest and upload them to a gallery of views. (The Inquirer, 10.12.2013.)*
- Electrolux i Philips – razvoj novih proizvoda
- Procter & Gamble (P&G) – objavljuje zadatke i cijenu
- NASA – programska podrška za obradu velikih količina podataka svemirskih istraživanja
- HP – predviđanja prodaje

Izrada po mjeri ili nabava gotovog rješenja ?

□ 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 !

Graditi ili kupiti gotovo (**build.vs.buy**) ?

□ Određivanje mogućih rješenja

- Identifikacija rješenja na temelju analize poslovnih zahtjeva
- Ulazi: specifikacija HW i SW te odabrana tehnička arhitektura
- Izlazi: moguća rješenja novog sustava i njihove karakteristike

□ Analiza izvedivosti alternativnih rješenja

- Procjena alternativa s obzirom na njihovu izvedivost
- Ulazi: karakteristike rješenja, cijene, reference i uvjeti dobavljača
- Izlazi: analiza izvedivosti za svako moguće rješenje

□ Prijedlog rješenja sustava koje će se oblikovati i ugraditi

- Odabir onog rješenja koje ima najbolju ukupnu kombinaciju izvedivosti
- Ulazi: analiza izvedivosti, 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 pisačima	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 pisač)	3500 kn	4000 kn	2200 kn	3500 kn
Cijena preporučenog računala (plus monitor, pisač 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 -tог rješenja

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

w_j = važnost ili težina j -tog kriterija

- **Što učiniti kada su sustavi (pod)jednako bodovani ?**
 - Primjer: Uprava-IzborAlata
- **Što učiniti ako pojedino svojstvo ima više podsvojstava ?**
 - <http://www.zpr.fer.hr/zpr/Projekti/IPIS/>

Primjer: analiza izvedivosti 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 pisačima	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-Studija izvedivosti**

		Komponente poslovno – upravljačkog IS-a		Skladište i materijalno		Opća služba		Urudžbeni zapisnik	
Nadogradnja	Operativna	3	3	3	3	1	1	1	0
	Tehnička	3	3	3	3	1	1	1	0
	Vremenska	3	3	3	2	1	1	1	0
	Ekonomска	3	3	3	3	1	1	1	0
	Ocjena alternative	3	3	3	2.75	1	1	1	1.75
Izrada vlastitog	Operativna	1	1	1	2	3	3	2	2
	Tehnička	1	1	1	2	2	2	1	2
	Vremenska	1	1	1	1	2	1	1	1
	Ekonomска	1	1	1	1	2	1	1	1
	Ocjena alternative	1	1	1	1.5	2.25	1.75	1.75	1.25
Nabava gotovog	Operativna	2	2	1	1	2	1	1	2
	Tehnička	3	2	1	1	2	1	2	2
	Vremenska	3	2	1	1	3	1	2	3
	Ekonomска	1	2	1	1	1	1	1	2
	Ocjena alternative	2.25	2	1	1	2	1	1.5	2
Konačan prijedlog alternative	Nadogradnja	x	x	x	x			x	
	Izrada vlastitog					x	x	x	
	Nabava gotovog								x



Je li izvediv projekt održiv ?

□ Klizanje dosega (creeping scope)

- polagano, ali značajno povećanje obujma uslijed pogrešne procjene, različitog tumačenja zahtjeva ili promjene zahtjeva
- granice projekta moraju biti definirane što je moguće preciznije
- klizanje možda neće biti uklonjeno 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 Integriranog PIS visokoobrazovnih učilišta
- Procjena održivosti projekta
 - SurvivalTest [McConnell, 1998], <http://www.construx.com/survivalguide/>
 - Primjer:  Planiranje\ McConnell-OdrzivostProjekta



Studijski slučaj

- Firma živi od aplikacije zasnovane na staroj tehnologiji (Clipper)
- Umorni vlasnik održava 30ak klijenata i 10x toliko krajnjih korisnika
- Ustrojena je ekipa od 11 programera koji rade na izradi nove verzije (korisnik-poslužitelj, objektni jezik i RBP sa 100 tablica)
- Modul M1 koji čini pola aplikacije je 90% dovršen i na njemu radi 7 ljudi
- Preostala 2 modula (M2 i M3) nisu započeta, može ih preuzeti 4 zaposlenika
- Po završetku slijedi primjena u više različitih poduzeća.

- Strategija ?
- Model razvoja ?

Zamijeniti staro novim ?

□ Koji je bolji u razdoblju od 5 godina

- stari projekt:
 - stara tehnologija i održavanje 10 kkn / mjesечно
- novi projekt:
 - nakon 3 godine besplatnog razvoja i prilagodbe
 - nabava licenci za 300 kkn uz održavanje 20% / god
 - dodatni trošak održavanja 100 kkn/god

Graditi ili kupiti ?

- Potrebno je izgraditi sustav koji se sastoji od 300 šifrarničkih formi, 100 formi s matičnim podacima i 50 složenih formi pri čemu za izradu jedne forme treba 4, 8, odnosno 30 sati rada. Uzimajući u obzir da razvojnik s bruto plaćom od 15 000 Kn mjesечно radi ukupno 200 sati uz produktivnost od 60%, izračunajte ukupni vremenski napor (u čovjek mjesecima) i optimalno trajanje (u mjesecima) razvoja, koristeći softversku jednadžbu. Odredite veličinu ekipe, to jest broj potrebnih razvojnika i ukupni trošak razvoja.
- Ako se na tržištu može kupiti gotovi sustav inicijalnog troška 150 000 Kn za čiju prilagodbu i implementaciju treba 2 mjeseca i 6 konzultanta uz trošak od 25 000 Kn mjesечно po konzultantu, izračunajte što se više isplati, graditi ili kupiti gotovo rješenje. Treba biti vidljiv postupak (računica) za svaku od alternativa.

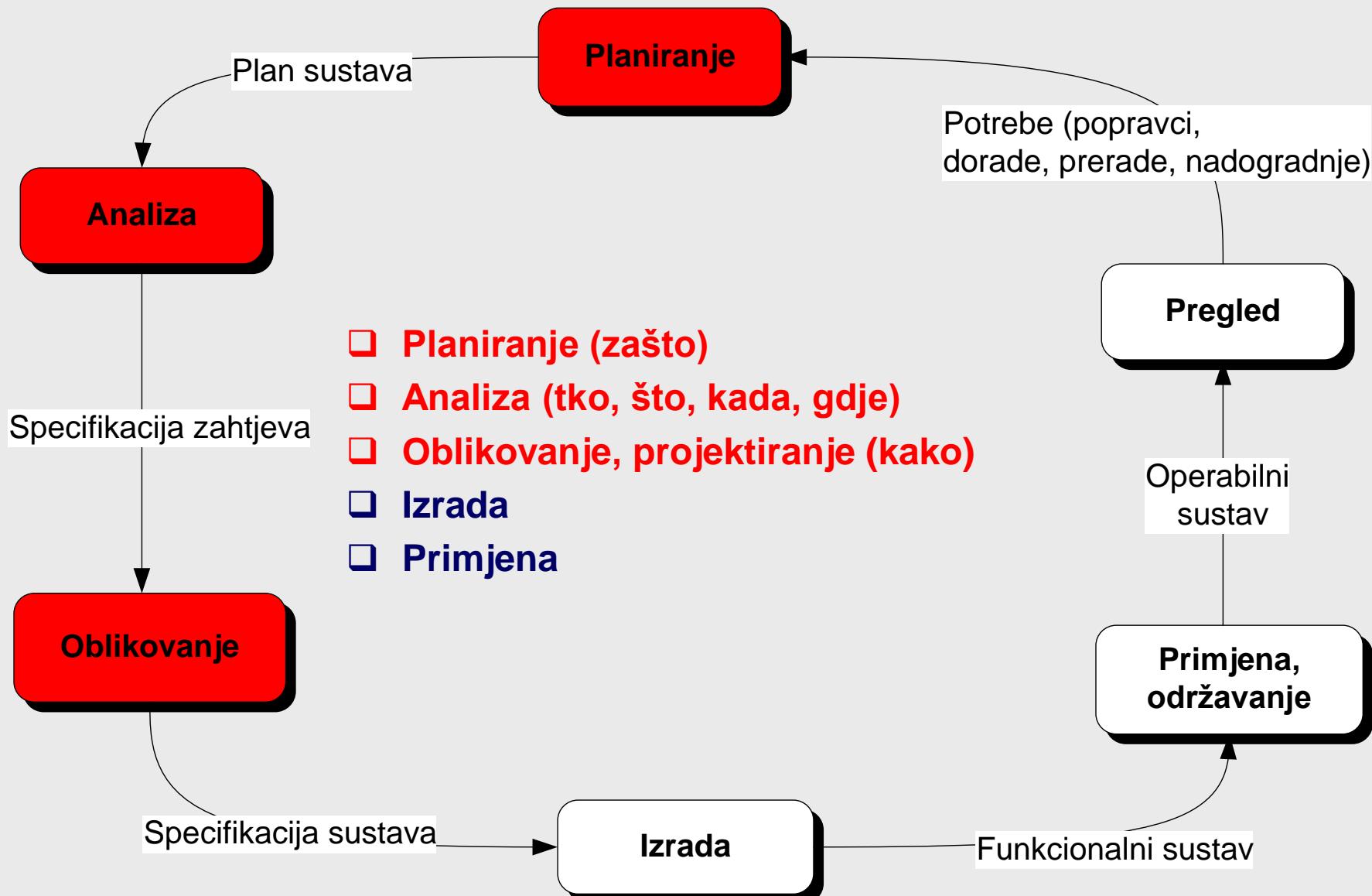
Reference

- **International Functional Point Users Group (IPUG)**
 - <http://www.ifpug.org/>
- **Modern Cost Estimation Tools**
 - <http://www.dacs.dtic.mil/techs/estimation/comparison.shtml>
- **T.T.Wilkens: Earned Value, Clear and Simple**
 - <http://handle.dtic.mil/100.2/ADA402619>
- **COCOMO**
 - <http://sunset.usc.edu/research/COCOMOII/>
 - <http://www1.jsc.nasa.gov/bu2/COCOMO.html>
- **Parametric Cost Estimating Handbook**
 - <http://cost.jsc.nasa.gov/pcehg.html>

Prikupljanje informacija i utvrđivanje zahtjeva

2013/14.03

Kontekst prikupljanja informacija



Postupci prikupljanja informacija

- **Intervjui s ključnim korisnicima i radne sjednice**
 - Lokalne informatičare svakako uključiti u analizu – sučeljavanje, rješavanje proturječja
- **Upitnici i ankete**
 - Nadomjestak za intervju ili prikupljanje informacija o resursima.
- **Analiza dokumentacije**
 - Treba prikupiti sve dokumente značajne za poslovanje (pravila, strukture podataka)
- **Promatranje**
 - Neposredni uvid u poslovne procese promatranjem radnih sredina
- **Ostale tehnike**
 - Prosudba postojećih aplikacija ili evidencija na računalu
 - analiza funkcionalnosti, strukture podataka te podataka koje treba “spasiti”
 - Prototipiranje – kada nema uzora ili korisnik ne zna što hoće
- **Postupak analize mora biti prilagođen korisniku !**

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

- niz pojedinačnih razgovora
- prema unaprijed dogovorenom planu i rasporedu → koordinator
 - Primjeri:  \Analiza\PlanRazgovora
- sporo i neučinkovito - organizacija svakog pojedinog razgovora
- neke razgovore (ponekad i seriju) treba ponoviti – nadopuna, proturječja
- broj razgovora ne može se unaprijed točno odrediti - prilagoditi situaciji

Tehnika intervjuiranja (2)

□ 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 (3)

□ 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 (prikljicanje informacija o ...)
- Vođenje razgovora
 - postavljanje pitanja i bilježenje odgovora
 - prikupljanje dokumentacije
- Završetak razgovora
 - približno 5-10 minuta prije isteka planiranog vremena
 - provjera potpunosti rečenog
 - dogovara se nastavak razgovora (dopunski razgovor) kada:
 - nisu odgovorena sva pitanja ili je razgovor otvorio nova
 - treba proširiti krug sugovornika
 - zahvala na razgovoru

Tehnika intervjuiranja (4)

- **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.")
 - više analitičara – voditelj razgovora bude zapisničar, a ostali učestvanti primjedbe
 - Sadržaj zapisnika
 - Vrijeme i mjesto, sudionici, zapisničar
 - Sadržaj razgovora (tekst zapisnika)
 - Popis prikupljene dokumentacije
 - 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 sugovornicima, koji potvrđuju zapisano
 - vlastoručnim potpisom (!?), mailom (!!)
 - po potrebi sugovornici korigiraju tekst
- **Primjeri:**  \Analiza\Zapisnik-*

Preporuke za vođenje intervjeta

□ Treba pitati o svemu što se smatra važnim

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

□ Repertoar i vrste pitanja

■ Pitanja zatvorenog tipa:

- Koliko ... obrađujete (u nekom razdoblju)?
- Na koji način obrađujete ... ?
- Kako se popunjava ... ?

■ 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 intervjeta (2)

□ 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 najbolje rješenje, a ne podupirati neki proizvod
- pažnja i jezgrovitost
 - “brzo misli, jasno govori”
- izbjegavanje sugestivnih pitanja
- nemametanje zaključaka
- praćenje reakcija !

□ Grupno intervjuiranje

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

Radne sjednice (združeni razvoj)

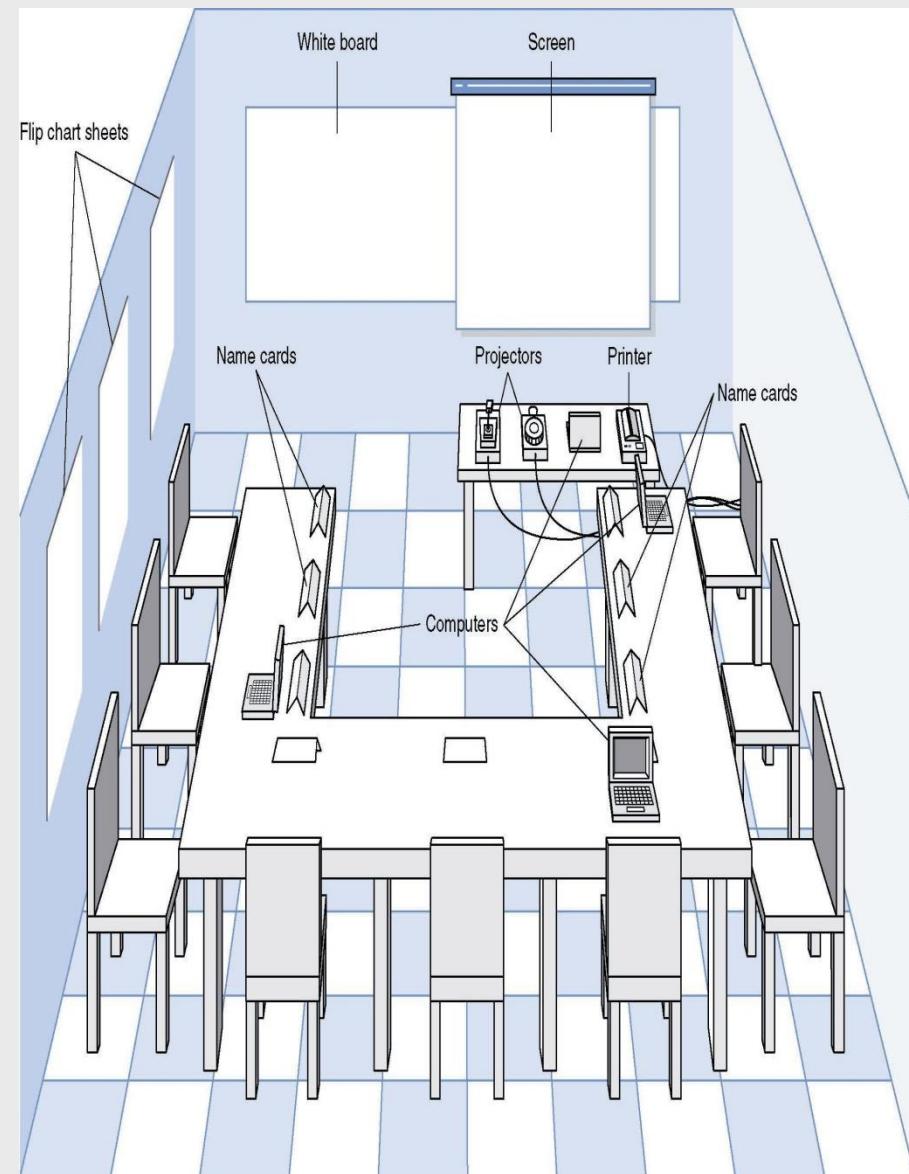
Slika: A.Dennis & B. Haley Wixom,
Systems Analysis and Design,
John Wiley & Sons, 2000

□ Radni sastanci, sjednice (workshop)

- analitičari i korisnici provode SA
- cilj - (zajedničko) pronalaženje rješenja
- poseban prostor i izolacija
- moderator, dnevni red i zapisnici

□ Genijalnost grupe (tzv. brainstorming)

- prikupljanje ideja i info. potreba
- aktivno i kreativno sudjelovanje
- svakog sudionika se stimulira da definira po njemu idealno rješenje
- svaki sudionik iznosi sve čega se sjeti, a odnosi se na problem
- od predloženih rješenja odabire se najbolje prema realnoj izvodljivosti
- postupak je učinkovit tamo gdje postoje korisnici koji dobro poznaju sustav ali teško prihvataju nove ideje



Radne sjednice (2)

□ Prednosti

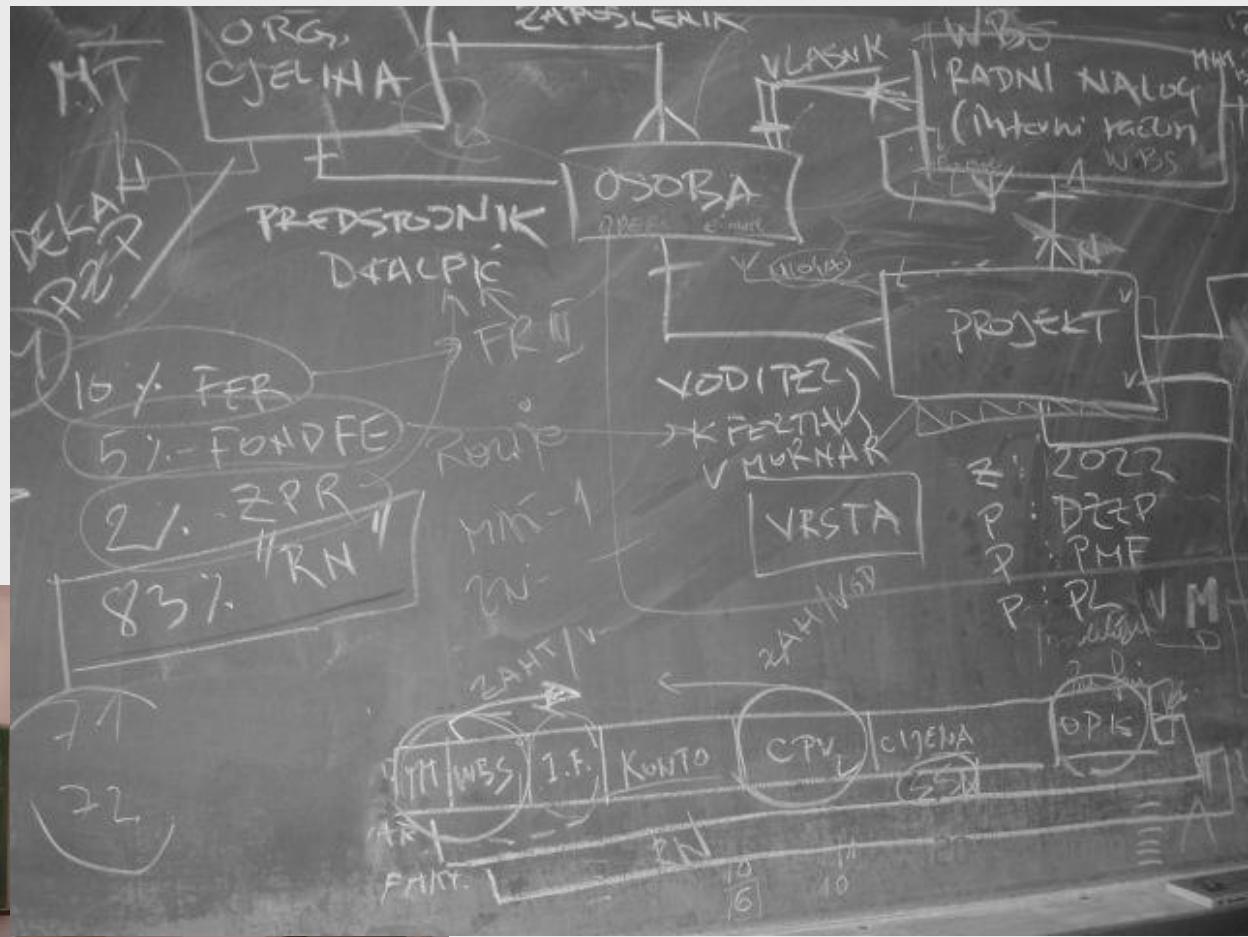
- rješavanje problema važnih za čitavu organizaciju ili veći dio poslovanja
- izbjegavanje specifičnih (egzotičnih) i nejasnih zahtjeva
- preciznije ustanovljavanje dosega projekta
- bolje uočavanje proturječnih zahtjeva

□ Nedostaci

- sadržaj i dinamika
 - pasivnost sudionika
 - “usitnjavanje” razgovora
 - udaljavanje od tema
- trajanje
 - više dana (5-10) u nekoliko tjedana
 - u praksi je teško udovoljiti zbog obveza sudionika
- otpor sjednici i pratećoj izolaciji
 - razmjeran je razini položaja pojedinog sudionika
 - izraženiji među korisnicima koji smatraju da je to informatički posao

□ Primjeri: Plastika, VisokoUčilište

□ IPIS visokih učilišta



□ Gigi ? Toni ? Vrt ?

Upitnici i ankete

□ Upitnik (pismeni intervju)

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

□ Anketa (inventar)

- može se obuhvatiti više ispitanika
- pitanja zatvorenog tipa – standardizacija obrade
- prikladna za popis resursa
- primjer:  \Analiza\Anketa-resursi

□ Intervjuiranje je elastičnije!

Analiza dokumentacije, proučavanje dokumenata

- **Prikuplja se sve do čega se može doći**
 - Analiza procesa
 - Tipični dokumenti: opis radnih procedura !?, pravilnici i zakoni
 - Analiza podataka
 - Tipični dokumenti: obrasci (formulari), izvješća
- **Vrijednost informacija dobivenih analizom (samo) dokumenata je niska**
 - praksa može odudarati od pravilnika i obrazaca
 - treba shvatiti zašto i kada dokumenti nastaju, kako se popunjavaju, koliko su potrebni
 - poželjno je da budu reprezentativni, tj. popunjeni na tipičan način, ali ...
 - reprezentativni dokumenti ne ukazuju na izuzetke - rjeđe bilježene podatke
 - stalno bilježenje nekih podataka ne mora značiti da su ti podaci stvarno potrebni
 - treba prikupiti više uzoraka iste vrste dokumenta (uzorkovanje) !
- **Primjeri:**  \Analiza\PrimjeriDokumenata\

Promatranje poslovnog sustava

- **Uvid u poslovne procese promatranjem radnih sredina**
 - promatranje lokacija, kretanja ljudi, koljanja dokumenata te tijek posla
- **Prednosti**
 - analitičar je u stanju realno sagledati poslovni proces
 - učinkovitost, ako se dobro provede
 - pouzdanost prikupljenih informacija
- **Nedostaci**
 - utrošak vremena
 - ometanje i neugoda promatranih osoba
 - mogućnost manipulacije promatrača
 - npr. prikrivanjem uobičajenog kršenja radnih procedura
 - problem određivanja dužine i učestalosti promatranja
 - informacije iz malog broja kratkotrajnih promatranja mogu biti nepouzdane
- **Promatranje je najteža metoda za pronalaženja činjenica.**
- **Primjer: Ministarstvo**

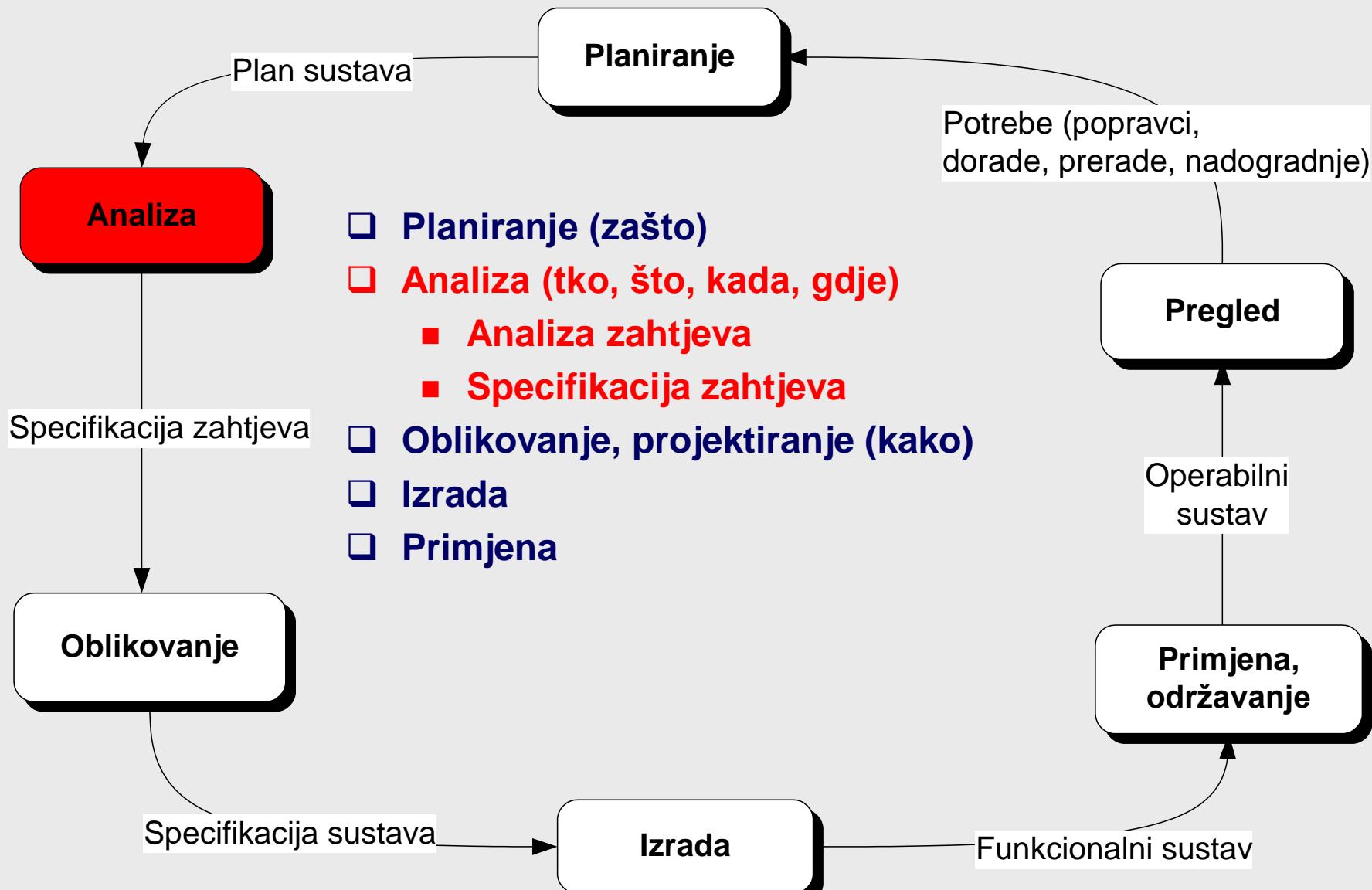
Izbor prikladne tehnike prikupljanja zahtjeva

- Tip informacije – područje primjene
- Dubina informacije – stupanj detaljizacije, mogućnost razumijevanja razloga
- Raspon informacije – širina, raznolikost
- Integracija informacije – mogućnost povezivanja ili provjere informacija dobivenih iz različitih izvora
- Sudjelovanje korisnika – stupanj uključenosti, angažiranost
- Trošak – organizacije, provedbe

	Intervju	Ankete	Analiza dokumenata	Udruženi razvoj	Promatranje
Tip informacije	as-is, pooboljšanja, to-be	as-is, pooboljšanja	as-is	as-is, pooboljšanja, to-be	as-is
Dubina informacije	veliki	srednji	mali	veliki	mali
Raspon informacije	mali	veliki	veliki	srednji	mali
Integracija informacije	mali	mali	mali	veliki	mali
Sudjelovanje korisnika	srednje	mali	mali	veliki	mali
Trošak	srednje	mali	mali	srednji do velik	mali do srednjeg

Analiza sustava

Kontekst analize 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

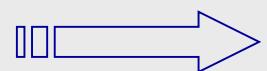
- Važno je shvatiti strukturu organizacije, tko u njoj radi, tko je kome podčinjen, kako surađuju različiti odjeli, itd.
- shema organizacijske strukture osobe i grupe koje obavljaju neki dio posla
→ Modeliranje funkcija

□ Analiza poslovanja

- Detalji poslovanja → Modeliranje procesa, modeliranje podataka

□ Svrha, cilj i dubina analize

- automatizacija poslovnih procesa,
- poboljšanje poslovnih procesa i
- preoblikovanje poslovnih procesa



Automatizacija poslovnih procesa

❑ Business Process Automation (BPA)

- Povećanje učinkovitosti korisnika primjenom računalne tehnologije

❑ Tehnike

- Analiza problema (problem analysis)
 - Otkrivanje problema sustava i predlaganje rješenja uz pomoć korisnika
 - Poboljšanja su mala i inkrementalna – učinkovitost, lakoća korištenja
- Analiza uzroka (root cause analysis)
 - Usmjereno na probleme (simptome), a ne na rješenja
 - Problemima se dodjeljuju prioriteti te traže svi mogući uzroci
 - Uzroci se analiziraju sve dok se ne utvrди pravi uzrok
 - Prednost imaju uzroci koji izazivaju više problema

Primjer: automatizacija poslovnog procesa

□ Primjer: Izrada putnih naloga

- izrađuje se 2700 putnih naloga godišnje
 - nalog za putovanje odobravaju voditelj, predstojnik te dekan
 - urudžbiranje (2 djelatnice) i knjiženje te isplata predujma (1 djelatnica)
 - po povratku putnik popuni zapisnik, nalog se obračunava (1 djelatnica) te se na blagajni (1 djelatnica) isplaćuje razlika / plaća povrat
- u pojedinim razdobljima broj putovanja značajno je povećan
 - gomilanje naloga u pisarnici
 - česte gužve na blagajni

□ Rješenja:

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

Poboljšanje poslovnih procesa

□ Business Process Improvement (BPI)

- manje promjene radi boljeg iskorištenja tehnologije i učinkovitosti/djelotvornosti

□ Tehnike

- Analiza trajanja (duration analysis)
 - Analiza vremena po svim koracima procesa
 - Integracija i paralelizacija procesa koje obavlja više ljudi
 - Integracija – smanjenje glavnog procesa da manje ljudi radi na ulazima
 - Paralelizacija – povećanje istovremenosti koraka
- 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 poboljšanja: integracija poslovnog procesa

- **Primjer: obračun nastave na poslijediplomskom studiju**
 - Stanje: velik broj sitnih transakcija ugovaranja i isplate, uz prijepis
 - po obavljenoj nastavi Studentska dostavlja evidenciju predavanja
 - izvođači zasebno potpisuju autorski ugovor u Pravnoj službi
 - nezavisni ugovori za vođenje (mentorstvo) i obrane doktorata
 - sve isplate temeljem ugovora provodi Računovodstvo

- **Poboljšanje:**
 - jednom godišnje (za nastupajuću ak. godinu) izrađuje se autorski ugovor s općim uvjetima - uključeni svi predvidivi izvođači
 - primjerak potписанog ugovora dostavlja se u Studentsku službu
 - naknadno angažirani izvođači potpisuju dodatak ugovoru
 - Studentska služba po obavljenom poslu evidentira obavljeno te generira nalog za isplatu s pozivom na broj autorskog ugovora

Primjer poboljšanja: 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

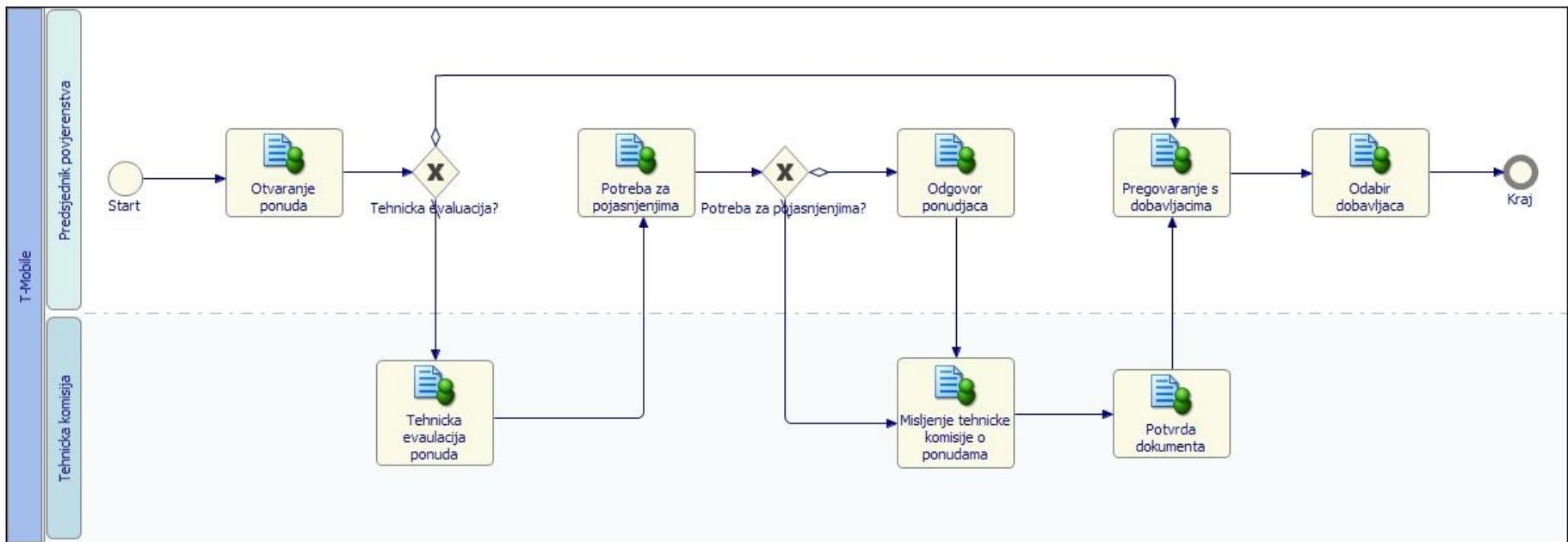
Preustroj, preoblikovanje poslovnih procesa

- **Business Process Reengineering (BPR)**
- **Business Process Redesign (BPR)**
 - radikalni redizajn analizom posljedica, procjenom alternativnih tehnologija, ukidanjem ili zamjenom aktivnosti, analizom troškova-koristi i analizom rizika
- **Tehnike**
 - Analiza posljedica (outcome analysis)
 - Razumijevanje rezultata koji predstavljaju vrijednost za korisnika
 - procjena što bi se moglo ponuditi krajnjim korisnicima
 - Analiza tehnologije (technology analysis)
 - Analitičari i voditelji razvijaju listu važnih i zanimljivih tehnologija
 - 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
- proces se sastoji od otvaranja ponuda, tehničke evaluacije, slanja pitanja ponuđačima, definiranja mišljenja tehničke komisije, ... i odabira dobavljača.
- predsjednik povjerenstva nabave (15 ljudi) i tehnička komisija (400 ljudi).



Primjer: heuristički preustroj procesa (2)

- simulira se izvođenje postojećeg procesa,
- vrši se preoblikovanje
 - npr. prepostaviti će se da nije potrebno tražiti dodatna pojašnjenja od ponuđača, jer su zahtjevi dobro definirani
- te se simulira izvođenje preoblikovanog 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 procese, poboljšava učinkovitost
- BPI poboljšava i učinkovitost i djelotvornost
- BPR radikalno poboljšava prirodu poslovanja

Trošak projekta (project cost)

- BPA ima ograničen doseg
- BPI ovisi o dosegu projekta
- BPR skup zbog utroška vremena i napora preoblikovanja

Doseg analize (breadth of analysis)

- BPA obično za pojedinačni proces
- BPI nekoliko procesa
- BPR vrlo širok doseg

Rizik pogreške (risk of failure)

- BPA mali do umjeren rizik jer je sustav jasan
- BPI mali do umjeren rizik jer je sustav jasan
- BPR slabo predvidiv i podložan riziku te se provodi uz odobrenje



Preporuke za analizu

Slika: Awad, 1985;
preveo Anonymous
(WWW)

□ Ispravni pristup

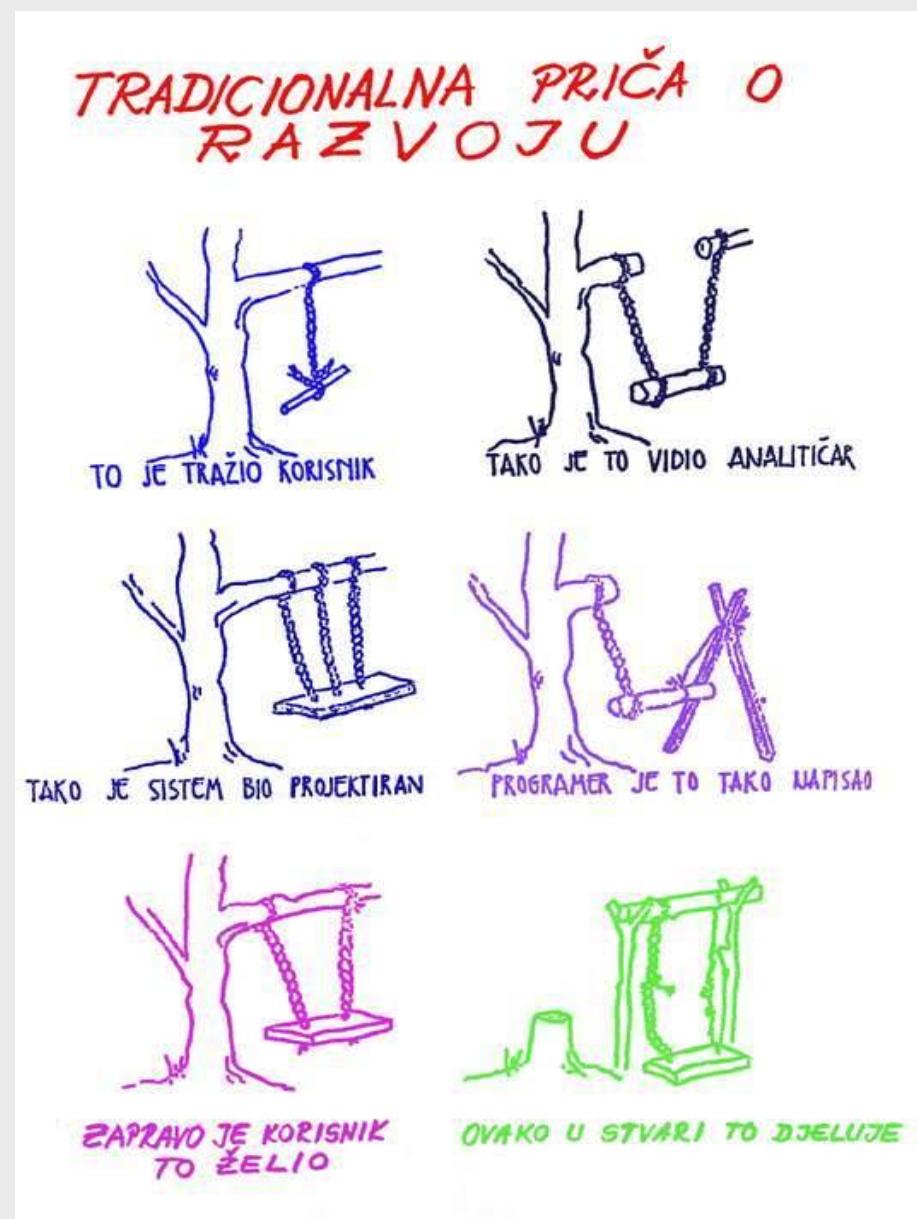
- analiza mora biti prilagođena korisniku
- ključno je razumjeti suštinu organizacije i način poslovanja
- Primjer: \Analiza\PreporukeZaAnalizu

□ korisnik želi <> korisnik treba

- umješnost je uvjeriti ga da je „sve sam smislio“
- da se priča o razvoju ne bi ponavljala - modeliranje

□ Paraliza analize

- analysis paralysis, paralysis of analysis
- vječito modeliranje „dok ne bude dobro“
- ne pretjerivati s modeliranjem postojećeg, onog što će biti izmijenjeno ili zamijenjeno



Inženjerstvo zahtjeva

Requirements engineering

Zahtjevi

□ Zahtjev – prema ISO / IEC / IEEE

- Systems and software engineering — Vocabulary [ISO/IEC/IEEE 24765:2010]

- (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.

□ Zahtjevi ne smiju sadržavati detalje dizajna ili implementacije

- ali ako su nepotpuni onemogućuju planiranje projekta i praćenje promjena

□ Ključni problemi [Leffingwell, 1997]

- 40-60% pogrešaka projekta uzrokovano je pogreškama postavljanja zahtjeva
- posljedica - "neispunjena očekivanja" (expectation gap)

- Naknadne prepravke mogu predstavljati do 40% troškova razvoja,
- od čega je 70-85% pripisivo pogrešnim zahtjevima

Vrste zahtjeva

□ Poslovni zahtjevi (zašto)

- ciljevi organizacije ili zahtjevi rukovodstva, poslovni problemi ili potrebe
- opisani u viziji i dosegu projekta
 - primjer: poslovna potreba, „Pridobivanje novih kupaca“
 - 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

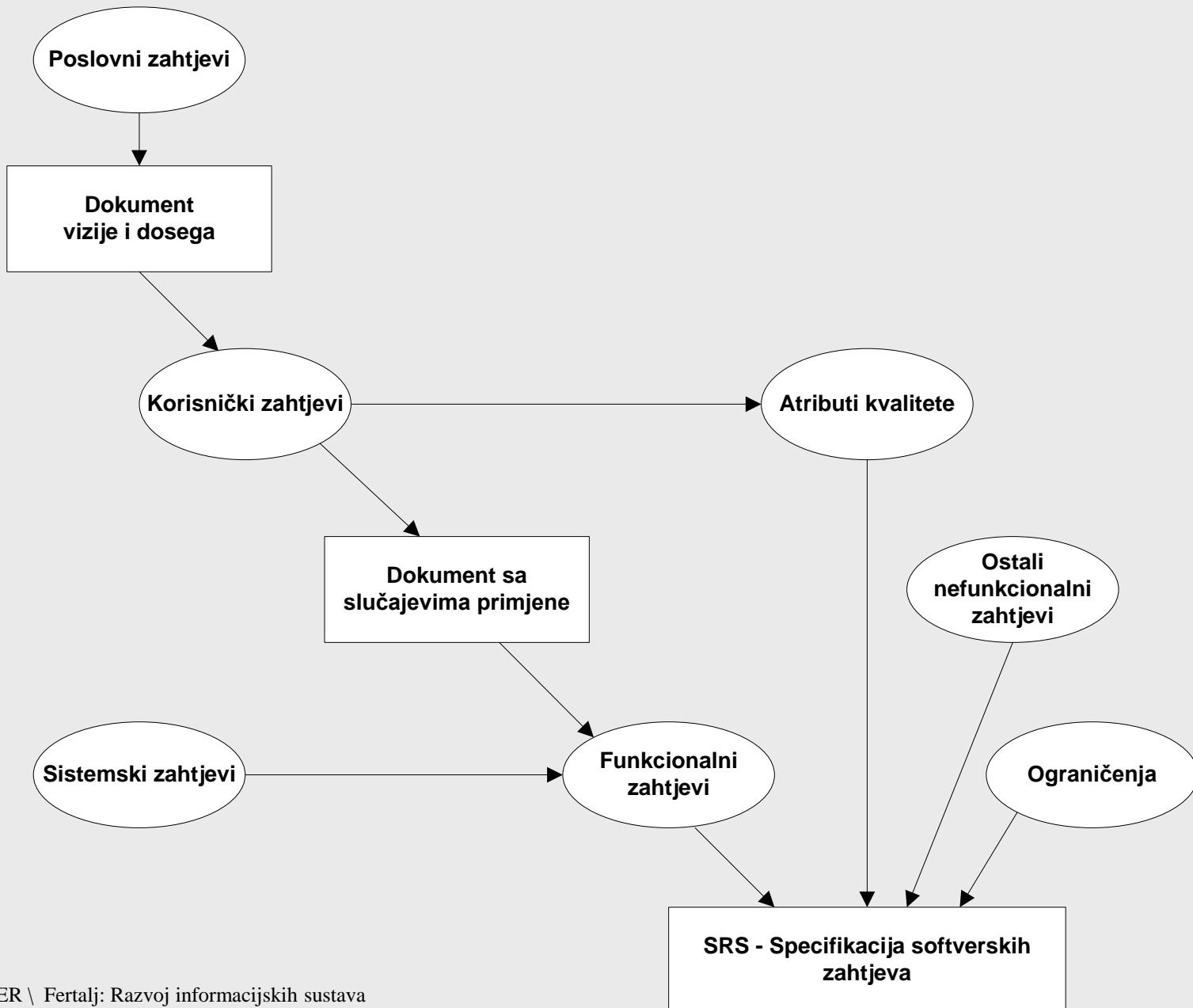
□ Funkcionalni zahtjevi (što)

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

□ Nefunkcionalni zahtjevi (kako ili kako dobro)

- sistemski zahtjevi, atributi kvalitete, ostali nefunkcionalni zahtjevi
- standardi, pravila i ugovori, opisi vanjskih sučelja, zahtjevi na performanse, ograničenja na dizajn i implementaciju ...

Kontekst postavljanja zahtjeva



Primjeri poslovnih zahtjeva

□ Očekivana novčana ušteda

- prava na subvencioniranu prehranu može koristiti samo student koji ih je stekao te 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

□ Idealni slučaj - zahtjevi naručitelja odgovaraju 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 ...*

□ Ukinuti plaćanje unaprijed

- *Treba izbjegići bilo kakvo plaćanje od strane studenata za potrebe ostvarivanja prava, a posebice unaprijed.*

□ Ukloniti nepotrebne postupke za ostvarivanje prava

- *Kad se studentu jednom zavedu prava na maticnoj ustanovi, nisu potrebna nikakva daljnja dokazivanja za ostvarivanje prava kod pružatelja usluga.*

□ Smanjiti rizik gubitka ostvarenih prava

- *Sustav mora onemogućiti zloporabu 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: 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 moguć
 - Storno nužno ne poništava cijeli iznos, nego ga korigira za iznos storna, ...
- 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.

□ Funkcionalni zahtjevi često sadrže poslovna pravila !

Sistemski zahtjevi (1)

□ Operativni zahtjevi

- Tehničko okruženje – HW, SW, mrežna infrastruktura
- Integracija sustava – interoperabilnost, npr. uvoz/izvoz Excel ili web servis
- Prenosivost – rad na različitim sustavima, npr. Linux, Windows
- Održavanje – promjene kojima se treba prilagoditi, pr. mjesecne verzije

□ Zahtjevi za performansama

- Brzina – obrade, odziva, pr. 7"za bilo koju transakciju, sink. svakih 30'
- Zahtjevi za kapacitetom – broj i promet korisnika, količina podataka
 - Vršno 100-200 istovremenih, 10k podatka/transakcija, ukupno 5000 korisnika, 2TB podataka
- Dostupnost i pouzdanost – 24x7, planirano održavanje 6h/mjesec

Sistemski zahtjevi (2)

□ Zahtjevi na sigurnost

- Procjene vrijednosti sustava – vrijednost sustava i podataka
 - Sustav nema kritičnu misiju, ali ispad košta 50 kkn/h
 - Potpuni gubitak podataka procjenjuje se na 20 Mkn
- Zahtjevi za kontrolu pristupa – ograničenje na pristup podacima
 - Voditelji mogu ..., operateri mogu ... ili anon/regi/admin mogu ...
- Zahtjevi za enkripcijom i autentifikacijom – kako, gdje i kada
- Zahtjevi za kontrolom virusa

□ Kulturni i politički zahtjevi

- Višejezičnost – podržani jezici na korisničkom sučelju
- Prilagodba – *customizacija*
 - pr. voditeljima projekata treba omogućiti definiranje korisničkih atributa
 - pr. omogućiti pohranu izgleda izvješća (*layout*)
- Eksplicitni navod postavki – pr. svi datumi formata *dd.mm.yyyy*.
- Pravni zahtjevi – zakoni i pravila, npr. Zakon o PDV-u, ZoZDiVO

Primjer: sistemske i ostale nefunkcionalne zahtjeve

□ Primjer: PISFER-Podprojekt

- sve informacije treba moći unijeti na mjestu nastanka
- omogućiti da sve organizacijske jedinice imaju pravo pristupa uz odgovarajuće dozvole
- omogućiti kontinuirani pristup podacima (ne samo krajem mjeseca)
- jednostavno i intuitivno korištenje sustava
- korištenje jedinstvene baze podataka za sve službe

□ Primjer: Viševalutni rad

- Devizni izvod poslovne banke sadrži iznose po tečaju poslovne banke.
- Za devizne uplate evidentirati će se uz kunski iznos i devizni iznos te valuta.

□ Primjer: Praćenje promjena nad podacima

- svaki zapis sadrži identifikator korisnika i datum unosa/promjene

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 (1)

Dostupnost (availability)

- postotak predviđenog vremena tijekom kojeg sustav treba biti funkcionalan
- 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 (efficiency)

- Stupanj iskorištenja resursa (procesor, disk, memorija)
- Primjer: „70% procesorskog kapaciteta i 85% memorije sustava će biti raspoloživo za vrijeme maksimalnog opterećenja sustava.“
- Potrebno je razmotriti minimalne hardverske konfiguracije

Prilagodljivost (adaptability)

- Lakoća dodavanja novih mogućnosti
- Važno za proizvode koji se rade inkrementalno

Zahtjevi na kvalitetu programske podrške (2)

□ Integritet (integrity)

- Integritet (ili sigurnost) – neovlašteni pristup, virusi, tajnost podataka, ...
- Primjeri: role-based security, single sign-on (SSO), ...

□ Interoperabilnost (interoperability)

- razmjena podataka ili usluga s drugim sustavima.
- koje druge aplikacije se namjerava koristiti i koji podaci će se razmjenjivati.
- Primjeri: „uvoz iz Accessa, izvoz u Excel”

□ Pouzdanost (reliability)

- vjerojatnost da će softver u nekom razdoblju raditi bez pogreške
- postotak ispravno izvedenih operacija, trajanje rada bez pogrešaka ili učestalost pogrešaka.
- proizvod koji zadovolji zahtjeve za pouzdanošću je isporučiv unatoč defektima
- Primjer: „Maksimalni očekivani mjesecni broj pogrešaka nakon zakrpe je 32”

Zahtjevi na kvalitetu programske podrške (3)

□ Robustnost (robustness)

- stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pogrešnih podataka, defekata u komponentama ili nepredviđenim operativnim uvjetima
- primjer: „pri učitavanju iz datoteke sustav će za redak s pogreškom prijaviti pogrešku i broj retka te pitati korisnika da li da nastavi od sljedećeg retka ili da otkaže učitavanje“

□ Upotrebljivost (user friendliness)

- mjera napora za pripremu, obradu ili tumačenje podataka, to jest lakoću korištenja
- krivulja učenja, korištenje početnika / iskusnih korisnika
- sukladnost standardima ili uzoru: "Sve kratice na u izborniku *File* će biti iste kao ..."

□ Lakoća održavanja (maintainability)

- mjera lakoće popravka pogreške ili izmjene
- prosječno vrijeme da se riješi problem i postotak obavljenih popravaka
- primjer: "Izvješće će biti izmijenjeno u roku tjedan dana od zaprimanja obrazaca"

Zahtjevi na kvalitetu programske podrške (4)

□ Prenosivost (portabilnost)

- napor prijenosa iz jedne operativne okoline u drugu
- projektantski principi slični iskoristivosti
- dijelovi koji moraju biti prenosivi te ciljane okoline

□ Ponovna upotrebljivost (reusability)

- mjera upotrebe u drugim aplikacijama.
- razvoj ponovno iskoristive komponente znatno skuplji
- softver mora biti modularan, dokumentiran, donekle općenit

□ Podložnost testiranju

- lakoća provjere softverske komponente
- ključna kod proizvoda sa složenim logikom obrade
- važno kod proizvoda koji će se često mijenjati - regresijsko testiranje

Među zavisnost zahtjeva

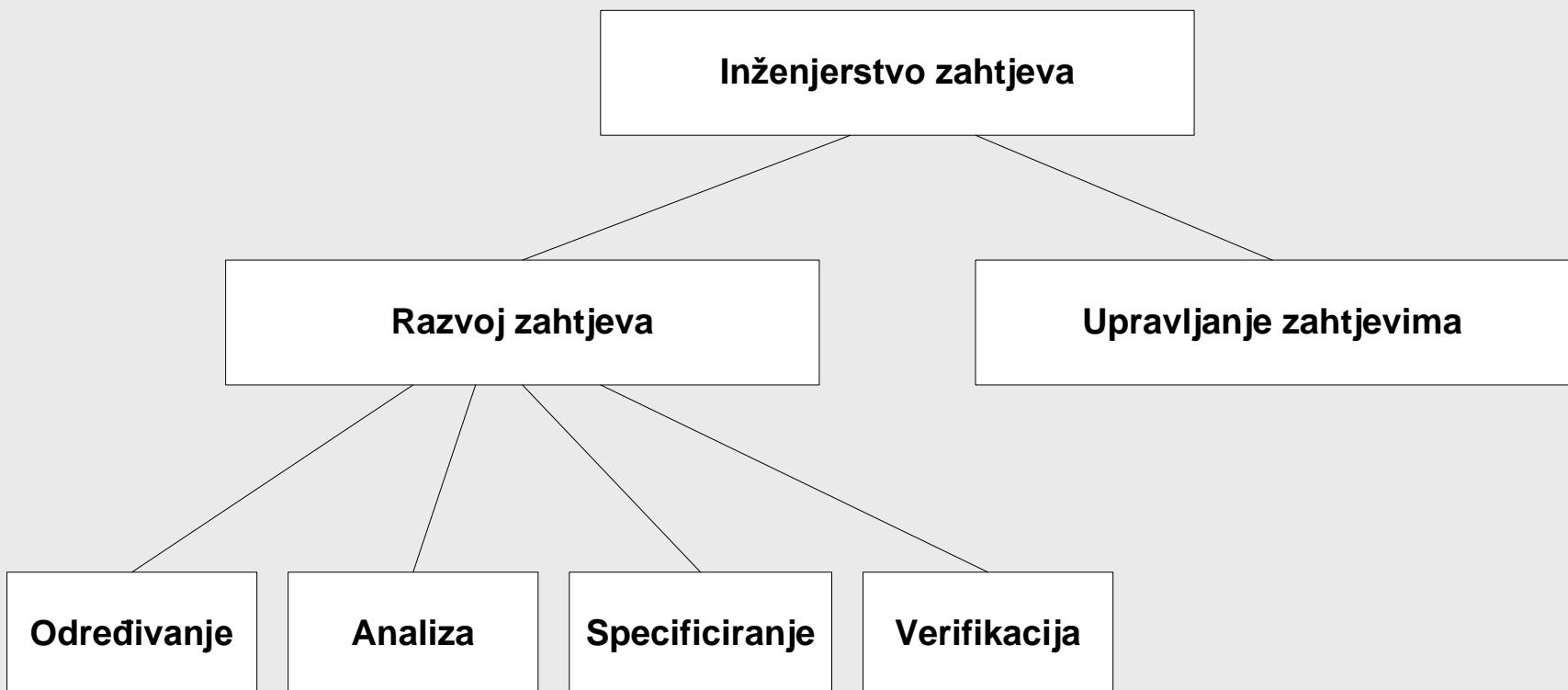
	Dostupnost	Učinkovitost	Prilagodljivost	Integritet	Interoperabilnost	Lakoća održavanja	Prenosivost	Pouzdanost	Ponovna isupotrebljivost	Robustnost	Podložnost testiranju	Upotrebljivost
Dostupnost												
Učinkovitost			-		-			-		-		-
Prilagodljivost		-		-		+		+		+		
Integritet		-			-				-		-	-
Interoperabilnost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

Inženjerstvo zahtjeva

Inženjerstvo zahtjeva

□ Inženjerstvo zahtjeva (requirements engineering)

- razvoj zahtjeva + upravljanje zahtjevima.
- razvoj zahtjeva: određivanje (iznalaženje) zahtjeva (elicitation), analiza, specificiranje (dokumentiranje) i verifikacija



Razvoj zahtjeva i upravljanje zahtjevima

□ Razvoj zahtjeva

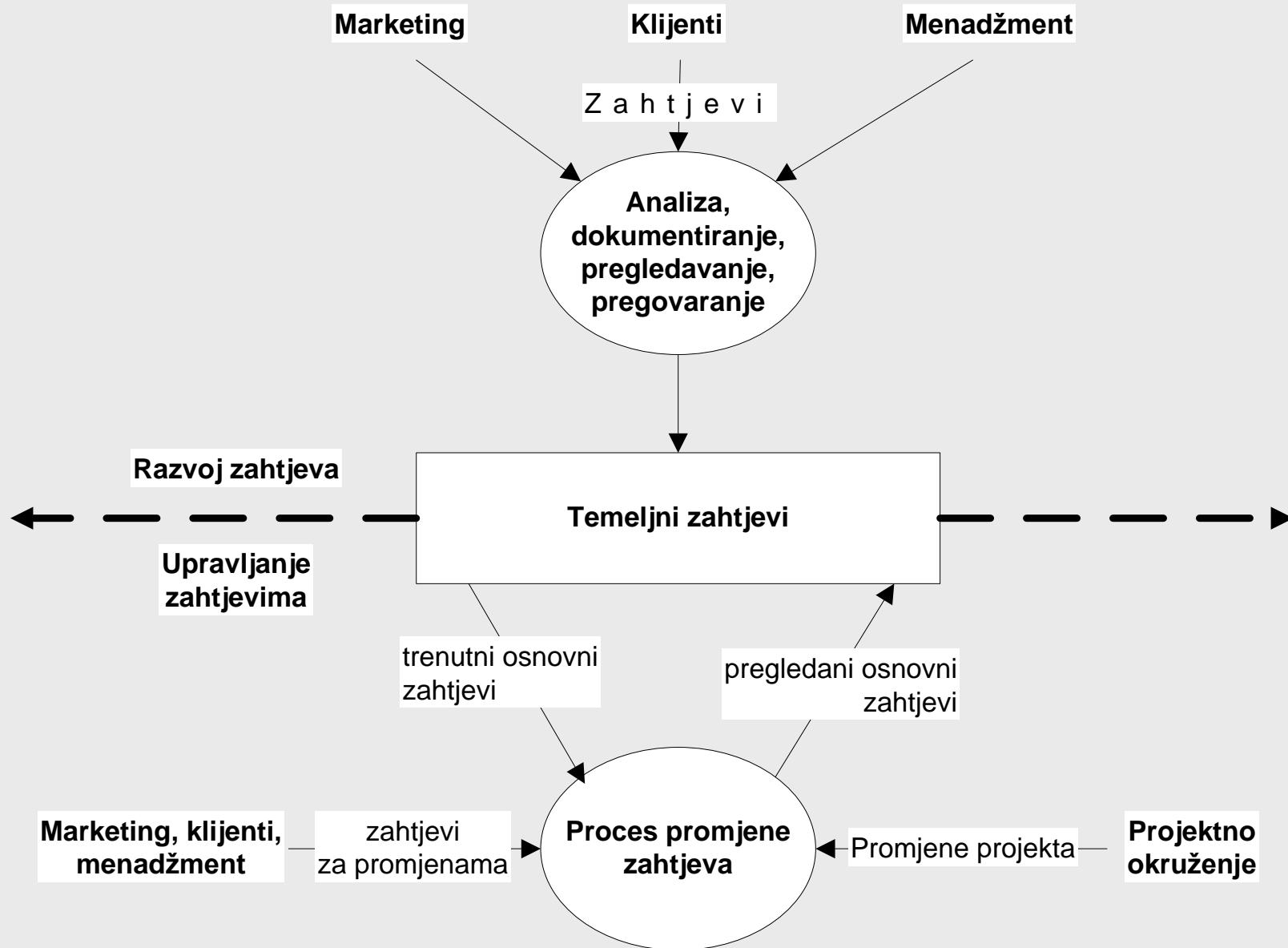
- Razredi korisnika ("najbolji", ključni)
- Poslovne potrebe, zadaci i ciljevi
- Analiza informacija, kategorizacija
- Razdioba s razine sustava na podsustave i udio u komponentama
- Razumijevanje relativne važnosti (težine) svojstava kvalitete
- Određivanje prioriteta ugradnje
- Modeliranje zahtjeva
- Dokumentiranje zahtjeva
- Provjera specifikacije zahtjeva

□ Upravljanje zahtjevima

- Rokovi prihvatanja zahtjeva
- Pregled promjena zahtjeva, procjena utjecaja, odluka o (ne)prihvatanju
- Kontrolirana ugradnja promjena
- Ažuriranje plana projekta
- Pregovori o rokovima isporuke
- Praćenje i povezivanje zahtjeva s izvornim kodom i primjerima za test
- Praćenje statusa aktivnosti i promjena zahtjeva tijekom projekta

□ Korisnik se mora moći predomisliti, ali to ne može trajati zauvijek

Razvoj zahtjeva naspram upravljanja zahtjevima



Određivanje zahtjeva (kako razlučiti)

□ Poslovni zahtjevi

- Sve što opisuje financijski ili drugi poslovni probitak za korisnike ili organizaciju, je najvjerojatnije poslovni zahtjev

□ Slučajevi korištenja ili scenariji

- Općenite izjave o poslovnim zadacima korisnika – slučajevi korištenja
- Specifični opisi zadataka - korisnički scenariji
- Specifične zadatke treba generalizirati u općenite slučajeve korištenja

□ Poslovna pravila

- izjava da neku aktivnost mogu obavljati samo pojedine osobe ili uloge, ili da se aktivnost obavlja pod određenim uvjetima
- operativni principi poslovnih procesa - nisu funkcionalni zahtjevi

Određivanje zahtjeva (2)

□ Funkcionalni zahtjevi

- Izjava „Korisnik mora moći ...“ ili „Sustav treba ... neko ponašanje“
- opisuju vidljivo ponašanje sustava i definiraju što će sustav raditi.
- **funkcionalni zahtjevi ponekad predstavljaju zastarjele/neučinkovite procese!**
 - treba biti jasno zašto sustav nešto „mora“

□ Atributi kvalitete

- vrsta nefunkcionalnih zahtjeva
- Izjave kako dobro sustav obavlja funkciju
- karakteristike: „brzinu, jednostavnost, intuitivnost, ...“ treba razjasniti s korisnicima

□ Zahtjevi vanjskih sučelja

- veze između sustava i vanjskog svijeta
- mehanizmi komunikacije za korisnike, hardver i druge sustave

Određivanje zahtjeva (3)

□ Ograničenja

- Uvjeti koji ograničavaju izbor rješenja
- Spadaju u nefunkcionalne zahtjeve
- Neka ograničenja mogu pomoći u zadovoljavaju atributa kvalitete
 - Primjer: prenosivost korištenjem samo standardnih naredbi

□ Definicije podataka

- opis formata, dozvoljenih vrijednosti, pretpostavljenih vrijednosti ili struktura

□ Ideje o rješenju

- Ako korisnik opisuje način interakcije
 - npr. „Kad kliknem, otvor se menu, ... a tu mi ugradi NN” (T.Nikolić)
- Prijedlog rješenja nije zahtjev!
- Usmjeriti se na ono što je potrebno obaviti, a ne na način realizacije
- Istražiti zašto korisnik predlaže određenu ugradnju – razumijevanje potrebe



Najčešći problemi pri određivanju zahtjeva

Taktika kuhinjskog sudopera

- korisnik navodi (preko)brojne potrebe: permutacije izvještaja, formi, sortiranja, ...
- pristup uzrokovani pomanjkanjem iskustva korisnika koji ne zna razlučiti bitno
- pristup: analiza, "normalizacija", prilagodljiv GUI / generika (dizajn)

Taktika dimne zavjese

- korisnik traži više mogućnosti nego što treba
- skriveni interes: postizanje bolje nagodbe ("neka se ima")
- pristup: redukcija na realne i izvedive zahtjeve, postavljanje prioriteta

Taktika "Treba mi isto ali bolje"

- korisniku nedostaje volja ili znanje, a ponekad oboje
- pristup: ustrajnost, iznošenje primjera, prototipiranje

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 ...)" - pravilo 80/20

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

- softversko "betoniranje" loših navika – "politika kuće" (više riječi kasnije)

Analiza zahtjeva

□ pročišćavanje, proučavanje i preispitivanje zahtjeva

- da budemo sigurni da svi sudionici razumiju što zahtjevi znače
- ujedno služi provjeri zahtjeva

□ Izrada dijagrama konteksta sustava

- jednostavni model – granice i sučelja + tok informacija i materijala

□ Izrada prototipova korisničkog sučelja

- model oponašanja kao sredstvo razjašnjenja zahtjeva

□ Analiza ostvarivosti zahtjeva

- procjena troškova, performansi, rizika, konflikata prema ostalim zahtjevima

□ Postavljanje prioriteta

- relativni prioritet ugradnje za slučaj korištenja, svojstvo proizvoda ili drugi zahtjev

□ Modeliranje zahtjeva

- izrada modela strukture, ponašanja i stanja

□ Izrada rječnika podataka

- središnji repozitorij za definiranje struktura podataka i zajedničku terminologiju

Analiza i modeliranje sustava

- **Na temelju izjava korisnika i prikupljene dokumentacije modeliraju se pojedine komponente sustava (procesi, podaci, događaji)**
 - preslikavanja imenica i glagola u elemente modela (n puta 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) - Ključno imati ?

- Sustav koji nema definirane nužne zahtjeve ne može ispuniti svoju svrhu
- Postoji tendencija da se previše zahtjeva proglaši nužnim!
- Testirati svaki zahtjev i nastojati ga rangirati
 - Ako se zahtjev može rangirati onda nije obvezan!

□ Poželjno svojstvo (should have) – Ispuniti do kraja projekta ?

- Međuverzije sustava mogu biti funkcionalne bez tih zahtjeva
- Poželjni zahtjevi mogu i trebaju biti rangirani.

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

- "bilo bi lijepo, ali se može bez"
- nisu pravi zahtjevi, ali mogu biti rangirani i odrediti kvalitetu

□ Nepotrebna svojstva (won't have)

Primjeri rangiranja prioriteta

oznaka	značenje	referenca
visok	<i>mission-critical</i> zahtjevi, potrebni za sljedeće izdanje (<i>release</i>)	Wiegers 99
srednji	podrška sistemskim operacijama; zahtijevano, ali može se odgoditi za sljedeće izdanje	
nizak	poboljšanje funkcionalnosti ili kvalitete	
neophodan	proizvod neprihvatljiv ukoliko ovi zahtjevi nisu ispunjeni	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>)	

Fertalj: 0, 1, 2, 3, 5, 9

Specifikacija 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
- ugradbeno nezavisan pogled na sustav
 - funkcionalni i nefunkcionalni zahtjevi te njihovi prioriteti
 - model organizacijske strukture (struktturni dijagrami)
 - opis protoka dokumenata (dijagrami toka rada)
 - model procesa (dijagram toka podataka, dijagrami slučajeva korištenja)
 - konceptualni model podataka (dijagram entiteti-veze)

Predložak specifikacije zahtjeva

- **Predložak** \Predlošci\SpecifikacijaZahtjeva – temeljem IEEE 830
 - specifikacija zahtjeva sadrži osnovicu zahtjeva
 - kasnije promjene zahtjeva ulaze u proces upravljanja promjenama

1. Uvod

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

2. Sveobuhvatni pregled

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

3. Zahtjevi za sučeljem

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

4. Svojstva sustava

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

5. Ostali nefunkcionalni zahtjevi

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

6. Ostali zahtjevi

- Dodatak A: Rječnik
- Dodatak B: Modeli i dijagrami
- Dodatak C: Lista nedovršenih/neodređenih zahtjeva

Označavanje zahtjeva

□ Uzastopni brojevi

- novi zahtjev dobiva raspoloživi serijski broj (npr. Z01, Z02, ... ili FZ01,...).
- nepregledno, teško za pratiti veći broj zahtjeva, nepraktično brisanje

□ Hjerarhijsko numeriranje (X.Y.Z)

- jednostavno uređivanje (Word), problem brisanja koje uzrokuje posmak
- poboljšanje: tekstovne oznake unutar brojčanih hjerarhija
 - Primjer: "3.2. - Funkcije editora", sa zahtjevima "ED-01", "ED-02", itd.

□ Hjerarhijske tekstovne oznake

- „objektno”, npr. ISPIS.KOPIJE.POTVRDA
- prednost - strukturiranost, jednostavnost, nedostatak: nezgrapnost

□ Potpunost

- nijedan zahtjev ili informacija ne smiju nedostajati – teško uočljivo
- kontrola „praćenjem” poslovnog procesa
- nepotpune posebno označiti (npr. TBD - "to be determined")
 - razriješiti prije ugradnje

Verifikacija zahtjeva

- ovjera da su zahtjevi precizni, potpuni i da zadovoljavaju traženo

□ Provjera dokumentacije sa zahtjevima

- jedna od najkorisnijih softverskih tehnika
- mikro ekipa (analitičari, projektanti, ... , korisnici) ispituje specifikacije i modele

□ Pisanje testova, probnih slučajeva

- prototipiranje, izvođenje scenarija korištenja – provjera očekivanog ponašanja
- slučajeve povezati s funkcionalnim zahtjevima da se osigura potpunost

□ Pisanje korisničkog priručnika

- skica korisničkog priručnika - kao specifikacija ili dio analize
- dobre upute opisuju svu vidljivu funkcionalnost – ostalo je u SRS

□ Definiranje kriterija prihvatljivosti

- definiranje (s korisnicima) uvjeta pod kojima će proizvod zadovoljiti zahtjeve
- na osnovu slučajeva/scenarija korištenja

Upravljanje zahtjevima

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

Upravljanje zahtjevima (2)

□ Uspostava vremenske osnove (baseline) i kontrole verzija

- definiranje slike dogovorenih zahtjeva u određenom trenutku
- nakon osnove, zahtjevi prolaze postupak promjena
- verzioniranje specifikacije – uklanjanje neodređenosti

□ Praćenje povijesti promjena zahtjeva

- zapis o vremenu, vrsti i sadržaju, razlozima, verziji i autoru promjene

□ Praćenje statusa zahtjeva

- status (*predložen, odobren, ugrađen, provjeren*) + statistika

□ Mjerjenje stabilnosti zahtjeva

- brojanje promjena – mjera da je problem shvaćen, opseg definiran

□ Korištenje alata za upravljanje zahtjevima

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

Reference

- Maciaszek L. Requirements Analysis and System Design: Developing Information Systems with UML, 3/ed, Addison Wesley Higher Education, 2007.
- Leffingwell, D., Widrig. D. Managing Software Requirements: A Use Case Approach (2nd Edition), Addison-Wesley Professional, 2003.
- http://www.jiludwig.com/Requirements_Management_Tools.html



□ Primjeri: DOO-Specifikacija, TK-upute

- dvije iteracije teksta, prototip, nekoliko? promjena pred isporuku

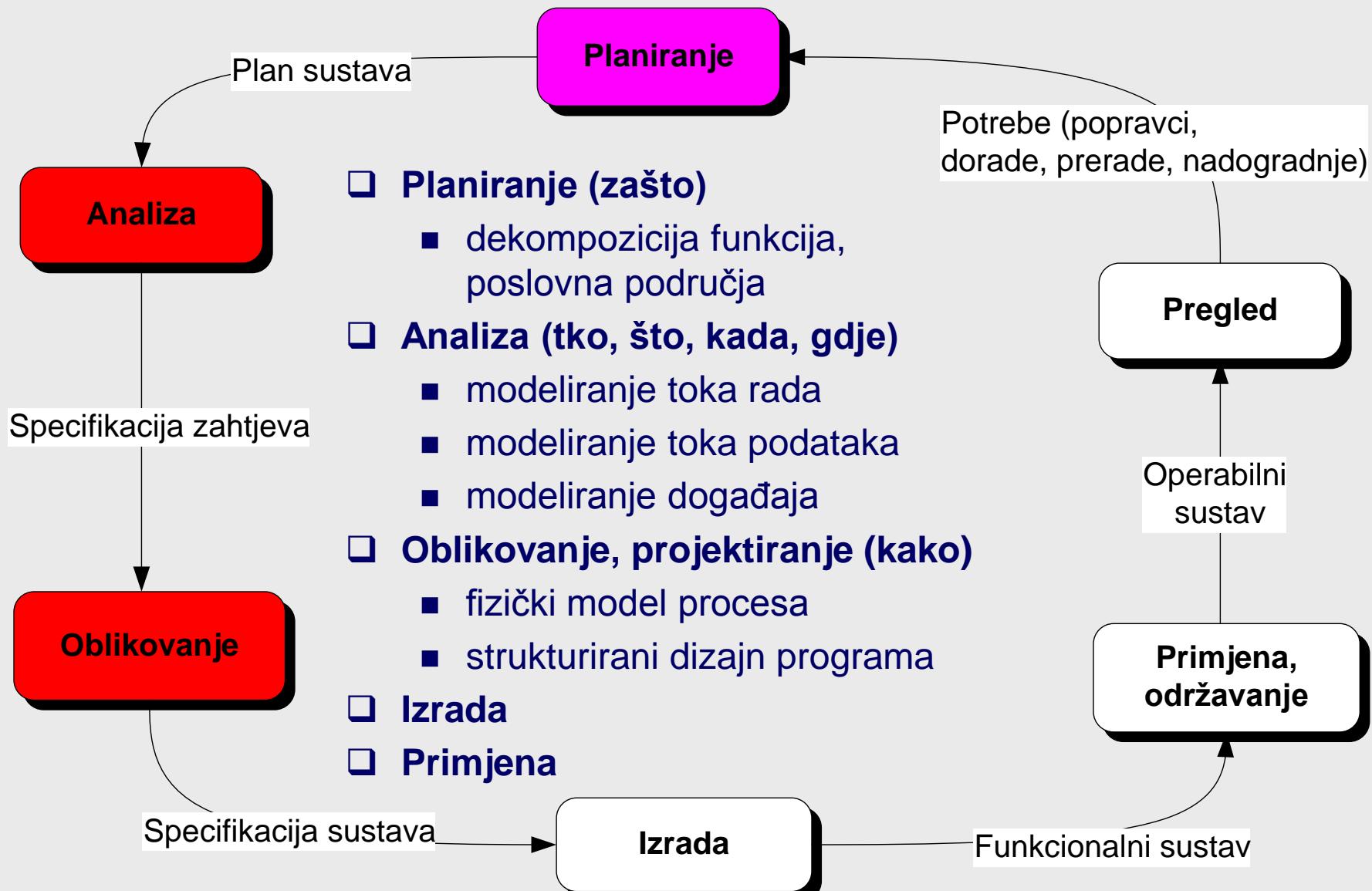
Strukturirana analiza i dizajn

2013/14.04

Sadržaj

- **Strukturirana analiza**
- **Dekompozicija procesa i funkcija**
 - Logički procesi
 - Poslovna pravila i poslovna politika
- **Modeliranje funkcija**
 - Dijagram dekompozicije funkcija
 - Hiperarhijski prikaz funkcija/procesa
- **Izrada dijagrama dekompozicije**
 - Dijagram organizacije
 - Razrada poslovnih procesa
- **Oblikovanje toka rada**
- **Modeliranje toka podataka**
 - Dijagram toka podataka
 - Pravila i ograničenja
 - Preporuke i metode
- **Modeliranje događaja**
 - Događaji i vrste događaja
 - Modeliranje vođeno događajima
 - Matrični prikaz modela događaja
 - Dijagram prijelaza stanja
 - Mape dijaloga

Strukturirana analiza i strukturirani dizajn



Strukturirana analiza

□ Moderna strukturirana analiza = Logički dizajn (česti sinonim)

- strukturirani proces i rezultati analize
- tehnika modeliranja poslovnih zahtjeva na sustav
 - usmjereni procesima, ali obuhvaća i podatke
- logički modeli – određivanje ŠTO je sustav i ŠTO mora raditi (ne KAKO)
 - dijagrami toka podataka za
 - prikaz zahtjeva nezavisno od tehničkih rješenja → **logički dizajn**
 - izražavaju suštinu sustava → esencijalni, konceptualni, poslovni modeli
- uključuje određivanje prioriteta zahtjeva

□ Postupak dekompozicije - strukturno raščlanjivanje

- podijeli pa s/vladaj (lat. divide et impera, eng. divide and conquer)
- hijerarhijski modeli, iterativno, s vrha prema dolje
 - funkcije i procesi, organizacija, podaci, softver

□ Aplikacijski model procesa = logički model procesa sustava ili aplikacije u fazi analize

Logički procesi

□ Funkcija (djelatnost, posao)

- skup logički povezanih trajnih poslovnih aktivnosti i zadataka
 - obavlja se stalno (nema određeni početak i kraj)
 - obavljaju osobe, grupe ili organizacijske cjeline
- primjeri: prodaja, proizvodnja, računovodstvo
- može se sastojati od desetina pa i stotina diskretnih procesa
 - hijerarhijsko razlaganje do razine procesa koji obavljaju neki zadatak

□ Događaj (poslovni događaj)

- logički dio posla koji se obavlja kao nedjeljiva cjelina → *transakcija*
 - pokretan diskretnim ulazom
 - završava nakon što proces odgovori odgovarajućim izlazom
- može se predstaviti procesom kojim sustav reagira na taj događaj
 - logički događaj dalje se razlaže do elementarnih procesa

Elementarni procesi

□ Proces (elementarni, primitivni proces)

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

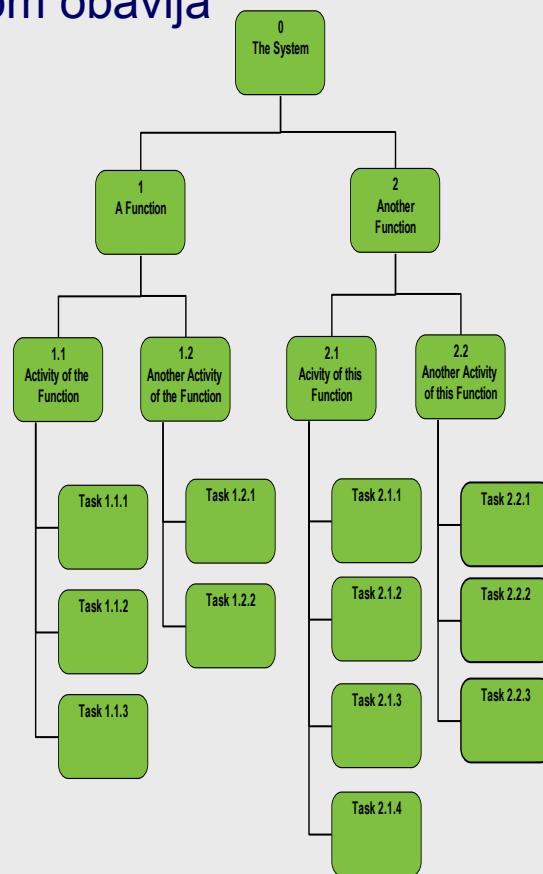
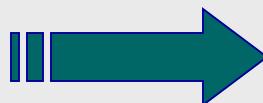
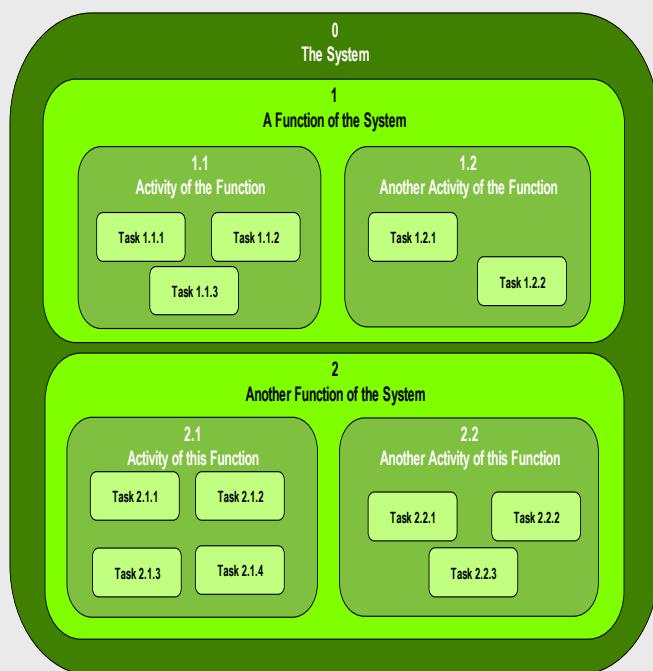
Poslovna pravila i poslovna politika

- **Poslovno pravilo - instrukcije i logika procedure 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**
 - *prvog u mjesecu dodjeljuju se prava ...*
 - *mjesечni iznos raspoloživ za subvenciju jednak je umnošku broja dana u mjesecu i dnevног subvencioniranog iznosa za razinu prava*
 - *Npr. za razinu 1 (domicilni) dnevni iznos je 16.70 Kn (cijena menija), pa student te razine ima raspoloživo u svibnju $16.70 * 31 = 517.7$ Kn.*
 - *subvencija se akumulira tijekom ak. godine - neiskorištena prava iz jednog mjeseca prebacuju se na slijedeći (!?)*
 - ...
 - **Kakav je to zahtjev ?**

Modeliranje funkcija

❑ Funkcionalna dekompozicija, dekompozicija funkcija

- izrada 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



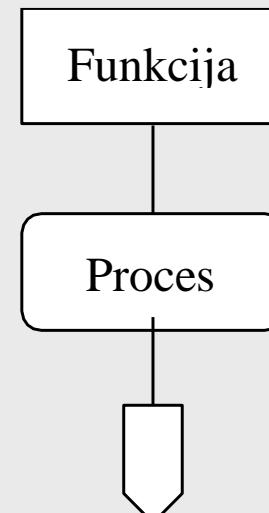
Dijagram dekompozicije funkcija

□ 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*

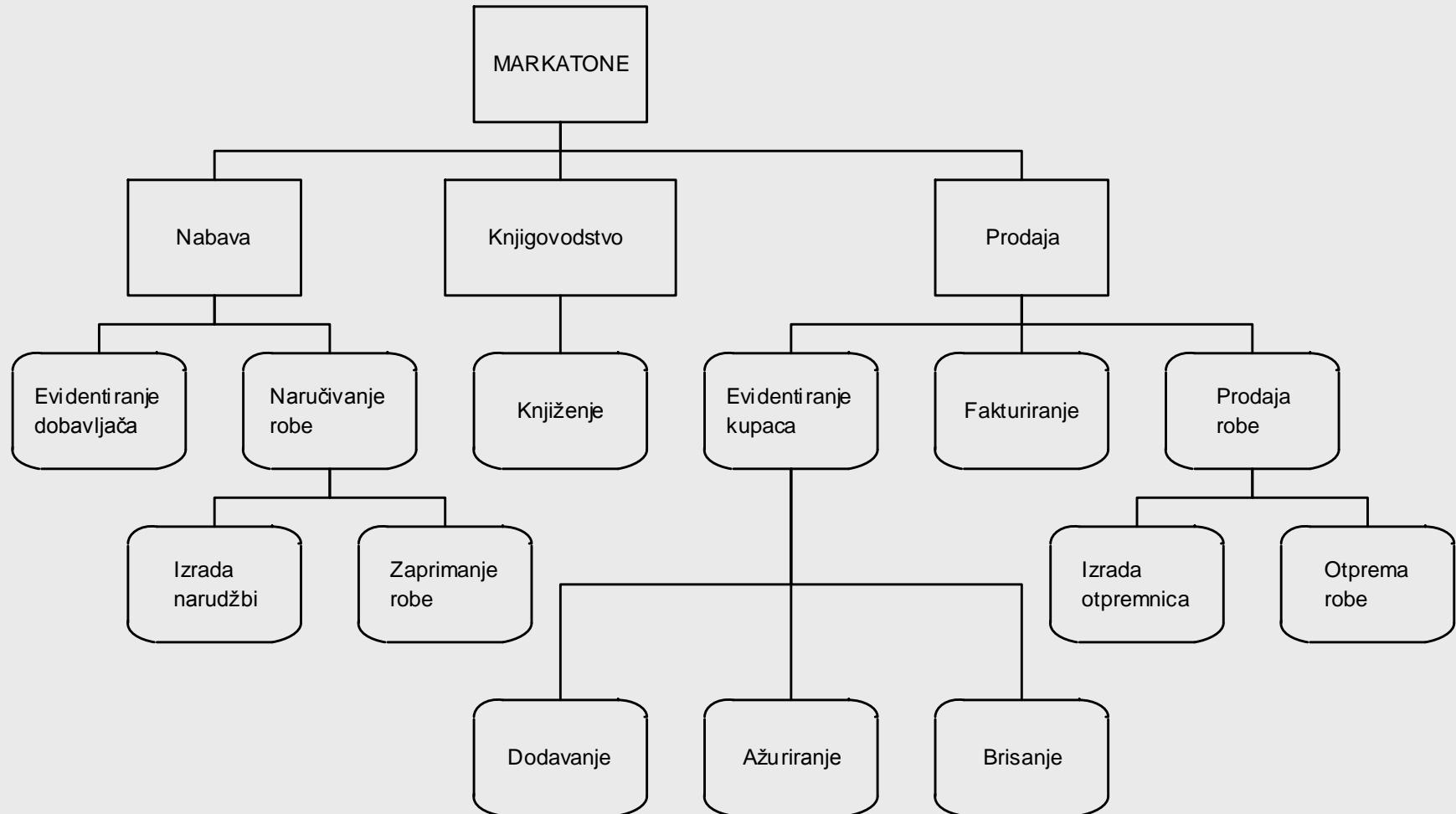
□ Elementi

- 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 - s dijelovima na drugim stranicama (off-page connector)



Dijagram dekompozicije funkcija - primjer

- Primjer, dijagram funkcija za jedan sustav/podsustav:



- Primjeri: [Modeliranje](#), [Primjeri Projekata](#)

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 mora biti dio hijerarhije
- roditelj mora imati barem dvoje djece

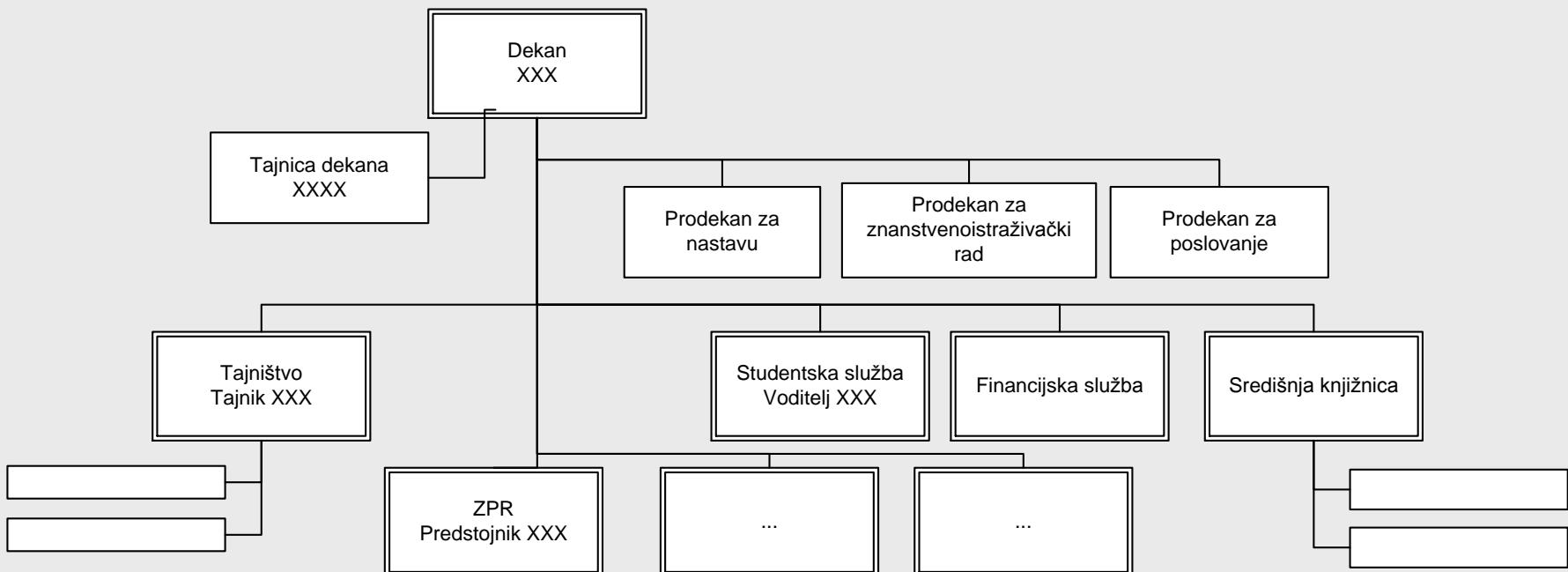
□ Preporuke

- izostaviti procese koji premještaju ili preusmjeravaju podatke, ali ih ne mijenjaju
- pažnju usmjeriti na procese koji
 - nešto računaju (npr. prosjek ocjena)
 - potpomažu odluke (npr. određivanje raspoloživosti robe pri naručivanju)
 - filtriraju ili agregiraju podatke (npr. računi kojima je istekao rok plaćanja)
 - organiziraju podatke u korisne informacije (npr. generiranje izvješća)
 - pokreću druge procese (npr. generiranje subvencije)
 - rukuju podacima (npr. stvaranje, čitanje, ažuriranje, brisanje) - Ne pretjerivati!

Dijagram organizacije

□ Shema, mapa, karta organizacije (Organization chart)

- prikaz strukture organizacije hijerarhijom pravokutnika ("kućica")
- pravokutnik reprezentira određenu ulogu ili odgovornost
 - problem: poistovjećivanje funkcije i organizacije: dekan = ured dekana
 - rješava se pročišćavanjem modela i odvajanjem uloga korisnika



Razrada poslovnih procesa

□ Tehnike modeliranja

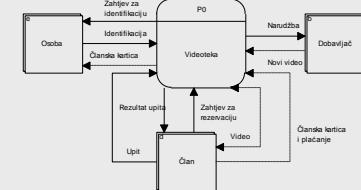
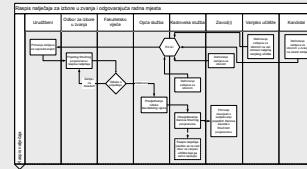
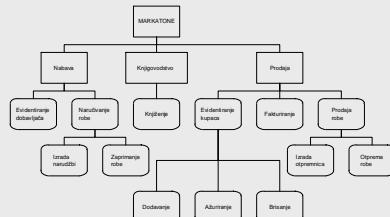
- u širinu - svaki dijagram se detaljizira prije dekomponiranja (breadth-first)
- u dubinu - identificira se hijerarhija, a zatim se detaljizira (depth-first)

□ Razina dekompozicije - Kada stati ?

- do dubine dovoljne za razumijevanje modela (!?)
- napredak do stanja u kojem ulazi i izlazi prevladaju na dijagramu
(predzadnji red u primjeru *Markatone*)

□ nastavak se može provesti

- modeliranjem toka rada (dijagram aktivnosti, Visio: Cross-Functional FlowChart, IDEF0) ili
- modeliranjem toka podataka (Data Flow Diagram - DFD)



Modeliranje toka rada i toka podataka

Modeliranje toka rada

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

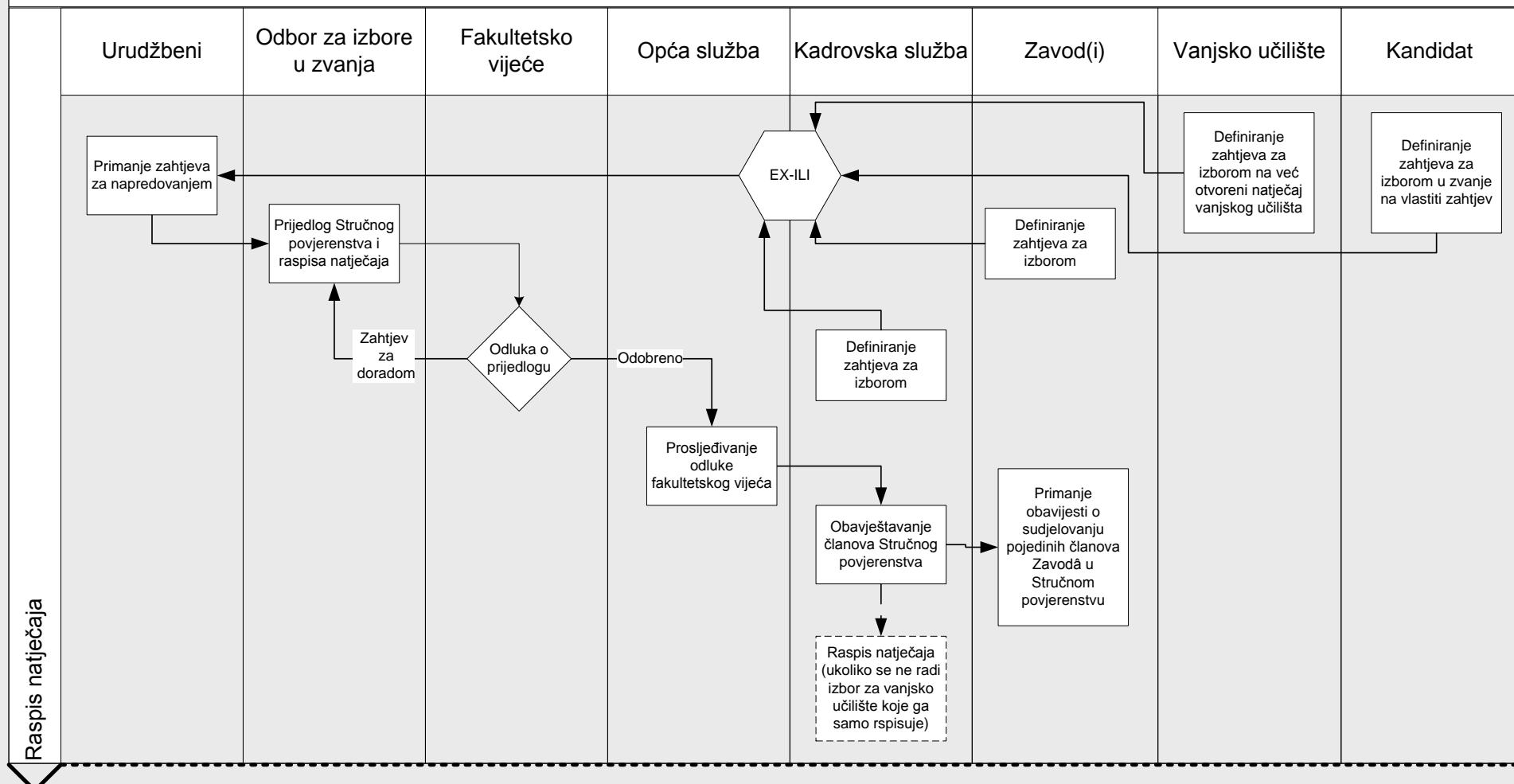
- slijed koraka obrade koji obrađuje jednu poslovnu transakciju
- definira logiku obrade i precizira nositelja
- može imati više varijanti (scenarija)

□ 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 / na osobni zahtjev ili na zahtjev organizacije. FER provodi postupak i na zahtjev s drugih organizacija koje nisu ovlaštene ...*
- *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*
- *Izbor na zahtjev pojedinca ili njegove organizacije ...*

Primjer: Raspis natječaja

Raspis natječaja za izbore u zvanja i odgovarajuća radna mjesta



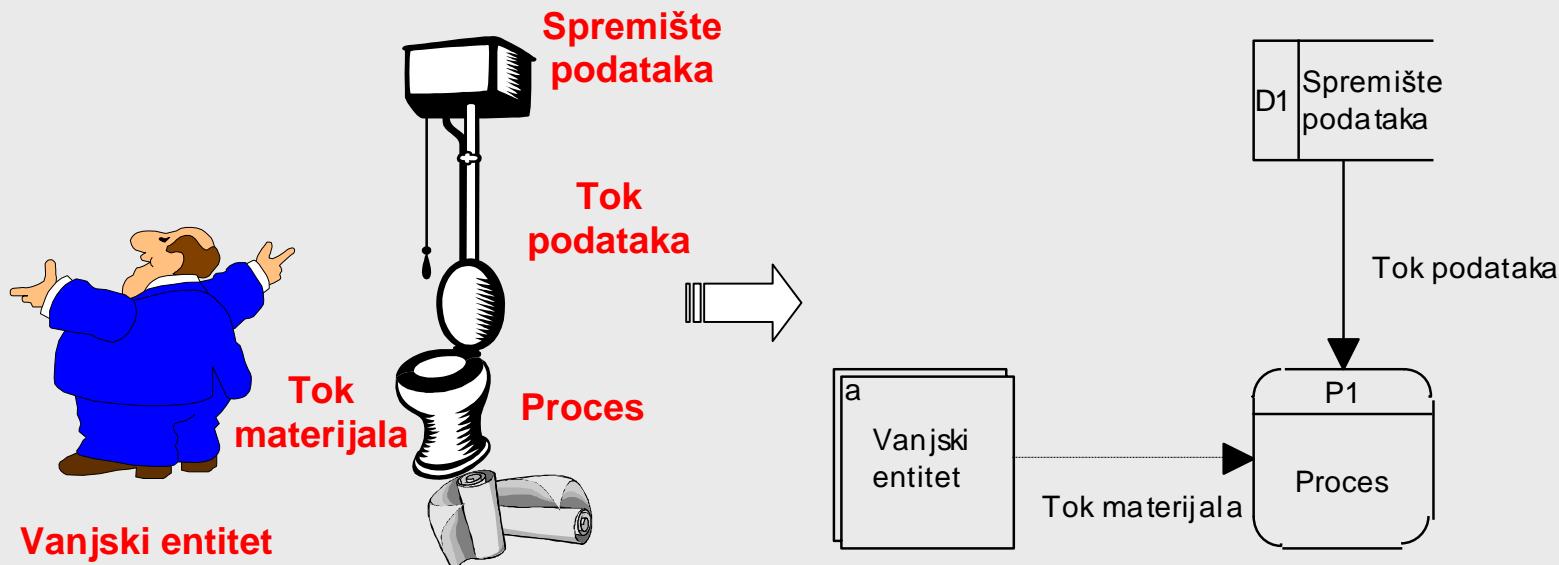
Slika vrijedi 1024 riječi

- može se automatizirati (u drugom ciklusu predavanja)

Modeliranje toka podataka

□ Dijagram toka podataka (DFD - Data Flow Diagram)

- prikaz protoka, strukture i obrade podataka
- dokumentiranje logike, poslovnih pravila i procedura
- sinonimi: transformacijski graf, mjehurasti dijagram (Bubble Chart)



- 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)

- skupovi podataka koji kolaju sustavom
- ulaze u procese (ulazni), mijenjaju se (ulazno/izlazni) ili nastaju (izlazni)
- nazivi oblika *imenica* ili *pridjev+imenica*, npr. JMBAG, Potvrđena prijavnica, Izlazni račun

□ Proces

- aktivnost pretvorbe tokova
- naziv oblika *infinitiv+objekt* (npr. Prijaviti ispit) ili glagolske imenice (npr. Prodaja, Prijava ispita)
- izbjegavati opće nazive (npr. Obavljanje računovodstvenih poslova)
- opis procesa sadrži opis aktivnosti (algoritam) njegovog djelovanja

□ Spremište podataka (data store)

- organizirani i trajni skup podataka
- mjesto pohrane, npr. mapa, registrator, „datoteka“
- mijenja se (CRUD) procesima
- naziv imenicom (ev. u množini), npr. Prijavnica (Prijavnice)

□ Vanjski entitet (external entity / agent)

- objekt vanjskog svijeta povezan s promatranim sustavom
- određuje granice sustava
- predstavlja izvorišta i odredišta podataka, (source, sink)
- mogu biti osobe, org. cjeline, ustanove, drugi sustavi ...
- nazivaju se imenicama, npr. Student, Kupac, Dobavljač

DTP naspram UCD ?

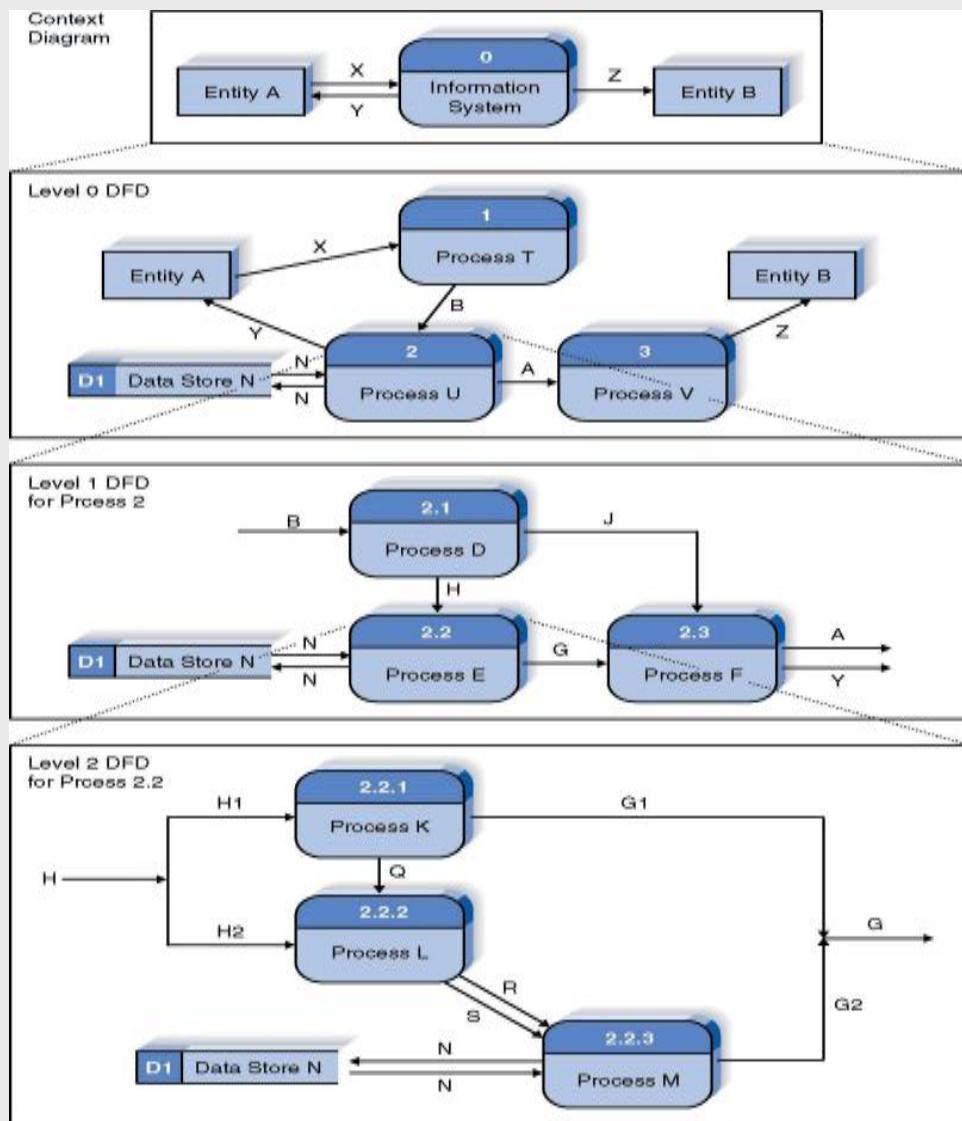
Izrada dijagrama toka podataka

□ Dekompozicija procesa

- polazni, dijagram konteksta (context diagram) hijerarhijski se razlaže na poddijagrame do osnovnih procesa
- niveličacija (leveling) – proces (parent) razrađuje se (explode) dijagramom na nižoj razini (child)
- preporuka: dijagrami s 2 do 9 procesa, a poželjno je slijediti "pravilo 7 ± 2 "
- zaustavljanje kada postane očigledna ugradnja procesa na najnižoj razini

□ Preporuke za označavanje elemenata

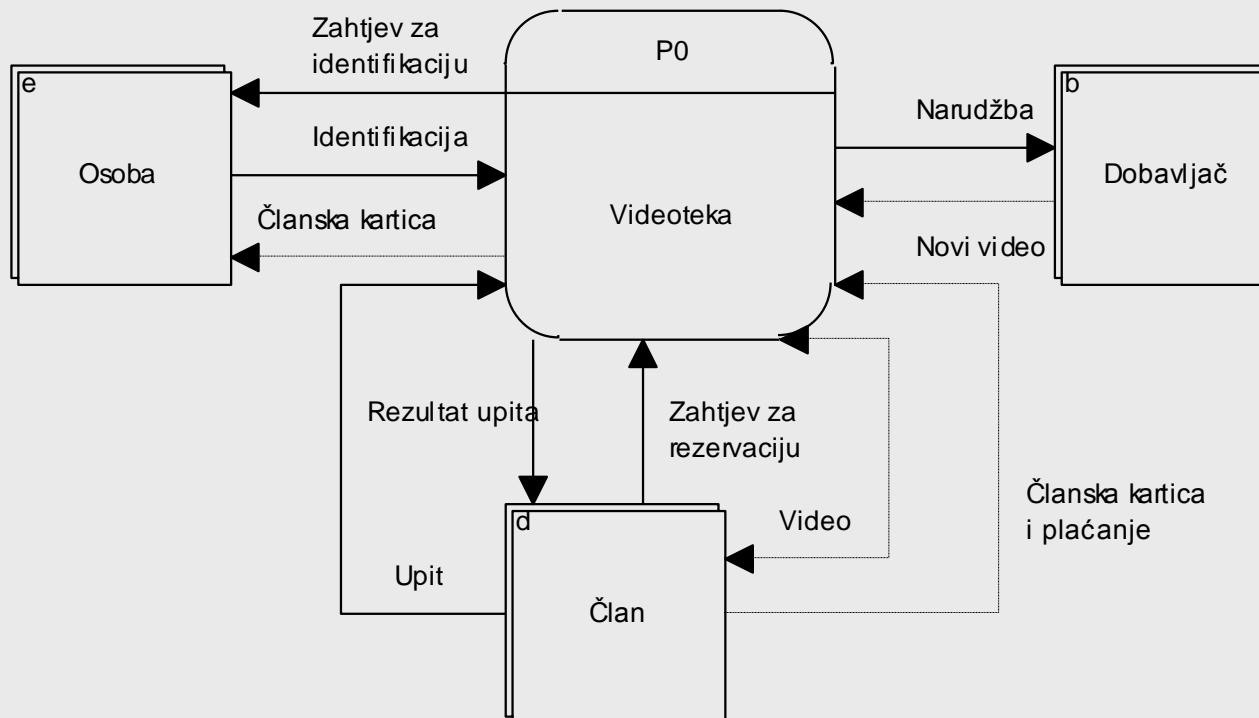
- hijerarhija, razina konteksta = 0
- spremišta, izvori i odredišta – nazivlje velikim slovima, oznake oblika *slово* ili *slово+broj* (pr.D1)
- procesi i tokovi - malim slovima



Izrada dijagrama toka podataka (2)

□ 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 cijelini
 - odrediti vanjske entitete i njihovu povezanost sa sustavom



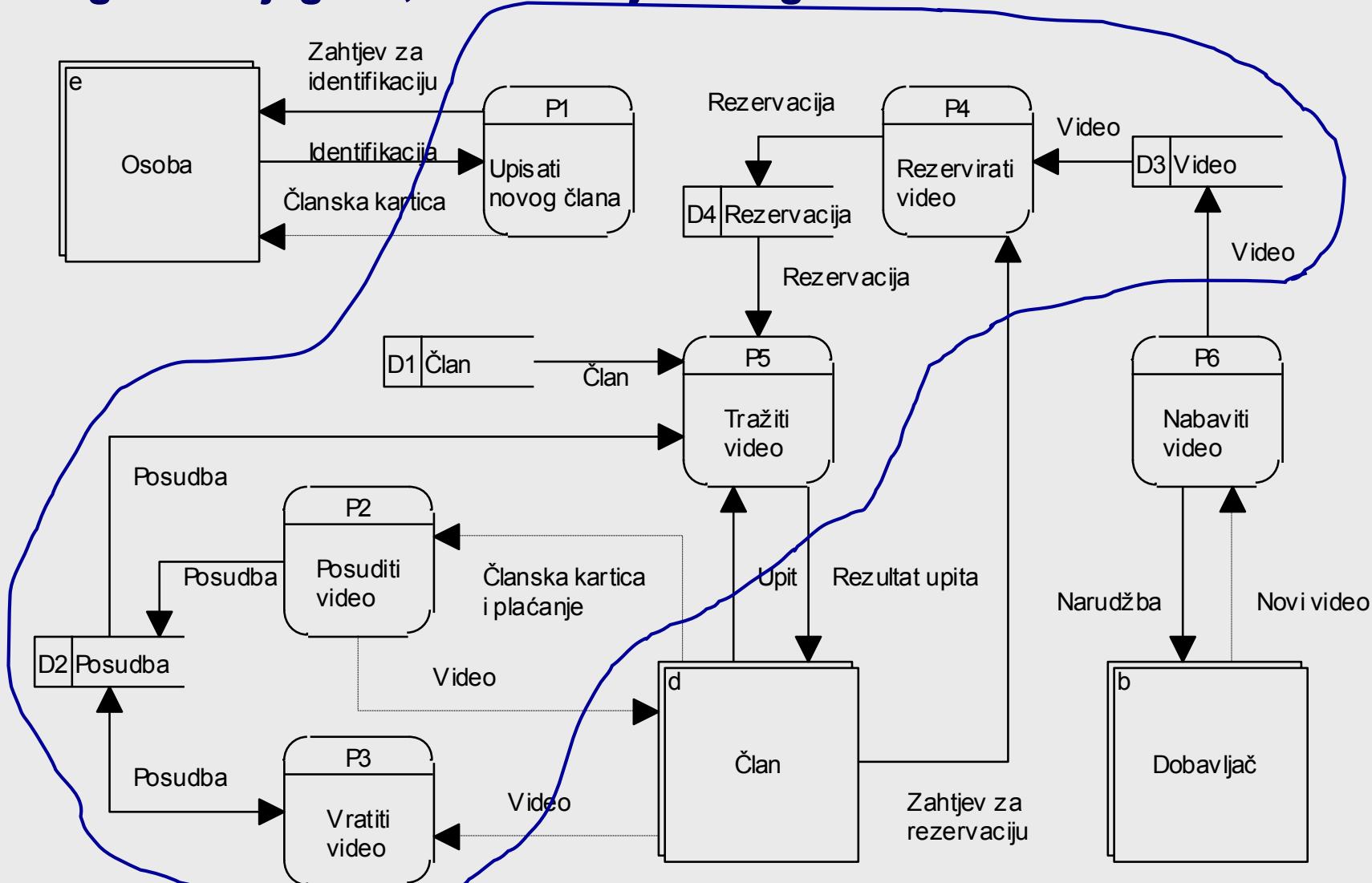
Izrada dijagrama toka podataka (3)

□ 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 (4)

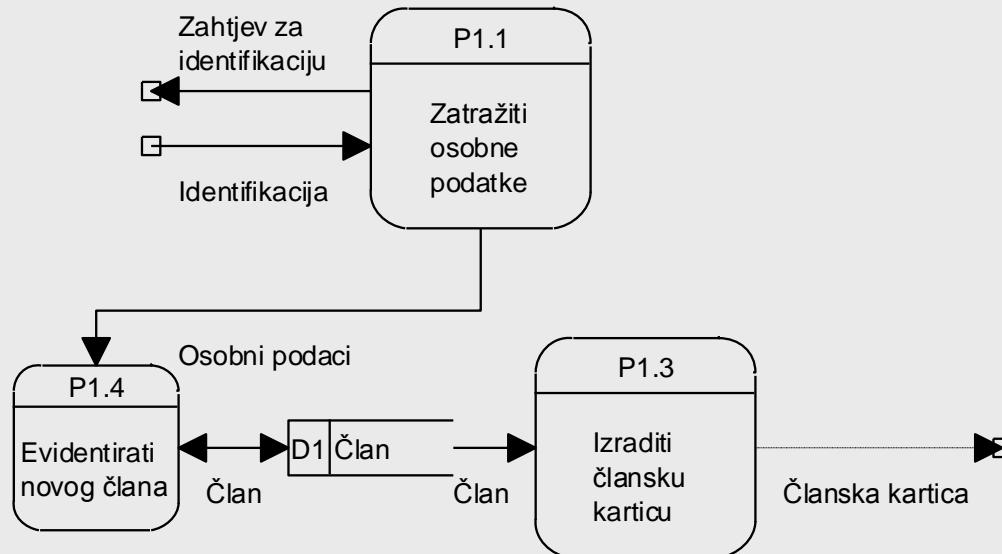
□ Pregledni dijagram, određivanje dosega



Izrada dijagrama toka podataka (5)

□ Razrada

- za svaki proces s preglednog dijagrama identificirati podaktivnosti
- na primjer, za proces Upisati novog člana:



□ 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 u većem broju ponavljanja!

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)

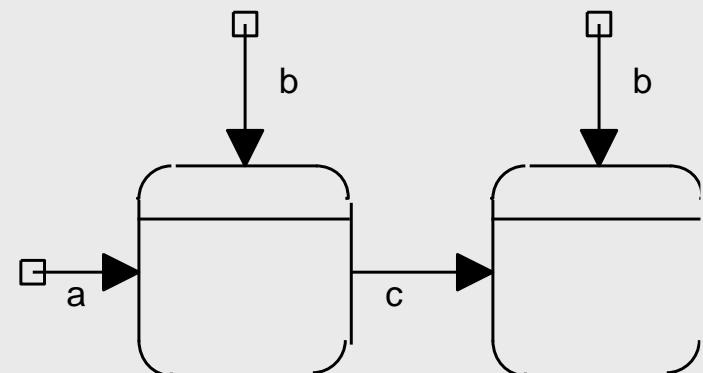
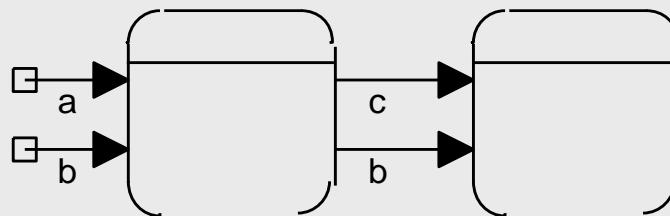
□ Ograničenja i posebni slučajevi

- Svi objekti moraju biti povezani. Nepovezanost = nepotpunost :
 - procesi bez ulaza i/ili izlaza (tzv. čuda i crne rupe)
 - izlazi bez dovoljno ulaza (tzv. sive rupe – najčešće)
 - „napuštena“ spremišta ili vanjski entiteti
- 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

Preporuke za izradu DTP

□ Treba pripaziti na:

- trivijalne tokove – izlazi iz procesa koji ne ulaze u spremišta ili odredišta
 - posebno značenje - prikaz posebnih stanja (npr. dojava pogreške)
- neposredno povezane procese
 - ako postoje - neki 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
 - primjer:



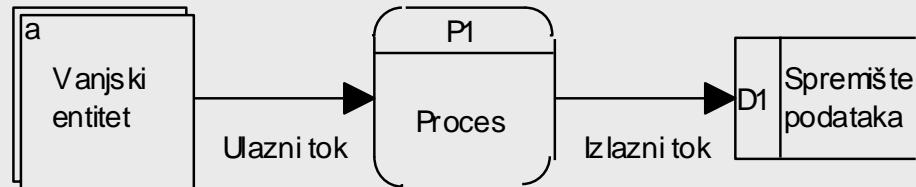
□ Procesi se mogu zbivati istovremeno

- DTP se ne smije tumačiti kao dijagram toka (flowchart)!

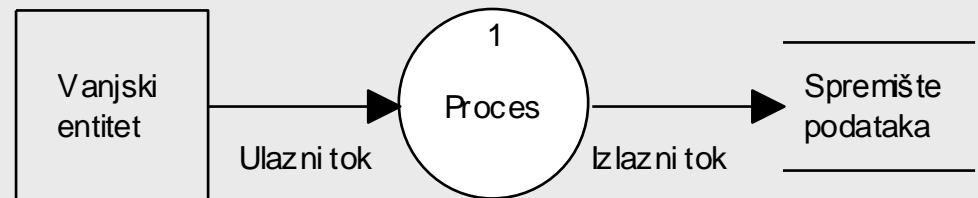
Metode koje koriste DTP

□ Notacije

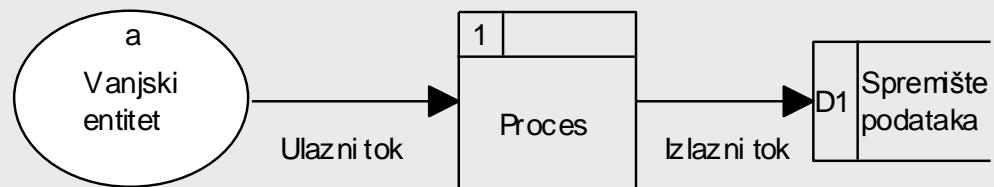
- Gane/Sarson (korištena u primjerima)



- Yourdon/DeMarco



- SSADM

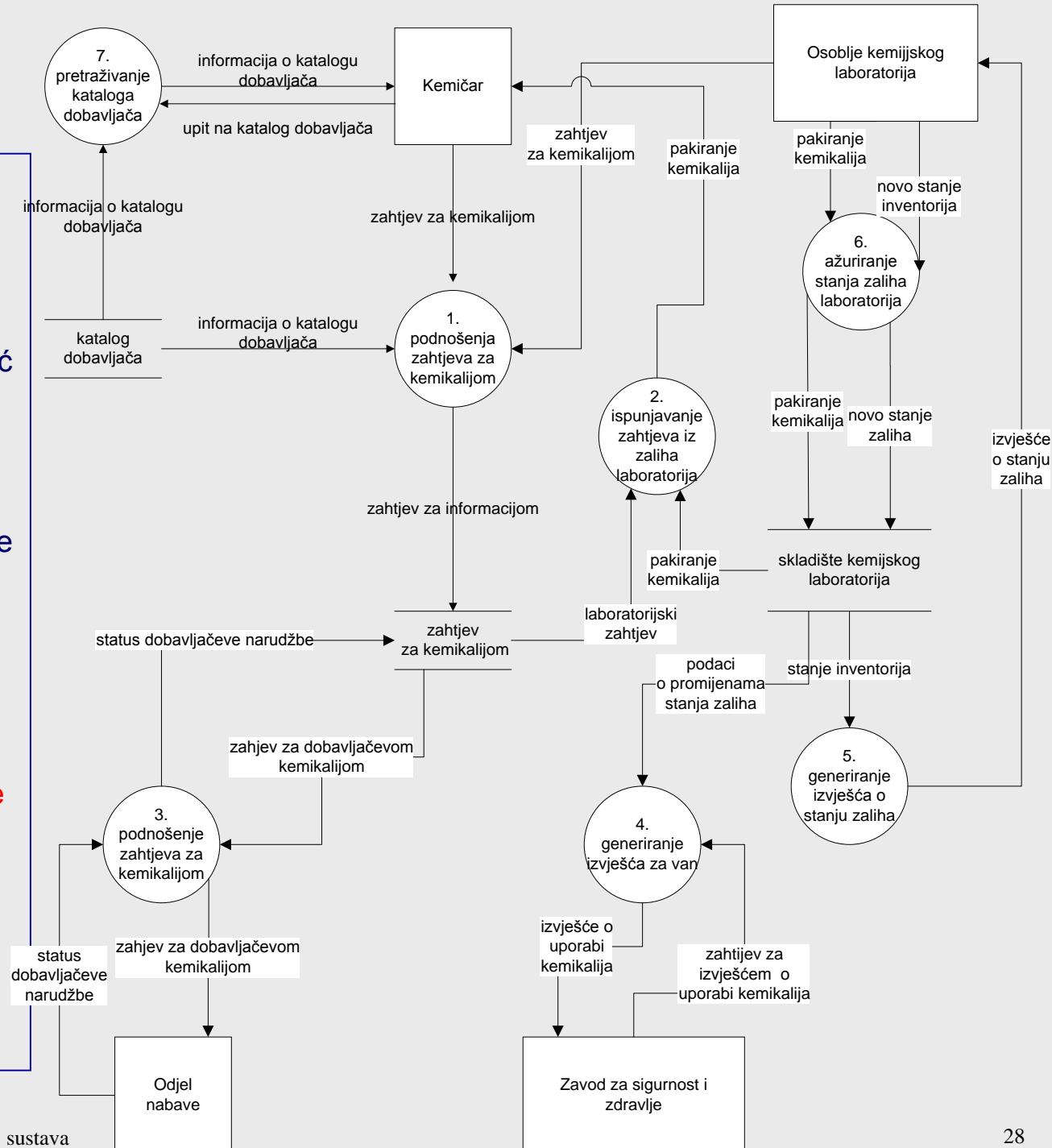


□ Proširenja modela

- 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, dokumenata ili upravljanja

DTP 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 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." ...



Primjer: Procesi sustava ISSP

□ ISSP se može podijeliti na sljedeće funkcionalne cjeline:

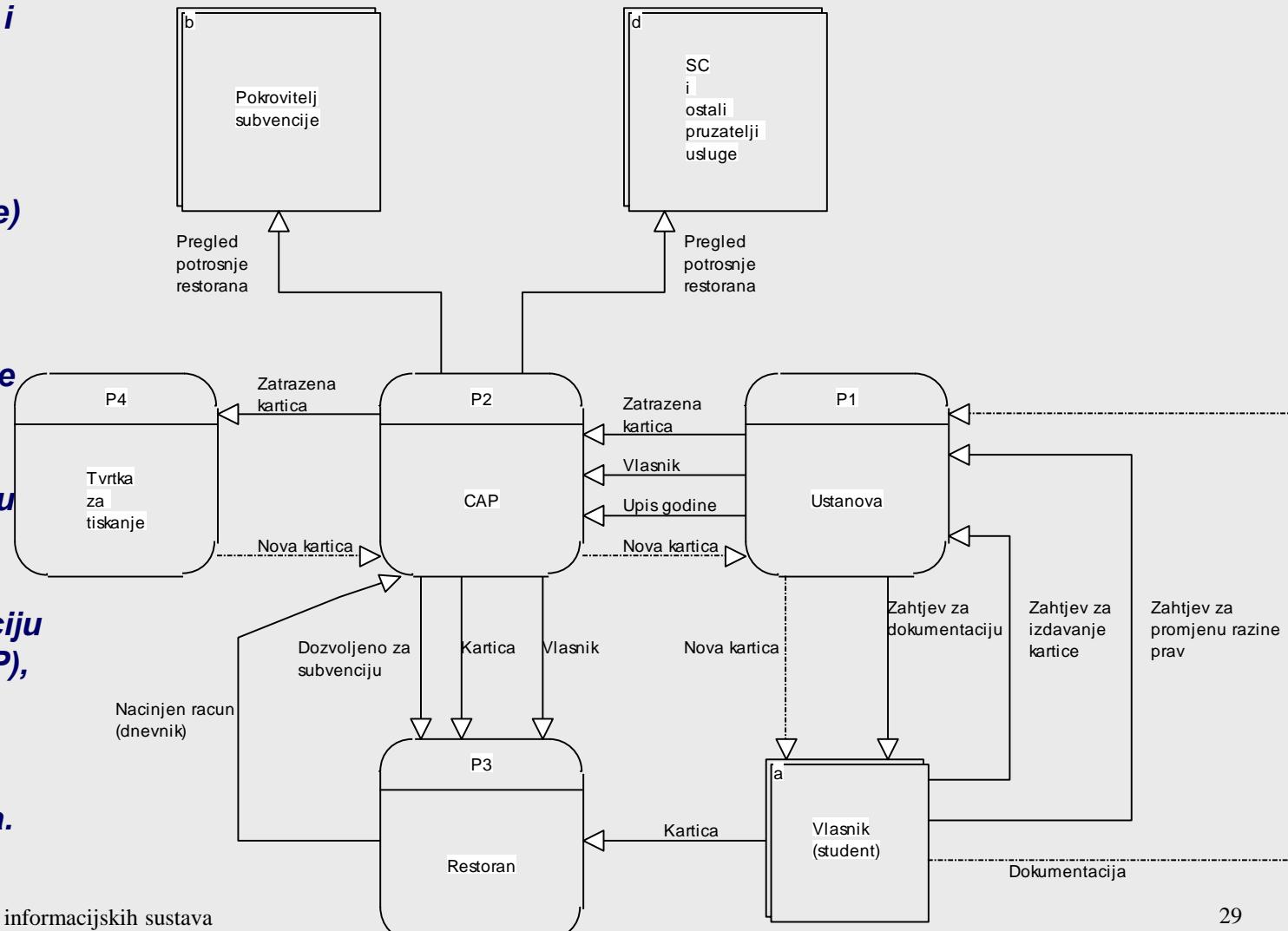
□ Članice sveučilišta i veleučilišta, čiji studenti ostvaruju pravo na subvencioniranu prehranu (ustanove)

□ Restorane koji pružaju uslugu prehrane prehranu članovima ustanove (mesta troška)

□ Tvrta zadužena za održavanje i nabavu opreme te tiskanje iskaznica članova

□ Centar za autorizaciju prava članova (CAP), kao ustanova zadužena za uvođenje i trajno održavanje sustava.

□ Unutar svake od cjelina obavlja se više specifičnih procesa koji ...



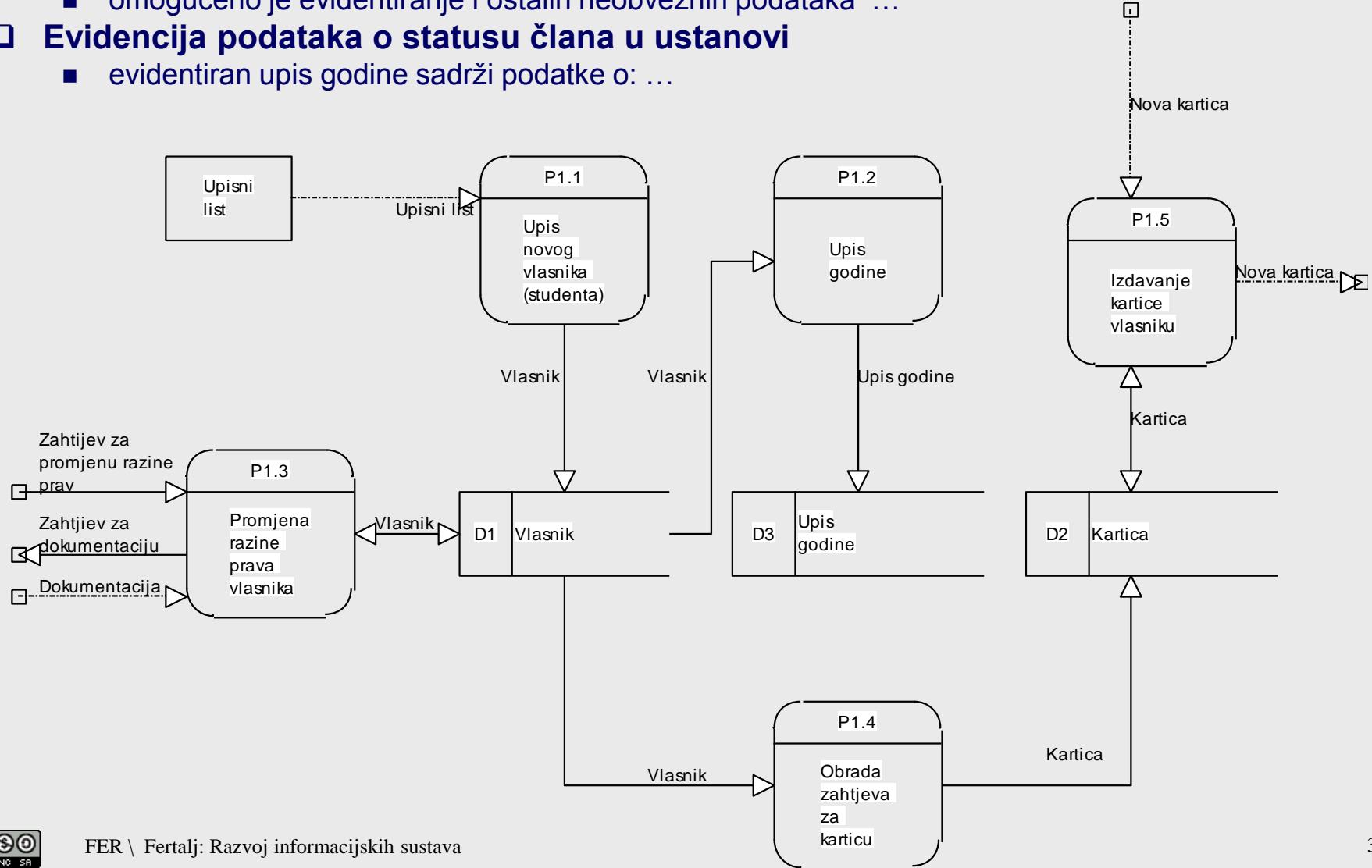
Primjer: razrada procesa ISSP

□ Evidencija matičnih podataka članova

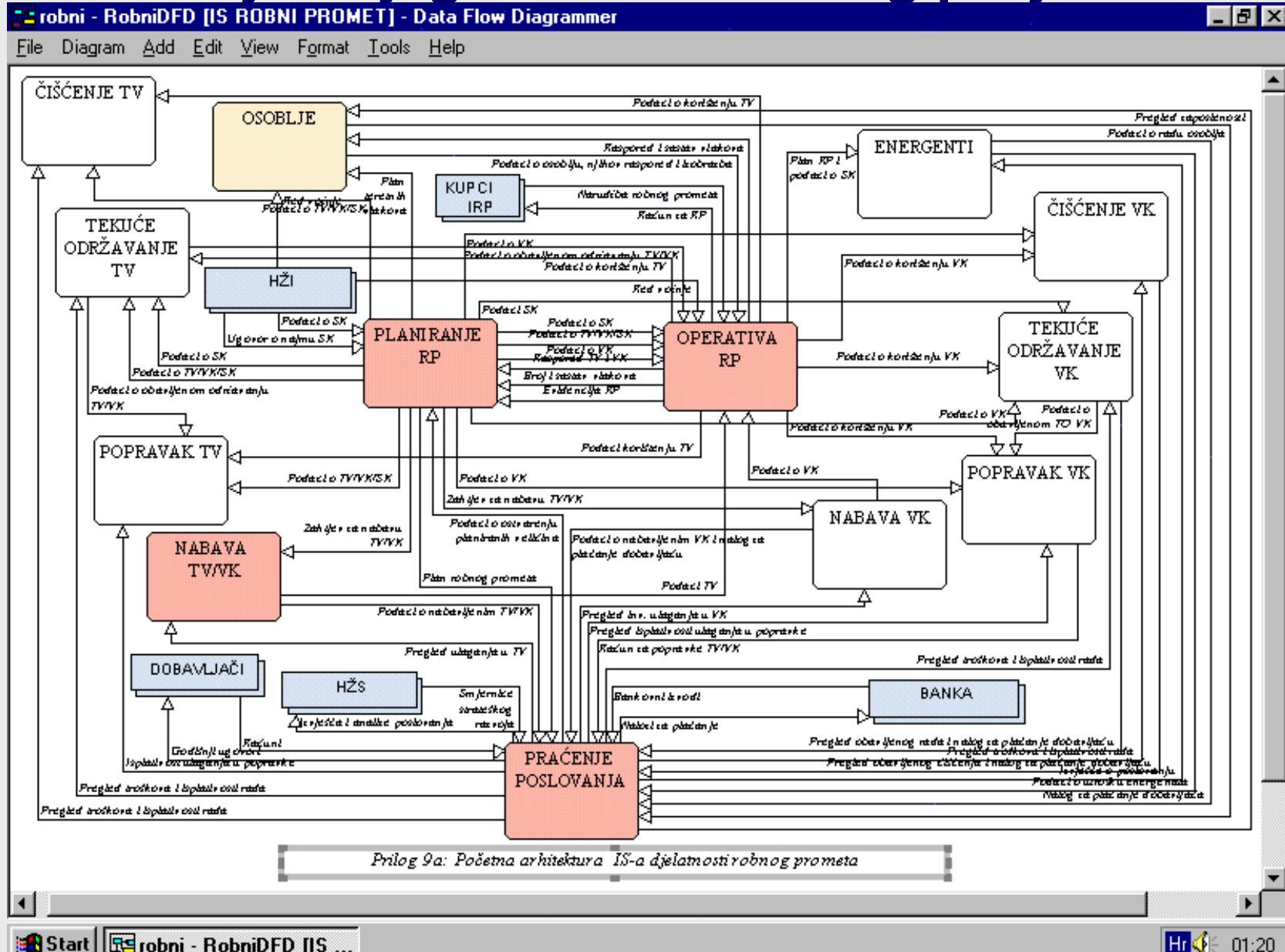
- matični podaci studenta : Ime, Prezime, JMBG, Matični broj (broj indeksa), Razina prava, JMBAG
- omogućeno je evidentiranje i ostalih neobveznih podataka ...

□ Evidencija podataka o statusu člana u ustanovi

- evidentiran upis godine sadrži podatke o: ...



Primjer dijagrama iz stvarnog projekta



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, čime 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

□ vanjski događaji

- potaknuti od strane vanjskih entiteta, koji zahtijevaju informaciju ili ažuriranje podataka → ulazni tokovi podataka
- naziv sadrži naziv vanjskog entiteta
- primjer: zahtjev za upis studenta ili zaprimanje narudžbe kupca

□ vremenski događaji

- vremenski uvjetovani (rok, učestalost) → ulazni upravljački tokovi
- naziv sadrži vremensku oznaku
- primjer: istek roka plaćanja računa, mjesecni obračun plaća, zaključivanje ispitnog roka

□ unutarnji događaji, događaji stanja

- posljedica prijelaza sustava iz jednog stanja u drugo, na način da to zahtjeva obradu → ulazni upravljački tokovi
- primjer: isporuka robe sa skladišta zahtjeva naručivanje nove robe

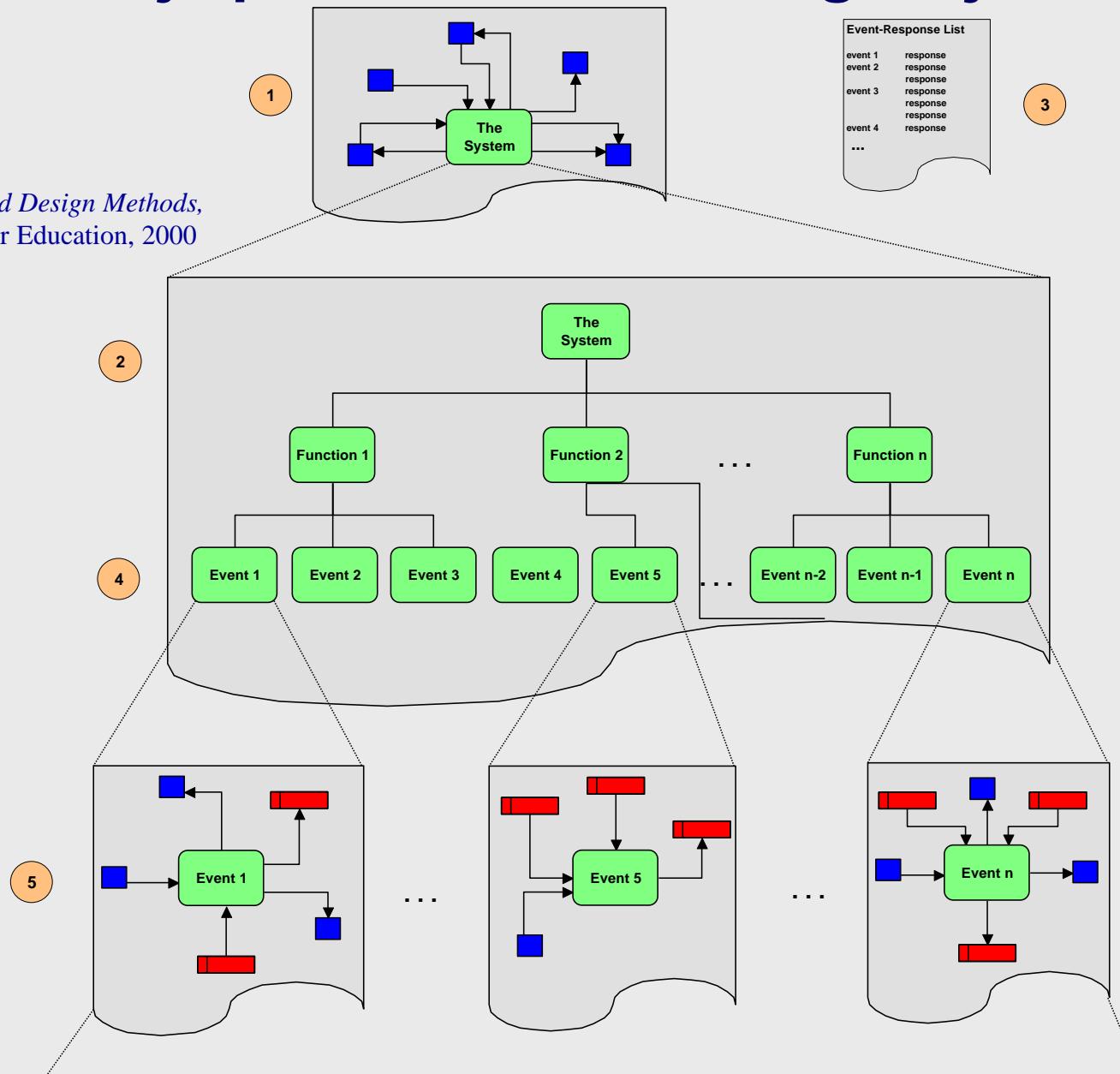
Modeliranje procesa vođeno događajima

□ Raspodjela događaja (event partitioning)

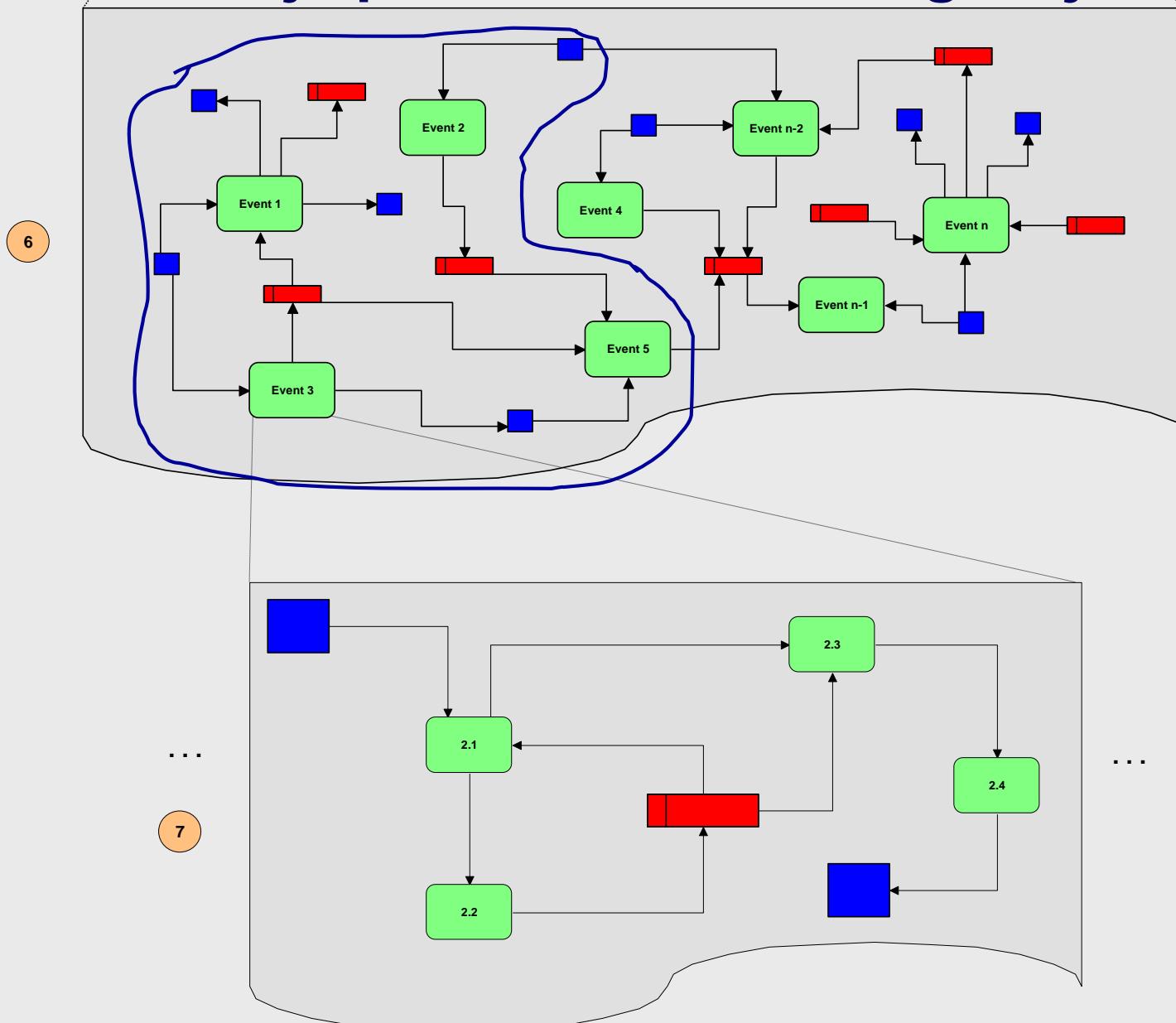
1. Izrada dijagrama konteksta sustava
 - postavljanje početnog dosega projekta
2. Izrada dijagrama funkcionalne dekompozicije
 - podjela sustava u logičke podsustave i/ili funkcije
3. Izrada popisa događaja i odziva
 - utvrđivanje poslovnih događaja na koje sustav mora odgovoriti
 - element popisa = 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 svaki događaj
6. Izrada dijagrama sustava
 - udruživanjem dijagrama događaja
7. Izrada primitivnih dijagrama
 - razrada DTP koji sadrži osnovne procese, spremišta i tokove za svaki pojedini događaj

Modeliranje procesa vođeno događajima (2)

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



Modeliranje procesa vodđeno događajima (3)



Tablica događaja u sustavu

□ Događaj

- Događaj zbog kojeg sustav obavlja određene aktivnosti

□ Okidač

- Kako sustav zna da se dogodio događaj?
- Za vanjske događaje, to budu podaci koji ulaze u sustav.
- Za vremenski uvjetovane događaje - trenutak koji aktivira djelovanje 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

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	Kreditni ured
				Potvrda narudžbe	Kupac
				Podaci o narudžbi	Otpremništvo
				Transakcija	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

Matrični prikaz modela događaja

□ Matrica Entiteti/Događaji

- Entity/Event Matrix,
Event/Enquiry, Entity Access
Matrix

- elementi 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 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		

Vrste matrica

□ Vrste prikaza u primjeni, 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

□ Matrično definiranje prava

- matrica Korisnik / Proces / X
- matrica Korisnik / Entitet / CRUD
 - Korisnik predstavlja grupu korisnika ili ulogu korisnika
 - 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

Klase Proces	K3	K54	K55	K38	K30	K39	K5	K31	K40	K7	K32	K41	K8	K42	K14	K63	K36	K50	K37	K46	K53	K12	K26	K61	K60	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										
P6	CRUD	R+	R-	CRUD	R+	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R										
P10	CRUD	R+	R-	R-	CRUD	CRUD	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R										
P11	CRUD	R+	R-	R-	CRUD	CRUD	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R										
P12	CRUD	R+	R-	R-	CRUD	CRUD	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R-	R										
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			
P32																				CRUD	CRUD	CRUD	R								
P44																				-R+	R+	CRUD	R								
P37																				-R+	R+	CRUD	R								
P22																				-R+	R+	CRUD	R								
P20																				CRUD	CRUD	R+	R-	CRUD	-	-	-	-			
P1																				R+	R-	CRUD	R								
P4																				R+	R-	CRUD	R								
P3																				R+	R-	CRUD	R								
P36																				R	R										
P48																				R	R										
P45																				R	R										
P18																				R	R										
P16	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			
P7	R		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	CRUD	R	R	R	R	R	R	R	R			
P24	R																			CRUD	R	R	R	R	R	R	R	R	R		
P30	R			R		R													R	R	R										
P25	R																		R	R	R										
P29	R																		R												
P28																			CRUD	R	R										
P33	R		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	CRUD	R	R	R	R	R	R	R	R	R			
P19	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			
P14	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			
P38	R			R		R		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			
P50																			R												
P52	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			
P51	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			
P47	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			
P46																			R	R	R	R	R	R	R	R	R	R	R		
P31																			R	R	R	R	R	R	R	R	R	R	R		
P15																			R												
P49																			R	R	R	R	R	R	R	R	R	R	R		
P26																			R	R	R	R	R	R	R	R	R	R	R		
P13	R																		R	R	R	R	R	R	R	R	R	R	R		
P27																			R	R	R	R	R	R	R	R	R	R	R		
P21	RU		RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	
P17	R																		R												
P2																			R	R	R	R	R	R	R	R	R	R	R		
P5																			R	R	R	R	R	R	R	R	R	R	R		
P35		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		
P34	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		
P41																															
P42																															
P39																															
P40																															
P43																															
P23																															

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 – kumulativ ponašanja nekog objekta (pravokutnik, krug ili elipsa)
- prijelaz – promjena uzrokovana događajem (usmjereni linija)
- događaj – uvjet promjene stanja i pokrenuta akcija (opis prijelaza oblika događaj/akcija)

□ DPS najčešće opisuje vremenski zavisno ponašanje čitavog sustava

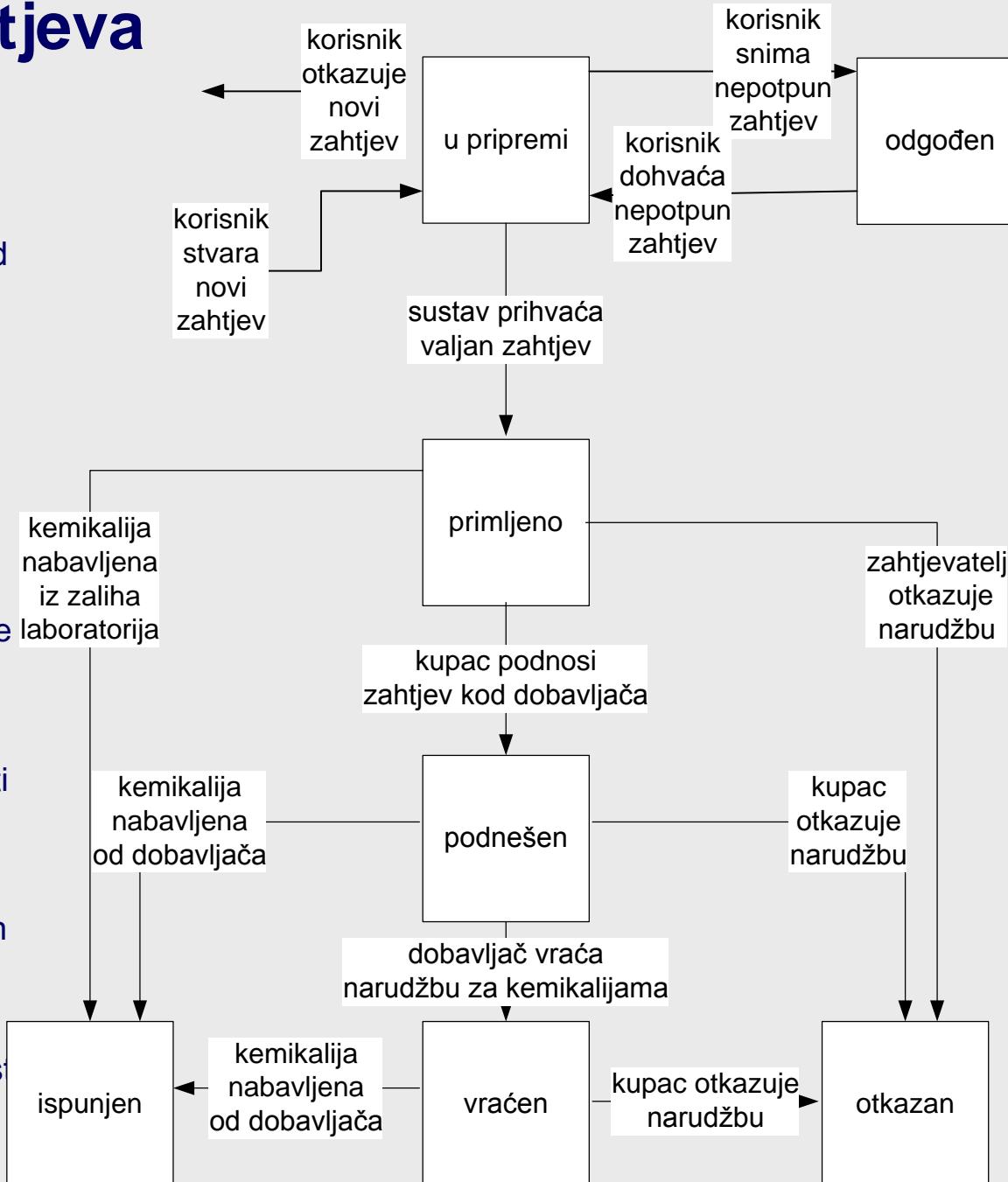
- manji sustavi mogu se prikazati jednim dijagramom
- veći sustavi razlažu se slično DTP

□ Primjena

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

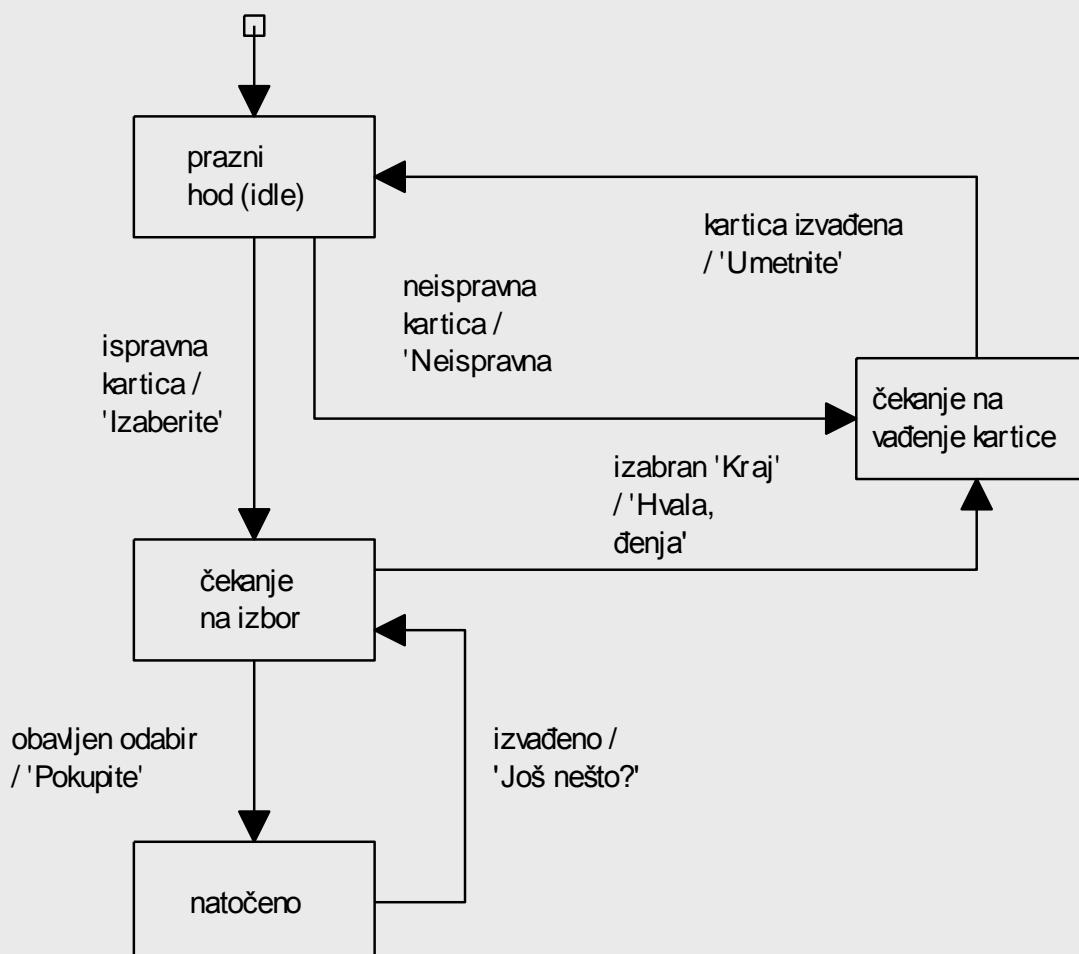
Primjer: Stanja zahtjeva

- 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
- Zaprimljen: 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
- Ispunjeno: zahtjev je zadovoljen bilo da je dostavljen iz zaliha kemijskog laboratorija ili da je dostavljen od strane vanjskog dobavljača
- Vraćeno: 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



Primjer: sustav za rad u stvarnom vremenu

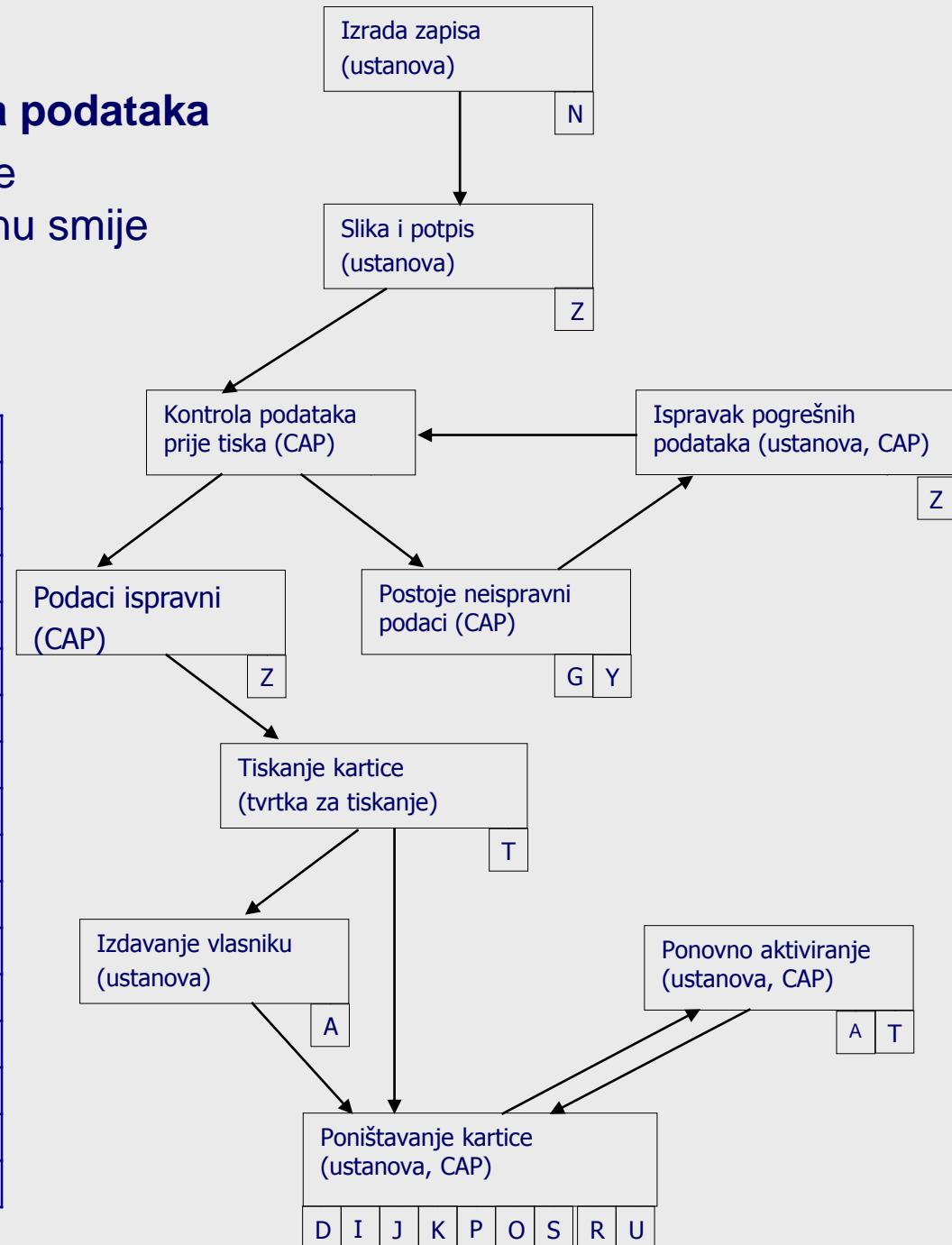
- Dijagrami sustava za rad u stvarnom vremenu razlikuju se po tome što sadrže posebno stanje "besposlen"
- Npr. Cugomat hotela *Proljeće* ili Bankomat banke *Oprosti nam duge naše*



❑ Primjer: prijelazi statusa podataka

- X-ice : 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 apsolventski 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



Mape dijaloga

□ Korisničko sučelje kao konačni automat

- Jeden element sučelja (forma, izbornik, dijalog) aktivan u trenutku
- pristup ograničenom broju drugih elemenata ovisno o akcijama korisnika
- broj putanji može biti velik, ali je konačan a mogućnosti poznate

□ Mape dijaloga kao analitička tehnika

- prikaz dijaloga i navigacije, ali se ne bavi dizajnom zaslona
- korisnici i razvojnici - usaglašavanje interakcije
- vizualizacija strukture web sjedišta (eng. site maps)
 - navigacijski linkovi kao prijelazi na mapi dijaloga

Primjena mapa dijaloga

□ Koristi se notacija dijagrama prijelaza stanja

- Uvjet pokretanja navigacije kao tekst na strelici
 - **Korisnička akcija**, npr. pritisak tipke, klik na link ili gumb dijaloga
 - **Podatkovna vrijednost**, npr. pogrešan unos koji izaziva poruku o pogrešci
 - **Sistemski uvjet**, npr. signal da je pisač ostao bez papira
 - **Kombinacija uvjeta**, npr. tipkanje opcije iz izbornika i pritisak na tipku *Enter*

□ Izostavljanje općih funkcija radi jednostavnosti

- pritiskanje tipke *F1* da bi se dobila pomoć
- standardni navigacijski linkovi (npr. *Natrag*, *Doma*)

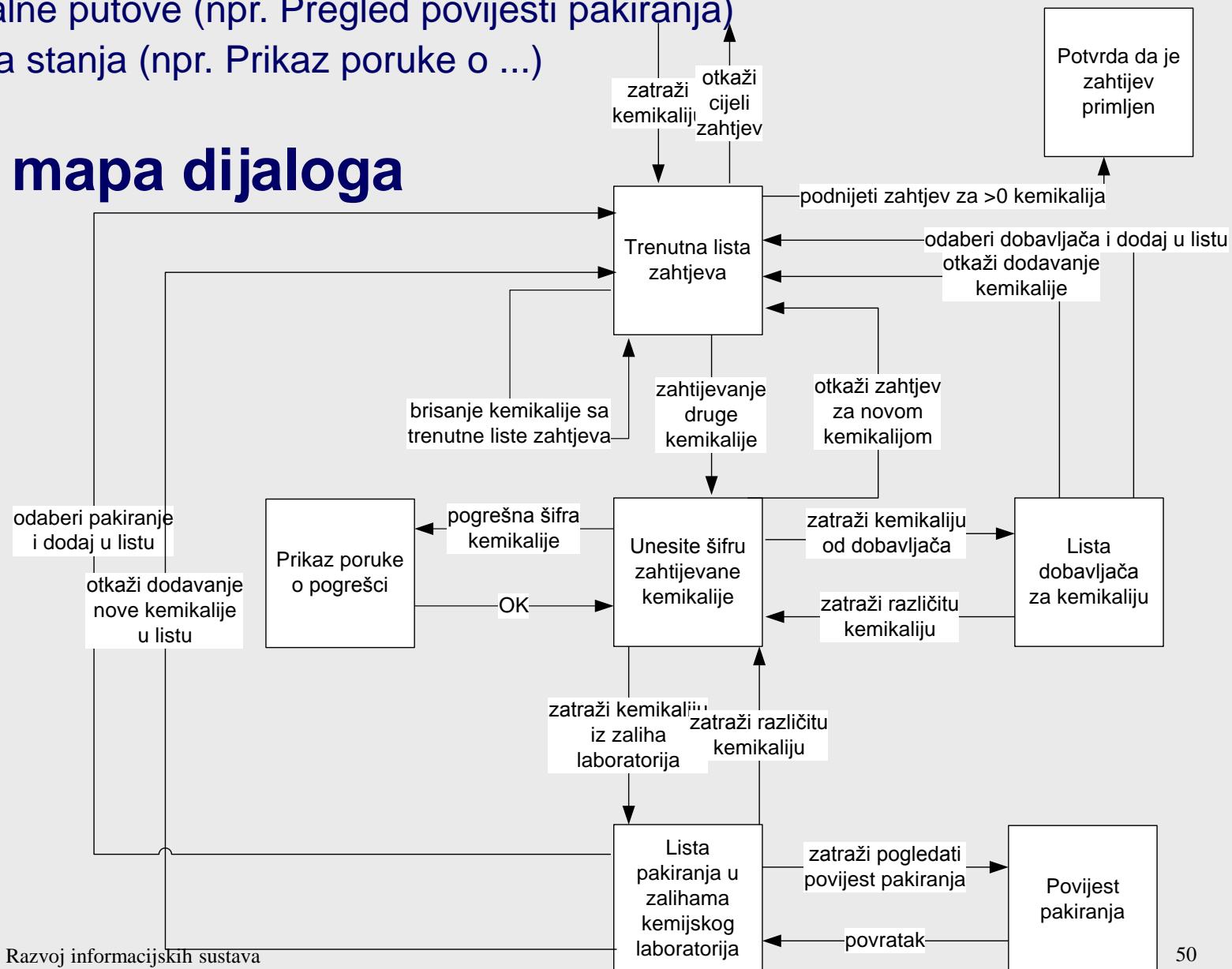
□ Pri analizi mapa predstavlja interakciju na konceptualnoj razini

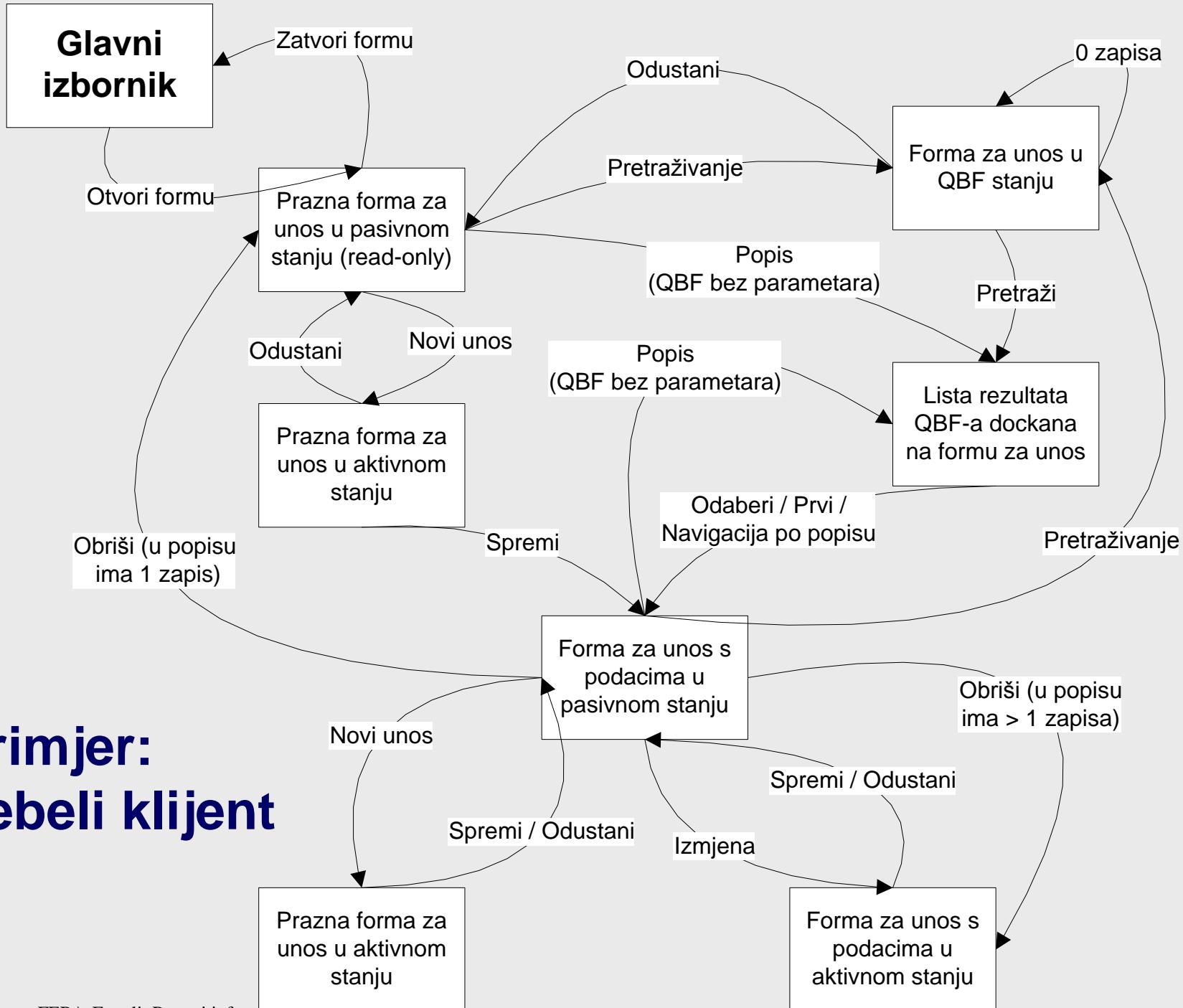
- Konkretna ugradnja može biti drugačija.
- Primjer:
 - Korisnik inicira slučaj korištenja odabirom opcije "zatraži kemikaliju" iz glavnog izbornika a što se ne vidi na dijagramu.
 - Nije nužno prikazivati detalje (npr. Potvrdu otkazivanja čitavog zahtjeva)
 - Ključno je da korisnik i razvojnik jednakо "razumiju" opisanu funkcionalnost.

Mape dijaloga mogu prikazati

- Alternativne puteve (npr. naručiti od dobavljača) koji odstupaju od uobičajenog puta (kemikalija iz zaliha).
- Opcionalne puteve (npr. Pregled povijesti pakiranja)
- Posebna stanja (npr. Prikaz poruke o ...)

Primjer: mapa dijaloga





Primjer: debeli klijent

Resursi i primjeri



Modeliranje\RIS-Preporuke

- preporuke za upotrebu različitih notacija izrade dijagrama

Modeliranje\StrukturiranaAnalizaDizajn

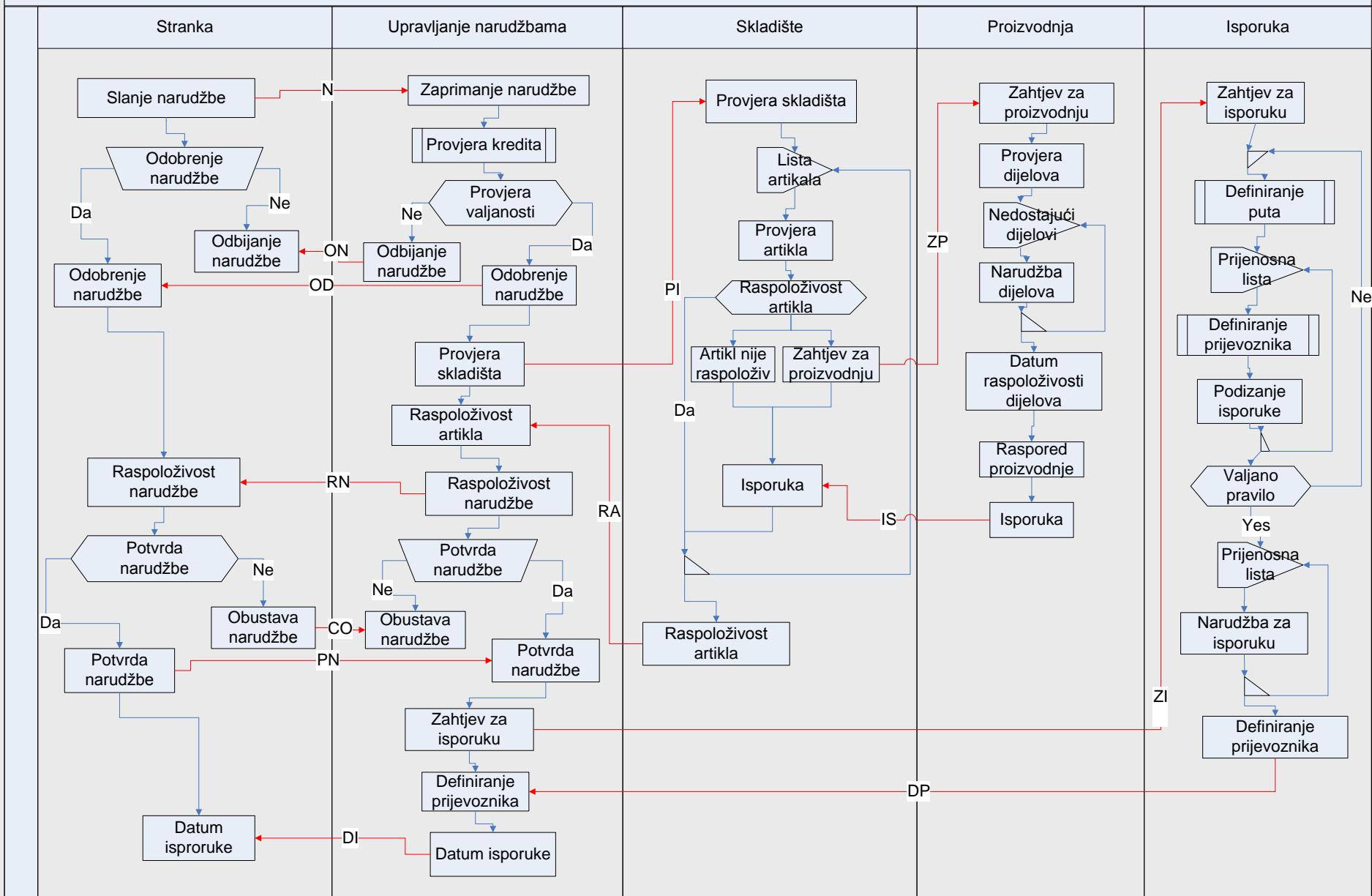
- primjer strukturiranog oblikovanja
- primjer analize izvedivosti
- primjer strukturiranog dizajna programa

Modeliranje\KakoRazmišljajuPoslovniLjudi

- Izvor: Howard Smith - The Third Wave of Business Process Management
- prijevod na sljedećem slajdu

Kako razmišljaju poslovni ljudi

Kako razmišljaju poslovni ljudi

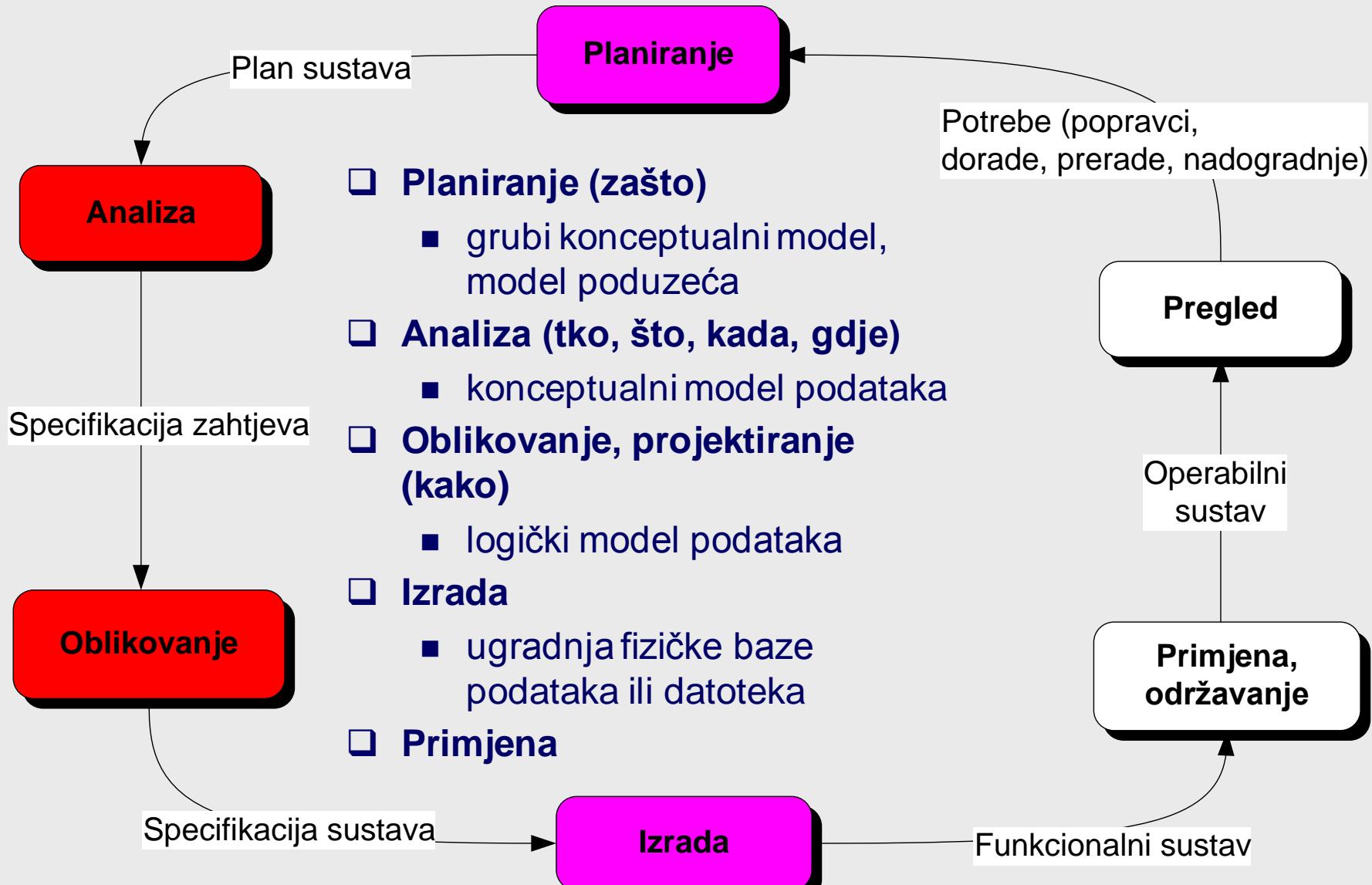




(konceptualno) Modeliranje podataka

2013/14.05

Kontekst modeliranja podataka



Vrste modela podataka

□ Konceptualni

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

□ Logički

- relacijski (najčešće), postrelacijski, objektno-relacijski
- uobičajeno podliježe relacijskoj teoriji
- potpuno definirani attribute i domene (logički tipovi vrijednosti)
- potpuno definirani ključevi
- potpuno normalizirani entiteti (samo veze 1:N)

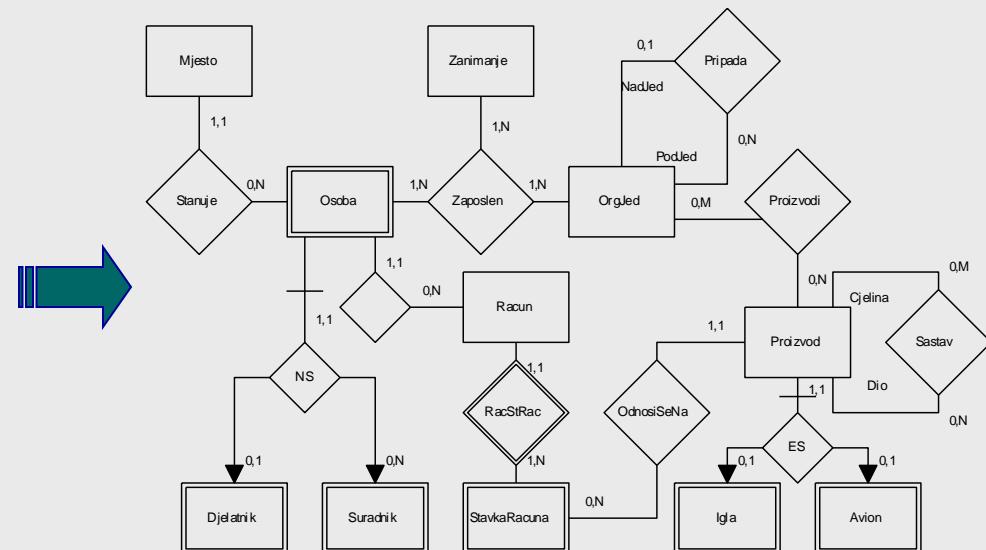
□ Fizički

- logički model preveden u model podataka za odabrani SUBP
- stvarni tipovi, indeksi, pogledi, ... te fizički parametri

Model entiteta-veze

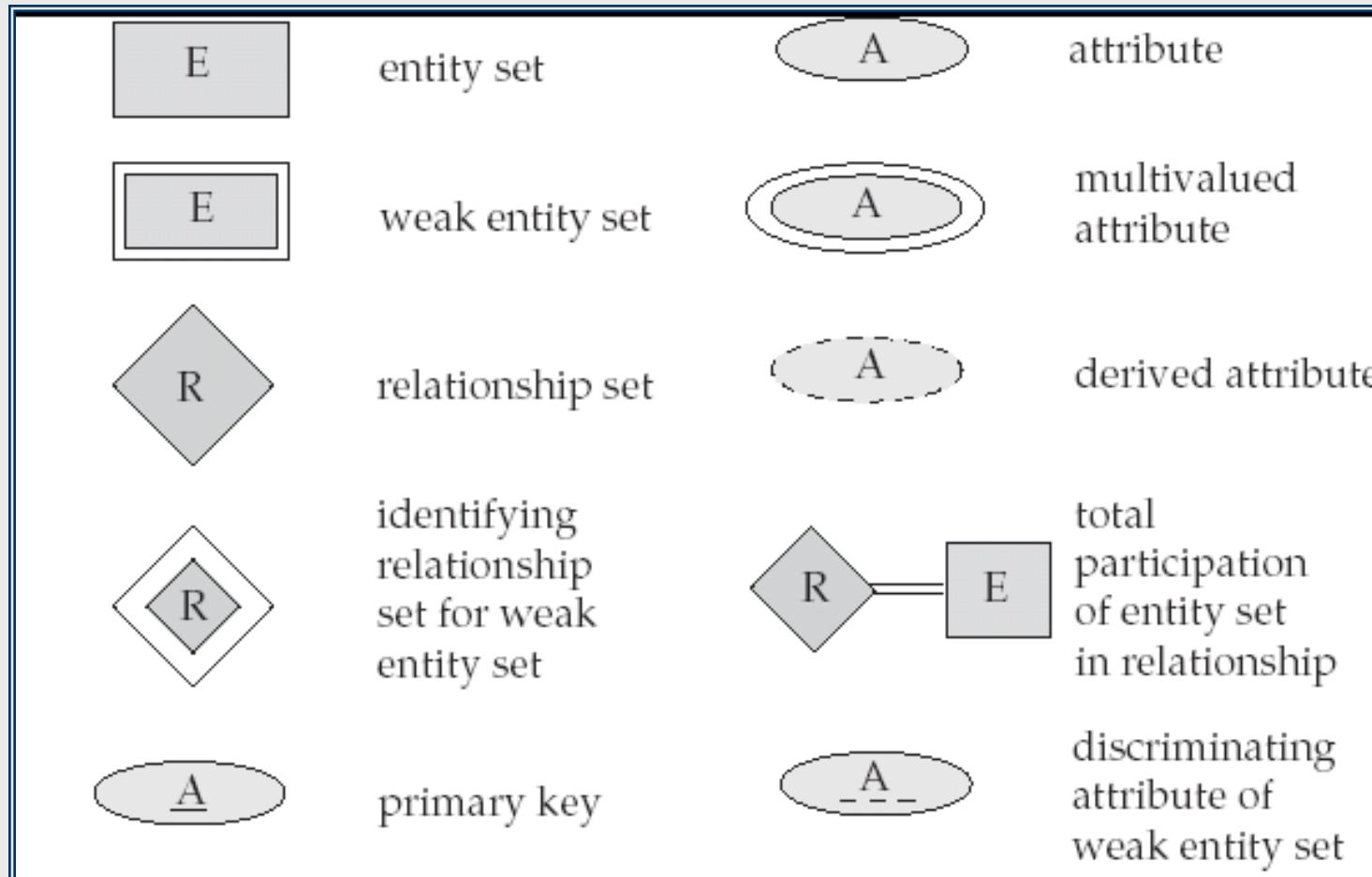
□ Model entiteta-veze

- model podataka koji opisuje entitete (pojavnosti), njihove atributе 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 entiteta-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



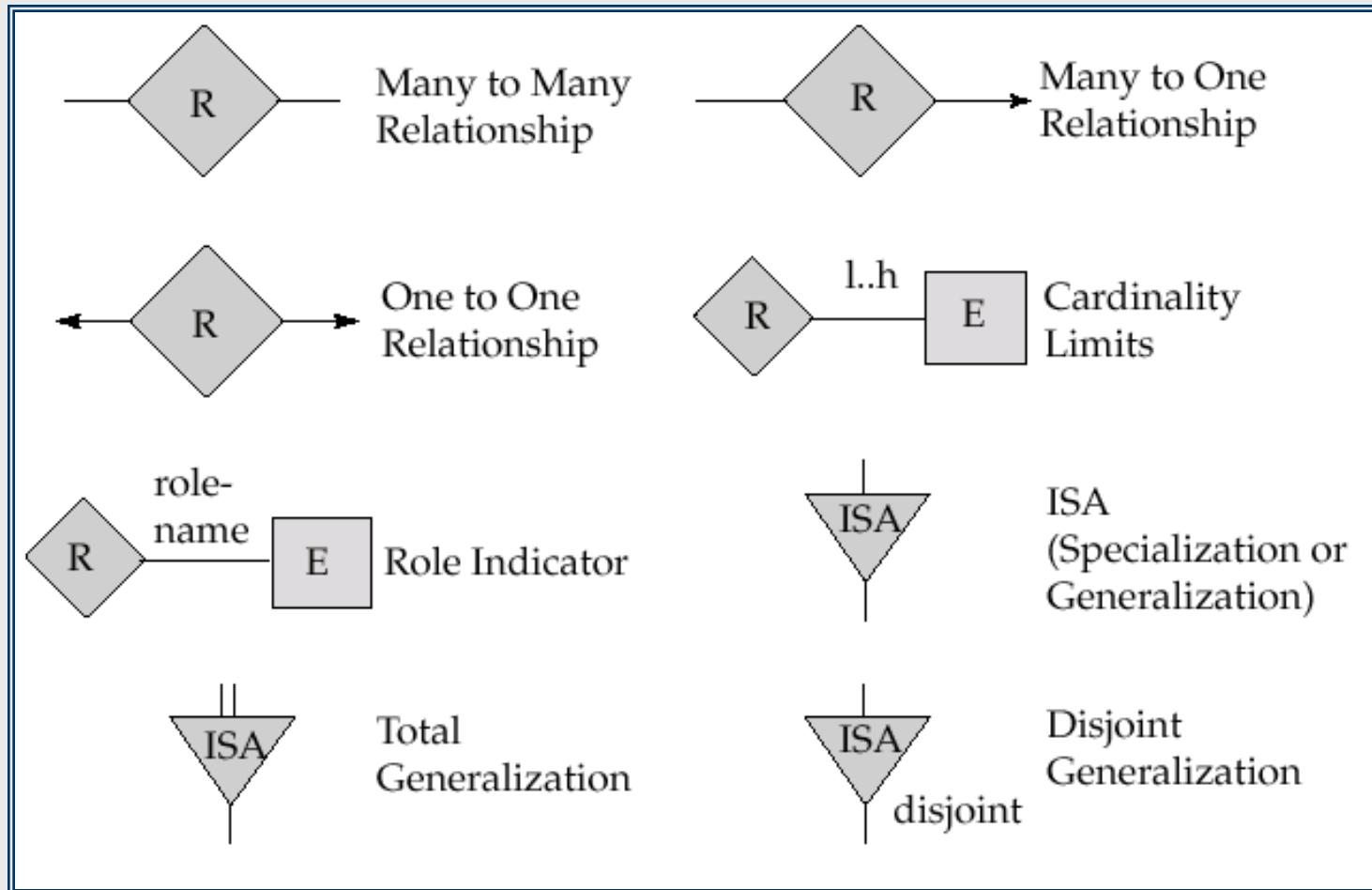
Proširena Chenova notacija (1)

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



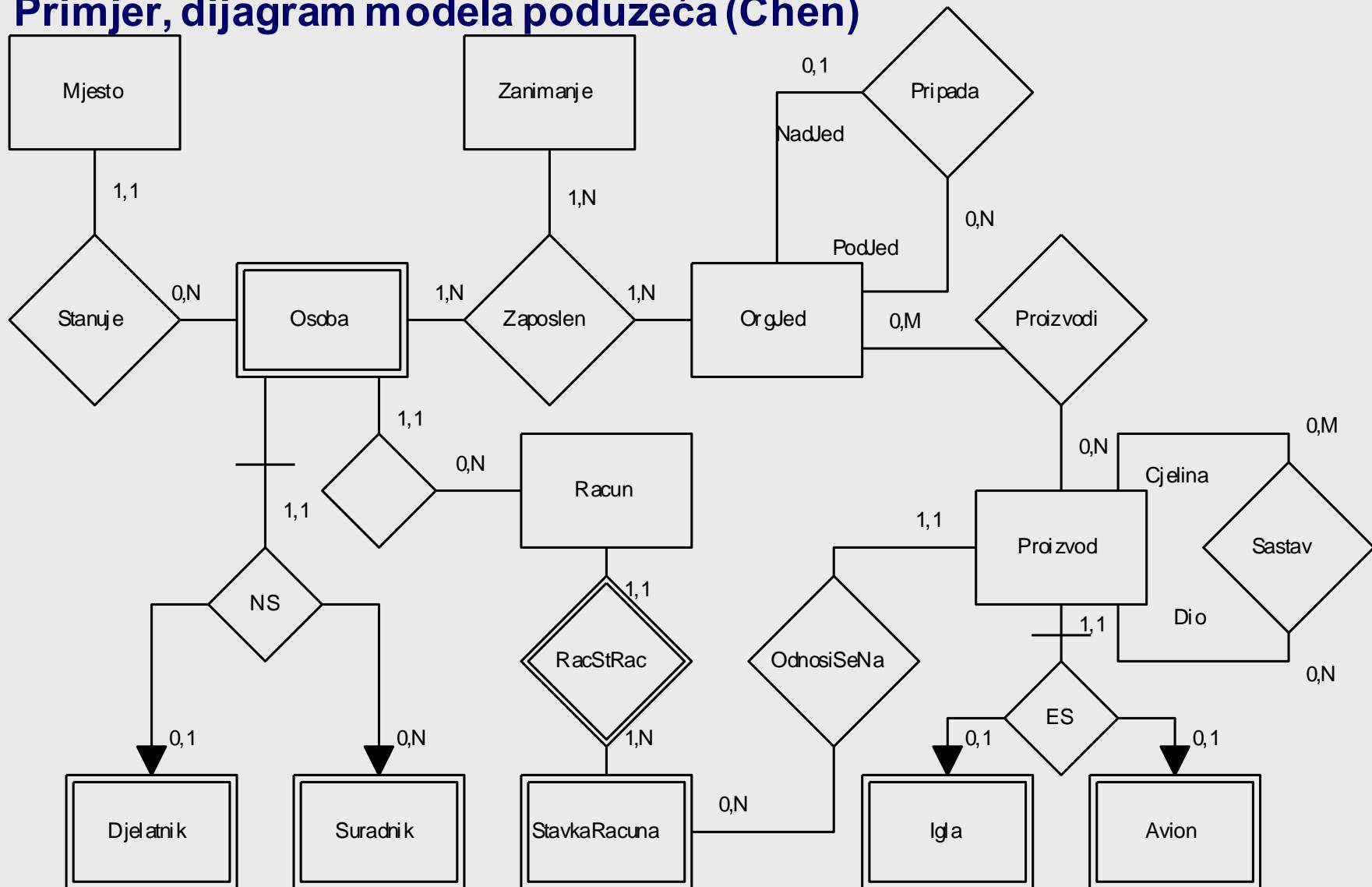
Proširena Chenova notacija (2)

- Potpuna generalizacija – entitet se mora specijalizirati



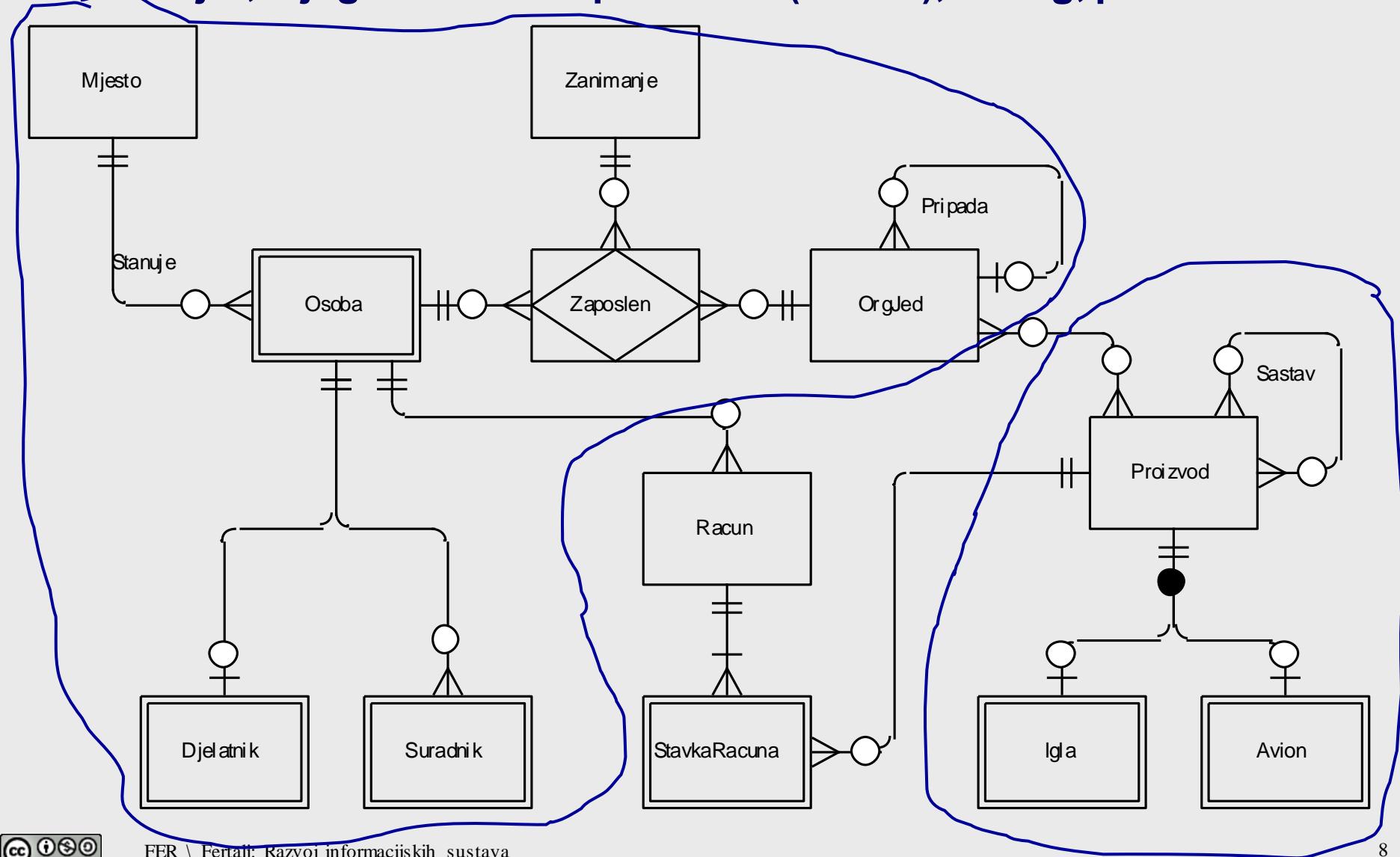
ERD notacija - Chen

□ Primjer, dijagram modela poduzeća (Chen)



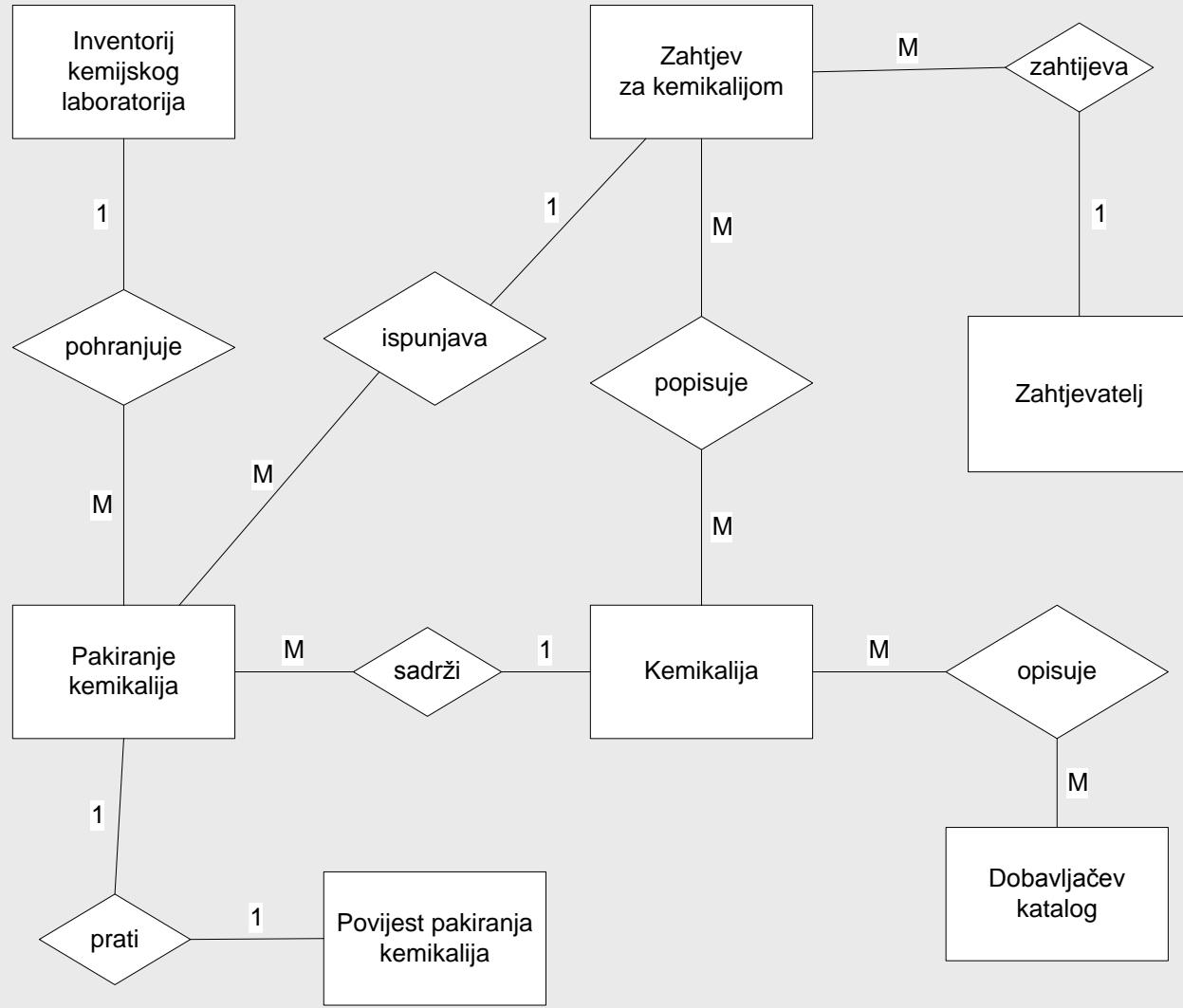
ERD notacija - Martin

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



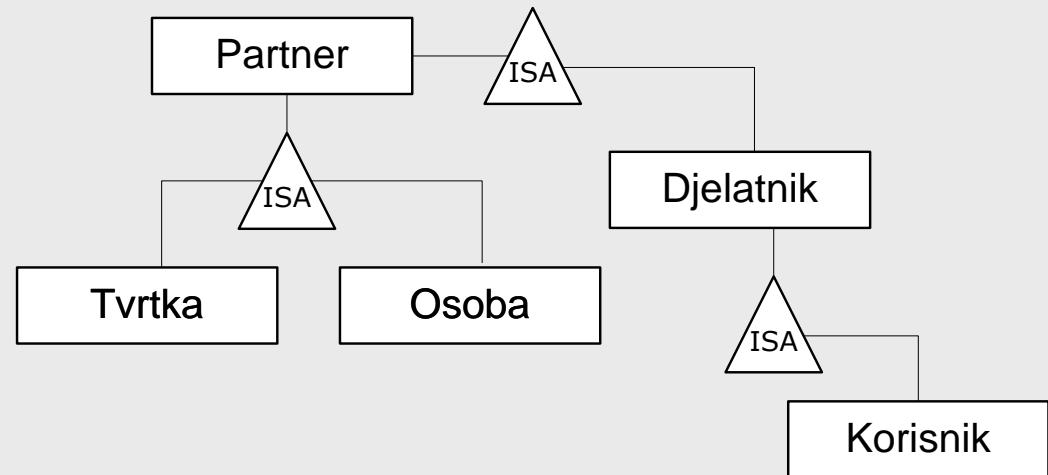
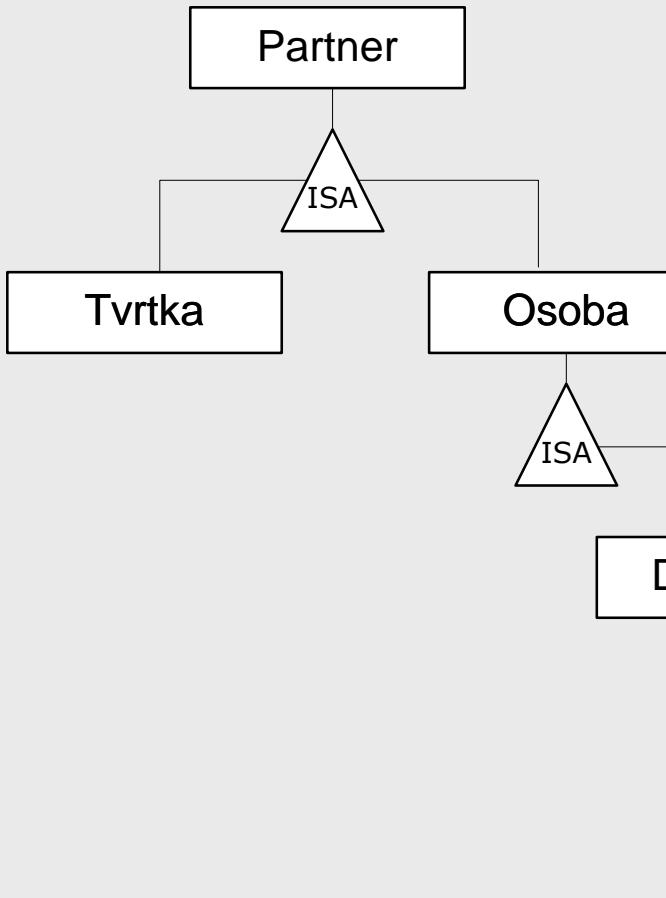
Izrada ERD analizom izjava korisnika

"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 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."



Primjer: hijerarhija specijalizacija

□ Diskusija: Može li bolje ?



□ Diskusija: Može li (još) bolje ?

Logičko modeliranje podataka

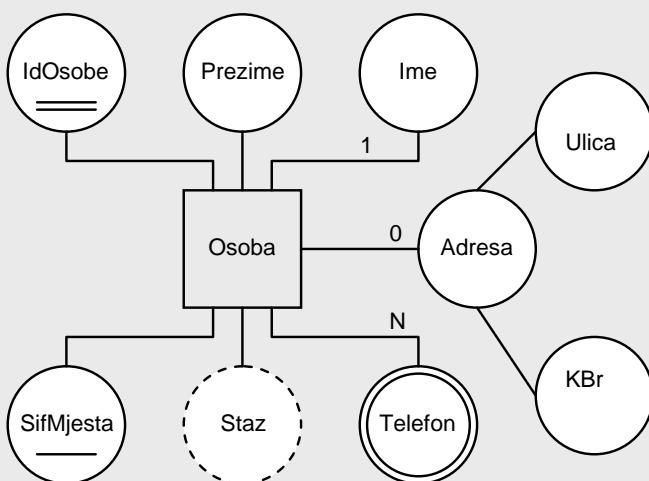
Pretvorba modela E-V u relacijski model

□ 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



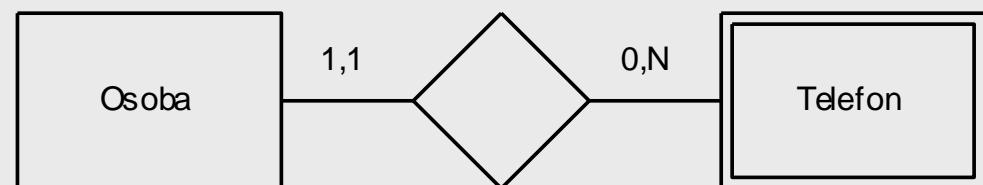
Osoba
IdOsobe
Prezime
Ime
Adresa
DatRod
Staz

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)

Osoba
IdOsobe
Prezime
Ime
TelefonNaPoslu
TelefonKodKuce
MobilniTelefon
Adresa
DatRod
Staz



Telefon
VrstaTelefona
BrojTelefona

Diskusija: Koje rješenje, kada ?

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

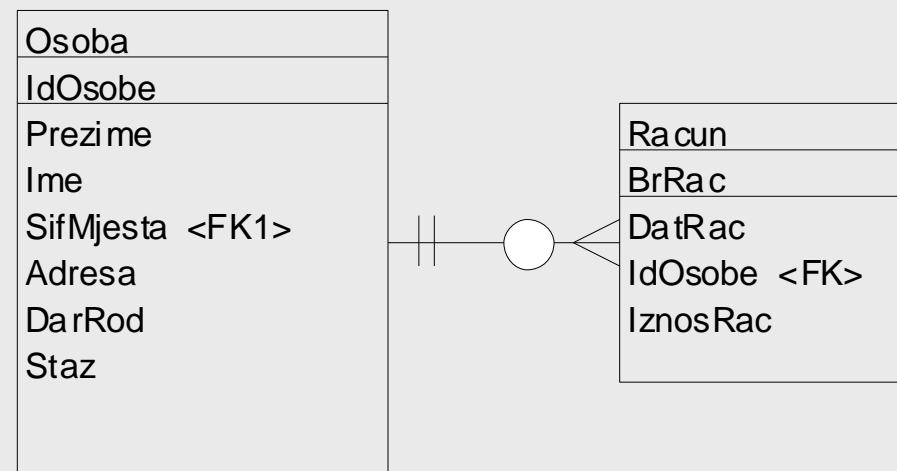
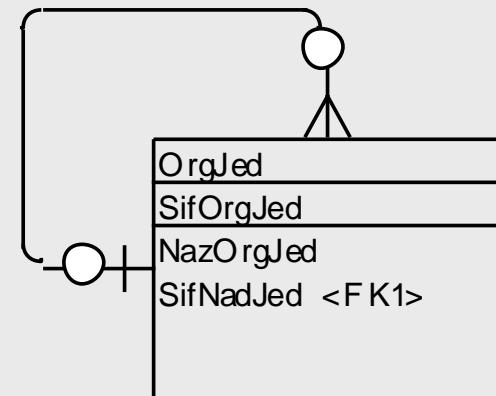
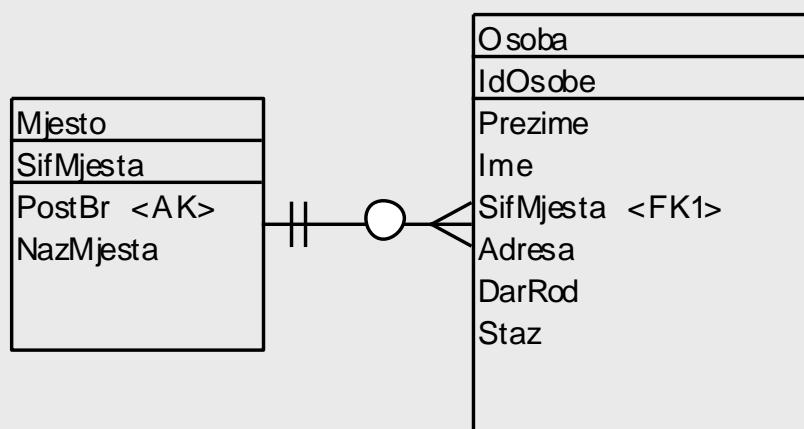
OsobaSKljuce m
IdOsobe
Prezim e
Im e
Adres a
DatRod
Staz

Mjesto
SifMjesta
PostBr <AK>
NazMjesta

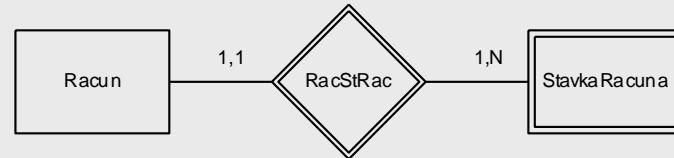
Pretvorba binarnih veza

□ Binarna veza 1:N → strani ključ

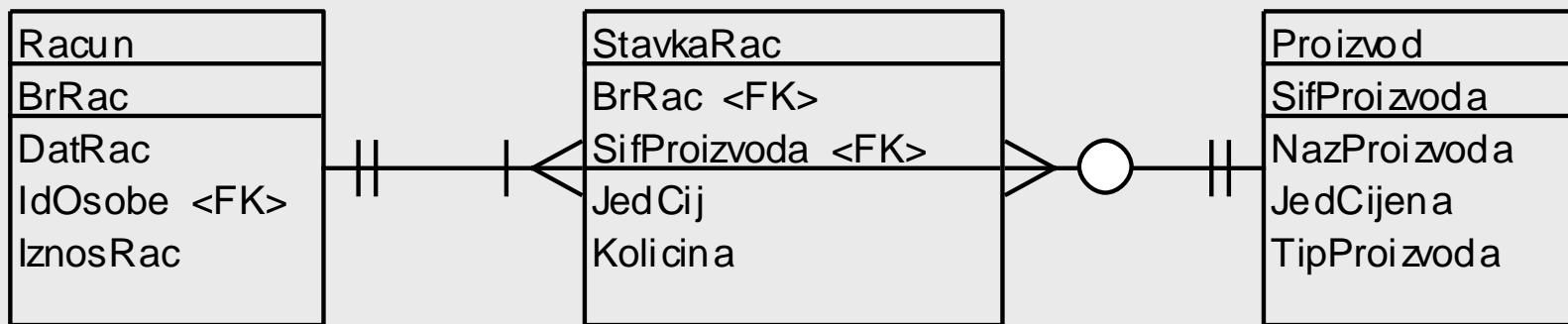
- egzistencijalno slabi entitet → obični strani ključ
 - npr. Stanuje → Osoba.SifMjesta, Pripada → OrgJed.SifNadJed, RacunOsoba → Racun.SifOsobe



Pretvorba identifikacijskih veza



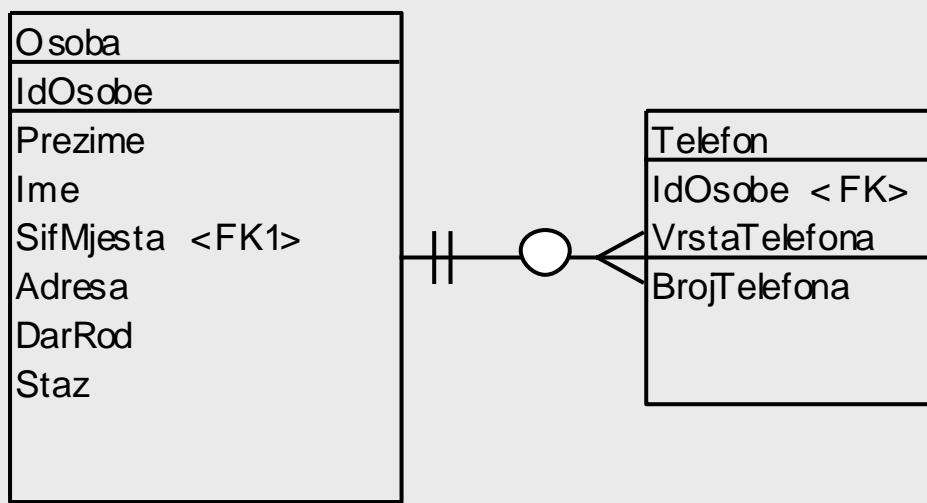
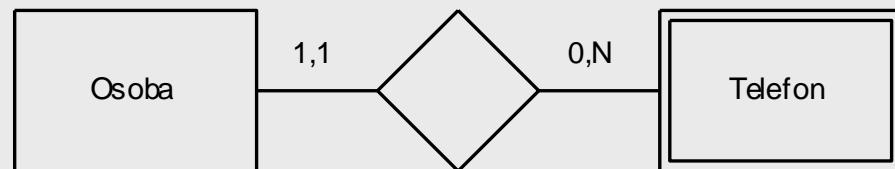
- identifikacijski slabi entitet → nasljeđuje ključ jakog entiteta
 - spojni ključ (compound key), npr. StavkaRacuna (BrRacuna, SifProizvoda, JedCijena, Kolicina)
» ili
 - kompozitni ključ (composite key), npr. StavkaRacuna (BrRacuna, RbrStRac, SifProizvoda, JedCij, Kolicina)



Diskusija: Koje rješenje, kada ?

Pretvorba identifikacijskih veza (2)

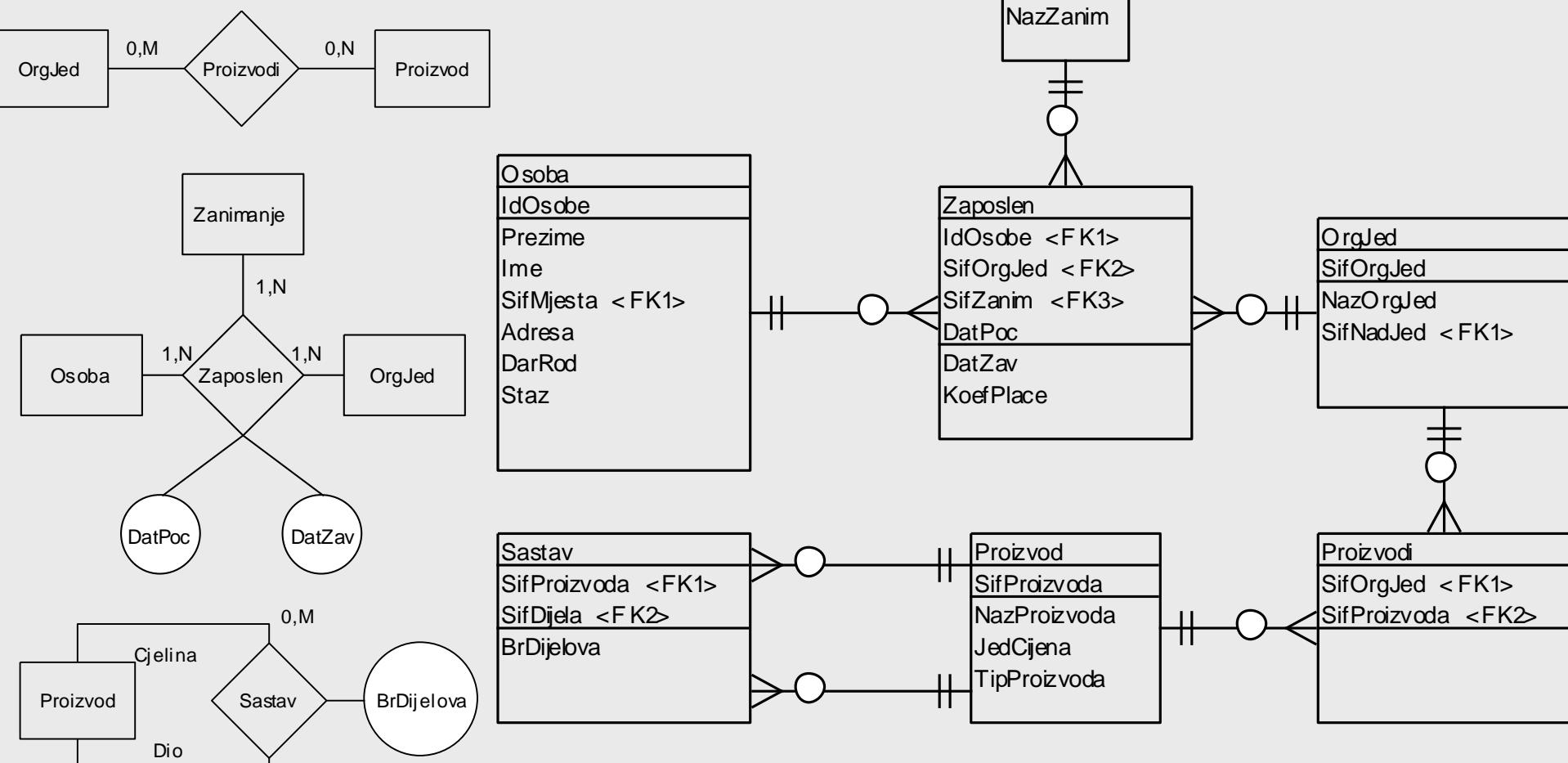
- Koliko i kakvih telefona osoba može imati ?
- Uvesti kompozitni ključ (pogledati prethodni slajd) ?
- Elegantnije rješenje ?



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

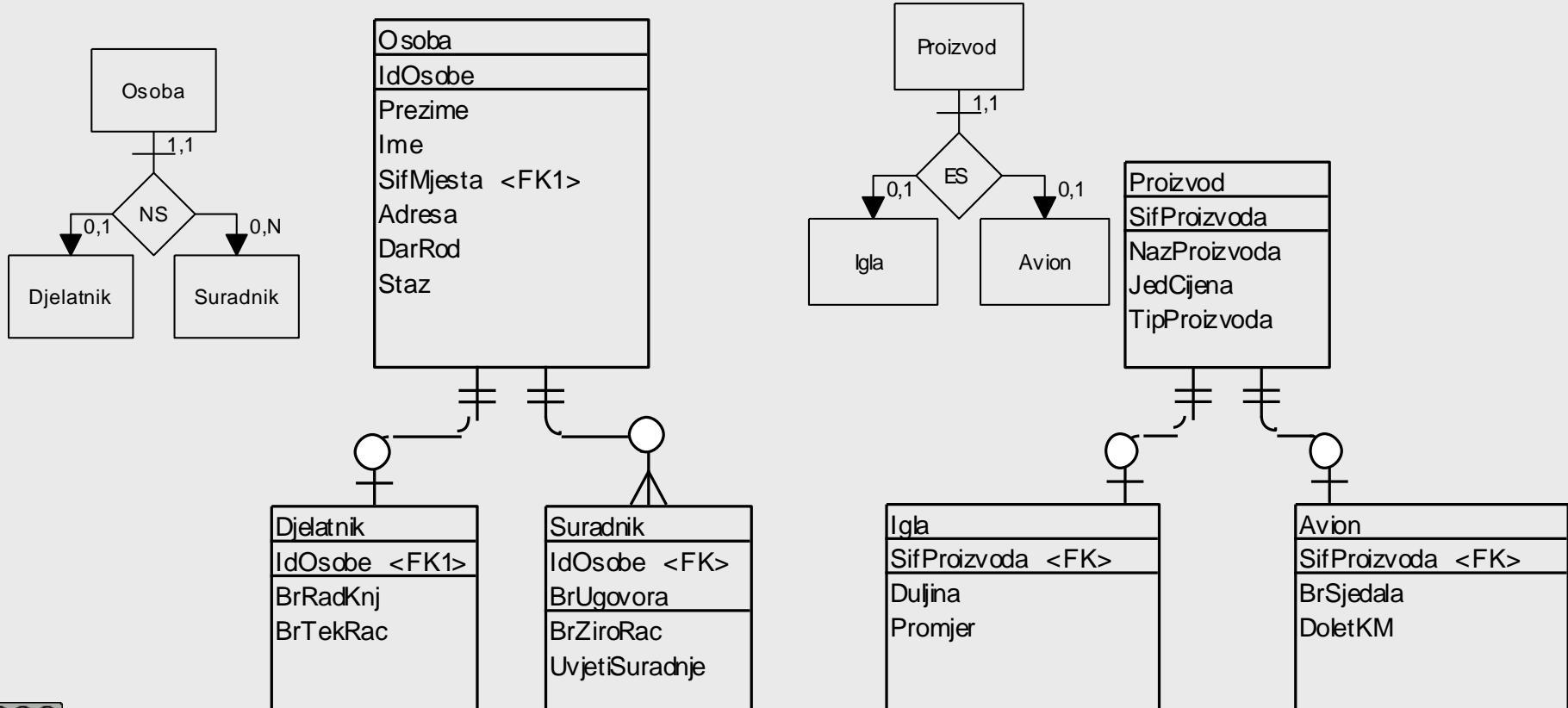


Pretvorba specijalizacija

□ Specijalizacija nadtipa u n podtipova $\rightarrow n$ binarnih veza

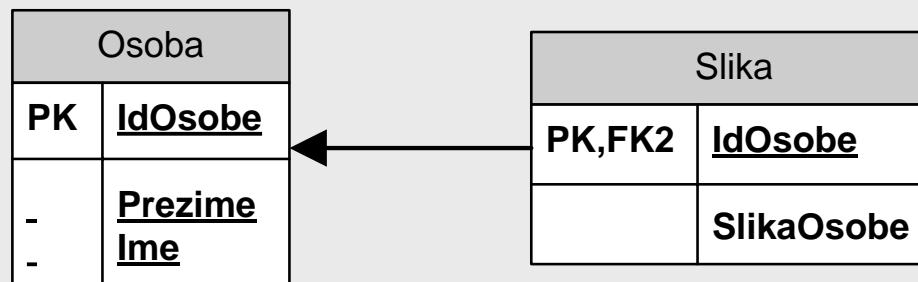
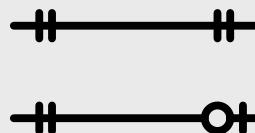
- nadtip \rightarrow (jaki) entitet, kojemu se po potrebi određuje klasifikacijski atribut, npr. Proizvod.TipProizvoda
- podtip \rightarrow (identifikacijski) slabi entitet, npr. Igla, Avion, Djelatnik, Suradnik

□ Da li i kada koristiti klasifikacijski atribut (teorijski, ne bi trebalo) ?

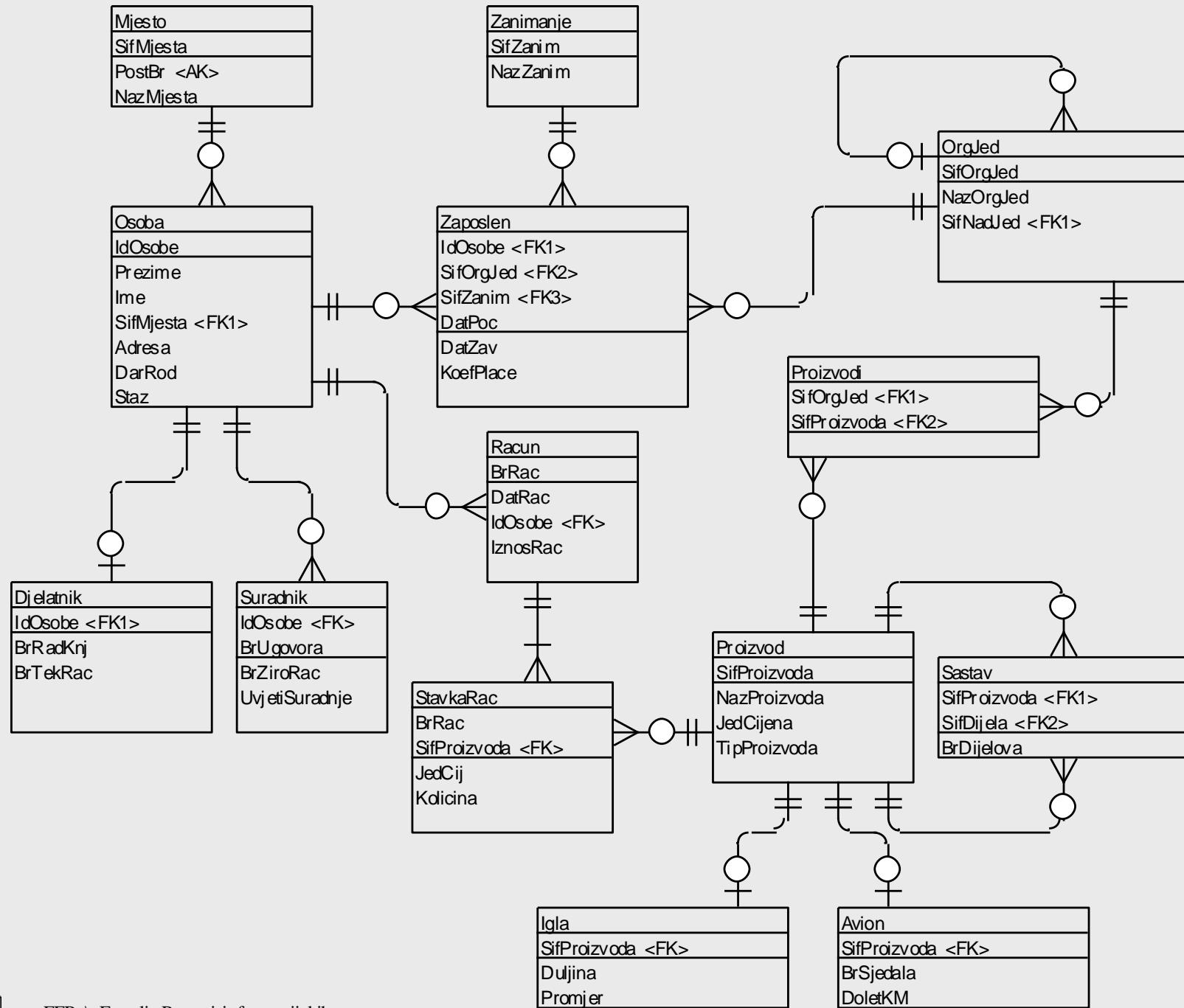


Pretvorba ostalih binarnih veza

- Binarna veza 1,1:0,1 i binarna veza 1,1:1,1 → strani ključ
 - identifikacijski slabi entitet bez deskriptora



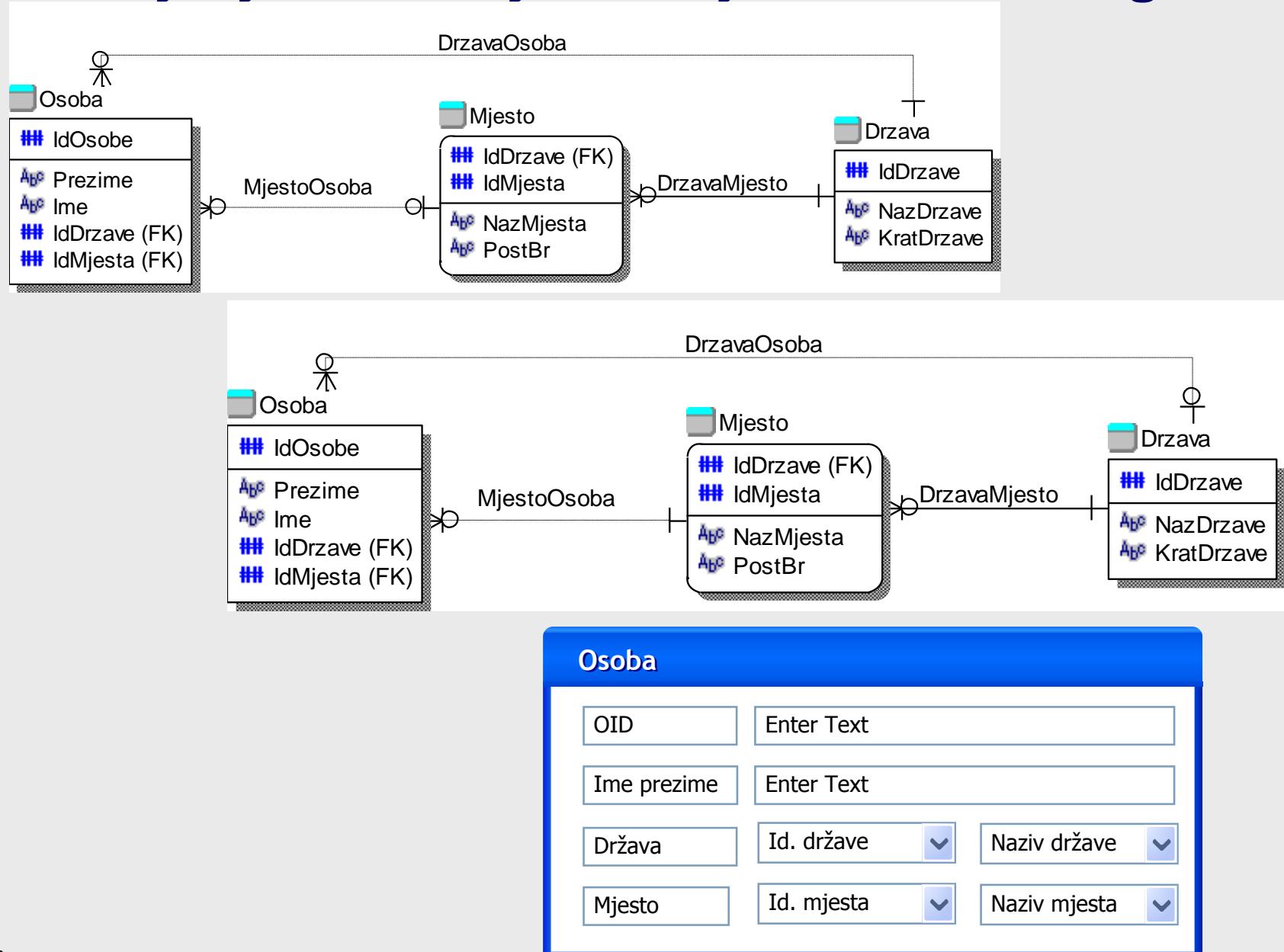
(bez diskusije, bude u nastavku)



Preporuke za izradu modela podataka

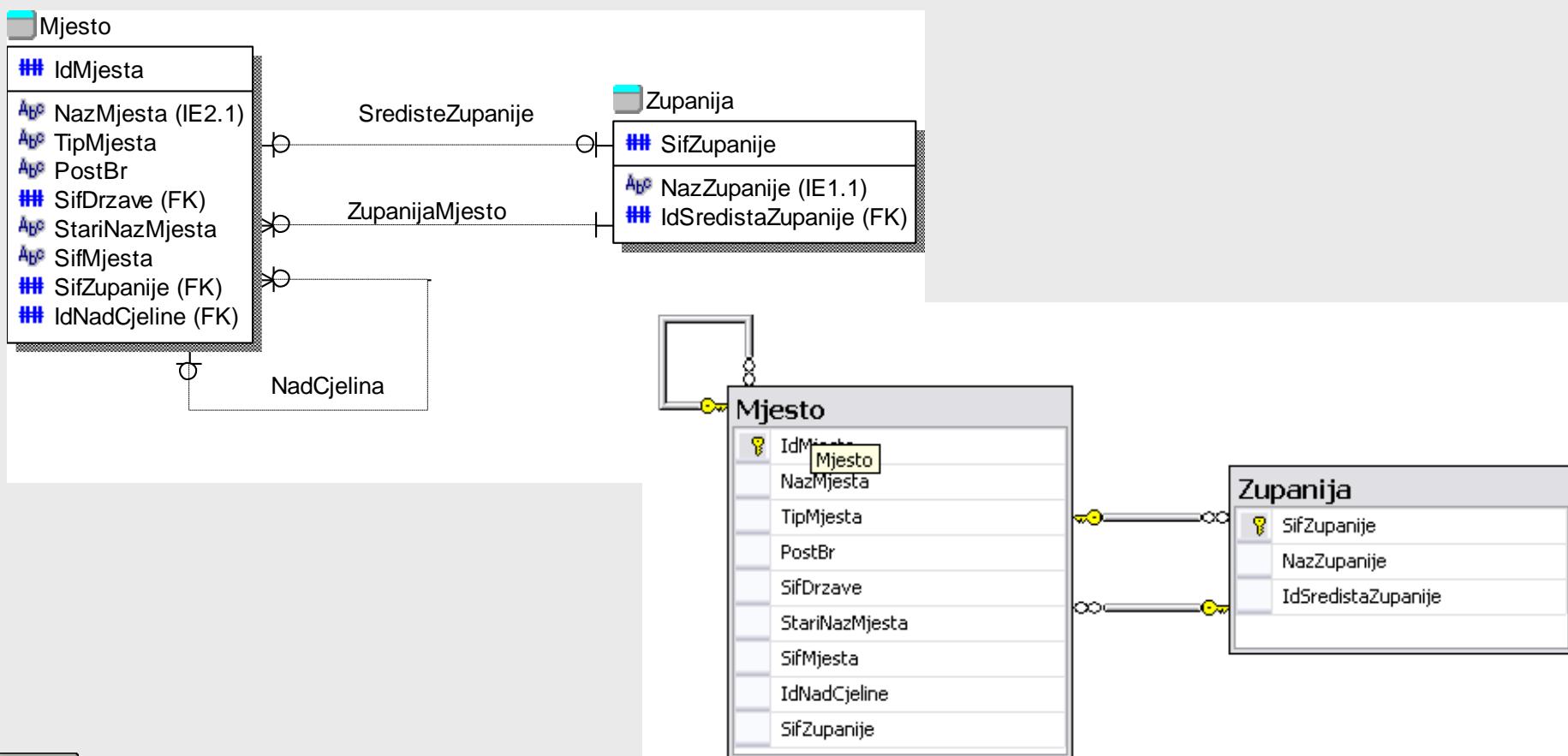
- **Sinkronizacija s modelom procesa**
 - spremišta podataka u modelu procesa postaju entiteti u modelu podataka
- **Provjera modela**
 - pripaziti na sinonime (različite nazine istih objekata)
 - pripaziti na homonime (jednake nazine različitih objekata)
 - provjeriti sumnjive i redundantne veze i po 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 atribute
 - mogući 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



Balansirane i nebalansirane binarne veze

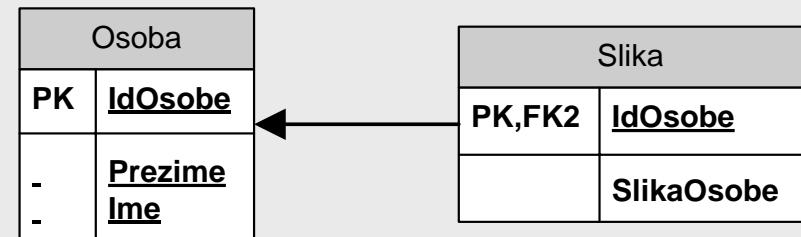
□ Udruživanje u zajedničku tablicu

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



□ Razlozi za udruživanje

- jednostavniji dohvati podataka
- ubrzanje upita



□ Razlozi za razdvajanje

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

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

Šifarski sustav (nastavak)

□ Samogovoreće 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.

□ Šifre s kontrolnom znamenkom

- Zadnja znamenka se izračunava iz svih ostalih
- Detektira se jednostruka pogreška u znamenci ili zamjena mesta dvije znamenke
- JMBG, IBAN (Tekući račun, Žiro račun), "poziv na broj", ...

Primjeri šifrarnika

□ Primjer, studentska prehrana

- Pravilnikom o studentskoj prehrani propisan je status upisa potreban za ostvariranje 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 - jedinstveni broj od upisa do kraja studija

- student upisan na dvije ili više ustanova zadržava JMBAG koji je dobio na prvoj
- broj se generira automatski
- JMBAG ima deset znamenki podijeljenih u dvije grupe te kontrolnu znamenku:
 - Prve četiri znamenke – matična ustanova vlasnika
 - Sljedećih 5 - oznaka vlasnika u ustanovi (matični broj vlasnika ili slijedno)
 - Unutar ustanove JMBG i matični broj (broj indeksa) su također jedinstveni.

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

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

601983 11 0036324986 0

A BC D E

Izrada šifarskog sustava

□ Izrada šifarskog sustava

- treba biti smislen i prikladan da dodavanje novih šifara bude jednostavno
- gdje je moguće treba preuzeti postojeće šifrarnike
 - ustanova ili sustava (državni zavodi, nacionalne institucije, ...)
- oznake definirane zakonom ili drugim propisima treba preuzeti i prilagoditi
- ostale šifrarnike 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

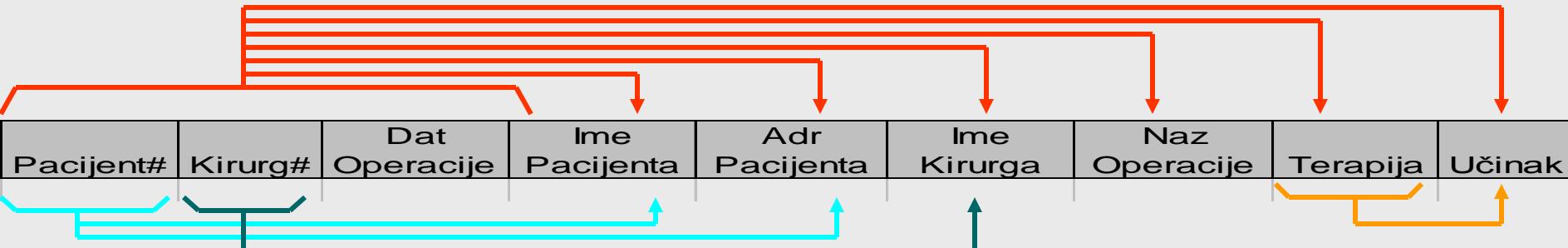
Normalizacija

□ 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 zavisan samo o PK
- Tako mi Codd pomogao! (Dr. E. F. Codd - otac relacijske teorije BP)
 - BCNF: ne postoji kompozitni i/ili preklapajući kandidati ključeva



Denormalizacija

- **Denormalizacija - redukcija modela podataka u nižu NF**
 - ne smije se miješati s pogreškom nedovođenja u jaču NF, lošim dizajnom
 - u tom slučaju denormalizacija mogla dovesti do još većih problema !
- **Razlozi za denormalizaciju**
 - poboljšanje performansi, npr. ubrzanje upita eliminacijom vanjskih spajanja
 - olakšanje pristupa nekim podacima koje je komplikirano 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 strukturirana te kako ju aplikacije koriste
- poznavanje frekvencije obrade podataka

□ Denormalizirati mjestimično i oprezno

- Prije denormalizacije ili za usporedbu probati druge tehnike
- izračunati/virtualni stupci (computed column), funkcije u bazi podataka

□ Analizirati postupke pristupa podacima

- indeksi, optimizacija upita, forsiranje plana izvođenja upita

□ Procijeniti fizičke resurse

- povećanje ili bolja raspodjela radne memorije
- dodavanje ili zamjena procesora
- fizička organizacija podataka (raspored datoteka, segmentacija, ...)
- kvaliteta i optimizacija diskovnog prostora (OS, BP, LOG, BAK)

Podešavanje (umjesto denormalizacije)

□ Postavljanje indeksa

- primarni ključevi implicitno indeksirani
 - ali im treba eksplicitno kreirati sortirani, tzv. clustered indeks
- strani ključevi indeksirani ?
- najčešće korištena polja – pretraživanje, grupiranje, sortiranje
- složeni ključevi "brzi" samo po prvom polju
- masovne obrade - ukloniti indekse, pa 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

□ **NULL vrijednost (razlikovati od broja 0), primjer Ispit.Ocjena:**

- Ne postoji uopće (predmet iz kojeg nema ispita)
- Još ne postoji (ispitivanje još nije završeno)
- Ne zna se (nastavnik nije upisao)
- Neće je ni biti (student nije izašao na prijavljeni ispit)

tumačenje ?

□ **Indeksi ne pomažu u slučaju nul-vrijednosti**

- opisna polja - zahtijevana vrijednost + prepostavljena vrijednost

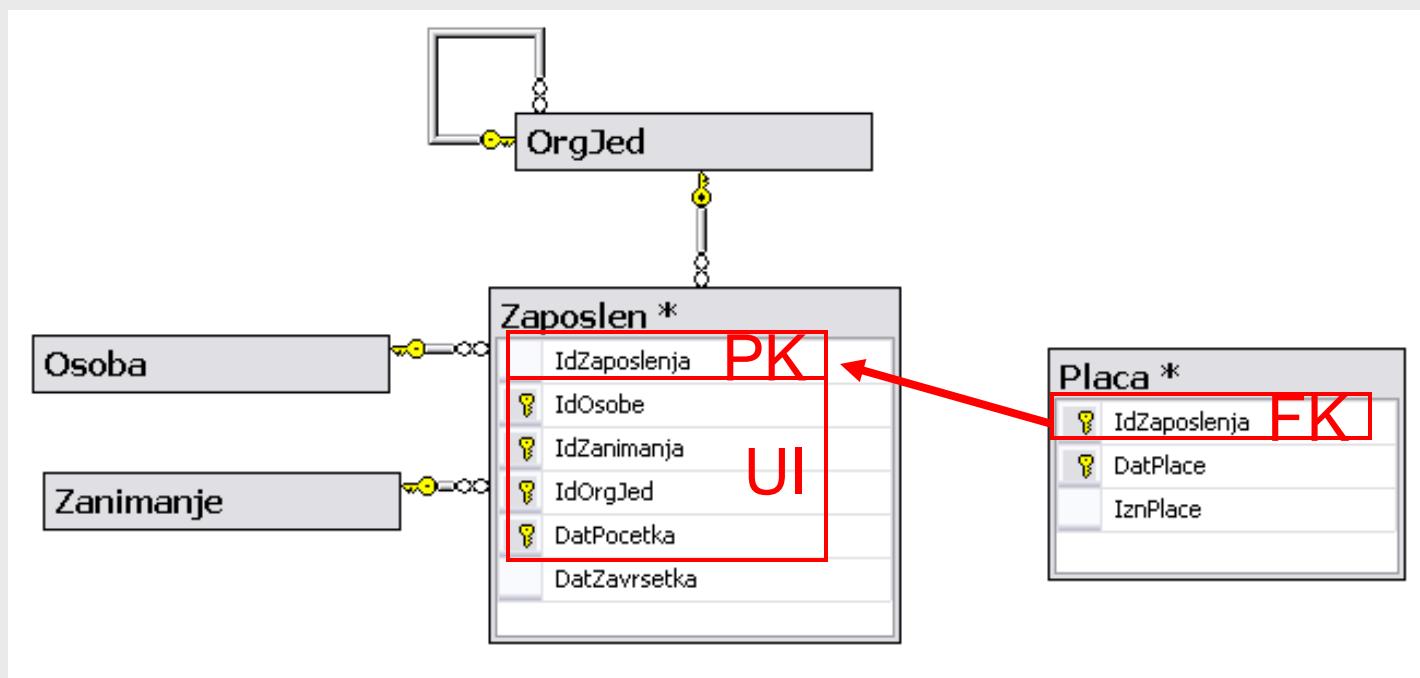
□ **NULL strani ključevi - vanjska spajanja koja ne koriste indekse**

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

Nadomjesni (surogatni) ključevi

☐ Nejasna ili izmjenjiva poslovna pravila, pojednostavljenje veza

- ispred ključa složenog od većeg broja atributa (npr. ≥ 3) umeće se ključ sa samopovećavajućim vrijednostima (serial), a na originalni ključ postavlja se jednoznačan (unique) kompozitni indeks
- teorijski se ne preporuča za asocijativne entitete, koji nasleđuju 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)

The screenshot shows the Microsoft Visual Studio IDE interface. At the top, there is a context menu for a database table named 'Dokument'. The table has columns: IdDokumenta, VrDokumenta, BrDokumenta, and DatDokumenta. Below the table, a dropdown list is displayed with the placeholder 'Prethodni:'. To the right of the dropdown, a 'ComboBox Tasks' window is open. It contains the following settings:

ComboBox Tasks	
<input checked="" type="checkbox"/>	Use data bound items
Data Binding Mode	
Data Source	prethDokumentBindingSource
Display Member	NazDokumenta
Value Member	IdDokumenta
Selected Value	dokumentBindingSource - I

At the bottom of the interface, there are buttons for navigating through the data source, showing '0 od 0' records. Below these buttons are three components: 'oleDbDataAdapterArtikl', 'oleDbTypeConnection1', and 'oleDbData'. The 'oleDbTypeConnection1' component is highlighted with a yellow icon.

"Čisti" dizajn

- **Takozvani "čisti" dizajn – dosljedna ugradnja nadomjesnog ključa sa samopovećavajućim vrijednostima u sve tablice baze podataka**
 - Prednost:
 - pojednostavljenje 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

Izvedeni podaci (derivable data)

□ Dodavanje atributa za vrijednosti koje se daju izračunati iz drugih

- atribut pohrane za vrijednost koja se može izračunati, 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 PPPP)
- 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)

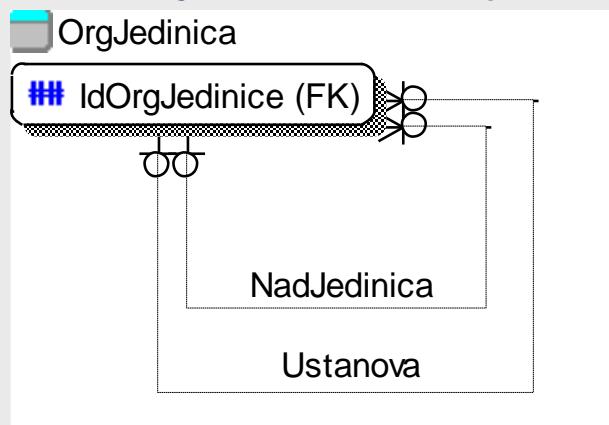
```
UPDATE Dokument SET IznosDokumenta =
  (SELECT SUM(KolArtikla * JedCijArtikla * (100 - PostoRabat) / 100)
   * (1 + Dokument.PostoPorez)
  FROM Stavka WHERE Stavka.IdDokumenta = Dokument.IdDokumenta)
```

Izvedeni podaci (2)

- Redundantna vrijednost koja se inače dohvaća složenim i/ili sporim a često postavljanim upitima
- Primjer, osoba s identifikatorom zadnjeg zaposlenja
 - može naravno i drugim operatorima, npr. $>= \text{ALL}$

```
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



Prethodno povezane tablice (prejoined tables)

- **Izrada osnovne tablice strukture spojnog upita**
 - kada se dvije ili više tablica često povezuje u različitim a (pre)sporim upitima
- **Alternativa**
 - u slučaju postavljanja većeg broja dovoljno brzih spojnih upita – VIEW
- **Varijanta : Kombinirane tablice (combined tables) u vezi 1:1**
 - SlikaOsobe (pogledati "Balansirane i nebalansirane binarne veze")
 - Firma: Partner+Tvrтka+Osoba
 - Poduzeće : Proizvod+Igla+Avion

Tablica izvješća (report table)

□ Tablica izvješća

- Može kombinirati i druge tehnike denormalizacije
- Česta primjena – TEMP TABLE

□ Primjer

```
INSERT INTO Report
    (ControlBreak, VrDokumenta, BrDokumenta, Promet)
SELECT 'dokument', VrDokumenta, BrDokumenta, IznosDokumenta
    FROM Dokument
UNION ALL
SELECT 'ukupno', VrDokumenta, NULL, SUM(IznosDokumenta)
    FROM Dokument GROUP BY VrDokumenta
UNION ALL
SELECT 'sveukupno', 'Z', NULL, SUM(IznosDokumenta)
    FROM Dokument
```

- SELECT FROM Report može zahtijevati sort, ovisno o tome kuda UNION smješta zapise s NULL (na vrh ili na dno grupe)
 - ORDER BY VrDokumenta, ControlBreak
 - u primjeru je radi sorta uvedena i izmišljena vrsta dokumenta 'Z'

Zrcalne tablice (mirrored tables)

□ Zrcalne tablice

- kopije originalnih tablica
- jednake strukture ili dodavanje dodatnih atributa – npr. vremenske dimenzije

□ Primjer:

- transfer u analitički sustav ili dio baze podataka za izvješćivanje
- da se izbjegne opterećenje ili zaključavanje izvornih tablica

□ Primjer:

- evidencija obrisanih zapisa

Dijeljenje, cijepanje tablica (table splitting)

□ Dijeljenje, cijepanje tablica

- vertikalno ili horizontalno – izvorna tablica u dvije ili više
- ako se zadrži izvorna – varijanta zrcaljenja
- razlog: lakše ili brže rukovanje podacima
- implicira kasnije kreiranje upita (pogleda) za agregaciju dijelova
 - UNION ALL za horizontalnu podjelu
 - INNER JOIN za vertikalnu podjelu

□ Kriteriji dijeljenja i primjeri

- fizički - po jedna tablica za svaki terminal
- prostorno - po državama, županijama
- vremenski - po jedna tablica za svaki mjesec ili godinu
- proceduralno - po jedna tablica za svaki korak poslovne procedure
- administrativno - po jedna tablica za svaki odjel umjesto za čitavo poduzeće

□ Neki problemi

- horizontalna podjela – konflikt primarnog ključa
- vertikalna podjela – simultano kreiranje zapisa s istim ključem u više tablica

Redundantni podaci (redundant data)

□ Redundantni podaci

- udruživanje čitave referencirane tablice u zavisnu
- primjer: Osoba + Spol gdje je Spol = @OznSpola INT, NazSpola CHAR
- anti primjer: Osoba + Mjesto

□ preporuke, uvjeti primjene:

- primjena samo na mali broj stupaca
- primjena samo na vremenski stabilne podatke
- pisanje programskog koda za održavanje

Ponavljajuće grupe (repeating groups)

□ Ponavljajuće grupe (repeating groups)

- izvorno: Naplata = @IdKupca, NazKupca, @TipIznosa, Iznos
- denormalizirano: Naplata = @IdKupca, NazKupca, Iznos1, ..., IznosN

□ preporuke, uvjeti primjene:

- kada je broj stupaca ponavljajuće grupe mali i stabilan
- kada se grupi pristupa kolektivno a ne individualno, tj. kada je grupa suvisla cjelina (n-torka) a ne skup vrijednosti pretvoren u atribut
 - pr.1 Osoba: @IdOsobe, ..., Državljanstvo1, Državljanstvo2
 - pr.2. KoodinataNalazišta: @IdNalazišta, OpisNalazišta, X, Y, Z
- postoji spremnost na rukovanje pojedinačnim NULL vrijednostima
- nije potrebno agregirati grupe unutar retka (zаписа)

Preopterećeni tipovi podataka (overloaded datatypes)

□ Udruživanje šifrarnika

- Sifrarnik<i> = @Sifra<i> <datatype>, Naziv<i>
 - pr. Mjesto = @PostBr INT, NazMesta VARCHAR
 - pr. Valuta = @KratValute VARCHAR, NazValute VARCHAR
 - pr. Koeficijent = @SifraKoeficijenta, VrijKoeficijenta DECIMAL(5,2)
- integriramo uvođenjem oznake šifrarnika (Vrsta) te nastaje
- Sifrarnik = @Vrsta VARCHAR, @Sifra VARCHAR, Naziv VARCHAR

□ prednost:

- izrada jedne (zajedničke) komponente za održavanje šifrarnika

□ problemi: glavni je, naravno, narušavanje 1NF

- konverzija tipa podatka izvornih ključeva i zavisnog atributa u VARCHAR
- validacija unosa
- strani ključevi zavisnih tablica
- ograničenje na šifrarnike s jednostavnim ključem
- sigurnost (kontrola pristupa)

Meta-modeliranje

**Rječnik podataka
Meta-baza
Primjena meta-modela u izradi aplikacija**

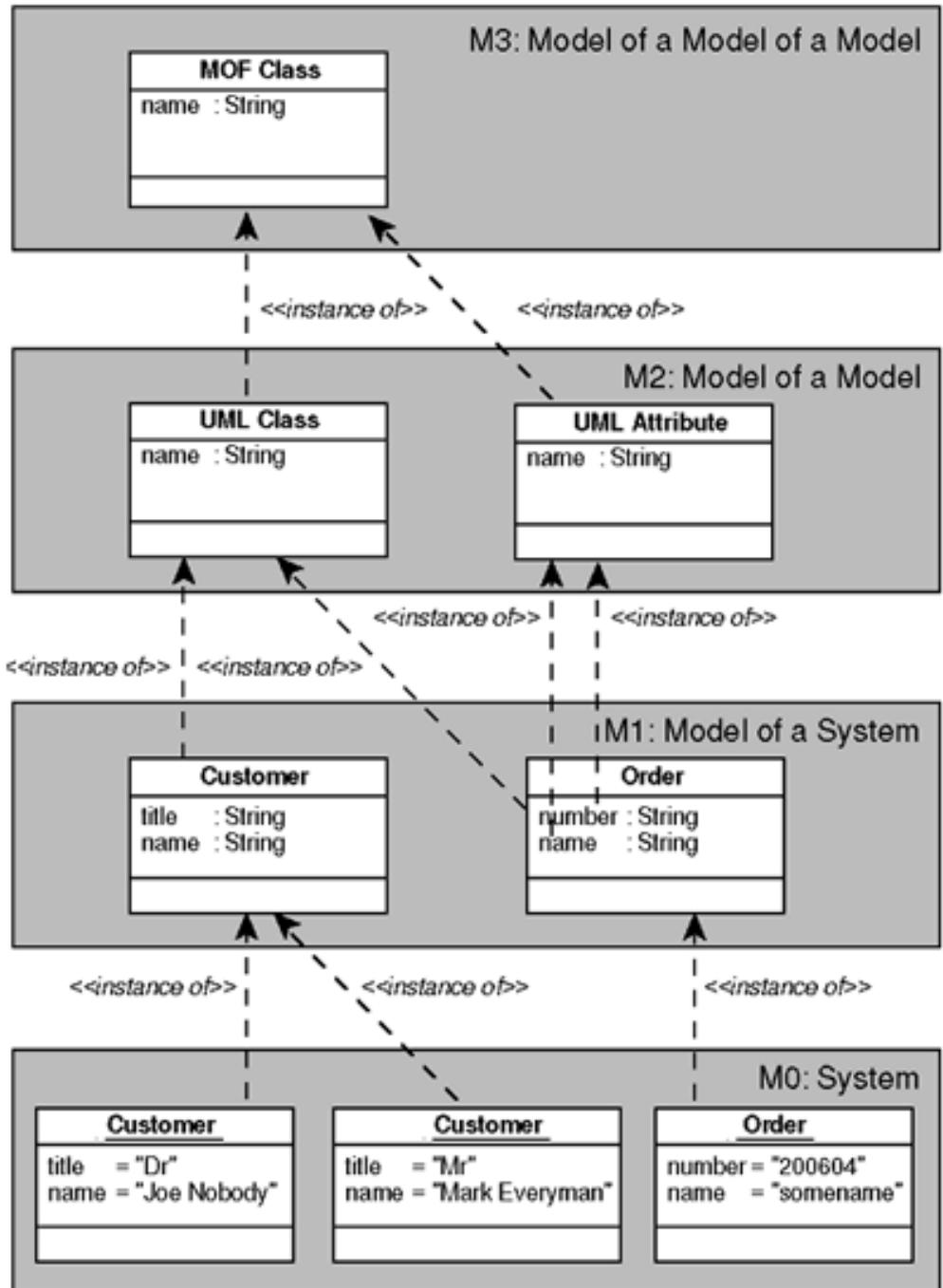
Meta-modeliranje

□ Meta-modeliranje = modeliranje podataka o podacima

- oznaka "meta" označava višu razinu apstrakcije
- metapodaci - podaci o podacima
- metamodel - model modela

□ Primjene

- opisivanje (baze) podataka
- generiranje aplikacija, podataka, meta podataka, ...
- dinamička prilagodba aplikacije podacima, supstitucija sličnih tablica, ...
- pohrana i razmjena podataka, npr. eLearning Digital Object

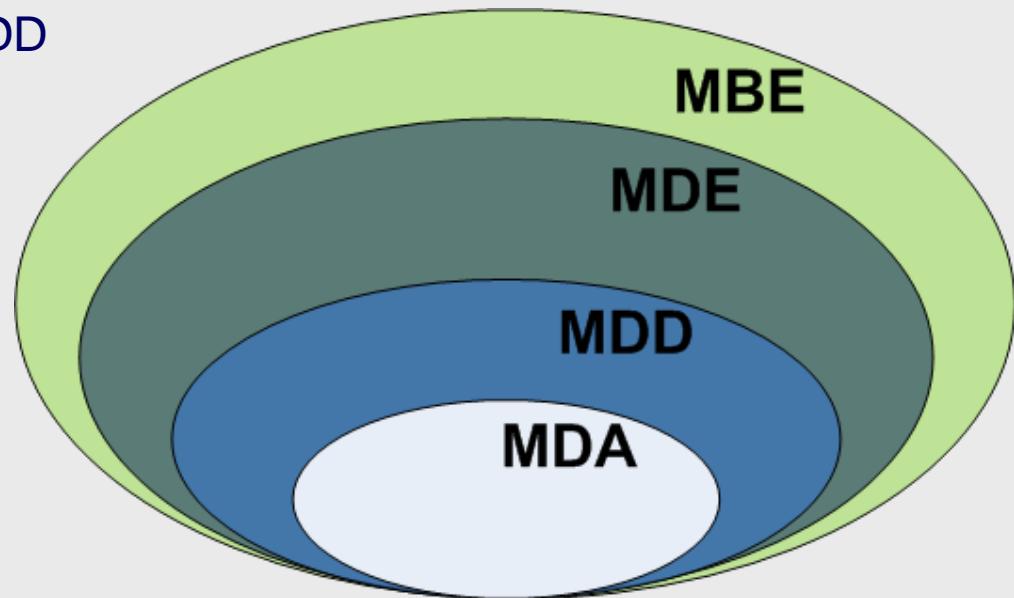


Opisivanje i pohrana (meta)modela

- **Metamodel – model (za opisivanje drugih) modela**
- **Metabaza – baza podataka za pohranu modela**
 - Rječnik podataka – dio SUBP s podacima za upravljanje BP
 - definicija strukture, evidencija dozvola, ...
 - Riznica, repozitorij (metapodataka) – dio CASE alata
 - pohrana specifikacija, definiranje transformacija, ...
- **Shema – instanca metasheme**
- **Metashema – instanca meta-metasheme**
- **Metamodeliranje – definicija i upotreba meta-metashema**

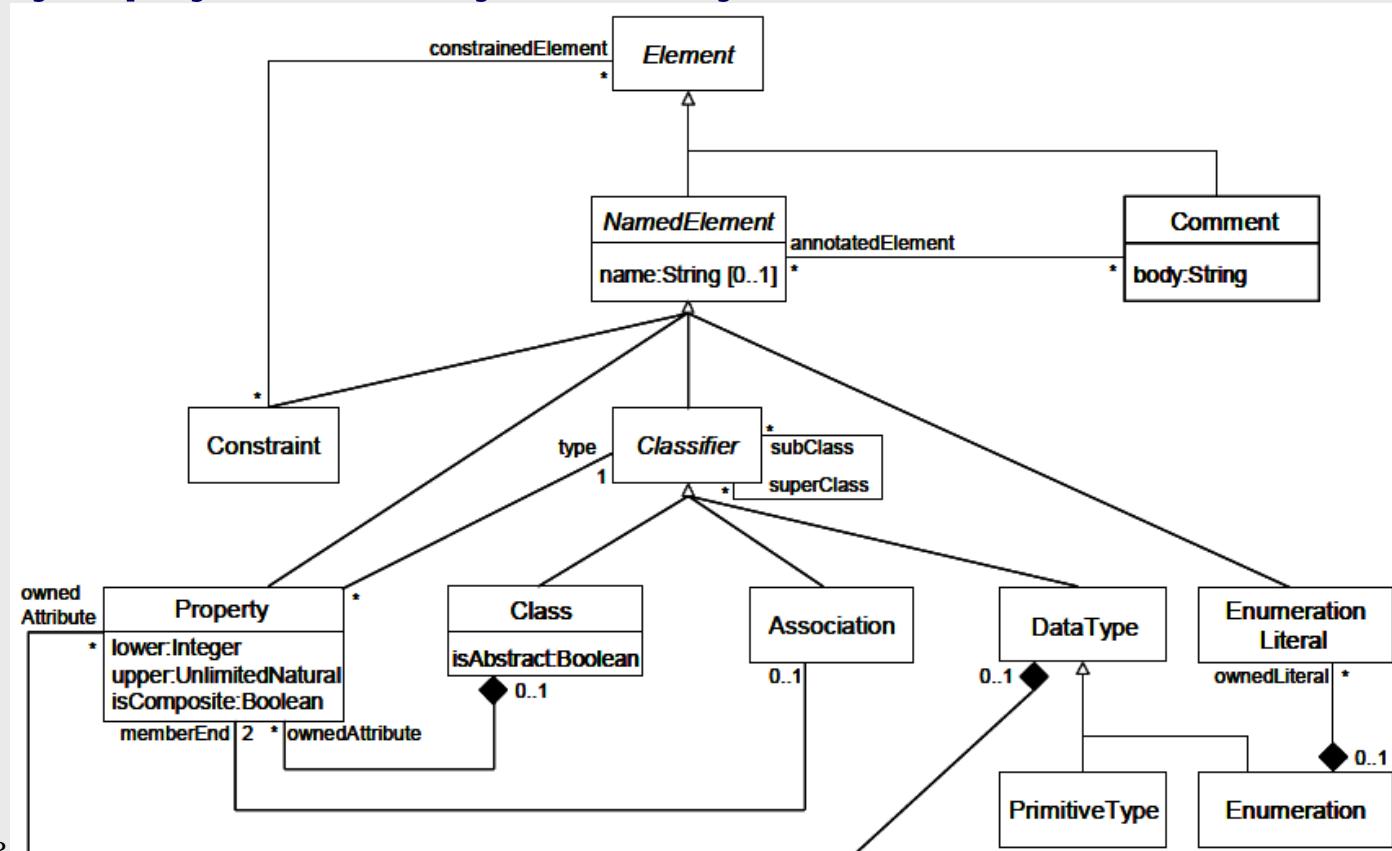
Modelom vođeno ...

- **Model-Based Engineering/Development**
 - Softverska verzija MDE, u kojoj modeli još nisu automatizirani
- **Model-Driven Engineering**
 - Nadilazi razvojni proces
- **Model-Driven Development**
 - Modeli „vode“ proces, (polu)automatskim generiranjem
- **Model-Driven Architecture**
 - OMGova vizija ili standard MDD



[jedan] Jezik za opis metamodela

- Meta Object Facility (MOF) - Object Management Group (OMG)
 - Standard za MDE, najpoznatija meta-metashema
 - Essential, EMOF - proširivi modelom vođeni okvir
 - Complete, CMOF – druge metasheme kao instance MOF (npr. UML2)
- Primjer, pojednostavljena verzija MOF



Životni ciklus MDA

□ CIM – računalno neovisan

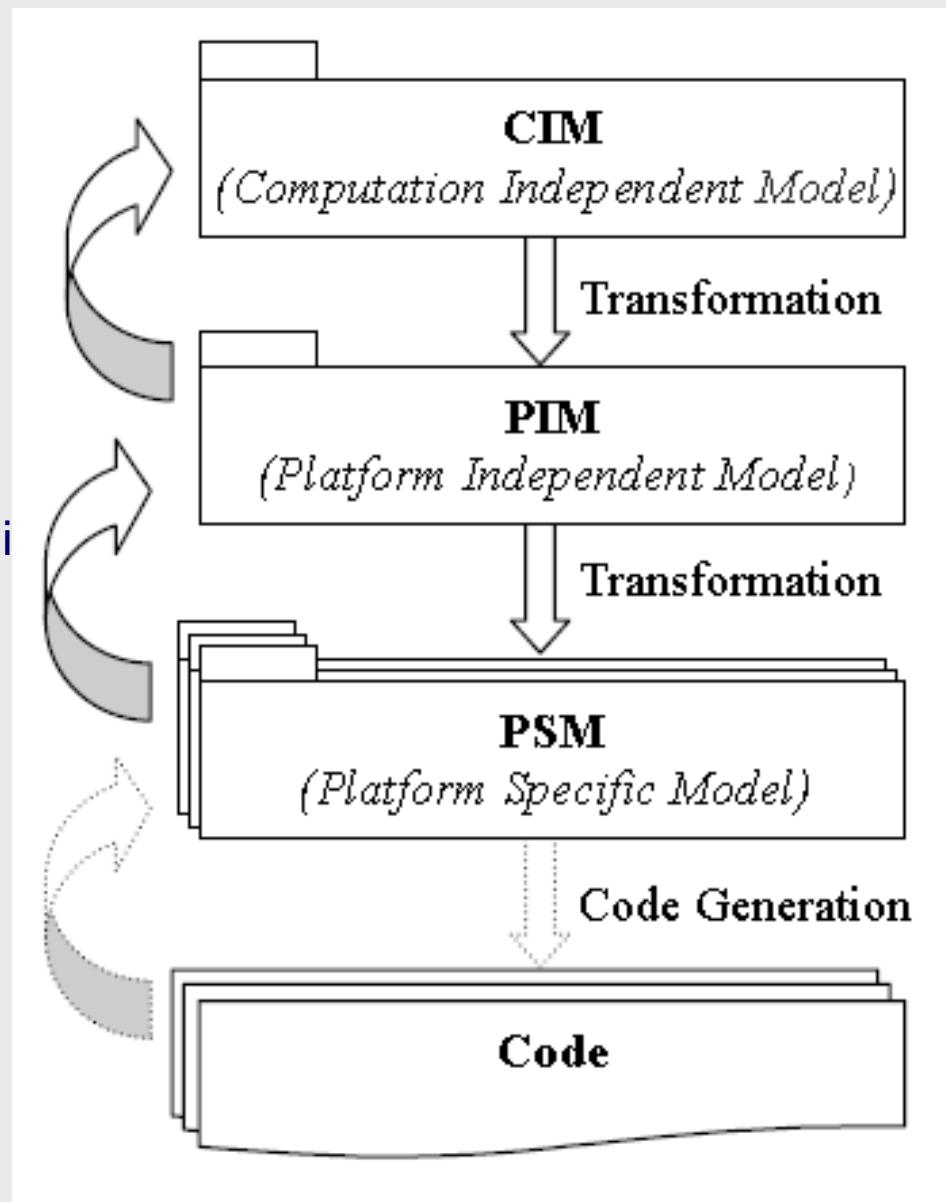
- Model domene, poslovni model

□ PIM – platformski neovisan

- Analiza i dizajn
- UML kao PIM: obični (essential) i izvršni (executable)

□ PSM - platformski ovisan

- implementacija



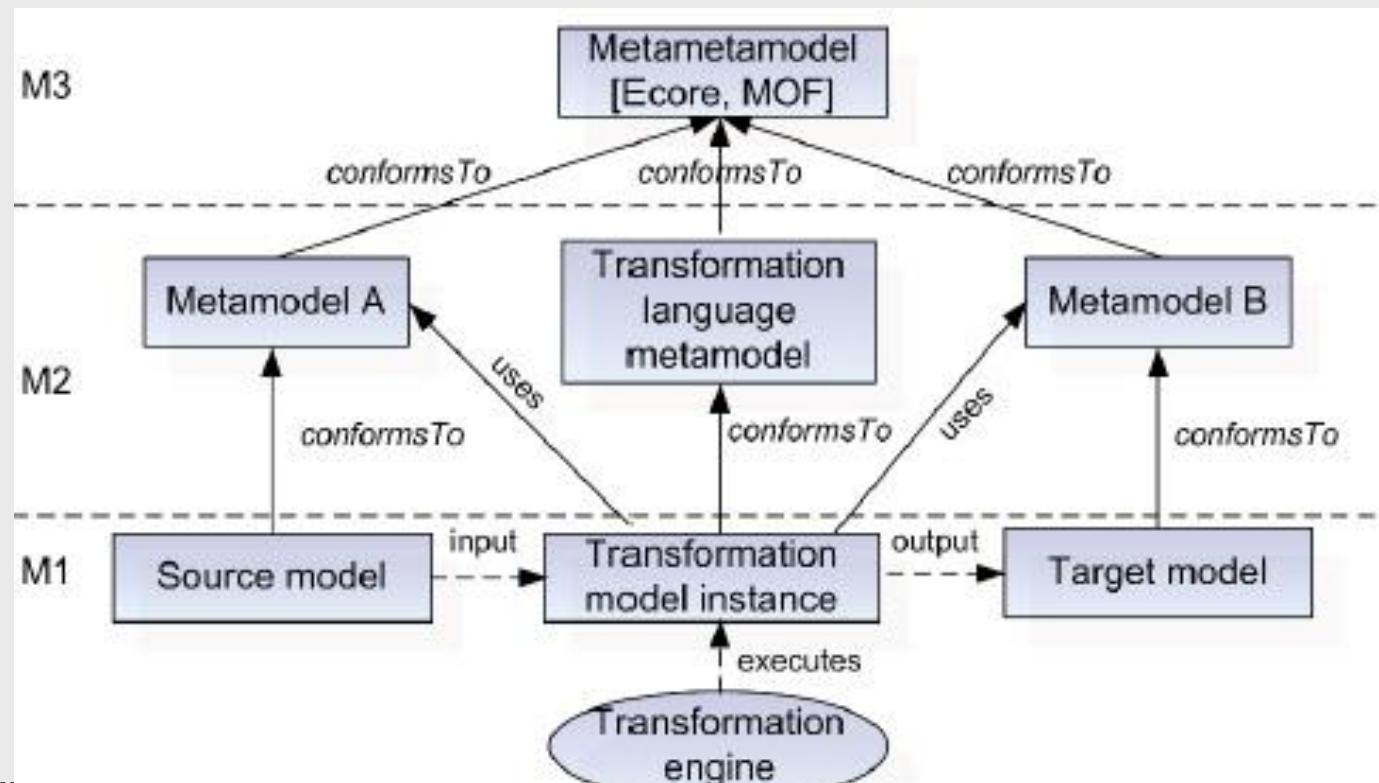
Transformacija modela

□ Transformacija

- jednog modela koji je sukladan s nekim metamodelom u drugi model sukladan nekom drugom metamodelu

□ Jezici za transformaciju

- QVT (Query/View/Transformation) – OMG standardiziran, opisan u MOF
- Atlas Transformation Language (ATL) – hibridni
- ...



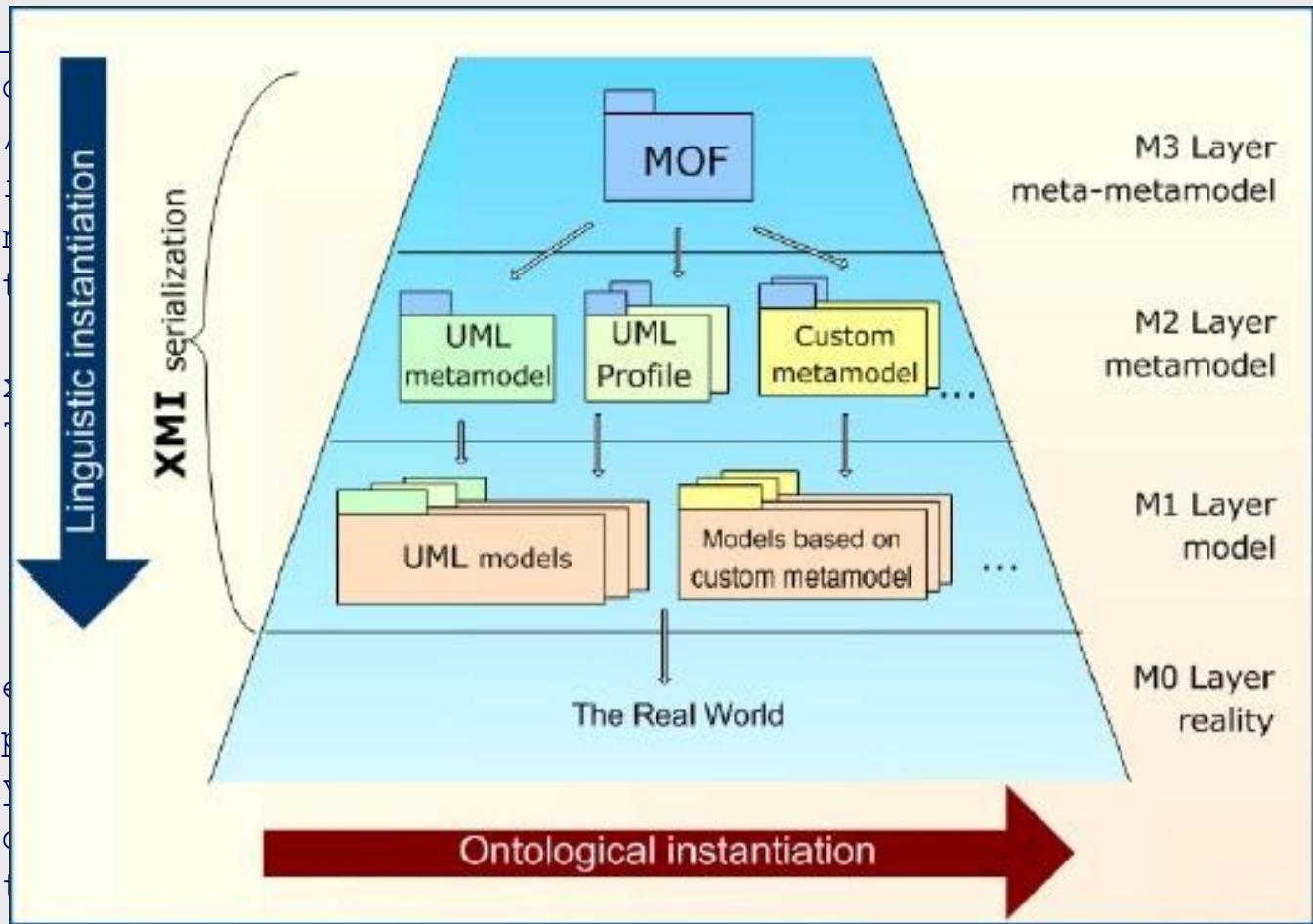
XML Metadata Interchange (XMI)

□ XML Metadata Interchange (XMI)

- OMG standard za razmjenu metapodataka XMLom
- Sadrži opis fizičke reprezentacije entiteta i veza

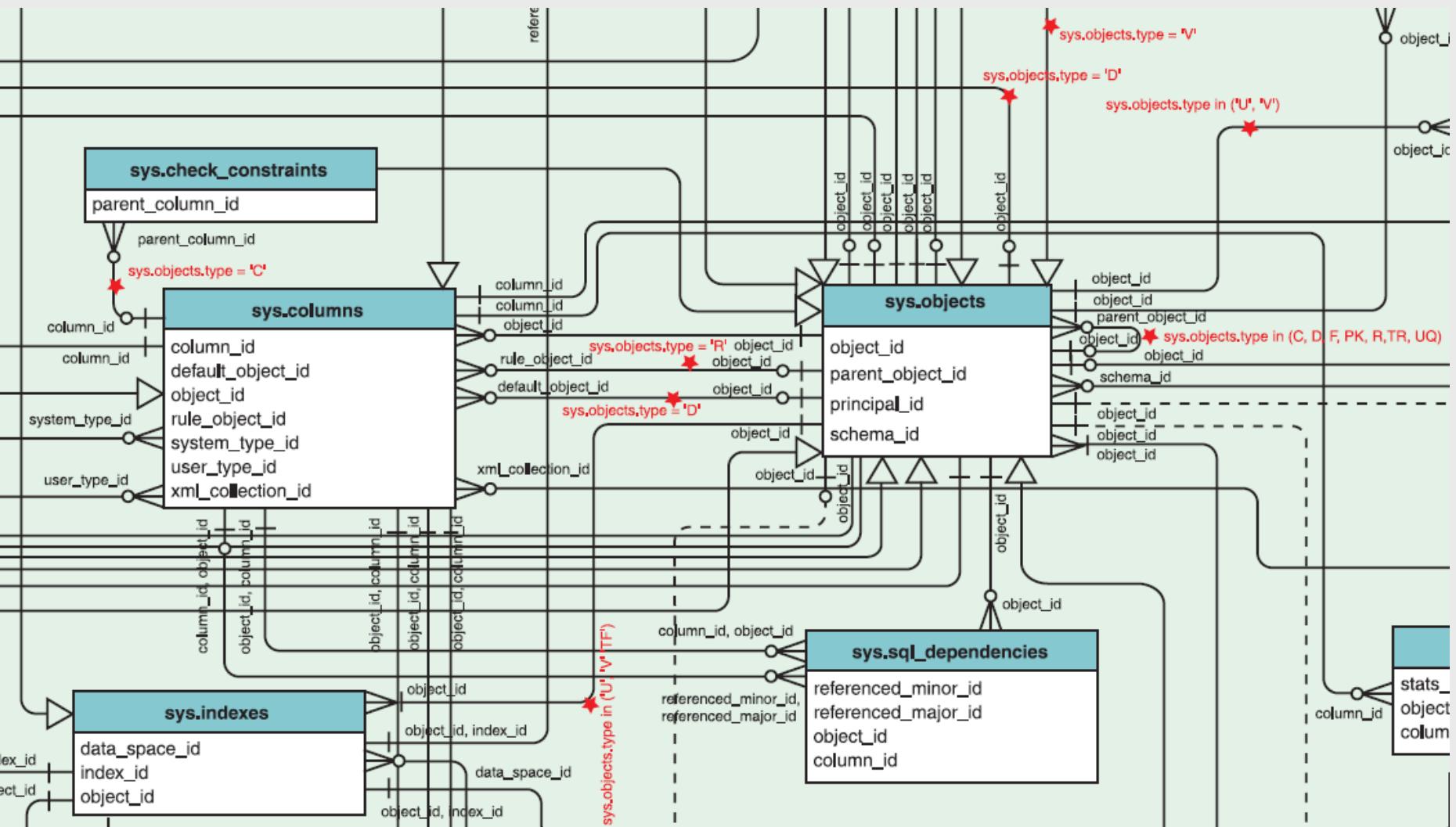
□ Primjer, XMI

```
<xmi:XMI xmi:version="2.0">
  xmlns:xmi = "http://www.omg.org/XMI"
  <OriginalBook xmi:id="1">
    language = "en"
    <isbn nil = "true"/>
  </OriginalBook>
  <TranslatedBook xmi:id="2">
    isbn = "84,721,2345,6"
    ...
    <Person xmi:id = "3">
      ...
      <DataType xmi:id="4">
        <ownedAttribute lower = 1 upper = 100>
          <type xmi:type="xsd:string" href="doc#Color"/>
        </ownedAttribute>
      </DataType>
    </Person>
  </TranslatedBook>
</xmi:XMI>
```



Shema 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 -- isto što i sys.objects  
ORDER BY xtype, name;
```

Korištenje rječnika

□ Primjer: baza podataka FirmaRIS, procedura ap_CountALL

```
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
```

	Results	Messages
1	Mjesto	9120
1	Partner	554
1	Osoba	63
1	Tvrtka	491
1	sysdiagrams	1
1	Dokument	857
1	Artikl	1531
1	Drzava	239

INFORMATION_SCHEMA

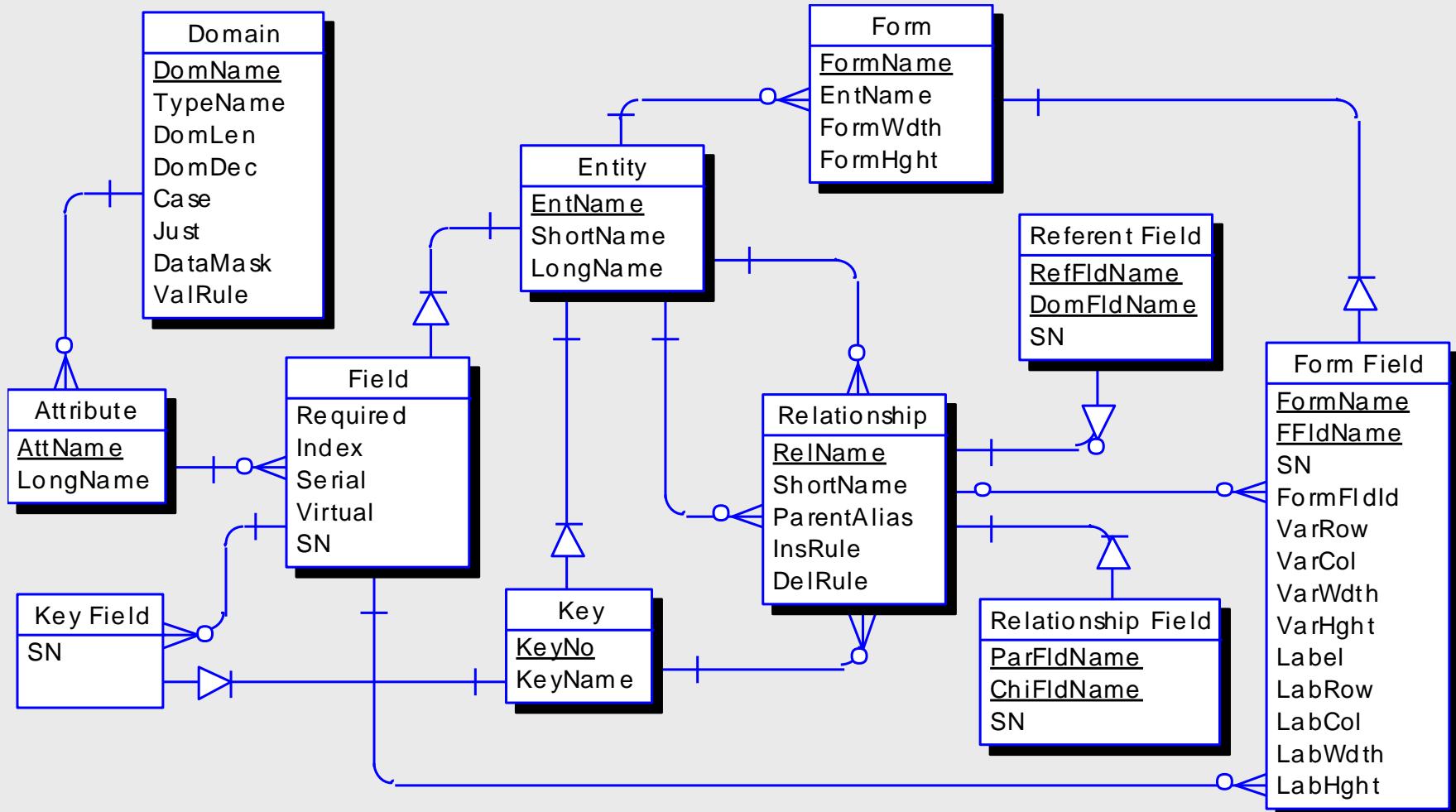
□ Sistemski nezavisni pogled na metapodatke - ISO standard

CHECK_CONSTRAINTS	REFERENTIAL_CONSTRAINTS
COLUMN_DOMAIN_USAGE	ROUTINES
COLUMN_PRIVILEGES	ROUTINE_COLUMNS
COLUMNS	SCHEMATA
CONSTRAINT_COLUMN_USAGE	TABLE_CONSTRAINTS
CONSTRAINT_TABLE_USAGE	TABLE_PRIVILEGES
DOMAIN_CONSTRAINTS	TABLES
DOMAINS	VIEW_COLUMN_USAGE
KEY_COLUMN_USAGE	VIEW_TABLE_USAGE
PARAMETERS	VIEWS

□ Primjer:

```
SELECT TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME,  
       COLUMN_NAME, COLUMN_DEFAULT  
FROM FirmaRIS.INFORMATION_SCHEMA.COLUMNS
```

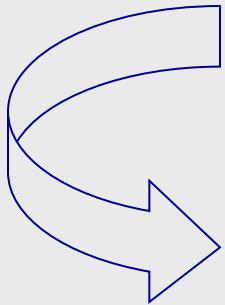
Model meta-baze generatora aplikacija



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
);
...
```

- Realizacije: XML, XSLT, ... SQL, LINQ, ..., XMI, QVT, ...

Meta-modeliranje aplikacije

□ Primjer: Firma

- Zahtjevnica, Upit, Ponuda, Narudžbenica, Otpremnica, Primka, Račun ...
 - Svi slične ili jednake strukture kao npr. Račun na slajdu „Pretvorba identifikacijskih veza“
- Zamijenimo ih s Dokument = VrstaDokumenta + BrojDokumenta
 - Vrsta Dokumenta $\in \{ Z, U, P, N, O, T, R \}$

□ Prednosti

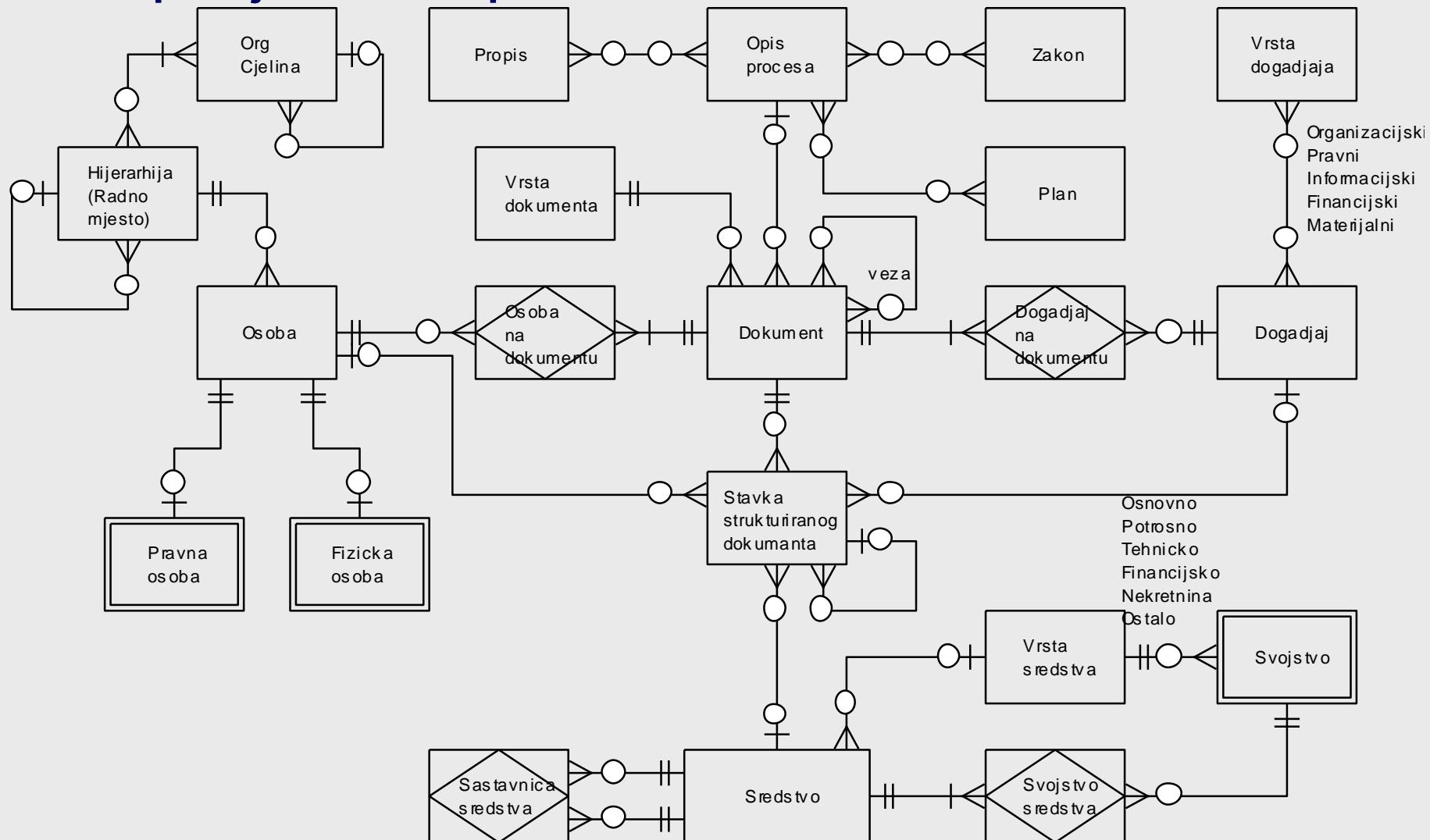
- Pojednostavljanje modela
- Smanjenje broja objekata

□ 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

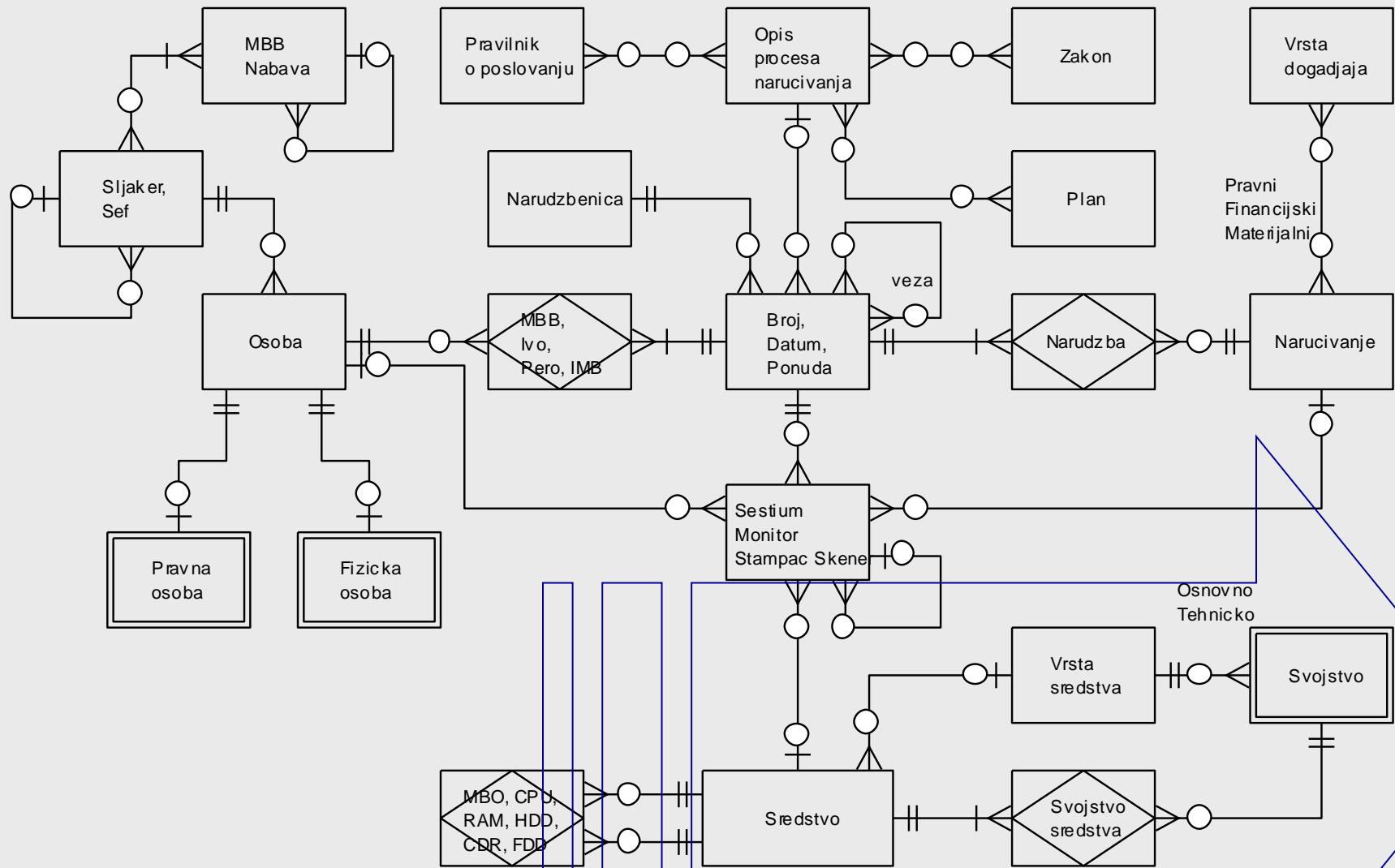
Monitor 15"	xx kom
-------------	--------

Štampač	xx kom
---------	--------

Skener	xx kom
--------	--------

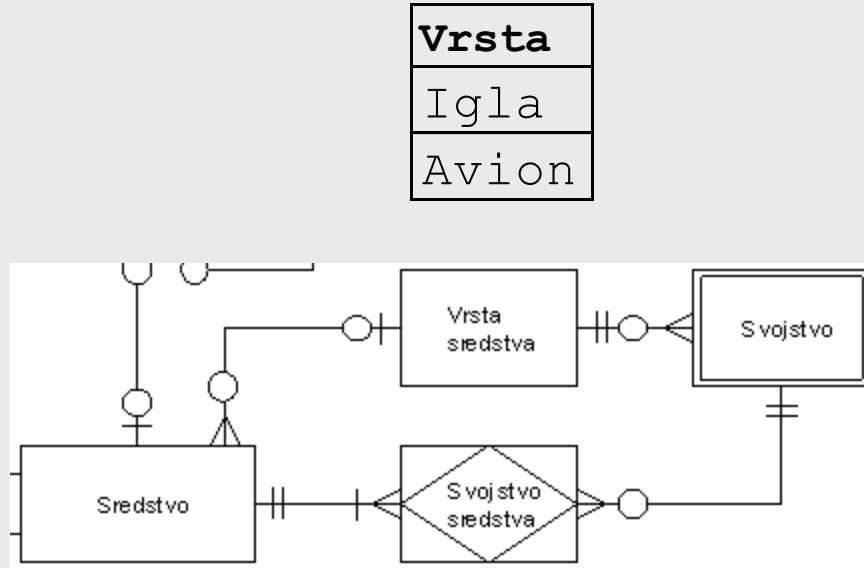
Ideja aplikacijskog meta-modeliranja (3)

□ Primjer, narudžba



Primjena na opisivanje podataka

□ Primjer: opisivanje podataka



Sredstvo	Vrsta
A380	Avion
B2	Avion

Vrsta	Svojstvo
Avion	Broj Sjedala
Avion	Dolet
Igla	Duljina
Igla	Promjer

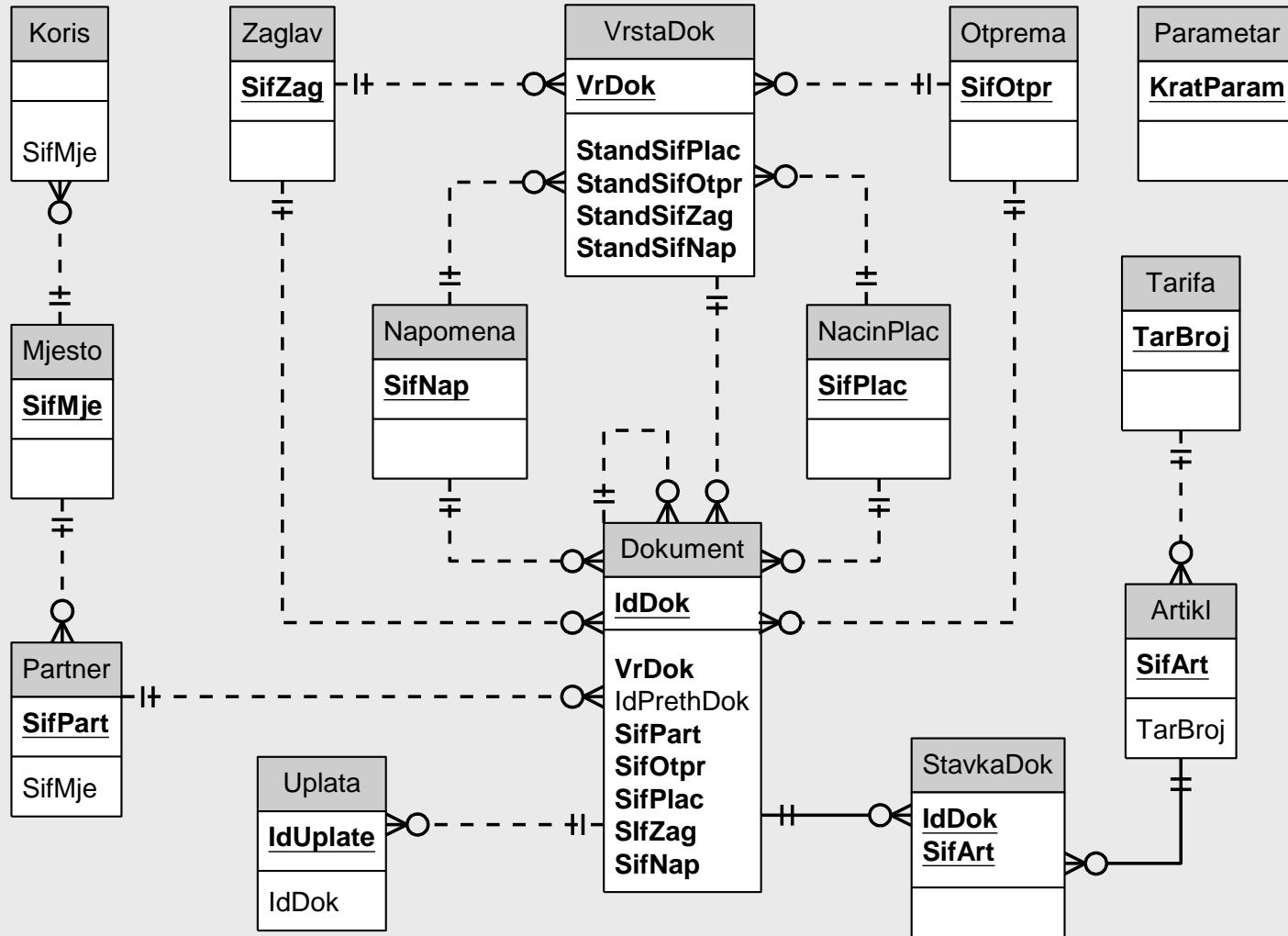
□ Primjer: unos podataka

Sredstvo	Svojstvo	Vrijednost
B2	Sjedala	2
B2	Dolet	18000
A380	Sjedala	853
A380	Dolet	15200

□ 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.plać	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	&Izlazni računi	-1	1	2	2	13	10
U	Ulazni račun	&Ulazni računi	0	-1	1	2	13	10
▶		Šifra zag			Tekst zaglavlja			
		▶ 0						
		13						
		18 Na temelju Vaše narudžbenice zaračunavamo Vam slijedeće:						
		20 Na temelju našeg ugovora o kupoprodaji video kasetu naručujemo:						
		25 Na temelju našeg dogovora o prodaji knjiga, molimo da nam pošaljete:						
		26 Na temelju Vaše pismene narudžbe zaračunavamo Vam slijedeće:						
		27 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 KOMISIONA PRODAJA						
		32 Na temelju Vašeg izvještaja o komisionoj 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 kasete :						

Primjer aplikacije nad meta-modelom

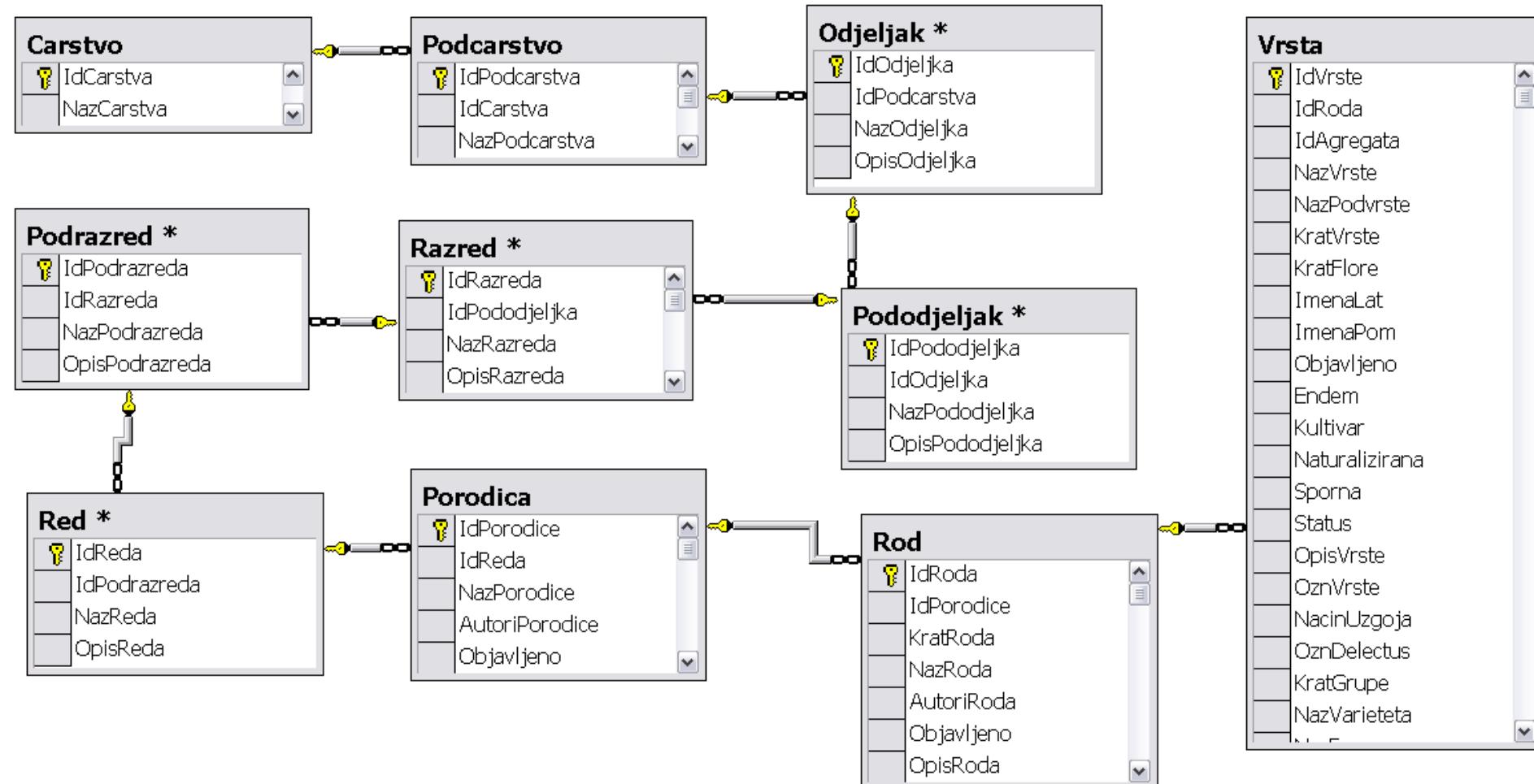
□ Primjer, prilagodba funkcionalnosti

□ Primjer: pojednostavljenje 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
```

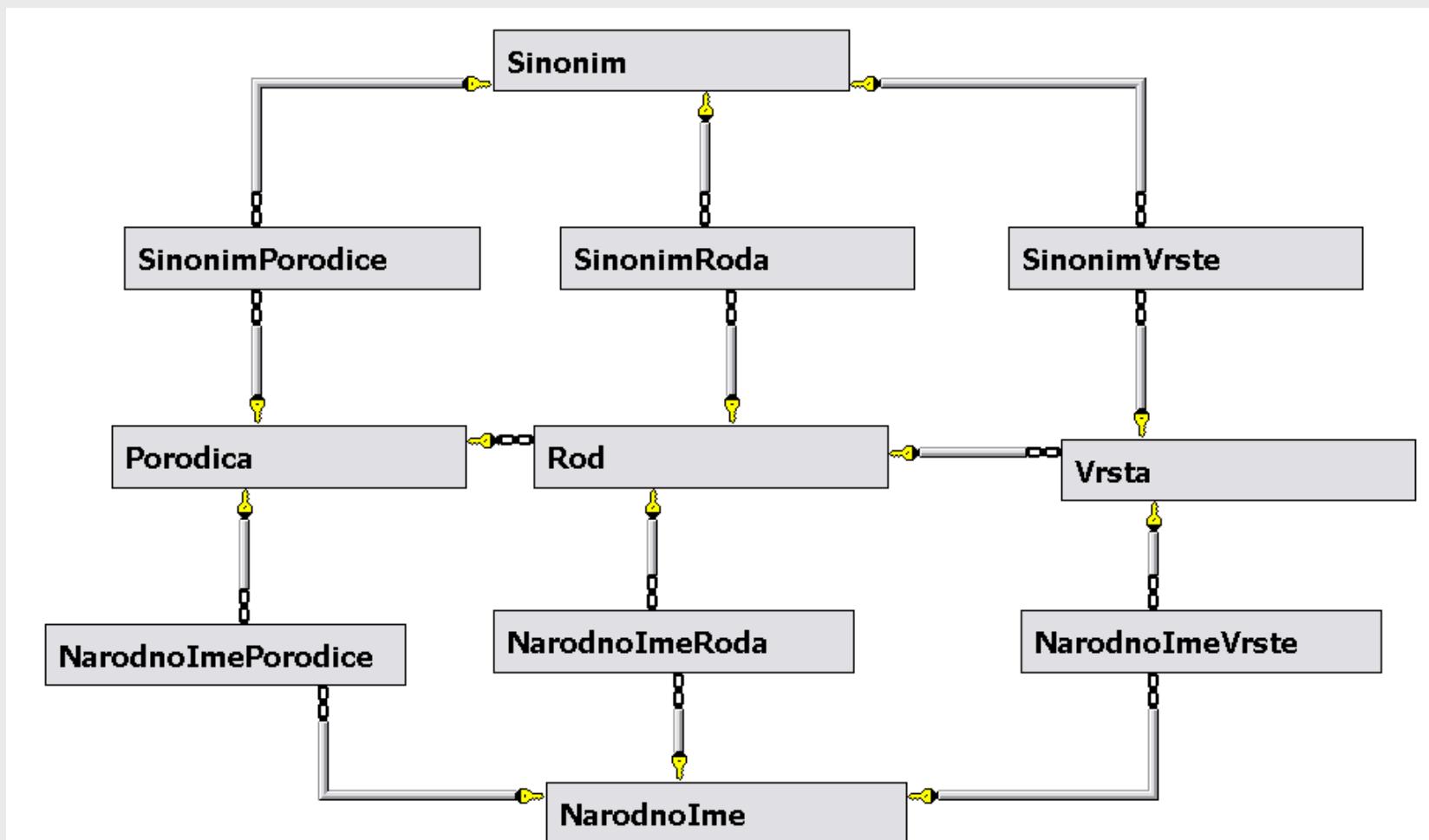
Primjer: kaskada sličnih entiteta

- Svi imaju identifikator i naziv, neki imaju opis i objavljeno ...
- Moguće rješenje: generalizacija (klasa) s 9 specijalizacija



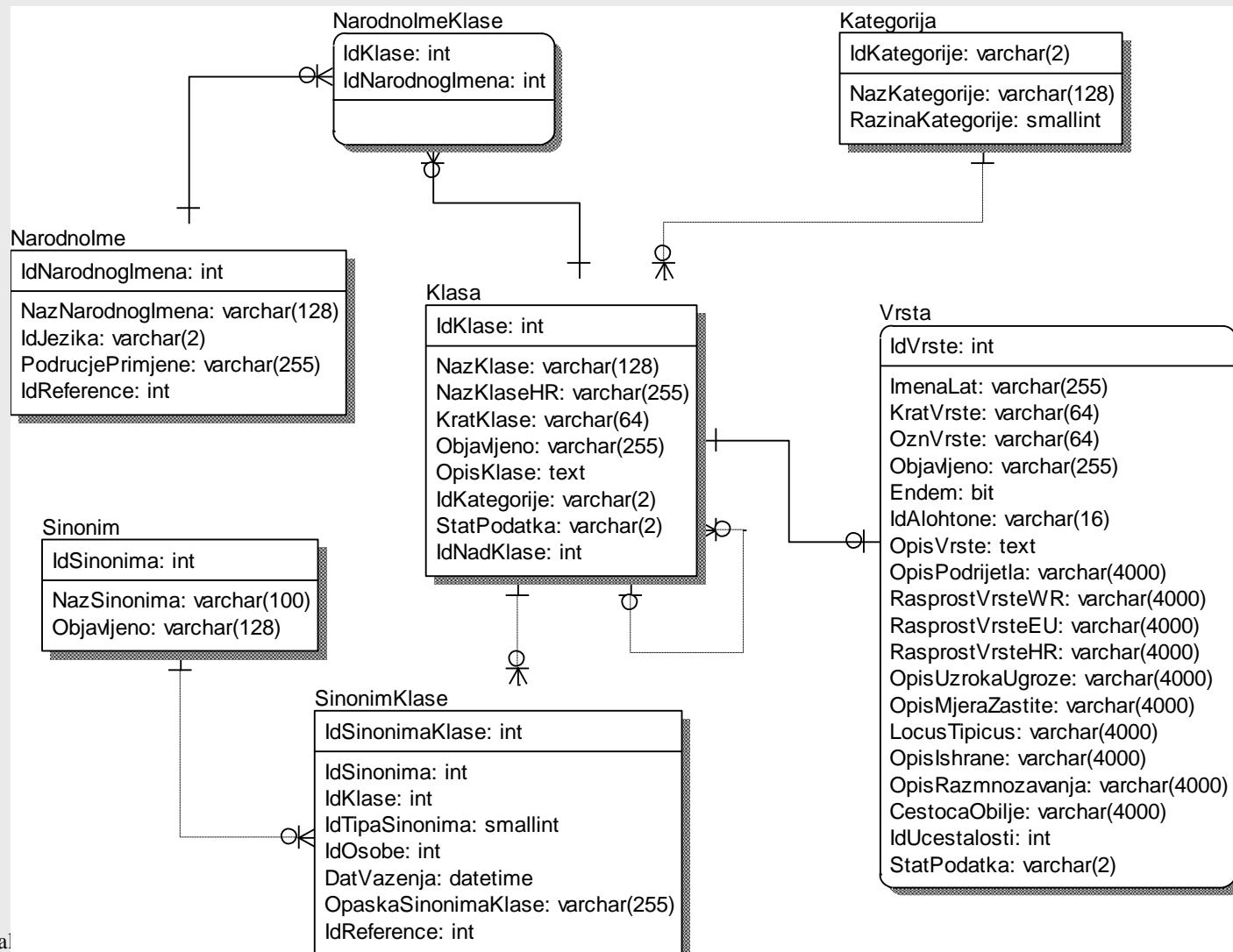
Primjer: kaskada sličnih entiteta (2)

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

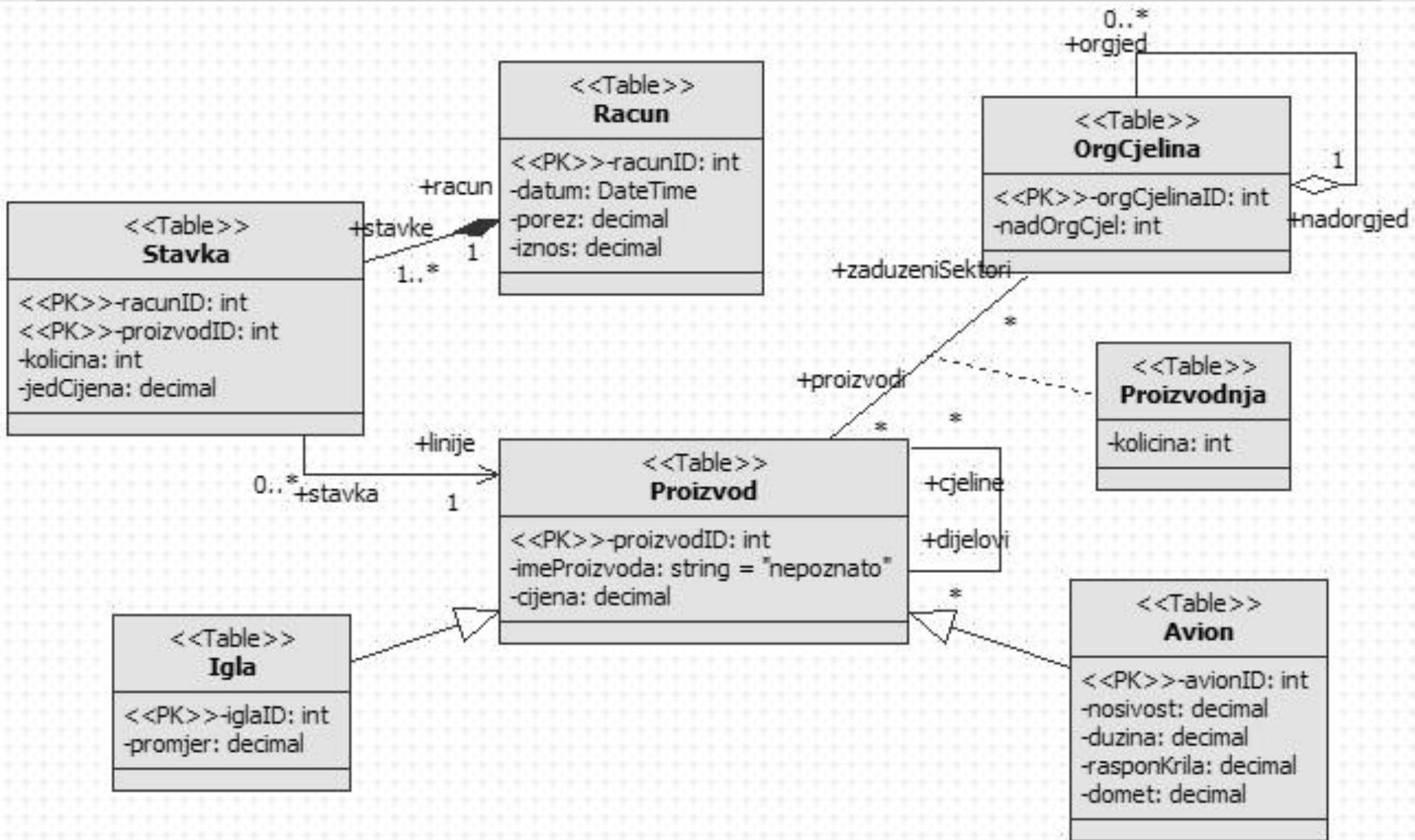


Primjer: kaskada nakon modeliranja

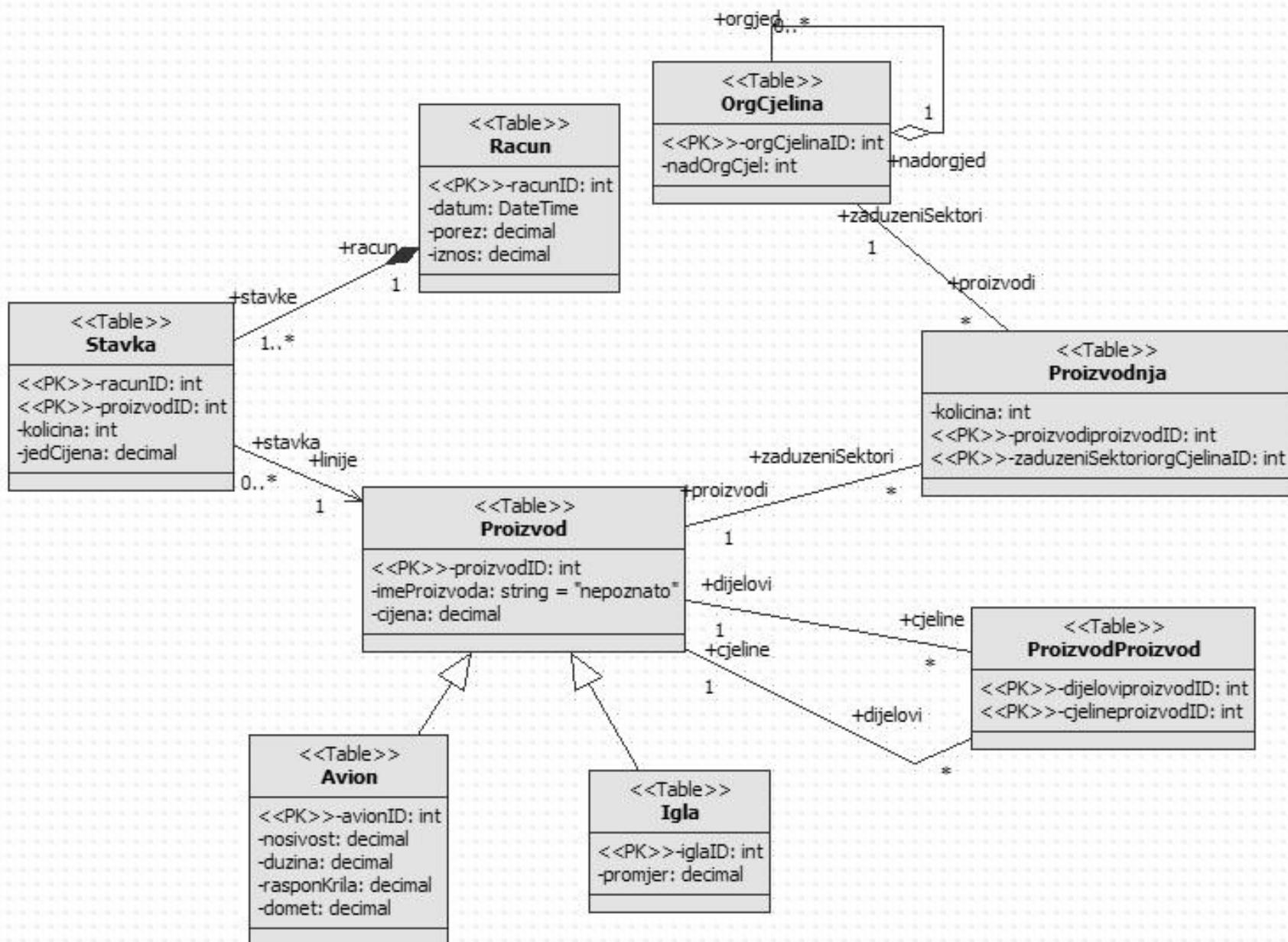
- rekurzivna hijerarhija, izjednačene strukture, jedna specijalizacija te spojni entiteti na narodna imena i sinonime (samo za entitete koji ih stvarno imaju)



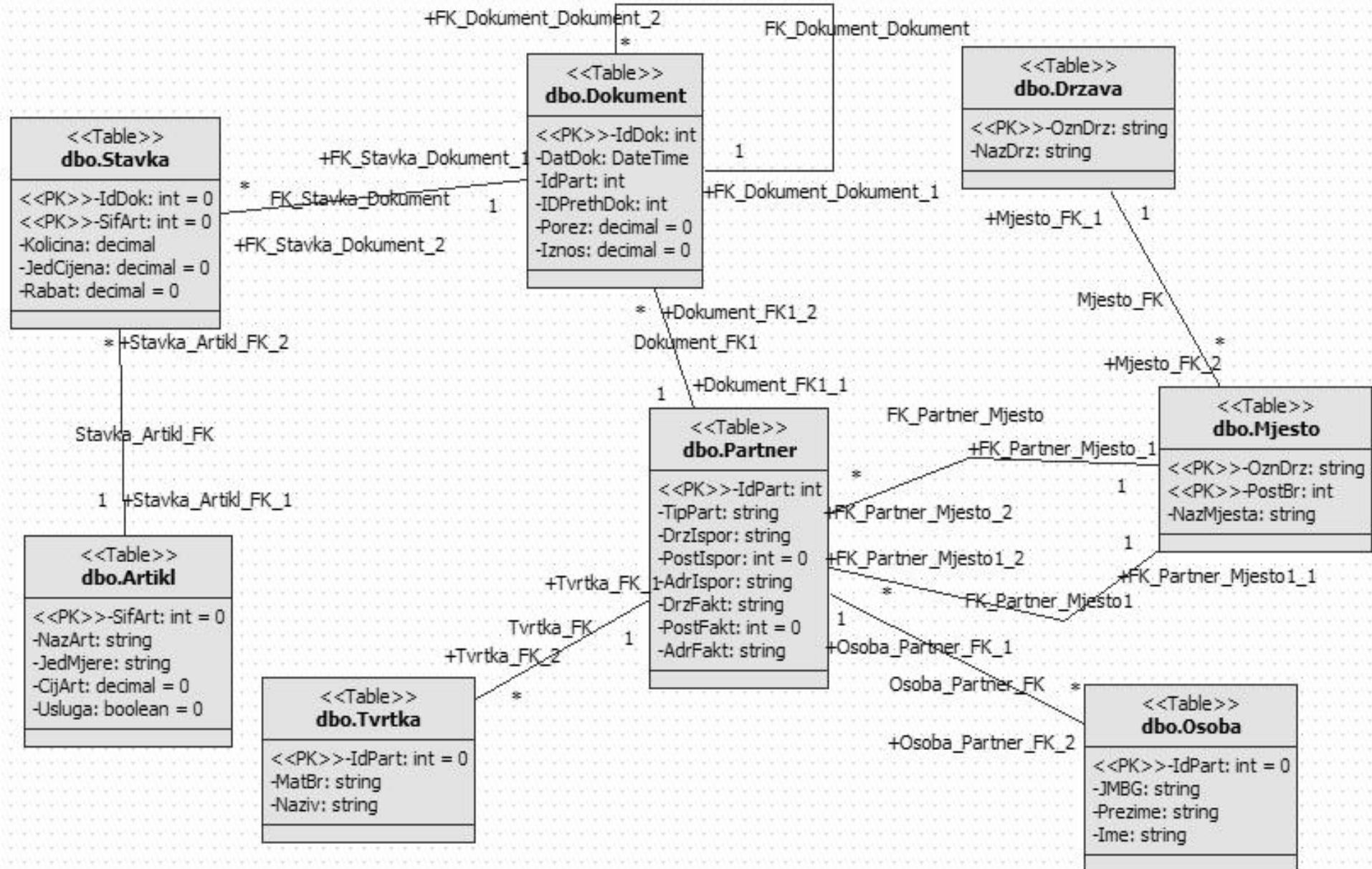
Za sladokusce: konceptualni UML model ...



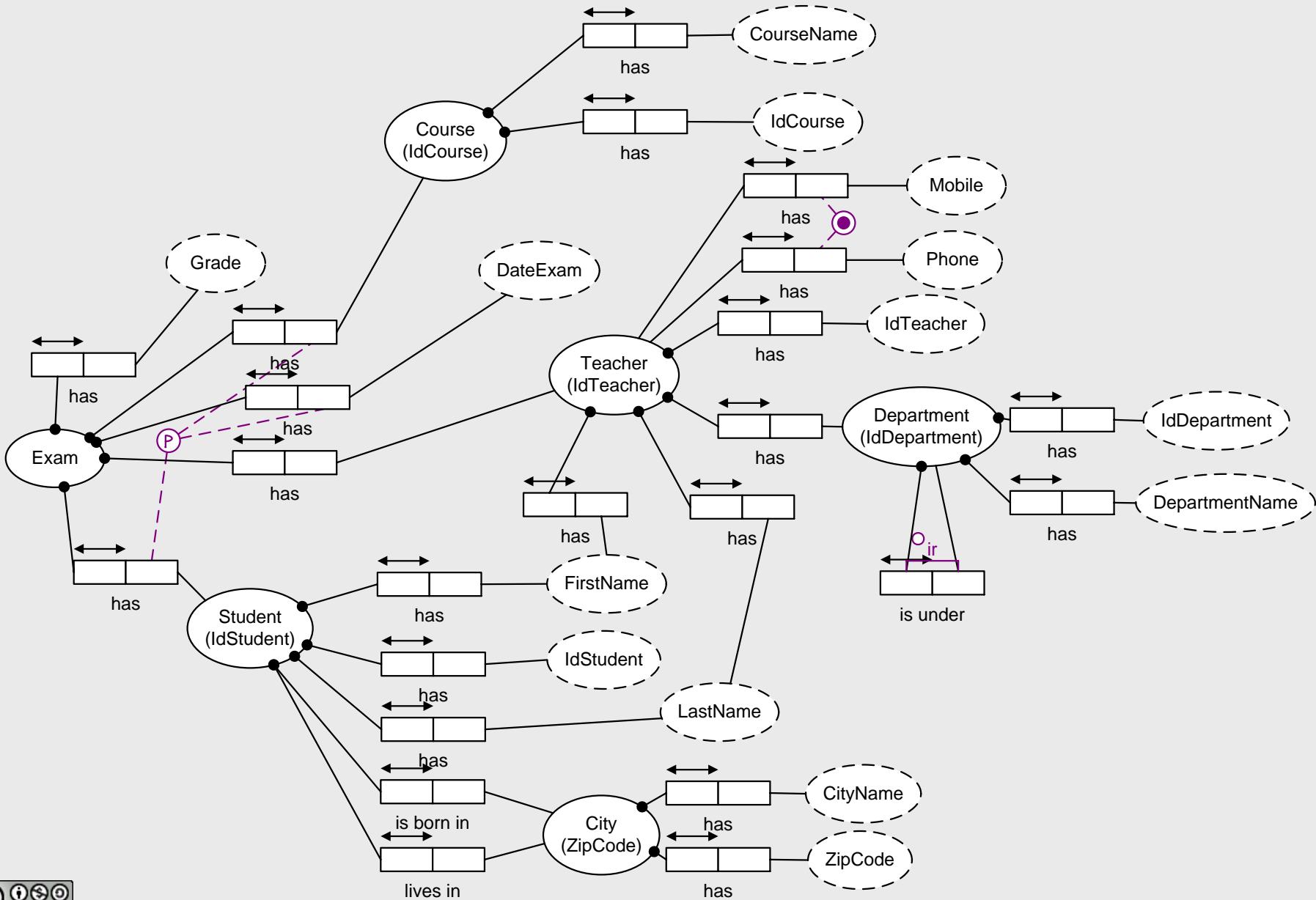
... model nakon transformacije ...



... i klasični relacijski model



Object Role Modeling (ORM)



Primjeri i zadaci



□ Primjer: Resursi\BazePodataka

- Poduzeće – temeljem ER modela s predavanja
- Organizacija – model iz jednog projekta
- FirmaRIS – primjeri aplikacija na narednim predavanjima
- Projekt – podloga aplikacija za naredna predavanjima

□ Dodatak: Dodaci\RIS05-dodatak

- ER notacija, razvoj modela

□ Zadaci za razgibavanje

- ER model i pripadni relacijski model cjenika u restoranu brze prehrane
 - Različite cijene samostalnog proizvoda i proizvoda u paketu
- ER model videoteke (Član, Posudba, Rezervacija, Video)
 - Na posudbu se plaća zakasnina



□ Domaća zadaća - čime crtati ?

- Primjer: Modeliranje\RIS-preporuke

[neke] Reference

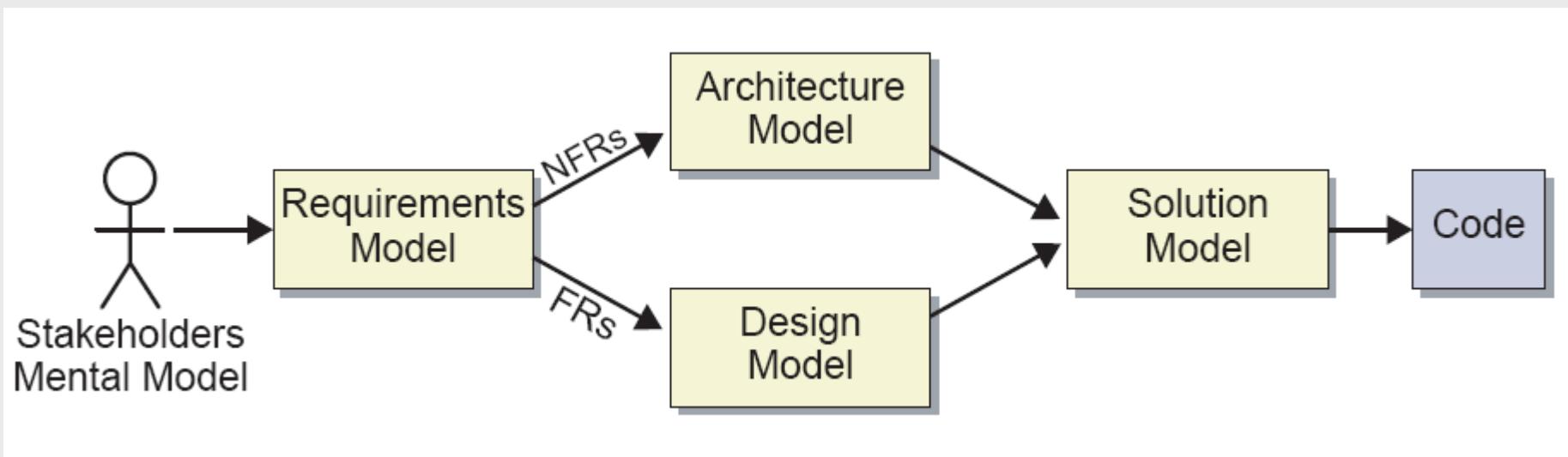
- A. Olive: **Conceptual Modeling of Information Systems**, Springer, 2007.
- A UML Profile for Data Modeling
 - <http://www.agiledata.org/essays/umlDataModelingProfile.html>
- OMG's MetaObject Facility
 - <http://www.omg.org/mof>

Objektno orijentirani razvoj

2013/14.06

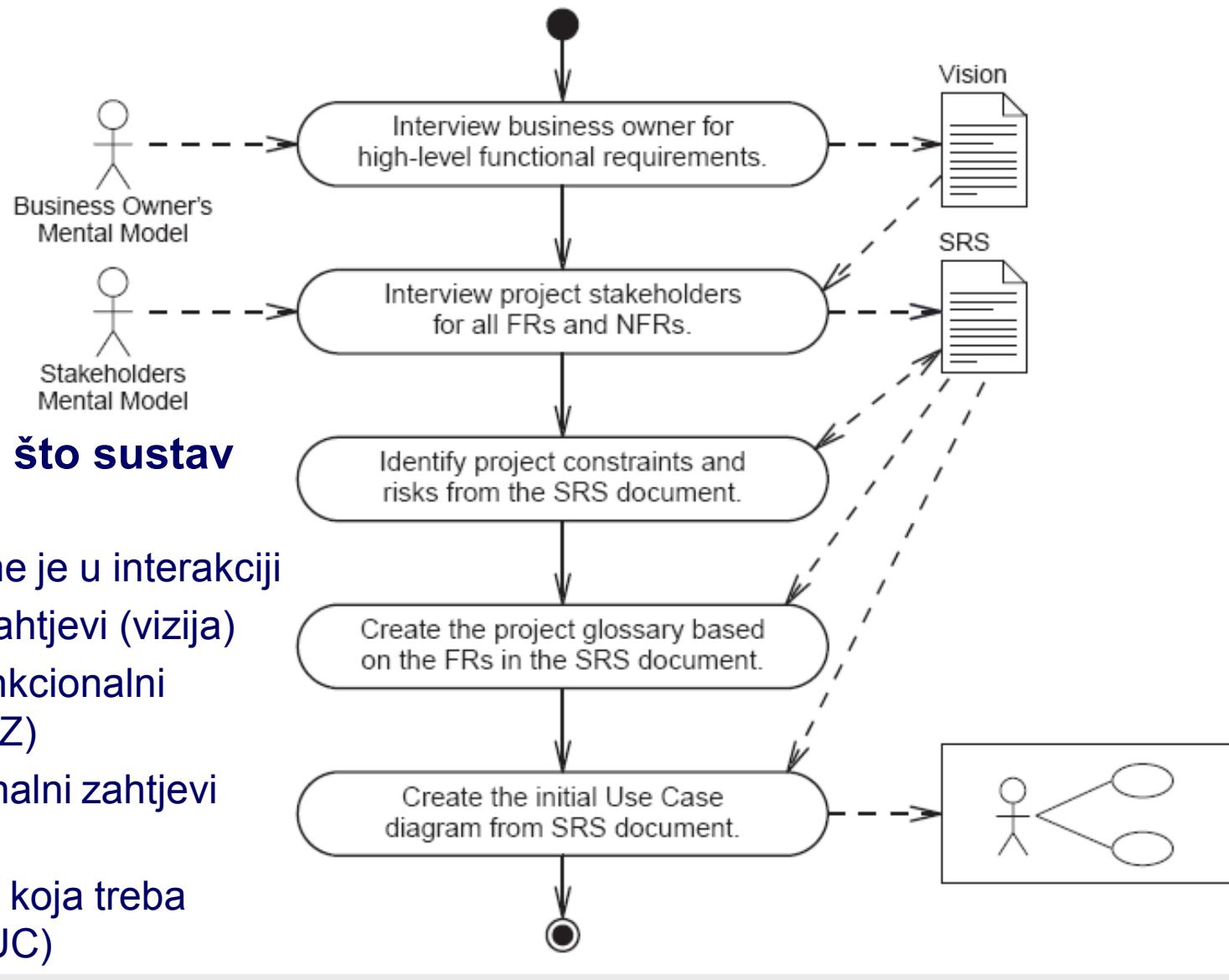
Objektno orijentirani razvoj softvera

- niz slijednih transformacija od modela do koda
- NFRs – programskim okvirima, ponovno iskoristivim knjižicama, ...
- FRs – projektiranjem prema zahtjevima korisnika



© Object-Oriented Analysis and Design Using UML
Sun Microsystems, Inc.

Radni tok prikupljanja zahtjeva



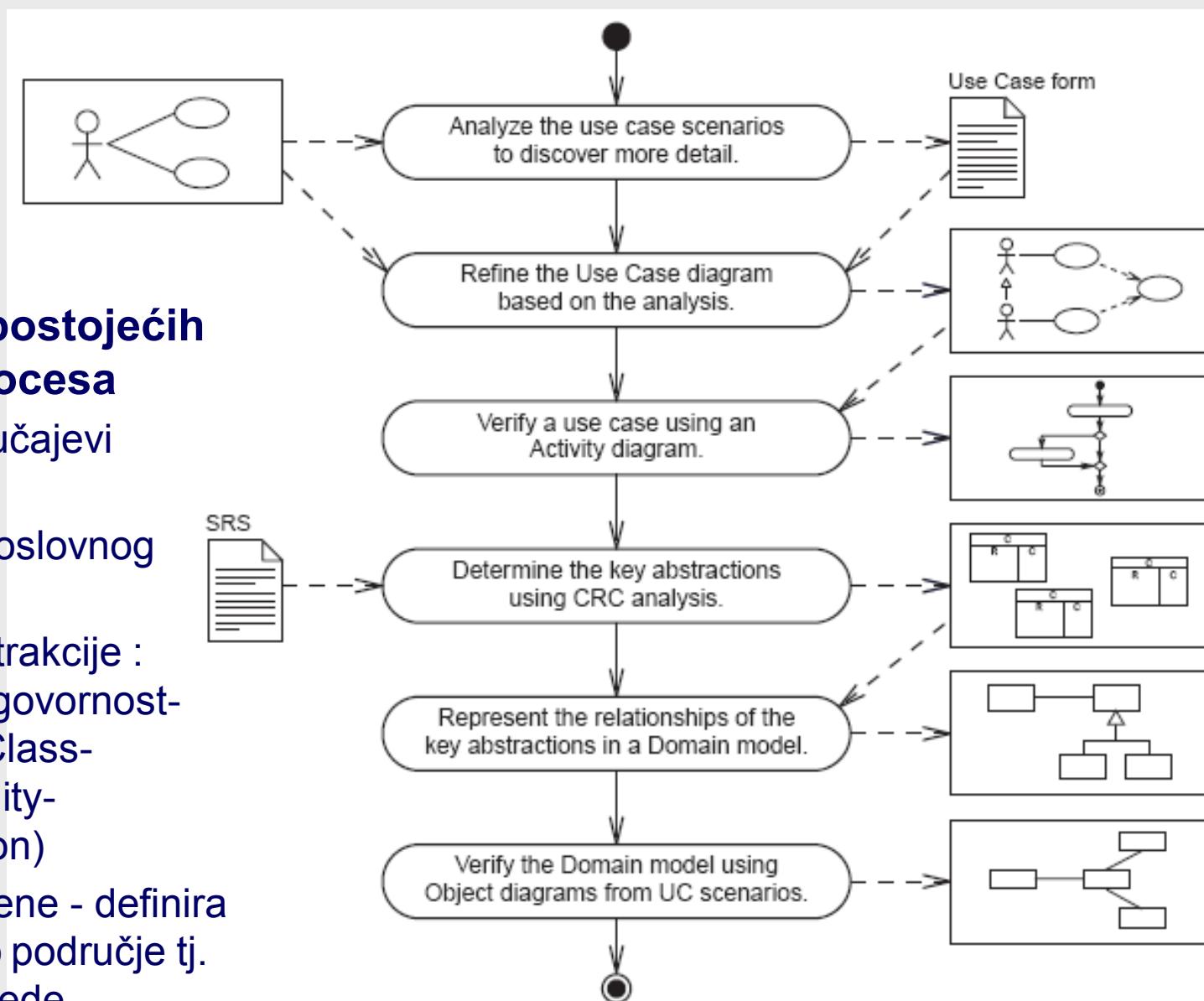
❑ Određivanje što sustav treba činiti

- s kime/čime je u interakciji
- poslovni zahtjevi (vizija)
- detaljni funkcionalni zahtjevi (FZ)
- nefunkcionalni zahtjevi (NZ)
- ponašanja koja treba podržati (UC)

Radni tok analize zahtjeva (→ modeliranje funkcija)

□ Modeliranje postojećih poslovnih procesa

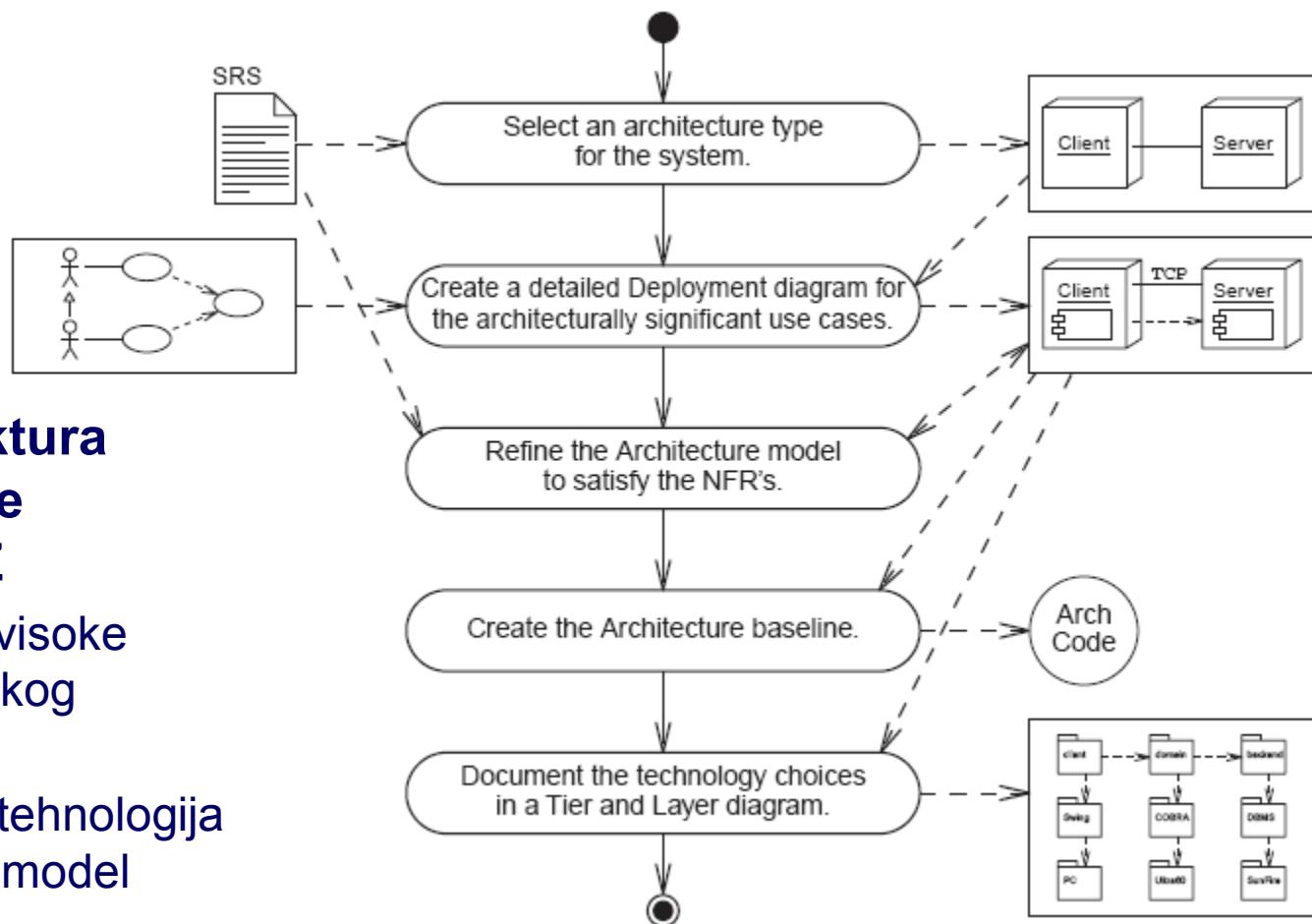
- rafinirani slučajevi korištenja
- aktivnosti poslovnog procesa
- ključne apstrakcije : Razred-Odgovornost-Suradnja (Class-Responsibility-Collaboration)
- model domene - definira problemsko područje tj. ključne razrede



Radni tok modeliranja arhitekture

□ Modeliranje struktura visoke razine koje zadovoljavaju NZ

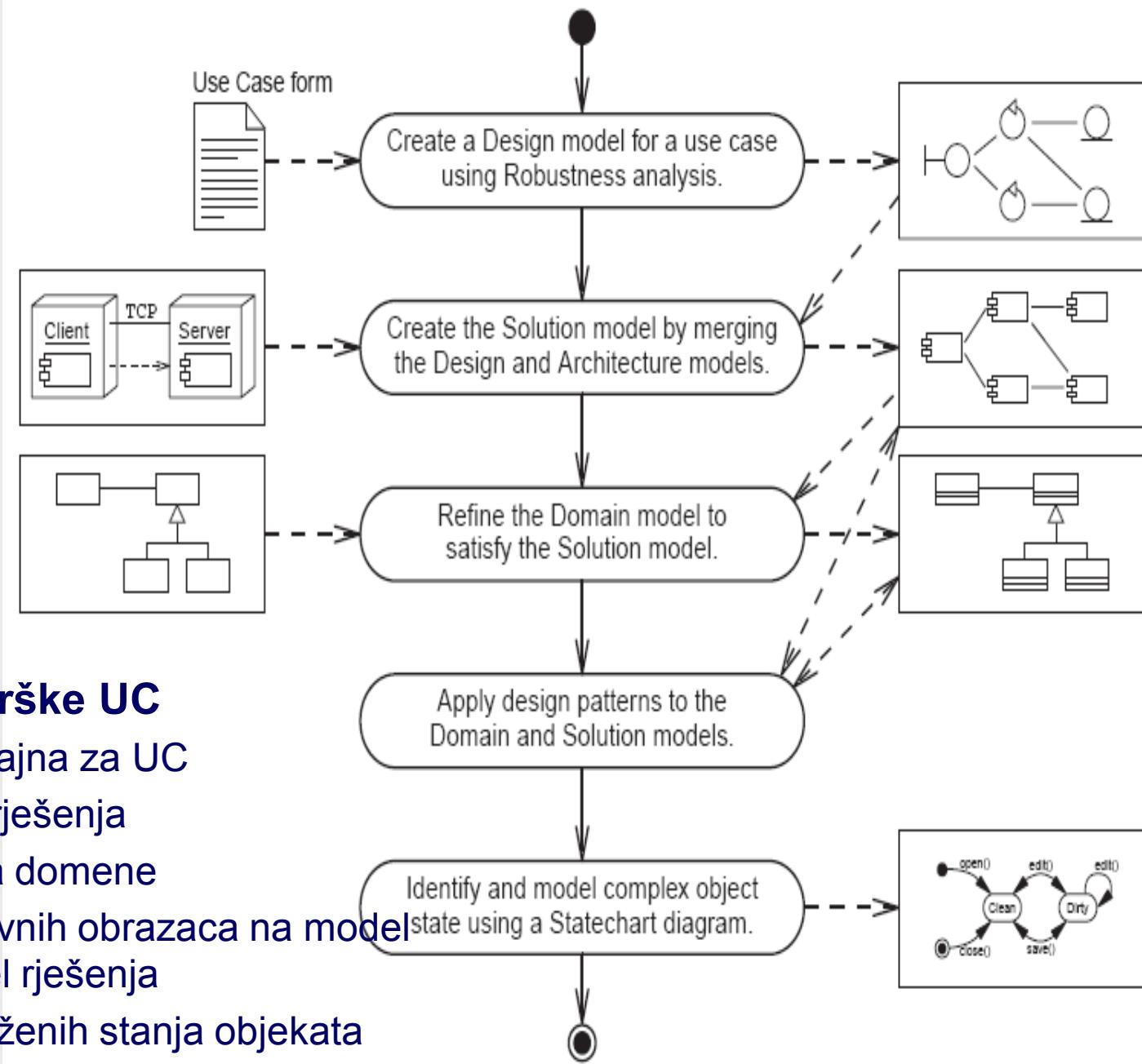
- razvoj struktura visoke razine programskog rješenja
- ustanavljanje tehnologija koje podržavaju model arhitekture
- pojašnjenje modela arhitekture obrascima arhitekture koji zadovoljavaju NZ



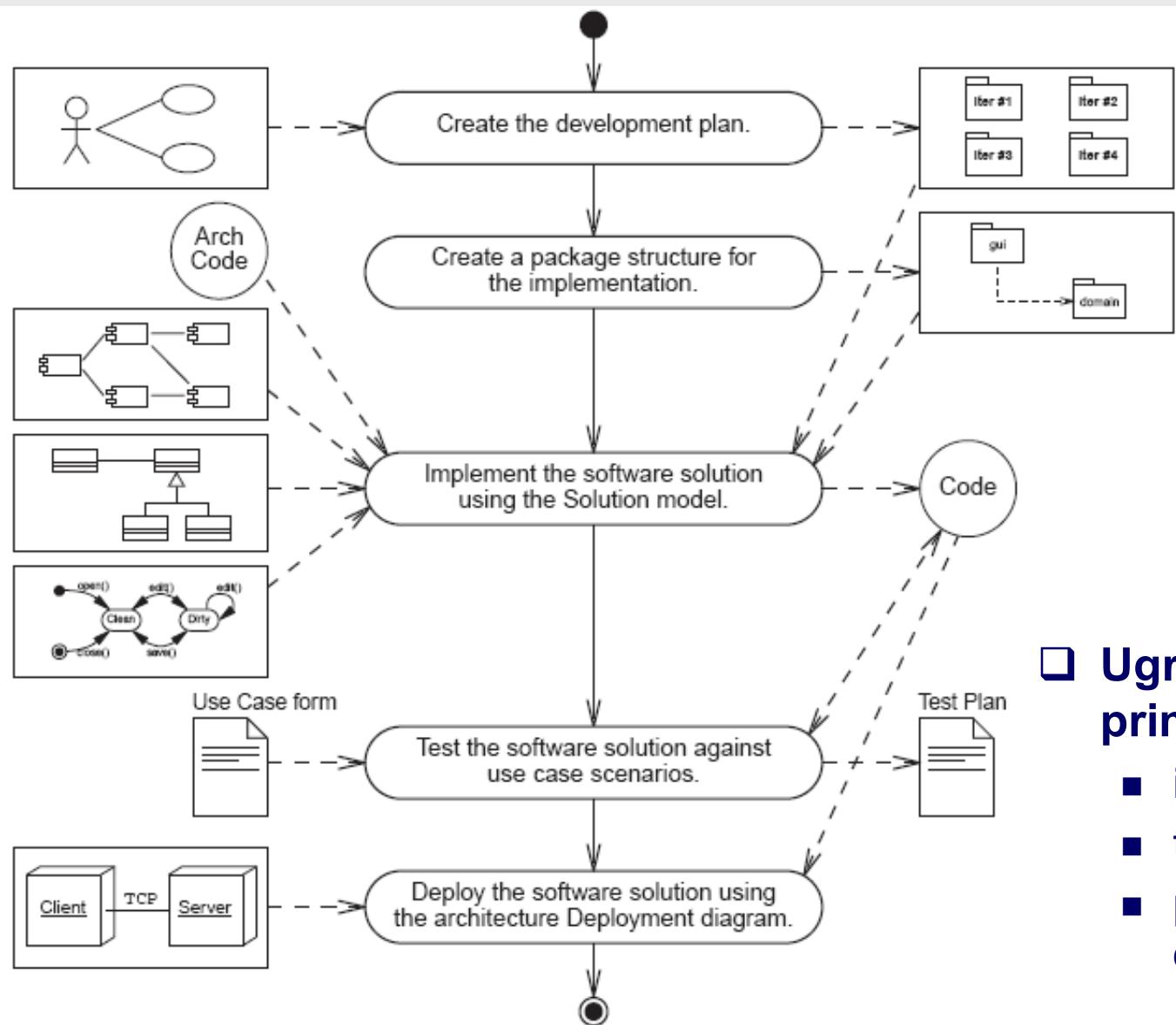
Radni tok dizajna

□ Modeliranje podrške UC

- modeliranje dizajna za UC
- izrada modela rješenja
- razrada modela domene
- primjena oblikovnih obrazaca na model domene i model rješenja
- modeliranje složenih stanja objekata



Radni tok konstrukcije



❑ Ugradnja, provjera i primjena sustava

- implementacija rješenja
- testiranje
- primjena u radnom okruženju

Dijagrami aktivnosti

Dijagram aktivnosti

□ Prikazuje

- ili tok rada (workflow) između objekata nekog slučaja korištenja
- ili tok upravljanja za neku operaciju

□ Osnovni simbol: aktivnost ili akcija

- akcija – jednostavna, nedjeljiva obrada
- aktivnost – skup akcija

□ Sadrži tok aktivnosti, ali ne i objekte

- sva (ili većina) stanja su stanja akcija
- sva (ili većina) prijelaza obavlja se po dovršetku akcija u stanju

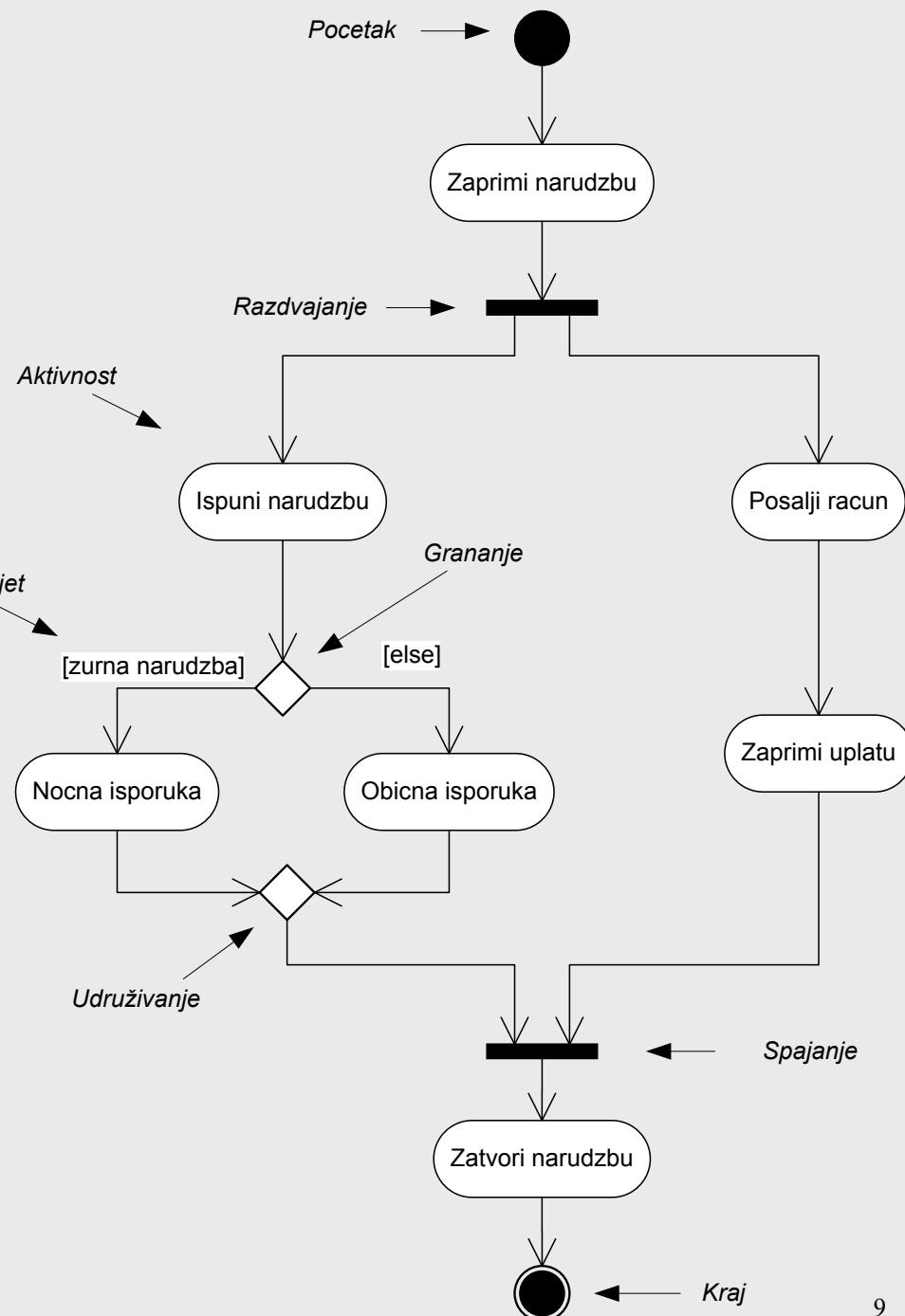
□ Odluka - uvjetovano ponašanje

- grananja (branches)
- udruživanja (merges)

□ Paralelizam, istovremenost procesa

- razdvajanje, račvanje (fork)
- spajanje, pripojenje (join)
- grana predstavlja nit izvođenja (thread)

□ Uvjet grananja (guard condition)



Paralelizam i uvjetne niti

□ Oznake za razdvajanje i spajanje moraju se poklapati

- za svako razdvajanje mora postojati spajanje niti nastalih razdvajanjem

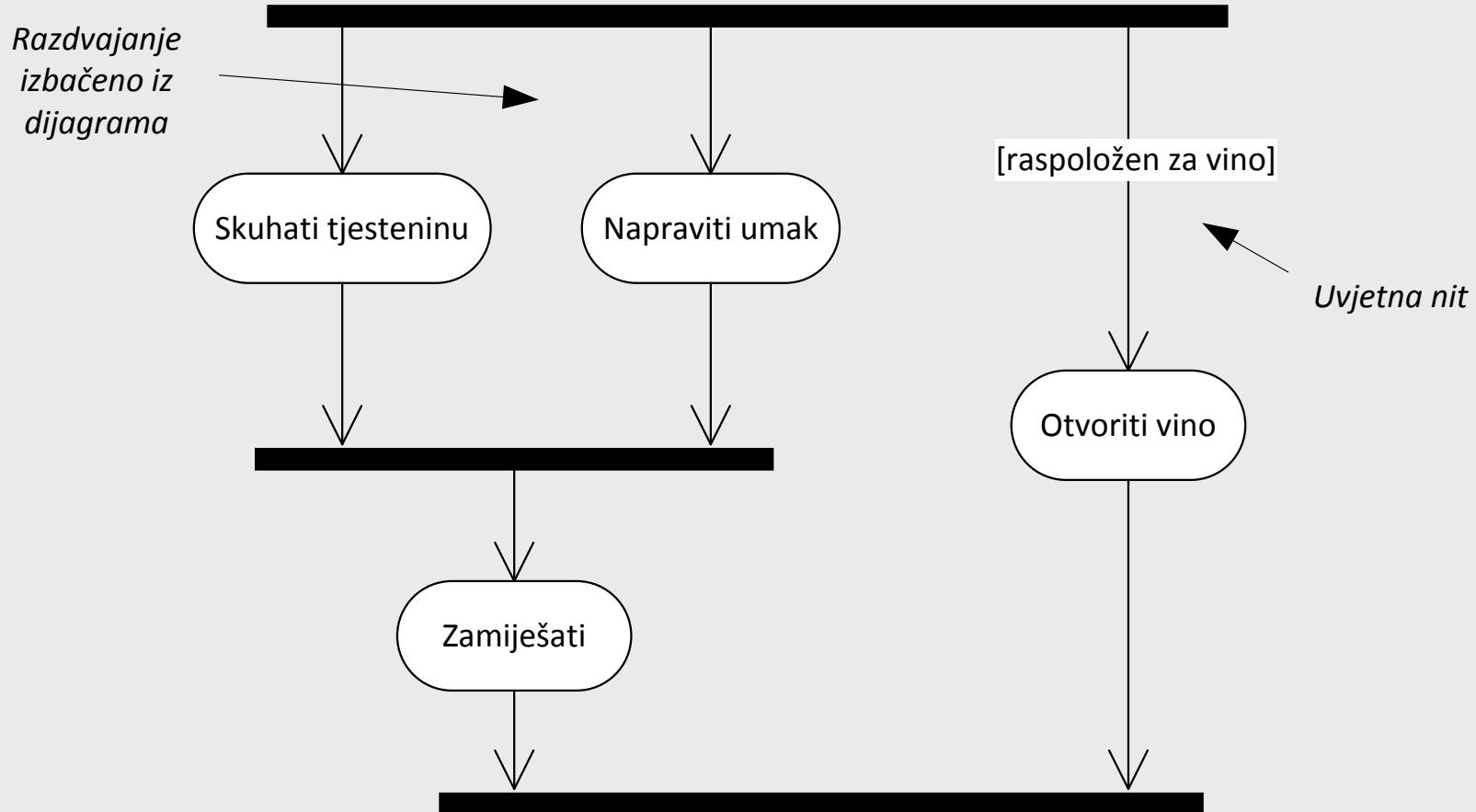
□ Dodatna pravila (sljedeći slajd)

- razdvojena nit može se i sama razdvojiti, npr. 1:2,3 zatim 3:4,5
- novonastale niti spajaju se obrnutim redom, npr. spoj 4,5 pa 2,3
- ako nit koja izlazi iz razdvajanja ide direktno u novo razdvajanje, dozvoljeno je izostaviti drugo razdvajanje a niti premjestiti da izlaze iz prvog
- analogno, mogu se udružiti neposredna slijedna spajanja
- postoji posebno stanje, tzv. sinkronizacijsko stanje (sync state) koje omogućava sinkronizaciju na mjestima na kojima bi pravilo o parovima razdvajanja i spajanja inače sprječavalo napredak

□ Uvjetna nit (conditional thread)

- ako uvjet u niti koja izlazi iz oznake razdvajanja nije ispunjen, smatra se da je ta nit završila

Primjer skraćenog prikaza i uvjetnih niti



Razlaganje aktivnosti

□ Dekomponiranje aktivnosti

- složena aktivnost - u podaktivnosti

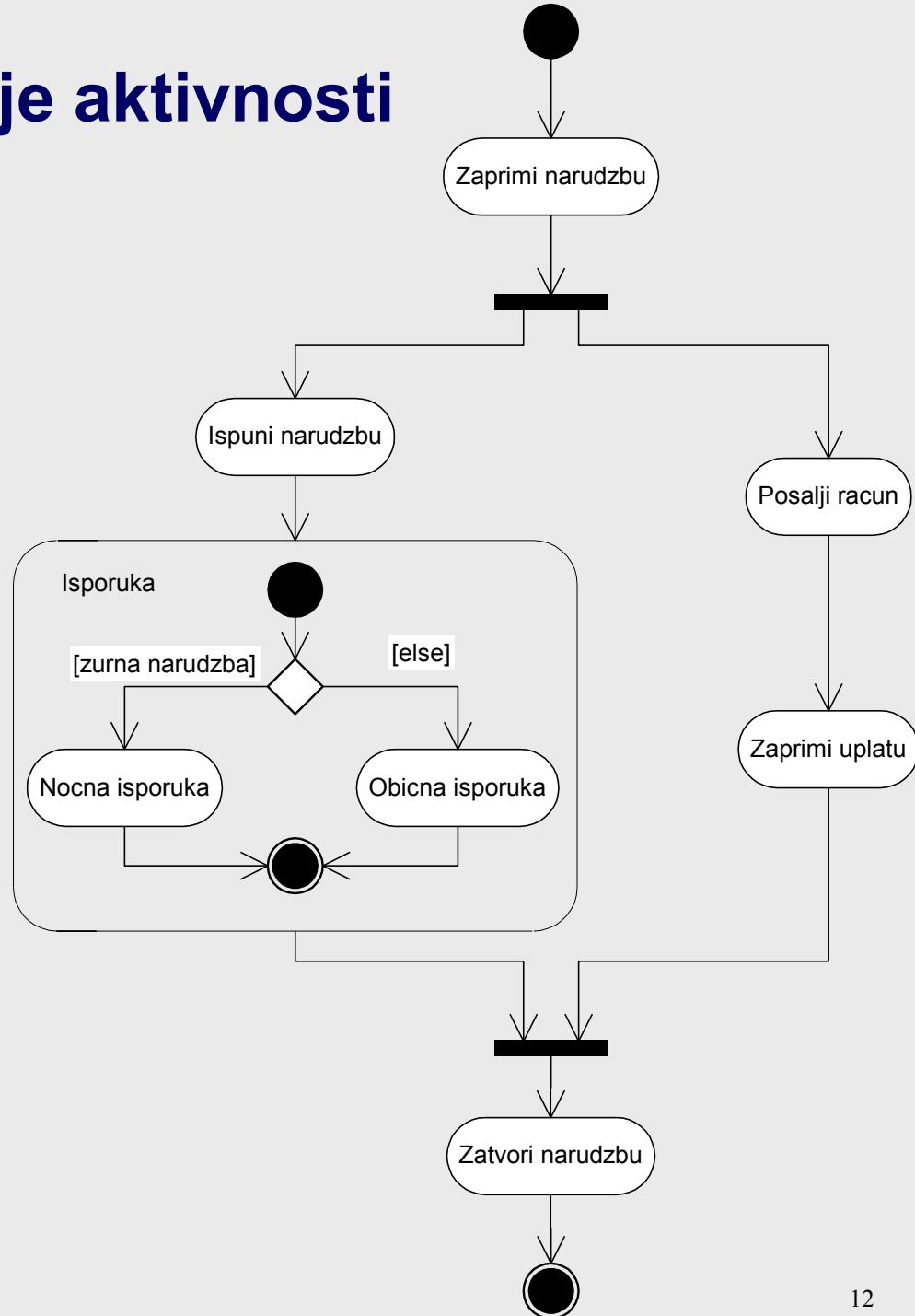
□ Alternative prikaza složene aktivnosti

- prikaz s podaktivnostima (na slici)
- prikaz bez podaktivnosti (vidjela bi se samo aktivnost Isporuka)

□ Unutar razložene aktivnosti mogu se prikazati točke početka i kraja

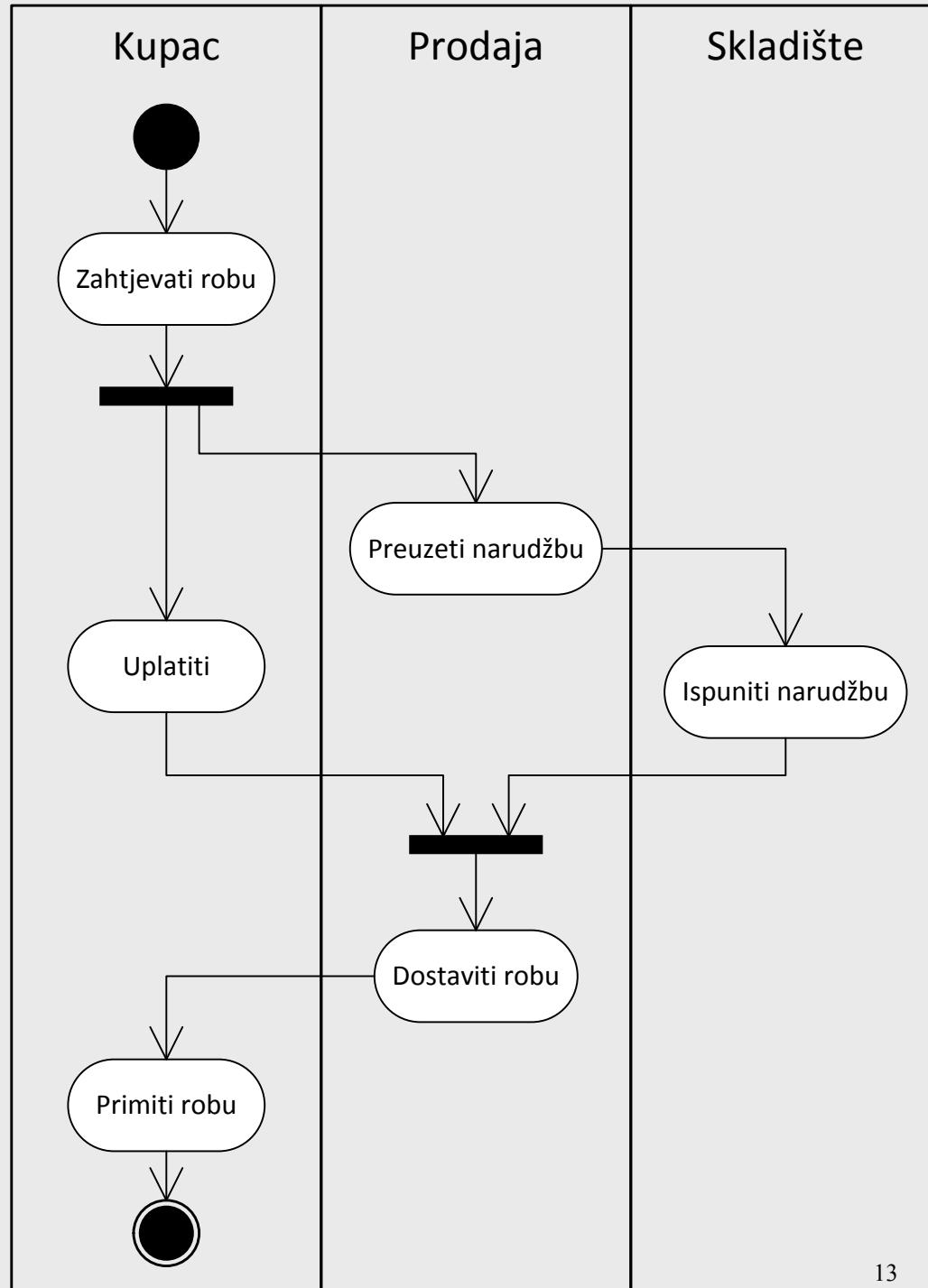
- ako se ne prikazuju tada prijelazi između stanja aktivnosti idu direktno u dijagram, odnosno iz poddijagrama.

□ Zašto je to važno ?



Plivaće staze

- Osnovni DA pokazuje koje se aktivnosti obavljaju i kojim redoslijedom ali ne i tko što radi
- Staze (swimlanes)
 - staze predstavljaju **odgovornost pojedinog razreda ili sudionika**-osobe, a najčešće su to organizacijske jedinice
- Svaka aktivnost pridružena je jednoj stazi
 - tokovi mogu prelaziti kroz više staza



Primjena dijagrama aktivnosti

□ Analiza slučaja korištenja

- DA dobro pokazuje slijed akcija **za nekoliko objekata i nekoliko UC**
- za prikaz ponašanja objekata za jedan UC bolji je dijagram interakcija
- na konceptualnoj razini nije ključno dodjeljivanje aktivnosti objektima
 - dobiti opću sliku ponašanja sustava, te odrediti međuzavisnosti u ponašanju – objekti se vezuju kasnije

□ Razumijevanje radnog, poslovnog procesa

- modeliranje radi analize procesa (mogućeg poboljšanja)
- potrebno uključivanje korisnika poznaje poslovni proces

□ Opisivanje složenih slijednih algoritama (obrade podataka)

- u ovom slučaju DA samo zamjena za dijagrama toka (tzv. blok dijagram)

□ Opisivanje višenitnih aplikacija

Modeliranje funkcija

Functional modeling

Modeliranje funkcija

□ Modeliranje funkcija (functional modeling)

- Opisuje poslovne procese i interakciju IS s okolinom
- Dokumentiranje zahtjeva i vanjskog ponašanja sustava

□ Logički modeli - problemskog područja (problem domain models)

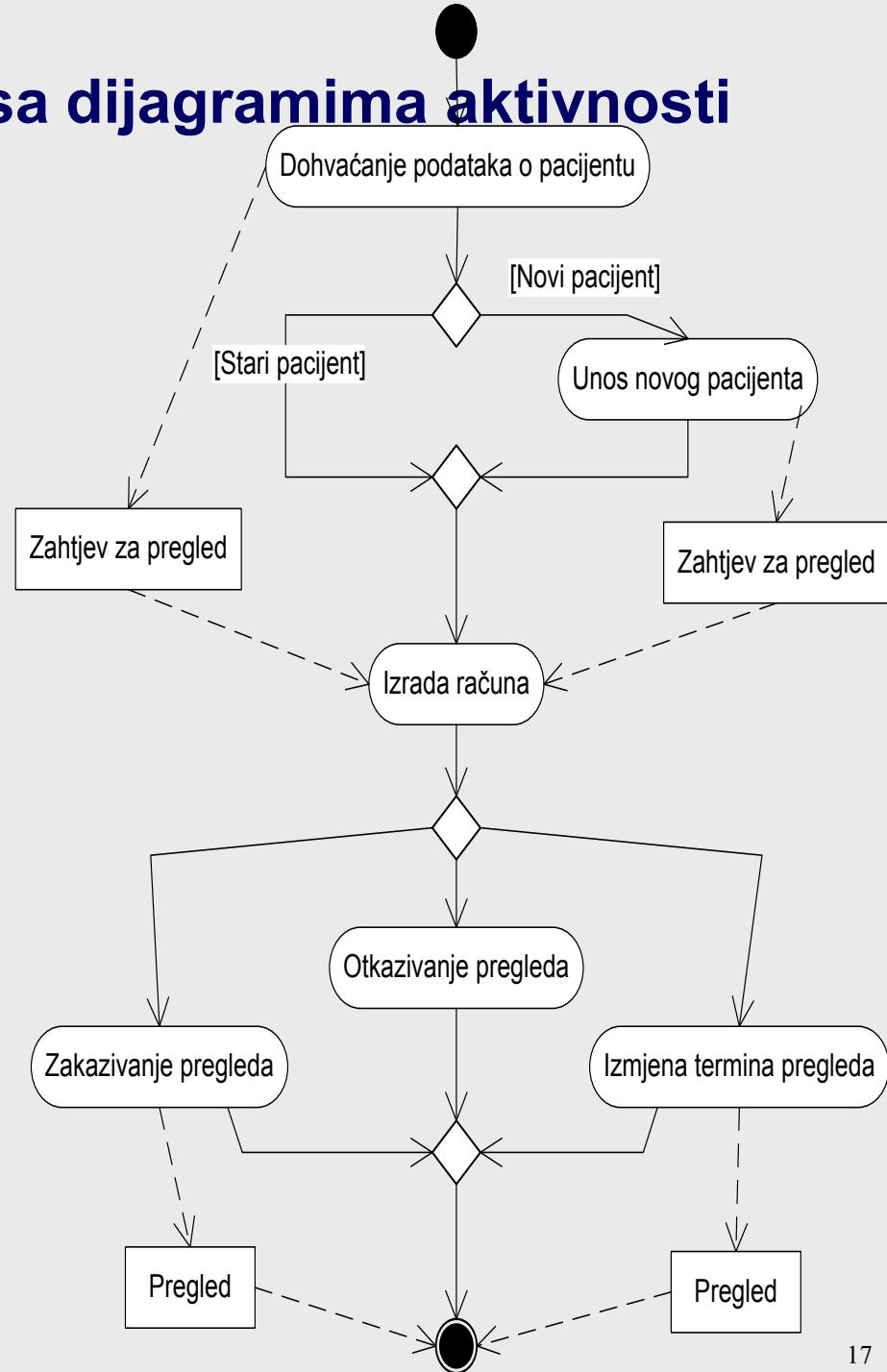
- opis aktivnosti bez detalja provedbe
- ne mora se znati izvode li se računalom ili ručno !
- **Dijagram aktivnosti**
 - logičko modeliranje poslovnih procesa i tokova rada
 - prikaz procesa ili aktivnosti te kolanja podataka
- **Slučaj korištenja**
 - prikaz aktivnosti korisnika
 - vanjski ili funkcionalni pogled na IS - interakcija sustava i okoline

□ Fizički modeli

- temeljem logičkih modela za vrijeme faze oblikovanja
- informacije potrebne za izgradnju sustava

Modeliranje poslovnih procesa dijagramima aktivnosti

- "sofisticirani dijagram toka podataka"
 - dodatni elementi AD
- Objekt (object node)
 - razred objekata povezan s tokom
- Upravljački tok (control flow)
 - puna strelica
 - slijed izvršenja
 - pridruživanje samo akcijama ili aktivnostima
- Podatkovni tok (object flow)
 - protok objekata između aktivnosti
 - isprekidana strelica
 - povezuje objekt i aktivnost



Slučajevi korištenja

Slučajevi korištenja

- **Analiza slučajeva korištenja (Use Case Analysis)**
 - tehnika analize poslovnih procesa iz perspektive korisnika
- **Use case = Slučaj korištenja (primjene)**
 - skup scenarija sa zajedničkim ciljem korištenja sustava
- **Scenarij**
 - niz koraka koji opisuju interakciju korisnik-sustav
- **Tipični sadržaj**
 - kako kreće i završava slučaj
 - normalni tok događaja
 - varijabilni (alternativni) ili iznimni tok događaja
- **Standard izrade UC-a ne postoji (nije dio UMLa) !**
 - alternative i iznimke mogu biti zasebni slučajevi korištenja

Primjer scenarija

Use Case: Posudba

Tipični tok

Aktivnosti sudionika	Odziv sustava
1. Član pristupa pultu s medijima koje želi posuditi.	
2. Član daje identifikacijsku karticu Djelatniku, koji u sustavu provjerava valjanost.	3. Prikaz informacije o članu i status prethodnih posudbi ako postoje, uključujući kašnjenja.
4. Za svaki primjerak Djelatnik unosi stavku evidencije o posudbi.	5. Prikaz liste posuđenih medija po datumima te iznos naknade i zakasnine.
6. Djelatnik informira člana o ukupnom iznosu za plaćanje.	
7. Član plaća gotovinom ili kreditnom karticom.	
8. Djelatnik evidentira plaćanje.	9. Za plaćanje karticom obavlja se autorizacija. 10. Ispis liste posudbe i računa.
11. Djelatnik izdaje račun i listu posudbe.	

Alternativni tok

7a. Član nema dovoljno gotovine. Zahtijeva se plaćanje karticom, opoziva prethodna transakcija ili smanjuje broj primjeraka za posudbu.

7b. Postoji neplaćena zakasnina. Član ju mora platiti prije posudbe ili se transakcija prekida.

9. Autorizacija kartice ne uspijeva (usluga nedostupna ili blokirana kartica). Plaćanje gotovinom.

Modeliranje funkcija slučajevima korištenja

- UC opisuju funkcije sustava iz perspektive korisnika
 - "Što sustav treba napraviti?"
- UC opisuje samo jednu funkciju interakcije korisnika i sustava
- UC može uključivati različite scenarije
 - Npr. scenariji za isti UC: pretraživanje knjiga po autoru, po naslovu, ...
- Svaki UC smije biti povezan s jednom i samo jednom ulogom
 - Npr. liječnička "sestra" dogovara pregledе, ispisuje uputnice i recepte, naplaćuje participaciju ...
- Različiti fizički korisnici mogu biti u istoj ulozi
 - stoga se slučajevi korištenja povezuju s ulogama a ne korisnicima

Tipovi slučajeva korištenja

□ Prema svrsi, namjeni:

- pregledni (overview) – usaglašavanje zahtjeva s korisnikom
 - na početku prikupljanja zahtjeva, sadrže samo osnovne informacije
 - identifikator, primarna uloga, tip, kratki opis
- detaljni (essential) – sve informacije potrebne za razumijevanje

□ Prema količini informacija koje sadrži:

- osnovni (essential) – analitički, implementacijski nezavisni
- stvarni (real) – opis specifičnih koraka
 - dovoljno detaljan da odredi ponašanje sustava kada bude ugrađen

Primjer opisa slučaja korištenja 1/3

Naziv slučaja korištenja: Zakazati pregled

ID: 2

Prioritet: Visok

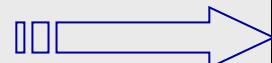
Glavni sudionik: Pacijent

Tip slučaja korištenja: Detaljni, osnovni

Sudionici i interesenti:

Pacijent – želi zakazati, promijeniti ili otkazati pregled

Liječnik – želi obraditi pacijenta u prihvatljivom vremenu



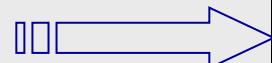
Kratki opis: slučaj korištenja opisuje kako zakazati, promijeniti ili otkazati pregled.

Okidač: Pacijent pozivom zakazuje novi pregled te mijenja ili otkazuje postojeći pregled.

Tip: Vanjski

Veze:

Asocijacija (association): Pacijent



Uključivanje (include): Izrada računa

Proširenje (extend): Unos novog pacijenta

Generalizacija (generalization):

Tok događaja:

1. Pacijent kontaktira ordinaciju radi progleda

Elementi opisa slučaja korištenja

- Pregled (overview) - osnovni podaci o slučaju korištenja
 - naziv, tip, sudionici, kratak opis, prioritet
 - glavni, primarni sudionik – pokreće scenarij
 - okidač – događaj pokretanja (vanjski, unutarnji, vremenski ...)

- Veze - odnos s drugim slučajevima korištenja
 - Asocijacija (association) – veza između slučaja korištenja i sudionika
 - Proširenje (<<extend>>) – proširenje funkcionalnosti uz dodatna pravila
 - npr. *Unos novog pacijenta*, izvodi se uvjetno, nije dio normalnog toka
 - Uključivanje (<<include>>) – uključivanje drugog UC, prije <<uses>>
 - omogućuje funkcionalnu dekompoziciju (slično DTP)
 - npr. *Izrada računa* smatra se dovoljno složenom da bude samostalan UC
 - Generalizacija (generalization) – poopćenje i nasljeđivanje UC
 - npr. *Zakazivanje pregleda starom pacijentu* i *Zakazivanje pregleda novom pacijentu* imaju generalizirano ponašanje u *Zakazivanje pregleda*

Primjer opisa slučaja korištenja 2/3

Uključivanje (include): Izrada racuna

Proširenje (extend): Unos novog pacijenta

Generalizacija (generalization):

Tok događaja:

1. Pacijent kontaktira ordinaciju radi pregleda.
2. Pacijent službeniku daje svoje podatke.
3. Službenik potvrđuje da pacijent postoji u bazi pacijenata.
4. Službenik izvršava slučaj korištenja "Izrada računa".
5. Službenik pita pacijenta želi li zakazati novi pregled, promijeniti ili otkazati postojeći pregled.

Ako pacijent želi zakazati novi pregled,
izvodi se podtok T-1: novi pregled.

Ako pacijent želi otkazati postojeći pregled,
izvodi se podtok T-2: otkaži pregled.

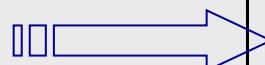
Ako pacijent želi promijeniti postojeći pregled,
izvodi se podtok T-3: promjeni pregled.

6. Službenik obavještava pacijenta o rezultatu transakcije.

Podtokovi:

T-1: Novi pregled

1. Službenik pita pacijenta za moguće termine pregleda.
2. Službenik uspoređuje željene termine pacijenta i dostupne termine te zakazuje novi pregled.



Elementi opisa slučaja korištenja (nastavak)

- Tok događaja (flow of events)
 - Opis pojedinih koraka unutar poslovnog procesa
 - normalni tok – standardni, uobičajeni koraci
 - podtokovi – zasebni slučajevi korištenja
 - definirani da bi se pojednostavnio normalan tok
 - slijede iz grananja dijagrama aktivnosti
 - mogu se oblikovati uključivanjem u normalan tok (<<include>>)
 - alternativni/izuzetni tokovi – mogu se dogoditi, ali nije uobičajeno
 - npr. potreba za unosom novog pacijenta prije nego mu se zakaže pregled

□ Navedeni su minimalni elementi, "zaboravljen je":

- procjena vremena potrebnog za realizaciju
- specifični tokovi podataka
- atributi, ograničenja ili operacije u vezi sa slučajem korištenja
- prepostavke (pre-conditions) i posljedice (post-conditions)
- ...

Primjer opisa slučaja korištenja 3/3

□ Primjer: Dizajn\UCprimjer

Izvodi se podtok T-3: promjeni pregled.

6. Službenik obavještava pacijenta o rezultatu transakcije.

Podtokovi:

T-1: Novi pregled

1. Službenik pita pacijenta za moguće termine pregleda.
2. Službenik uspoređuje željene termine pacijenta i dostupne termine te zakazuje novi pregled.

T-2: Otkaži pregled

1. Službenik pita pacijenta za termin zakazanog pregleda.
2. Službenik pronađe zakazani pregled i otkazuje ga.

T-3: Promjeni pregled

1. Službenik izvodi podtok T-2: otkaži pregled.
2. Službenik izvodi podtok T-1: novi pregled.

Alternativni/izuzetni tokovi:

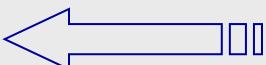
3a: Službenik izvodi slučaj korištenja "Unos novog pacijenta".

T-1, 2a1: Službenik u skladu s dostupnim terminima predlaže alternativne termine pregleda.

T-1, 2a2: Pacijent se odlučuje za jedan od predloženih termina ili odlučuje ne zakazati pregled.

Preporuke za pisanje opisa slučajeva 1/2

- Napisati svaki korak u obliku *Subjekt-Predikat-Objekt*
- Razjasniti tko je pokretač (subjekt), a tko primatelj (objekt) u koraku
- Utvrditi da je skup koraka slučaja smislen (vidi primjer UC 2/3)
 - glavni sudionik pokreće izvršenje slučaja slanjem zahtjeva (podataka) u sustav (npr. koraci 1 i 2)
 - sustav brine da je zahtjev valjan (korak 3)
 - sustav obrađuje zahtjev te moguće mijenja svoje unutarnje stanje (koraci 4 i 5)
 - sustav vraća rezultate obrade (korak 6)



Smjernice za pisanje opisa slučajeva 2/2

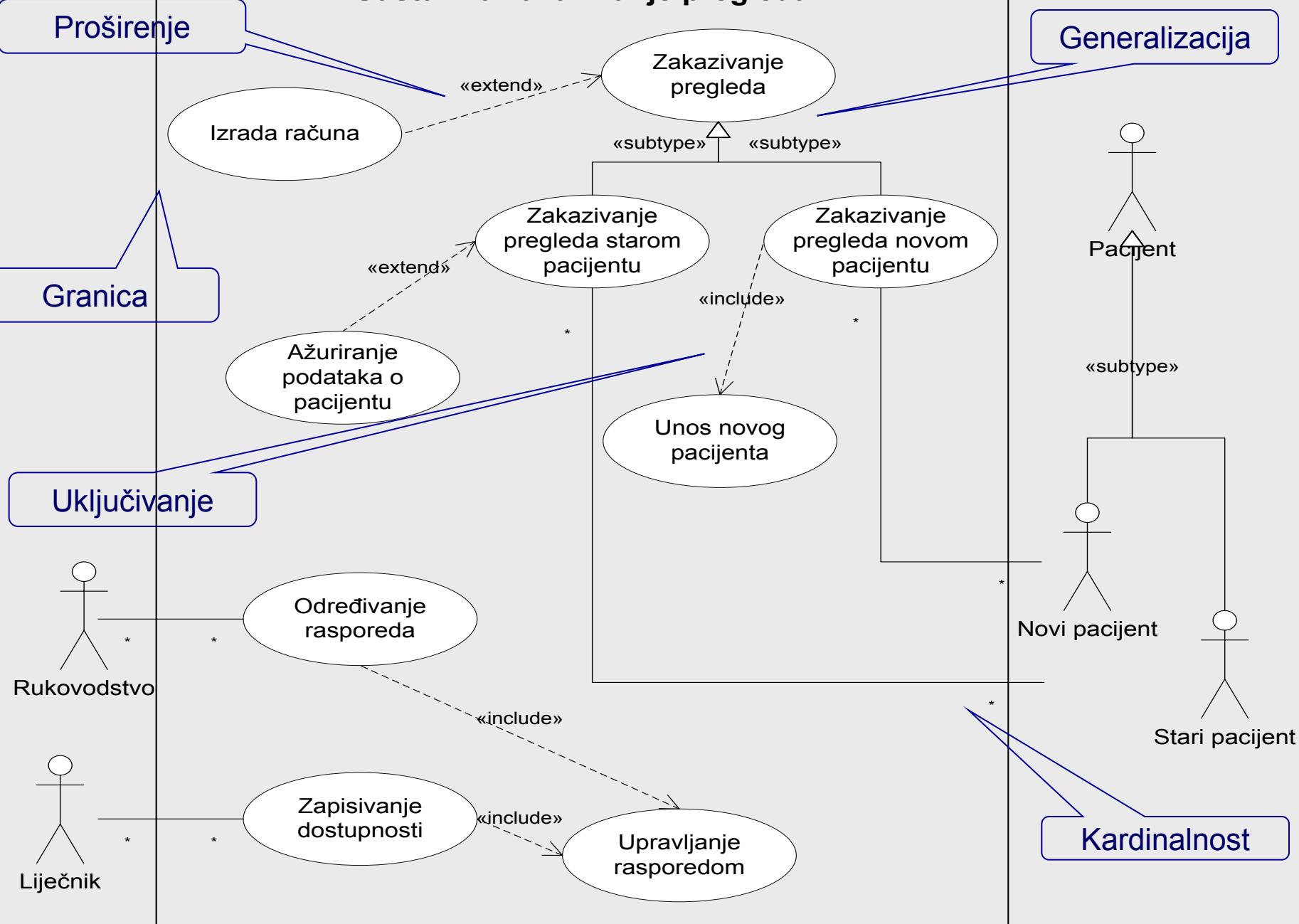
□ Pravilo KISS (Keep It Simple, Stupid / Keep it Short and Simple)

- presložen ili predug slučaj treba podijeliti u više manjih
- ako normalni tok postane složen, treba ugraditi podtokove
- ali ne pretjerivati s dekompozicijom !

□ Za korake koje treba ponavljati – napisati uputu

- Primjer: Novi pregled
 - Službenik pita pacijenta za mogući termin pregleda
 - Službenik provjerava je li termin dostupan
 - **Ponavljati korake 1 i 2 sve dok se ne pronađe dostupan termin**
 - Službenik unosi datum i vrijeme sastanka
 -

Sustav za zakazivanje pregleda



Komentar dijagrama i daljnji koraci

□ Granica sustava (subject boundary)

- razdvaja vanjske i unutarnje dijelove
- primijetiti da na prethodnoj slici nedostaje *Službenik*, jer je dio sustava (unutarnji sudionik)

□ Mogući daljnji koraci

- Rafiniranje veličine projekta UC točkama - slično funkcijskim točkama
- Modeliranje struktura - u nastavku

□ Primjeri: Dizajn\UC*

Modeliranje struktura

Structural modeling

Modeliranje struktura

□ Model strukture (structural model)

- formalna prezentacija objekata kreiranih i korištenih u poslovnom sustavu
- opisuje strukturu podataka koji prate poslovni proces

□ Modeliranje strukture

- modelira se iterativno od konceptualne do detaljne razine
- u fazi analize bez ulazeња u detalje pohrane podataka

□ Primjenjive tehnike

- CRC kartice
- dijagrami razreda
- dijagrami objekata

CRC kartice

□ Razred-Odgovornost-Suradnja (*Class-Responsibility-Collaboration*)

□ CRC kartica

- modelira odgovornosti jednog razreda i njegovu suradnju s drugim razredima
- može se koristiti umjesto dijagrama suradnje
- preporuča se da sadrži najviše tri odgovornosti visoke razine
 - u protivnom – razmisliti o razdvajanju razreda u nekoliko manjih

□ Odgovornosti

- odgovornost znanja (podaci)
- odgovornost radnje (operacije, postupci)

□ Suradnja

- klijent (client) – instanca koja šalje zahtjev drugoj instanci za izvođenjem
- poslužitelj (server) – instanca koja prima zahtjev
- ugovor (contract) – formalizacija interakcije između klijenta i poslužitelja
 - npr. pacijent dogovara pregled s liječnikom
 - oba sudionika imaju obvezu dolaska na pregled
 - ukoliko se pacijent ne pojavi, otpada plaćanje računa

Primjer CRC kartice

Prednja stranica:

Razred: Pacijent	ID: 3	Tip: Konkretni, Domenski
Opis: Pojedinac koji treba ili prima medicinsku pomoć.		Slučajevi korištenja: 2
Odgovornosti Zakazati pregled _____ Odrediti zadnji posjet _____ Promijeniti status _____ Dohvatiti povijest bolesti _____ _____ _____		Suradnici Pregled _____ _____ Povijest bolesti _____ _____

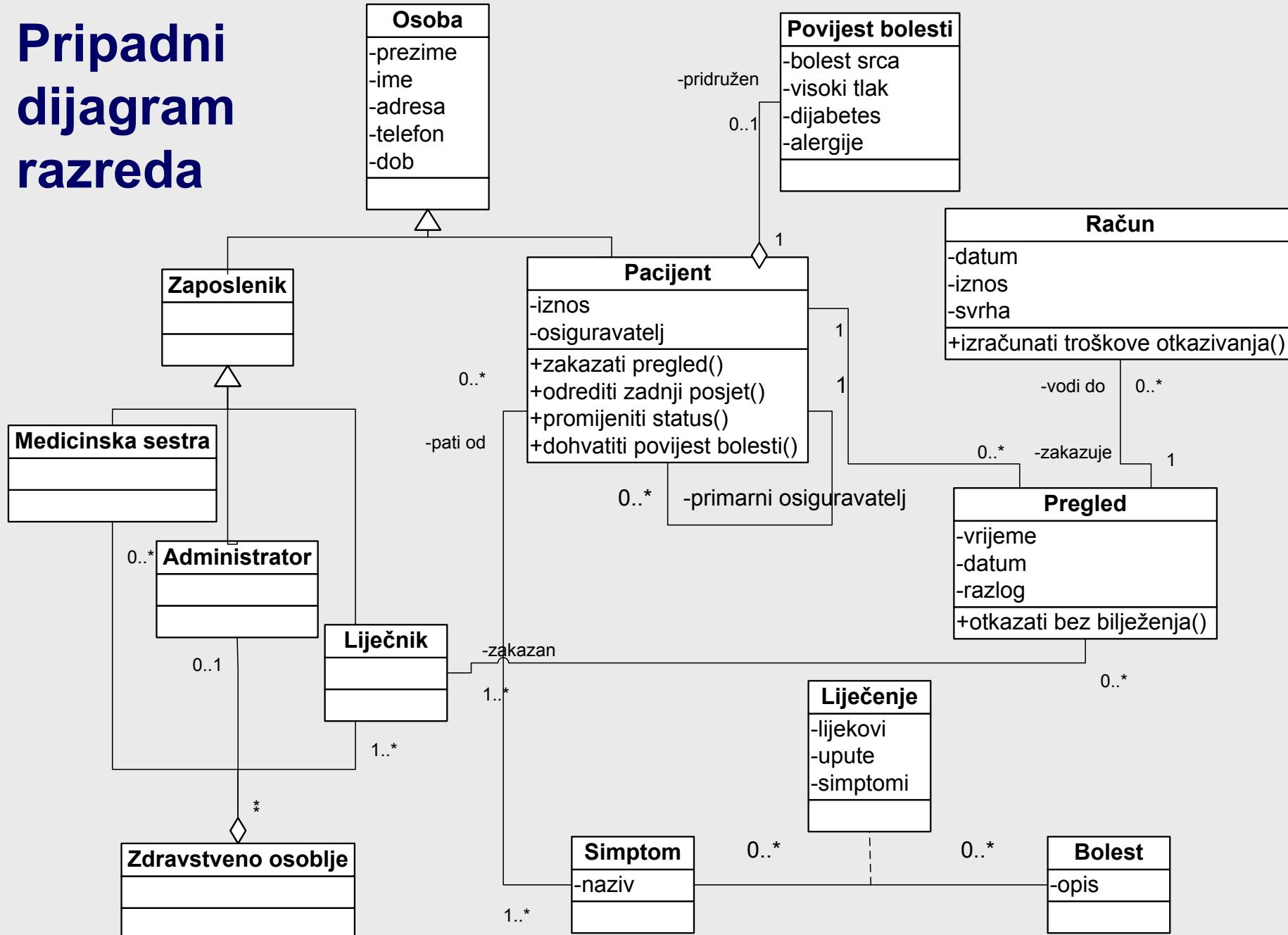
Stražnja stranica:

Atributi:	Iznos (double) _____ Nositelj osiguranja (text) _____ _____ _____
Veze:	Generalizacija (a-kind-of): Osoba _____ Agregacija (has-parts): Povijest bolesti _____ Ostale veze : Pregled _____

Koraci modeliranja strukture

- **Izrada CRC kartice analizom opisa UC**
 - određivanje razreda, atributa, operacija i asocijacija
- **Analiza skupne liste objekata (common object list approach)**
 - opipljive stvari (tangible things) – npr. oprema, materijal, roba
 - incidenti (incidents) – događaji, npr. sastanci, letovi, akcidenti
 - uloge (roles) – uloge osoba, npr. liječnici, pacijenti, administratori
 - interakcija (interaction) – transakcije, npr. akcija prodaje
- **Glumljenje CRC karticama (role-play the CRC cards)**
 - testiranje UC – antropomorfizam – igrač uzme karticu i glumi razred
 - Tko si ili što si ? Što znaš ? Što možeš ?
- **Primjena obrazaca (incorporate patterns)**
 - "kako se to radi"
 - analitički obrasci, npr. *Partner-Dokument-Stavke-Artiki*
 - "kako realizirati", "kako ponovno iskoristiti"
 - **oblikovni obrasci, aplikacijski okviri, knjižnice razreda**

Pripadni dijagram razreda



Dizajn vođen odgovornostima

Dizajn vođen odgovornostima

□ Responsibility-Driven Design

- postupak oblikovanja softvera koji naglašava modeliranje uloga i odgovornosti objekata te suradnje objekata

□ Načela dizajna vođenog odgovornostima

- Maksimiziranje apstrakcije
 - početno se skriva (zanemaruje) razlika između podataka i ponašanja
 - odgovornosti objekta su "znanje", "činjenje", "odlučivanje"
- Distribucija ponašanja
 - poboljšavanje odabrane upravljačke arhitekture
 - pametni objekti – inteligentno ponašanje, a ne samo pohrana podataka
- Očuvanje elastičnosti, fleksibilnosti
 - projektiranje objekata (razreda) koje se može brzo i lako mijenjati

Pristup oblikovanju vođenom odgovornostima

□ Postupak oblikovanja vođenog odgovornostima

- Analiza slučajeva korištenja ili korisničkih priča
- Izrada liste potencijalnih poslovnih objekata i njihovih veza
- Određivanje (jedinstvene) odgovornosti objekata
- Postupnim pročišćavanjem nastane dizajn razreda
- Temeljem istih slučajeva korištenja projektira se relacijski model
.... (kodiranje, refaktoriranje)

Analiza slučajeva korištenja ili korisničkih priča

- **Objekti postoje da bi podržali specifični UC ili korisnički scenarij**
 - posljedično, objekti sadrže poslovnu logiku i podatke
 - koje korisnik treba da bi ispunio neki zadatak – slučaj korištenja
 - ovaj zahtjev može umanjiti ponovnu iskoristivost (reusability) !
- **Svaki zadatak je slučaj korištenja – dekompozicija**
 - veće zadatke treba razdijeliti u manje od kojih je svaki zaseban UC
 - neki će slučajevi stoga ovisiti o drugima (*uses, extends*)
- **Krajnji cilj**
 - svaki slučaj korištenja niske razine (elementarni slučaj) opisuje jedan samostalan zadatak, tj. scenarij koji može biti jasno zapisan

Objekti s odgovornostima

- Nakon što su određeni elementarni slučajevi ustanovljavaju se objekti potrebni za implementaciju slučaja korištenja
- Pažnja se usmjerava na to što objekt radi a ne na podatke !
- Koji je posao ili uloga objekta unutar slučaja korištenja ?
 - preuzimanje (unos) podataka, provjera podataka, izračun, ...
- **Dizajn jednostrukе odgovornosti**
 - Pojedini objekt treba imati jednu jednostruku odgovornost unutar UC
 - Objekt može imati više ponašanja (svojstava, postupaka), ali takvih da objekt podupire jedinstvenu odgovornost

Restrukturiranje i optimiranje dizajna

□ Restrukturiranje dizajna

■ Faktoriranje

- postupak izdvajanja novog modula (razreda, metode) iz postojećih
- npr. da u primjeru ordinacije nije bio identificiran Zaposlenik, izvjesno bismo ga izlučili iz Sestra, Administrator ili Doktor

■ Normalizacija

- ustanovljavanje razreda koji nedostaju
- pretvorba asocijacijskih razreda i agregacija u attribute

□ Optimiranje dizajna (**refaktoriranje - u fazi kodiranja**)

- skraćenje pristupnih putova između objekata
- smanjenje broja metodom poslanih poruka
- redoslijed izvršenja naredbi (npr. redoslijed pretraživanja)
- uvođenje izvedenih atributa umjesto ponavljanog izračuna

Zavisnost objekata i normalizacija ponašanja

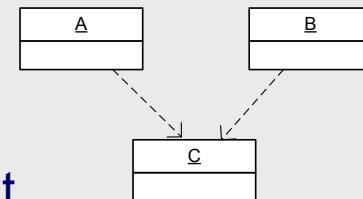
□ Npr. Objekt A i Objekt B podupiru svaki svoj UC

- A treba neko ponašanje od B (coupling)
- svaki od objekata podržava vlastiti UC
- rizik da će se B promijeniti radi promjene zahtjeva na UC koji podržava



□ Normalizacija ponašanja

- postupak postizanja ponovne iskoristivosti i kontrole kopčanja
- izdvajanje koda koji se želi ponovno iskoristiti u zaseban objekt
- npr, ponašanje iz B izdvaja se u C, te A i B nastavljaju nezavisno kolaborirati



□ Ponovnu iskoristivost forsirati samo ako se može izbjegići kopčanje !

Primjer dizajna vođenog odgovornostima

Modeliranje analizom više slučajeva korištenja

Slučajevi korištenja sustava evidencije projekata

□ Dodavanje projekta (provodi upravitelj projekta)

- podaci o projektu sadrže naziv, opis, datum početka i datum završetka
- projekt ima jedinstveni broj, ali se korisnik ne treba njime posebno baviti
- projekt se identificira nazivom, pa je to zahtijevana vrijednost
- ostale vrijednosti su opcionalne
- moguće je dodati projekte koji još nisu započeti
- započeti projekt ne mora imati datum završetka
- datum početka mora biti manji od datuma završetka
- svaki projekt ima listu sudionika (pogledati Pridjeljivanje sudionika)
 - pojednostavnjena evidencija - općenitije dodavali bi se i drugi resursi

□ Ažuriranje projekta (provodi upravitelj projekta)

- moguće je ažurirati postojeći projekt odabirom s liste projekata
- može se ažurirati datum početka, datum završetka i opis
- moguće je ažurirati listu sudionika (pogledati Pridjeljivanje sudionika)

□ Brisanje projekta (upravitelj projekta ili administrator)

- brisanje se obavlja odabirom projekta s liste projekata

Evidencija sudionika

- Dodavanje osoba (upravitelj projekta ili rukovoditelj)**
 - pohranjuju se ime i identifikator zaposlenika, uz obvezan unos
 - treba omogućiti evidenciju projekata osobe (zaduženja) pri dodavanju osobe (Pridjeljivanje sudionika)
- Ažuriranje osobe (upravitelj projekta ili rukovoditelj)**
 - dozvoliti izmjenu imena
- Brisanje osobe (upravitelj projekta ili rukovoditelj)**
 - zaposlenik može prekinuti posao ili otići u drugi dio organizacije
 - brisanje se obavlja odabirom osobe s liste osoba
- Pridjeljivanje sudionika (upravitelj projekta)**
 - proces se pojavljuje u više drugih procesa, pa je izdvojen u zaseban slučaj
 - osoba se dodjeljuje projektu
 - evidentira se uloga osobe, odabirom s liste uloga, te datum zaduženja
 - treba dozvoliti promjenu uloge osobe
 - osoba može sudjelovati u više projekata
 - treba omogućiti brisanje zaduženja (pogledati Brisanje osobe)
- Evidencija uloga (administrator)**
 - omogućiti održavanje uloga u odgovarajućem šifrarniku

Projektiranje objekata (Object Design)

- kombiniranjem dekompozicije ("traženjem imenica") i CRC karticama
- Početni dizajn - ustanovljavanje potencijalnih entiteta
 - neki od nabrojanih entiteta nisu objekti nego npr. uloge korisnika sustava (Upravitelj projekta, Administrator, Rukovoditelj), a neki su atributi

Upravitelj
Naziv projekta
Administrator
Osoba
Rukovoditelj
Lista uloga
Lista osoba

Projekt
Datum početka
Lista projekata
Naziv zaposlenika
Lista projekata osobe
Zaduženje
Lista sudionika projekta

Broj projekta
Datum završetka
Zaposlenik
Id. zaposlenika
Uloga
Datum zaduženja

- Lista entiteta se reducira te se dodaju odgovornosti i suradnici
- Napomena: dalje se nećemo baviti aspektima sigurnosti i zaštite

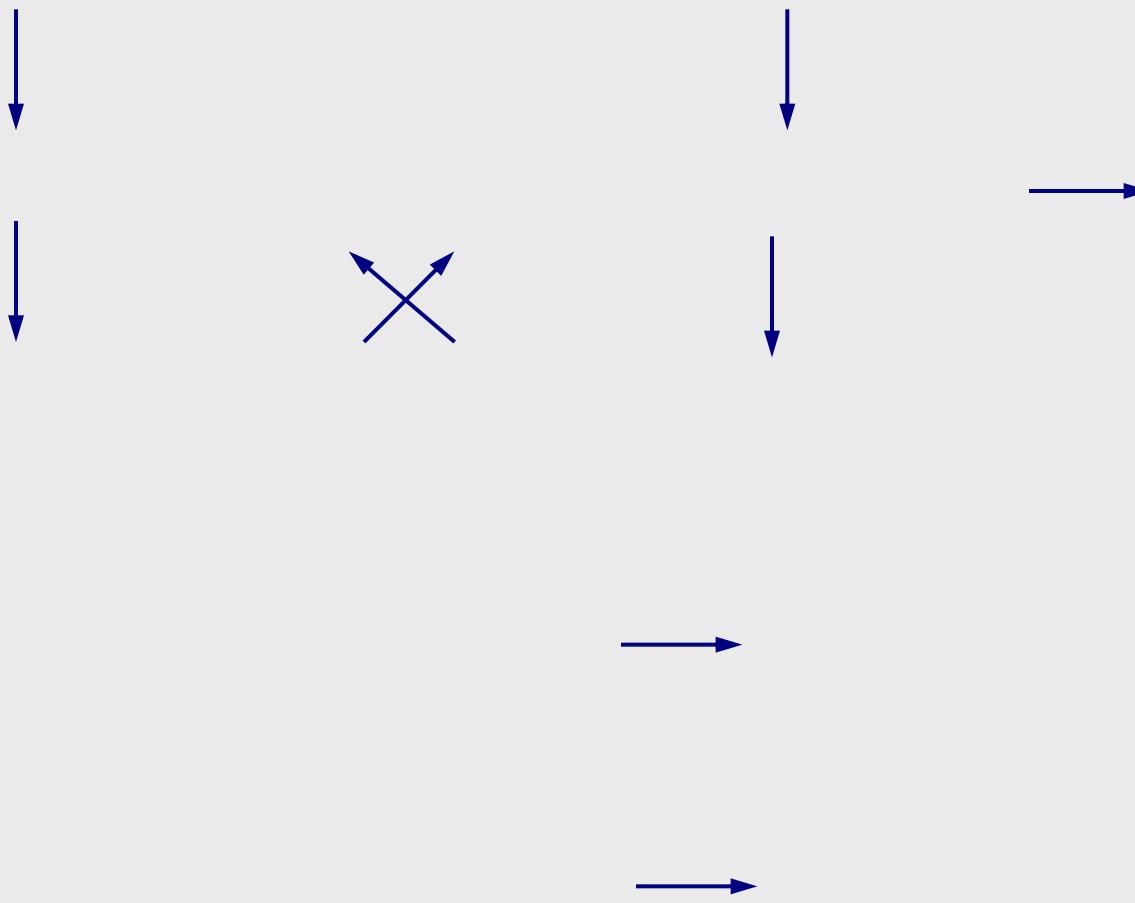
Razredi potencijalnih objekata

□ CRC kartica visoke razine za potencijalne objekte:

- odgovornost treba biti izražena kratko i precizno
- ako nije, to može značiti da je riječ o složenom objektu koji treba rastaviti

Potencijalni razred	Odgovornost	Suradnici
Projekt	dodaje i ažurira ispravan projekt	SudionikProjektaList
Osoba	dodaje i ažurira ispravnu osobu	ProjektOsobeList, Zaposlenik
Zaposlenik	dodaje i ažurira ispravnog zaposlenika	
ProjektList	dobavlja read-only listu projekata	Projekt
OsobaList	dobavlja read-only listu osoba	Osoba
SudionikProjektaList	održava listu osoba dodijeljenih projektu	Osoba, UlogaList
ProjektOsobeList	održava listu projekata na kojima je osoba	Projekt, UlogaList
UlogaList	dobavlja read-only listu uloga	Uloga
Uloga	read-only vrsta uloge	
UlogaEditList	održava listu uloga	UlogaEdit
UlogaEdit	dodaje i ažurira ispravnu ulogu	

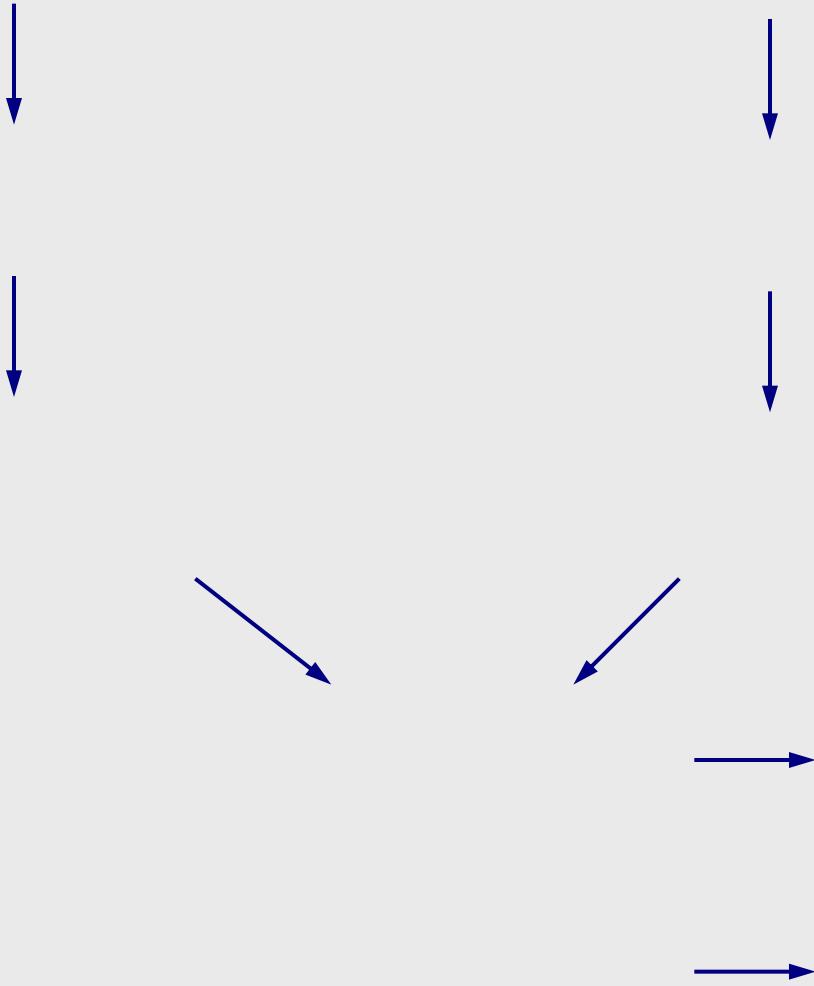
Dijagram potencijalnih razreda



Revizija dizajna

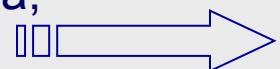
- **Osoba i Zaposlenik su potencijalni duplikati**
 - nema naznaka da osoba dodaje bilo što zaposleniku
 - Osoba i Zaposlenik mogu se udružiti
- **UlogaList i Uloga**
 - UlogaList je jednostavna lista oblika šifra/naziv (tzv. name/value list)
 - Uloga je jednostavan šifrarnički zapis (tzv. name/value placeholder)
 - može se kasnije pojednostavniti odgovarajućim razredom (npr. Csla.NameValueListBase)
- **Projekt, SudionikProjektaList, Osoba, ProjektOsobeList**
 - komplikirane, cirkularne veze – potencijalne beskonačne petlje
 - nedostaje podatak o ulozi osobe na projektu
 - osoba treba sudjelovati na više projekata ili u više uloga
 - uloga za sada uloga nepovezana
 - rješenje: dodavanje novog razreda Zaduzenje

Revidirani dijagram



Problem zaduženja

- **Razred Zaduzenje koristi se u dva konteksta**
 - s liste sudionika projekata i s liste projekata osobe
 - zajedničko dijete različitih kolekcija komplicira ugradnju i testiranje
- **Odgovornost razreda Zaduzenje ? - nejednoznačno**
 - pridruživanje osobe projektu
 - pridruživanje projekta osobi
- **Problem s podacima**
 - Projekt koristi SudionikProjektaList da dohvati listu resursa projekta
 - podrazumijeva da Zaduzenje sadrži informaciju o osobi
 - Osoba koristi ProjektOsobeList da bi dohvatio listu projekata osobe
 - podrazumijeva da Zaduzenje sadrži informaciju o projektu
- **Očigledno postoji konflikt ponašanja ali i konflikt podataka**
- **Moguća rješenja**
 - udruživanje SudionikProjektaList i ProjektOsobeList u npr. ZaduzenjeList
 - problem bi se izgledno preselio razinu više (pa se nećemo time baviti)
 - alternativa: dijeljenje razreda Zaduzenje u dva zasebna razreda,
 - SudionikProjekta i ProjektOsobe



Alternativa (1.dio) : Dodjela osobe projektu

- **Temeljem UC, korisnik odabere projekt i dodaje osobu projektu**
 - podrazumijeva da projekt ima kolekciju pridijeljenih osoba

- **Koja ponašanja (odgovornosti) očekivati od SudionikProjektaList ?**
 - vratiti listu osoba pridruženih projektu
 - pridruživanje nove osobe projektu, npr. postupkom Dodijeli (IdOsobe)
 - davanje podataka za pregled i uređivanje

vip

Alternativa (2.dio) : Dodjela projekta osobi

- Analogno prethodnoj dodjeli, dodaje se projekt odabranoj osobi
 - podrazumijeva da osoba ima kolekciju pridijeljenih projekata

- Koja ponašanja (odgovornosti) očekivati od ProjektOsobeList ?
 - vratiti listu projekata pridruženih osobi
 - pridruživanje projekta osobi, npr. postupkom DodjeliProjektu()
 - davanje podataka za pregled i uređivanje

Što dalje: Udruživanje sličnih razreda ?

- SudionikProjekta i ProjektOsobe sadrže slične podatke, ali im je ponašanje različito
- Smislenost udruživanja u zajednički razred ProjektOsoba ?
 - Naziv projekta nema smisla kada se objekt referencira iz konteksta projekta
 - slično vrijedi za neka druga svojstva, ovisno o kontekstu
- Rješenje?
 - poslovna logika koja će baciti iznimku ukoliko se objektu ProjektOsoba, ovisno o kontekstu, pristupa na pogrešan način
 - ostaje, međutim, problem da ProjektOsoba ima različite odgovornosti
 - *Spaghetti code – povjesno dokazano loša implementacija (dakle ne tako)*

Vladanje zajedničkim ponašanjem i informacijama

- **Kako koristiti iste podatke bez umnažanja poslovne logike ?**
 - normalizacija ponašanja
- **Osnovna ideja relacijskih baza podataka je smanjenje zalihosti**
 - svaki elementarni podatak smije postojati samo jednom
- **Cilj objektnog dizajna je osiguranje jedinstvenog ponašanja**
 - svako ponašanje mora postojati samo jednom u modelu, ali smije biti korišteno od strane različitih objekata

Vladanje zajedničkim informacijama

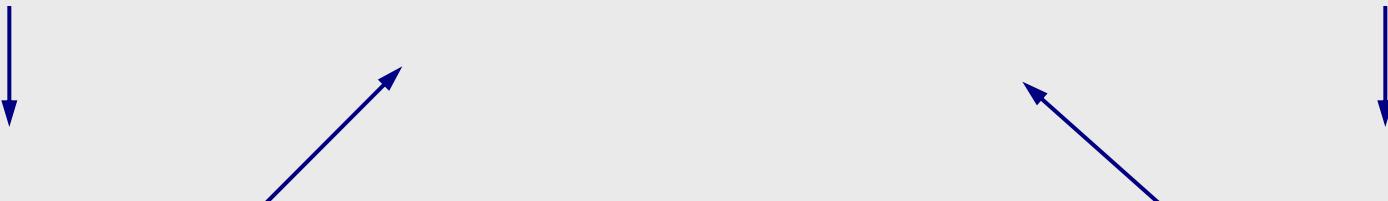
- Dat Dodjele Uloge i Id Uloge ne treba smještati u zajednički objekt
 - jer ne razmišljamo relacijski, ne normaliziramo podatke
- Pitanje je da li ti atributi imaju zajedničko ponašanje (poslovno pravilo ili poslovnu logiku) koje se može staviti u zajednički objekt
 - Primjer: validacija dodijeljene uloge

130

Vladanje zajedničkim ponašanjima

□ Odgovornost razreda `Zaduzenje` je upravljanje vezom između projekta i osobe kao resursa projekta

- `Zaduzenje` treba podržati pridruživanja osobe projektu i obrnuto
- Posao dodjele vodit će SudionikProjekta i ProjektOsobe
- kolekcijski razredi samo će dodavati i uklanjati ove objekte
- naravno, koristiti i druge postupke razreda `Zaduzenje`



Poboljšanje performansi

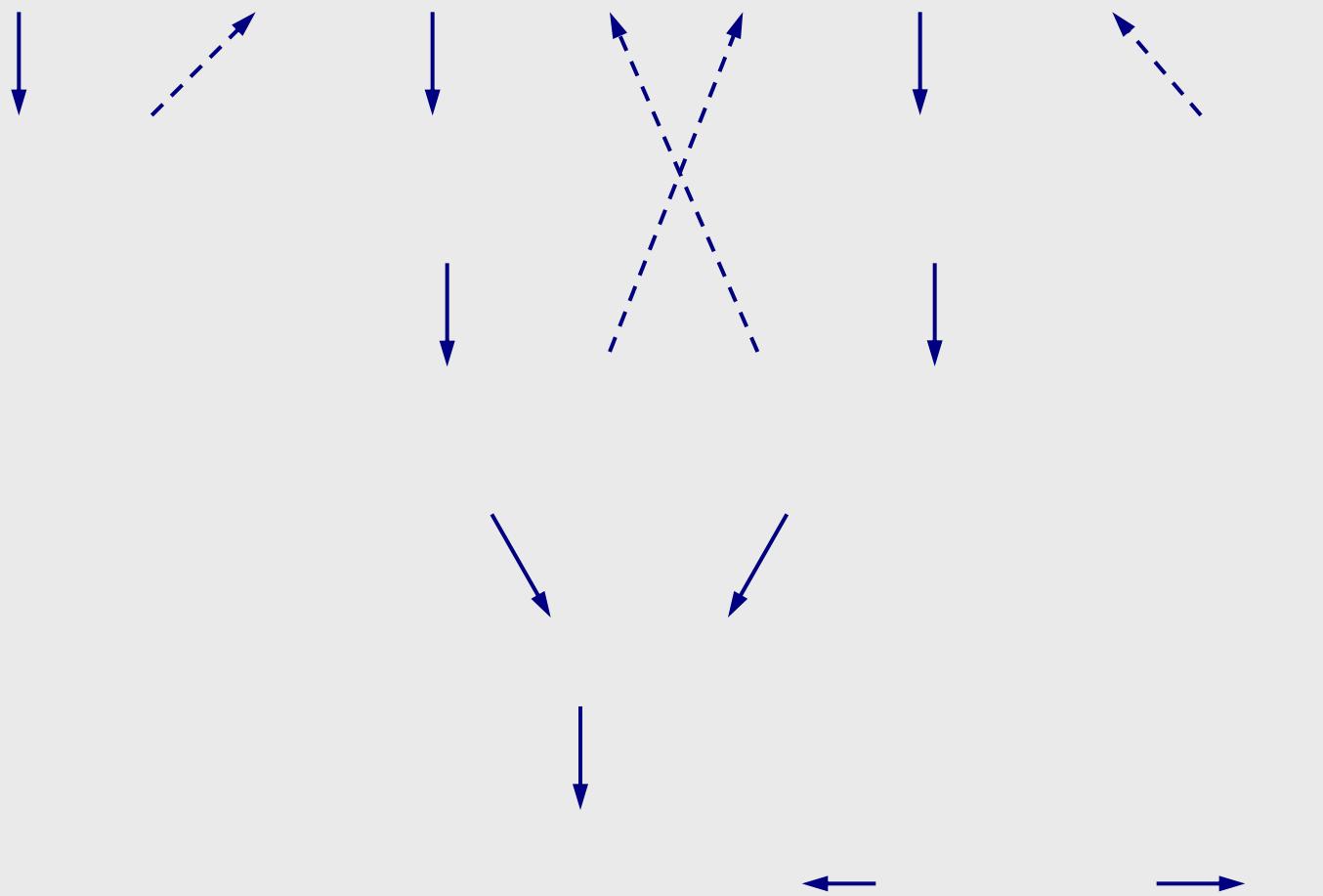
- SudionikProjektaList i ProjektOsobeList dohvaćaju kolekcije poslovnih objekata Projekt i Osoba samo da bi ih prikazali u listi
 - učitavanje objekta Projekt ili Osoba povlači učitavanje lista zaduženja ... (kaskadno)
- Umjesto učitavanja hijerarhija - formirati **read-only** liste
 - Uvode se potrebni read-only objekti za prikaz osoba i projekata ProjektInfoList, ProjektInfo, ...



Ponovna procjena dizajna

- **Završni korak - usporedba konačnog modela s izvornim UC**
- **U nastavku je prikazan dijagram koji**
 - ilustrira suradnju objekata (pune poveznice)
 - pokazuje navigaciju između objekata (isprekidane poveznice)
 - npr. s `ProjektInfo` objekta moguća je navigacija prema objektu `Projekt`, postupkom `GetProjekt()`
 - primijetiti da je `ProjektList` zamijenjen s `ProjektInfoList`, a `OsobaList` s `OsobaInfoList`
- **Slijedi konačna lista objekata i njihovih odgovornosti**

Konačni objektni model evidencije projekata



Konačna lista objekata i odgovornosti

Razred	Odgovornost	Suradnici
Projekt	dodaje i ažurira ispravan projekt	SudionikProjektaList
SudionikProjektaList	održava listu osoba dodijeljenih projektu	SudionikProjekta
SudionikProjekta	upravlja pridjeljivanjem osobe projektu	Zaduzenje, Osoba
Osoba	dodaje i ažurira ispravnu osobu	ProjektOsobeList, Zaposlenik
ProjektOsobeList	održava listu projekata na kojima je osoba	ZaduzenjeOsobe
ZaduzenjeOsobe	upravlja pridjeljivanjem projekta osobi	Zaduzenje, Projekt
Zaduzenje	održava vezu projekta i osobe	UlogaList
ProjektInfoList	dobavlja read-only listu projekata	ProjektInfo
ProjektInfo	read-only informacija o projektu	Projekt
OsobaInfoList	dobavlja read-only listu osoba	OsobaInfo
OsobaInfo	read-only informacija o osobi	Osoba
UlogaList	dobavlja read-only listu uloga	
Uloge	održava listu uloga	Uloga, UlogaList
Uloga	dodaje i ažurira ispravnu ulogu	



Na kraju

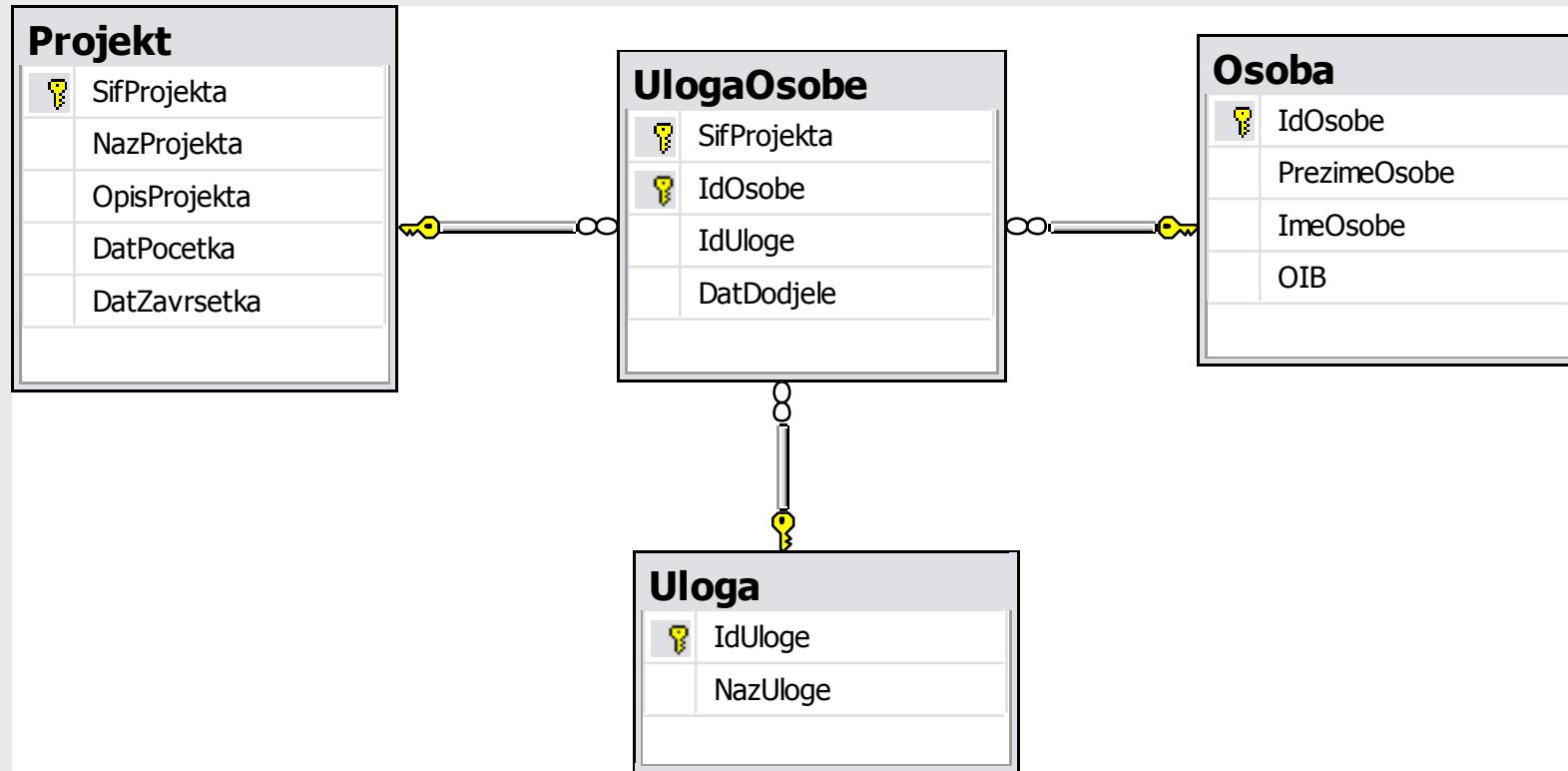
❑ Relacijski dizajn i OO dizajn

- 2 različita procesa koji rezultiraju različitim modelima
- Usklađivanje ovih modela provodi se objektno-relacijskim mapiranjem

❑ Object Relational Mapping (ORM, OR/M)

- tehnika konverzije podataka između relacijske BP i OO jezika

❑ Diskusija : code-first programming



Prilozi i reference

Dodaci

RIS06-dodatak

Resursi

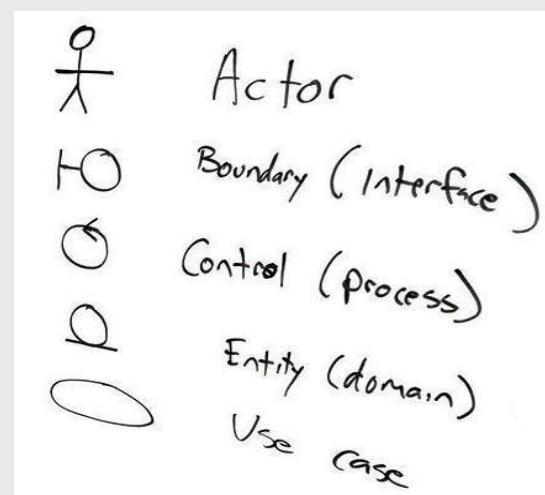
- Rockford Lhotka: Expert C# 2008 Business Objects, Apress, 2008

Dizajn

- Rebecca Wirfs-Brock: A Brief Tour of Responsibility-Driven Design
 - http://www.wirfs-brock.com/PDFs/A_Brief-Tour-of-RDD.pdf

Agilno modeliranje

- <http://www.agilemodeling.com>
- .../artifacts/robustnessDiagram.htm



UML dijagrami

□ Dijagram ponašanja

- Use Case Diagram – dijagram slučajeva korištenja

□ Dijagrami strukture, Strukturni dijagrami

- Class Diagram – dijagram razreda
- Object Diagram – dijagram objekata
- Component Diagram – dijagram komponenti
- Package Diagram – dijagram paketa
- Deployment Diagram – dijagram ugradnje

□ Dijagrami dinamike

- Sequence Diagram – slijedni dijagram
- Collaboration Diagram – dijagram kolaboracije, suradnje
 - Communication Diagram – dijagram komunikacije (UML 2.0)
- Statechart Diagram, State Diagram, State Machine Diagram – dijagram stanja
- Activity Diagram – dijagrami aktivnosti

□ Ostali

- Composite Structure Diagram
- Interaction Overview Diagram
- Timing Diagram

Projektiranje arhitekture

2013/14.07

Aplikacija

- **Aplikacija – skup programskih komponenti koje čine logičku cjelinu**
 - npr. dvoslojni debeli klijenti, višeslojni pametni klijenti, web aplikacije
 - naspram toga SOA model – poslovni sustav sastavljen od aplikacija i servisa
 - korisnička aplikacija i servis kao nezavisne cjeline
 - servis je također aplikacija – XML umjesto GUI ili HTML sučelja
- **Logička arhitektura – razdvajanje tipova funkcionalnosti**
 - npr. sučelje, poslovni sloj, podatkovni sloj
 - sloj logičke arhitekture (layer) : n-layer architecture
- **Fizička arhitektura – fizička reprezentacija glavnih komponenti IS**
 - raspodjela logičkih komponenti na fizičke uređaje
 - sloj fizičke arhitekture (tier) : n-tier architecture

Veza logičke i fizičke arhitekture

- Zabluda – pretpostavka da se logička arh. preslikava u fizičku 1:1**
 - logička arhitektura odnosi se na (samo na) odvajanje funkcionalnosti (npr. korisničko sučelje, pristup podacima)
- Broj slojeva logičke \geq broj slojeva fizičke arhitekture !**
 - više logičkih slojeva može biti smješteno na isti fizički uređaj
- Prilikom kreiranja aplikacije važno je odabrati logičku arhitekturu koja će omogućiti kasniji odabir fizičke arhitekture**
 - sadrži slojeve koji organiziraju komponente u diskrete uloge
 - ima barem toliko logičkih slojeva koliko će imati odabrana fizička arhitektura
- Komunikacija slojeva – odvajanje nije uvijek jednako moguće**
 - npr. BL i DL – moderne aplikacije odvajaju podatke od ostatka
 - npr. UI i BL – nepraktično, nepoželjno zbog povezivanja sučelja (binding)

Fizičke arhitekture i složenost aplikacija

- **Višeslojne fizičke arhitekture smanjuju složenost ako je zadovoljen neki od uvjeta**
 - aplikacija je velika ili složena
 - postoji velika ili složena kombinacija više jednostavnih aplikacija
 - velika ili složena okolina (ugradnje, podrške)
- **Višeslojne fizičke arhitekture povećavaju složenost ako je istinito sve navedeno**
 - aplikacija je mala ili relativno jednostavna
 - aplikacija nije dio većeg sustava povezanih aplikacija
 - jednostavna okolina
- **jednostavne aplikacije mogu rasti i usložnjavati se → skalabilnost**

Karakteristike fizičke arhitekture (kriteriji uslojavanja)

□ Performanse - brzina odziva

- najveća kad su komponente na istom računalu

□ Skalabilnost – prilagodba kapaciteta s obzirom na opterećenje

- smanjenje opterećenja odvajanjem poslovne logike na aplikacijski poslužitelj
- za većinu aplikacija (ipak) nije dovoljan razlog za prijelaz na tri sloja
 - jer moderne BP lako poslužuju stotine istovremenih korisnika u 2slojnoj

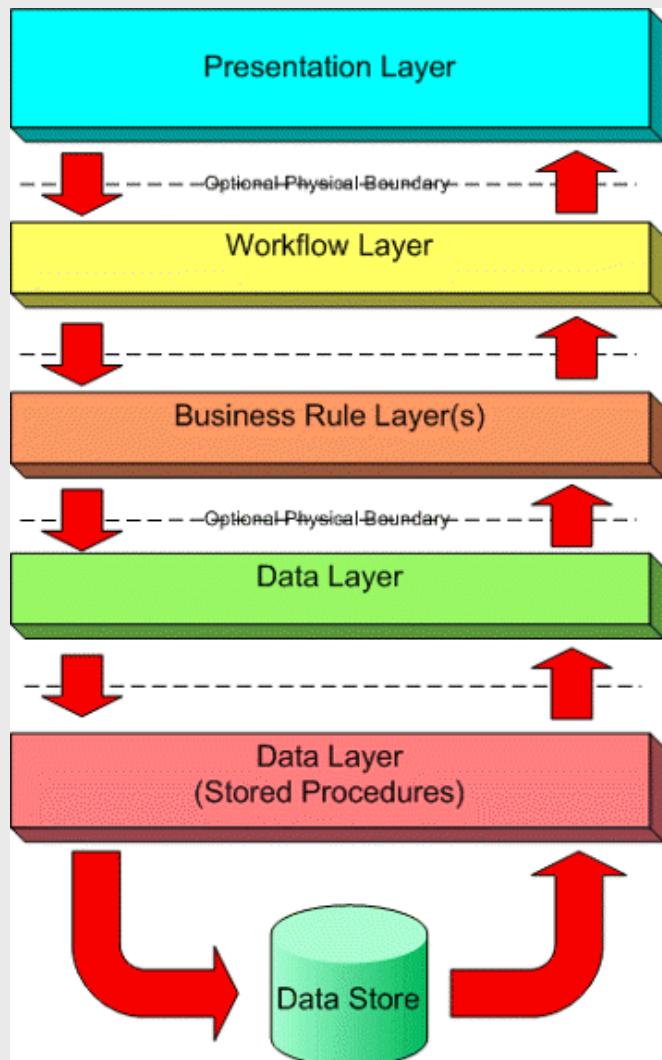
□ Sigurnost – glavni argument

- broj fizičkih slojeva ne utječe na autentifikaciju/autorizaciju korisnika
- nego na fizički pristup strojevima pojedinog sloja (pr. izloženost konekcija)
- troslojna je općenito sigurnija ali unutar iste LAN teže branjiva
 - poželjno odvajanje vatrozidom

□ Tolerancija pogrešaka

- bolja ako su fizički dijelovi zamjenjivi, npr. redundantnim hardverom

Petroslojna logička arhitektura



- Korisničko sučelje (UI)**
 - prezentacijski sloj (PL)
- Kontrola sučelja (interface control - IC)**
 - kontrola radnog toka (workflow)
- Poslovni sloj (business layer – BL)**
 - poslovna ograničenja, validacijska pravila
- Sloj za pristup podacima (data layer – DL)**
 - data access layer - DAL
- Sloj za upravljanje pohranom i upravljanje podacima (data storage and management layer – DSML)**
- Fizička pohrana (data storage)**

Slojevi petoslojne logičke arhitekture

□ Korisničko sučelje

- prikaz podataka, unos od strane korisnika
 - npr. web preglednik, SOA XML poruka, Windows sučelje
- vrlo slično i blisko kontroli sučelja

□ Kontrola sučelja

- generiranje izlaza, interpretacija / obrada ulaza
 - npr. Web, WPF ili WF pozadinski kod, MVC controller
- vođen događajima (event-driven)
- prihvat unosa korisnika i prosljeđivanje u BL na validaciju i obradu

□ Poslovni sloj

- poslovna pravila, validacija, obrada, autentifikacija, *database lookups*, ...
 - npr. MVC model
- odvajanje od sučelja olakšava održavanje i ponovnu iskoristivost
- fizički bliže PL – klijentsko računalo, interaktivne aplikacije, npr. validacija
- fizički bliže DL – nefunkcionalni procesi, npr. obrada plaća, izračun zaliha

Slojevi petoslojne arhitekture (nastavak)

□ Podatkovni sloj

- surađuje sa slojem pohrane i rukovanja podacima
- dohvaća, umeće, ažurira, briše podatke ali ne upravlja i ne pohranjuje
- sučelje između BL i DSML
- fizički se odvaja uslijed istog razloga kao BL – performanse i robusnost
- odvajanje povećava fleksibilnost – neovisnost o sustavu pohrane (BP)
 - npr. ADO.NET, LINQ, EF, ...
- omogućuje objektno-relacijsko preslikavanje (ORM)

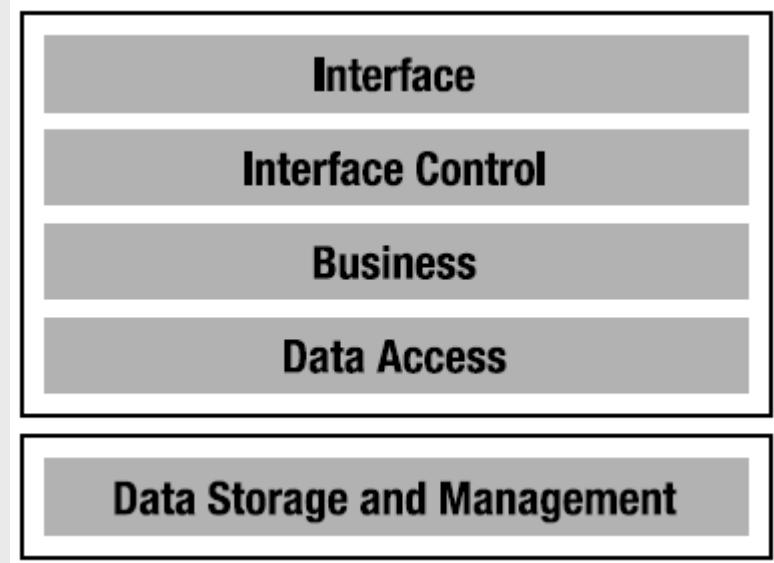
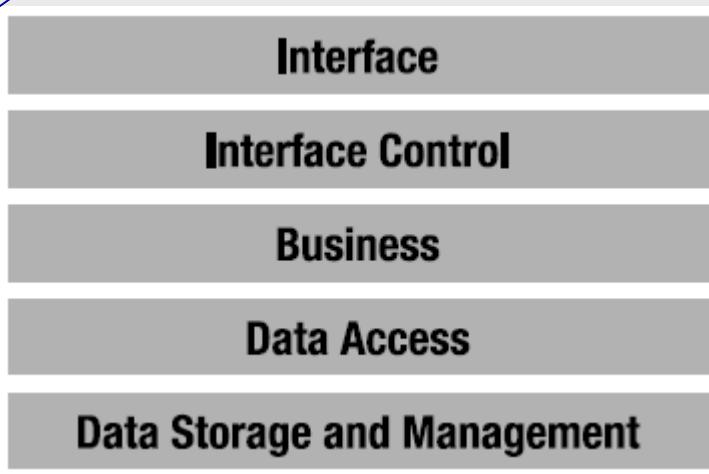
□ Sloj pohrane i rukovanja podacima

- upravlja fizičkim stvaranjem, dohvatom, ažuriranjem i brisanjem podataka
- mehanizmima RSUBP ili XML servisa implementira CRUD mehanizme
- može replicirati dio poslovnih pravila
 - npr. pravila integriteta koji su varijanta poslovnih pravila

Fizička ugradnja logičke arhitekture

□ Pametni (smart) klijent s optimalnim performansama

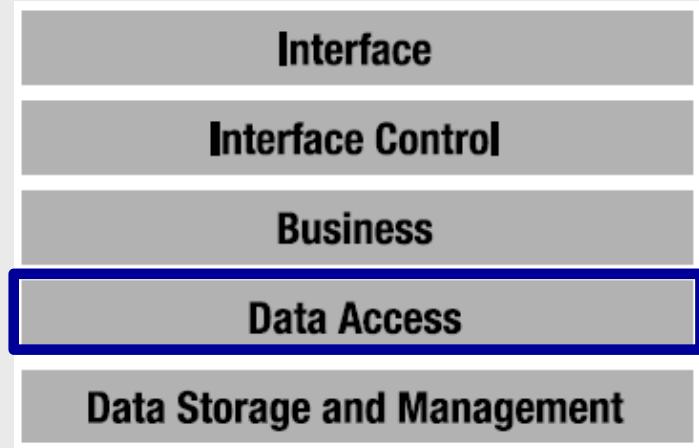
- Samostalna aplikacija
 - svi logički slojevi mogu biti smješteni na istom stroju
 - npr. blagajna (Point of Sale – POS) uz periodički transfer podataka
 - u osnovi isto što i
- Debeli klijent
 - odvajanje DSML – premještaj baze podataka na server zahtijeva samo promjenu priključka (konekcije) na BP
 - najviše okvirno 300 istovremenih korisnika



Visokoskalabilni pametni klijent u troslojnoj fizičkoj arhitekturi (1)

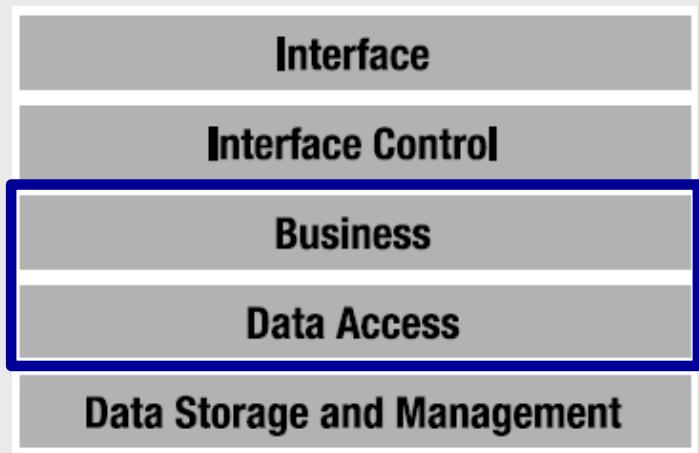
□ Odvajanje DL na aplikacijski poslužitelj

- dobro pravilo za odvajanje DAL : više od 50-100 istovremenih korisnika
- bolja sigurnost – DAL sadrži kôd pristupa BP, koji se seli na aplikacijski poslužitelj pa neće biti izložen na korisničkom računalu
- bolja optimizacija pristupa (connection pooling) kad korisnici koriste isto korisničko ime, 150-200 istovremenih pristupa s 2-3 konekcije



□ Odvajanje BL na aplikacijski poslužitelj

- dobro za neinteraktivne aplikacije
- ipak, većina modernih aplikacija zahtijeva korisničku interakciju
- nedostatak: latencija, pad performansi

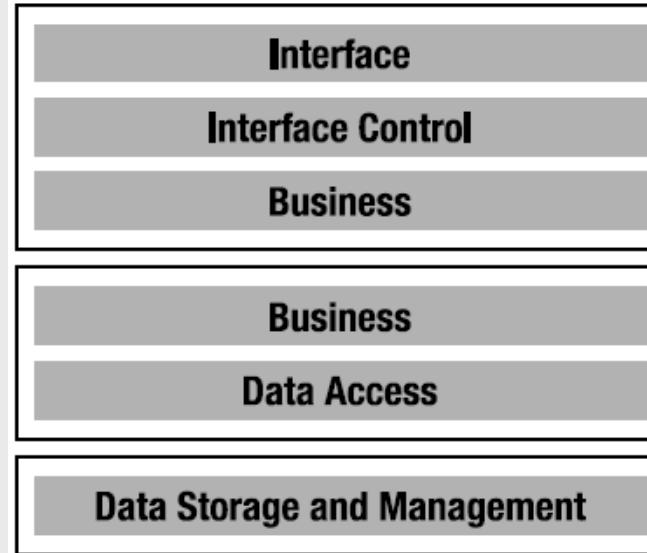


Visokoskalabilni pametni klijent u troslojnoj fizičkoj arhitekturi (2)

budući da se isti logički slojevi mogu raseliti na više fizičkih ...

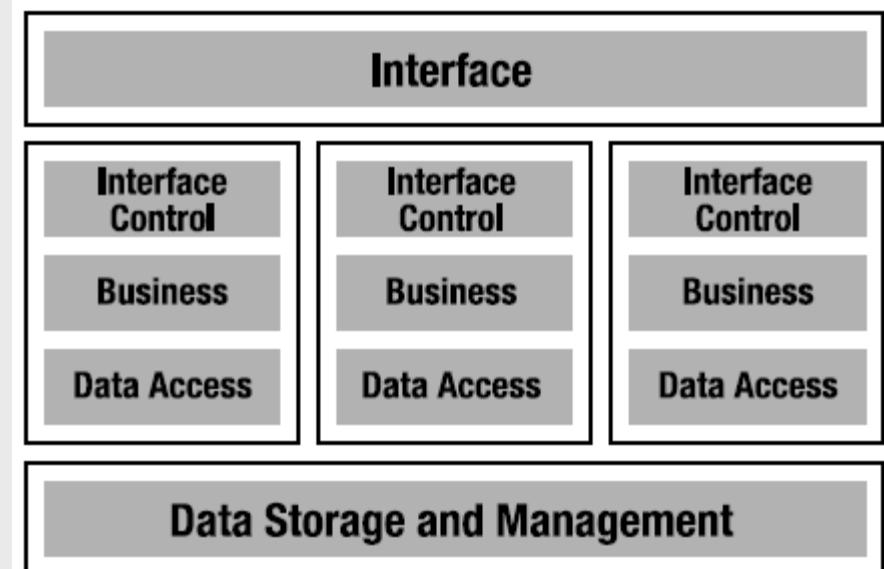
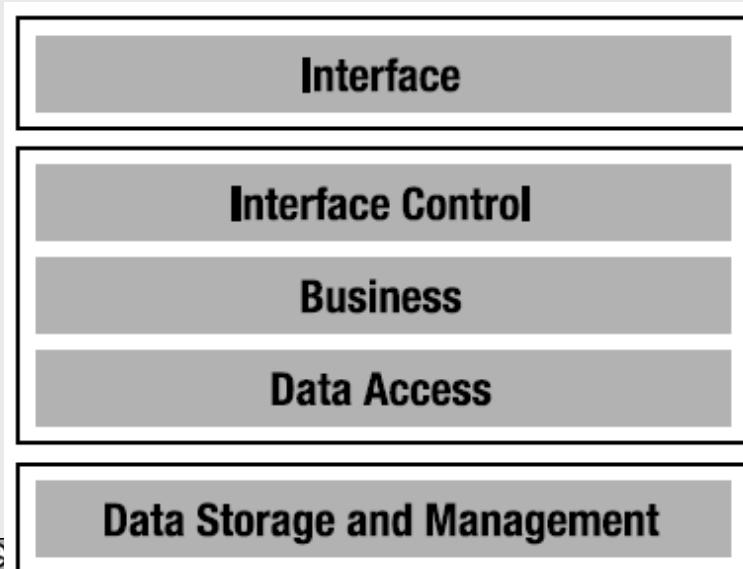
□ DL na aplikacijskom poslužitelju, BL i na klijentu i poslužitelju

- iskorištenje snage oba računala
 - Validacija na klijentu
 - Procesna brada na poslužitelju
- preko 1000 istovremenih korisnika
- vrlo velika skalabilnost



Web-klijent s optimalnim performansama

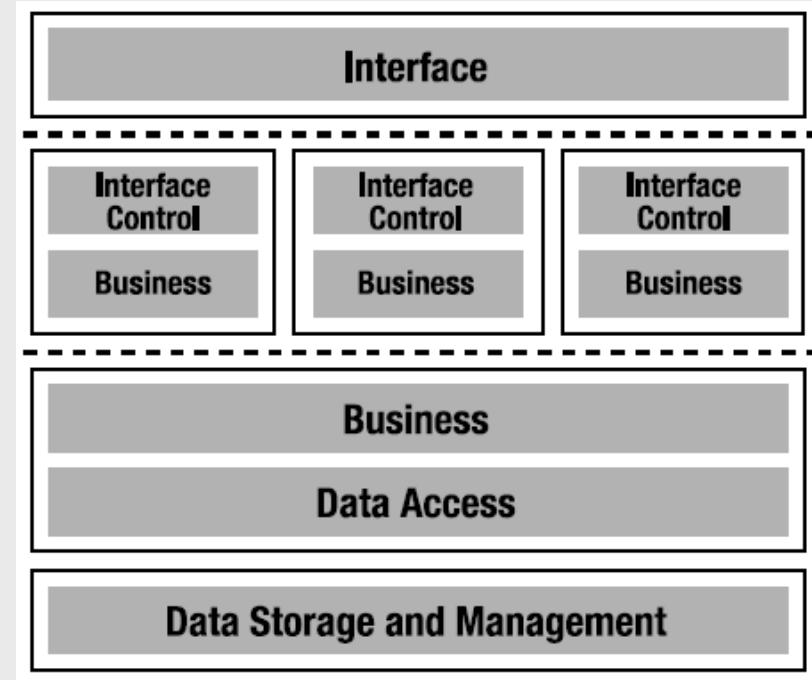
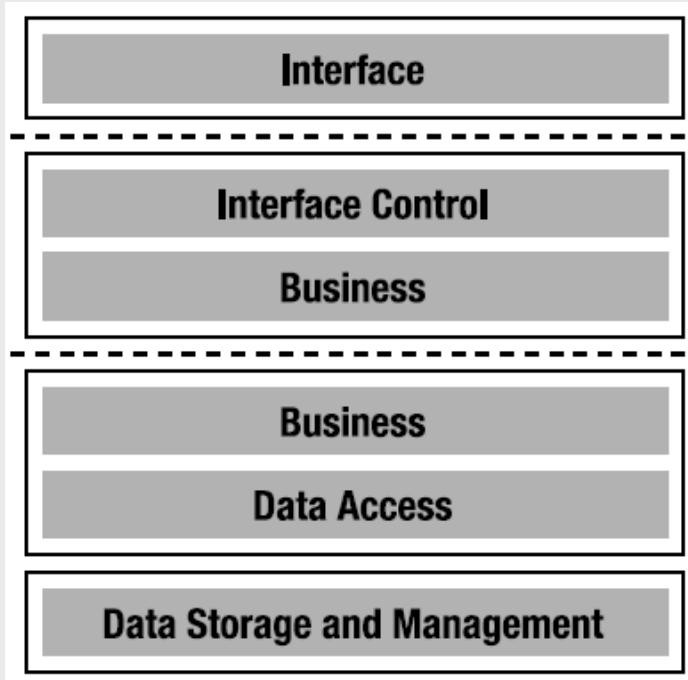
- Najbolje performanse postižu se smanjenjem broja fizičkih slojeva
- Moguće istovremeno poboljšanje performansi i skalabilnosti na uštrb sigurnosti
- Osnovna varijanta
 - IL fizički odvojen jer se izvodi u pregledniku
 - ostali slojevi na istom stroju/procesu
 - manje mrežno opterećenje, bolje performanse
- Poboljšanje
 - povećanje skalabilnosti
 - web farma na kojoj više web poslužitelja izvodi isti kod
 - dobar pristup (pooling) BP jer svaki web poslužitelj obradi stotine korisnika pa proslijeđuje zahtjeve



Vrlo sigurni Web-klijent

□ Web poslužitelj u demilitariziranoj zoni (DMZ)

- stješnjen između vanjskog vatrozida i unutarnjeg vatrozida
- komunicira s drugim poslužiteljem na kojem je BP
- slično skalabilnoj troslojnoj arhitekturi – dobro je imati BL na web poslužitelju i aplikacijskom poslužitelju
 - po cijenu 50% smanjenja performansi !
- povećanje skalabilnosti - web farmom



Odabir arhitekture s obzirom na tip aplikacija

□ Općenito, mogu se primijeniti sljedeći kriteriji:

Arhitekture	Aplikacije
Dvoslojne K/S arhitekture s tankim klijentima	<ul style="list-style-type: none"><input type="checkbox"/> Naslijeđeni, vlastiti (legacy) sustavi gdje je nepraktično i neisplativo odvajanje aplikacijske obrade i upravljanje podacima.<input type="checkbox"/> Procesno ili podatkovno zahtjevne aplikacije (pretraživanje i upiti) s vrlo malo ili bez aplikacijske obrade ("više prikazuju i računaju nego obrađuju i spremaju").
Dvoslojne K/S arhitekture s debelim klijentima	<ul style="list-style-type: none"><input type="checkbox"/> Aplikacije gdje se aplikacijska obrada izvodi na klijentu COTS (<i>Commercial Off-The Shelf Software</i>) programskom podrškom<input type="checkbox"/> Aplikacije koje zahtijevaju računalno zahtjevne obrade podataka (npr. vizualizacija podataka – interaktivno ili izvješćima).<input type="checkbox"/> Aplikacije s relativno čvrstom krajnje-korisničkom funkcionalnošću korištene u okolini gdje je dobro uspostavljeno upravljanje sustavom.
Troslojne ili višeslojne K/S arhitekture	<ul style="list-style-type: none"><input type="checkbox"/> Aplikacije velikog opsega sa stotinama ili tisućama klijenata.<input type="checkbox"/> Sustavi u kojima su i podaci i aplikacije promjenljivi.<input type="checkbox"/> Aplikacije u kojima se integriraju podaci iz višestrukih izvora.



Programski primjeri aplikacija s ORM

Debeli klijent nad Entity Framework

- Primjer: Prilozi \ RIS07-FirmaEF

Developer Express komponente:

<http://web.zpr.fer.hr/DevExpressComponents-13.2.5.exe>

Objektno relacijsko mapiranje

- Primjer: Dodaci \ RIS07-ORM.*
 - Linq to SQL (DLINQ), Nhibernate, Entity Framework, XPO

Komponentizacija, ugradnja

Aplikacijski okvir

□ Aplikacijski okvir (Application Framework) – okosnica aplikacije

- A software framework is a re-usable design for a software system (or subsystem).
- Građevni blokovi aplikacije
- Skup osnovnih softverskih rutina koje čine temeljnu strukturu za razvoj aplikacije u koju se ugrađuju aplikacijski specifične komponente (poslovni objekti, poslovna pravila, zaslonske maske, ...)

□ Okvir poduzeća (Enterprise Framework)

- cjelovito okruženje za razvoj i ugradnju složenih informacijskih sustava
- sadrži prethodno napravljene (pre-built) aplikacije i razvojne alate za prilagodbu i integraciju tih aplikacija te pisanje novih
- primjer: okviri za ERP, CRM
- mogu imati komponente za upravljanje radnim tokovima (workflow component)

Primjeri aplikacijskih okvira

- FirmaWin (RPPP)
- CSLA.NET (Component-based Scalable Logical Architecture)
 - The framework enables developers to leverage the power of object-oriented design as the basis for creating powerful applications.
 - Platforms: .NET, Windows Runtime (WinRT for W8), Windows Phone 8, Silverlight 5, Android / iPhone (using Xamarin), Linux/OSX (using Mono).
- Microsoft Enterprise Library
 - The Microsoft Enterprise Library is a collection of reusable software components (application blocks) designed to assist software developers with common enterprise development cross-cutting concerns (such as logging, validation, data access, exception handling, and many others).
- Ostali : class library, MFC, Struts, AFC, JFC, OWL

Elementi aplikacijskog okvira CSLA.NET

□ Prezentacijski sloj

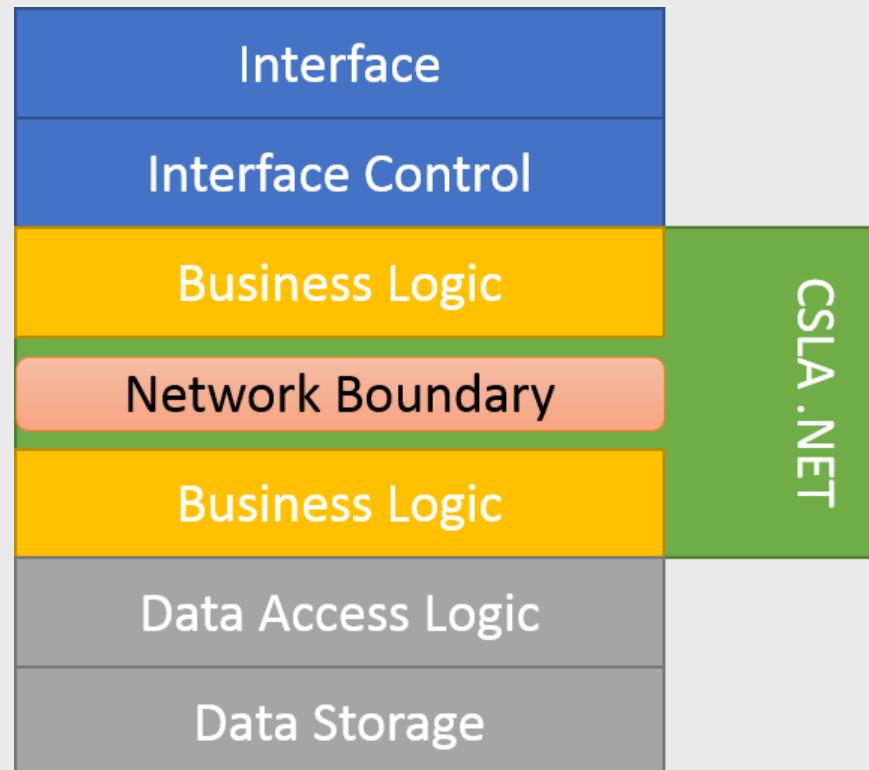
- sučelje glavne forme
- bazne forme i kontrole
- logika ponašanja aplikacije, komunikacija između formi, komunikacija s glavnom formom

□ Poslovni sloj

- bazne poslovne klase
- pomoćne klase (validacija, kontrola pristupa podacima)
- automatizacija upravljanja poslovnim objektima (stvaranje, uništavanje, prijenos između klijenta i poslužitelja)

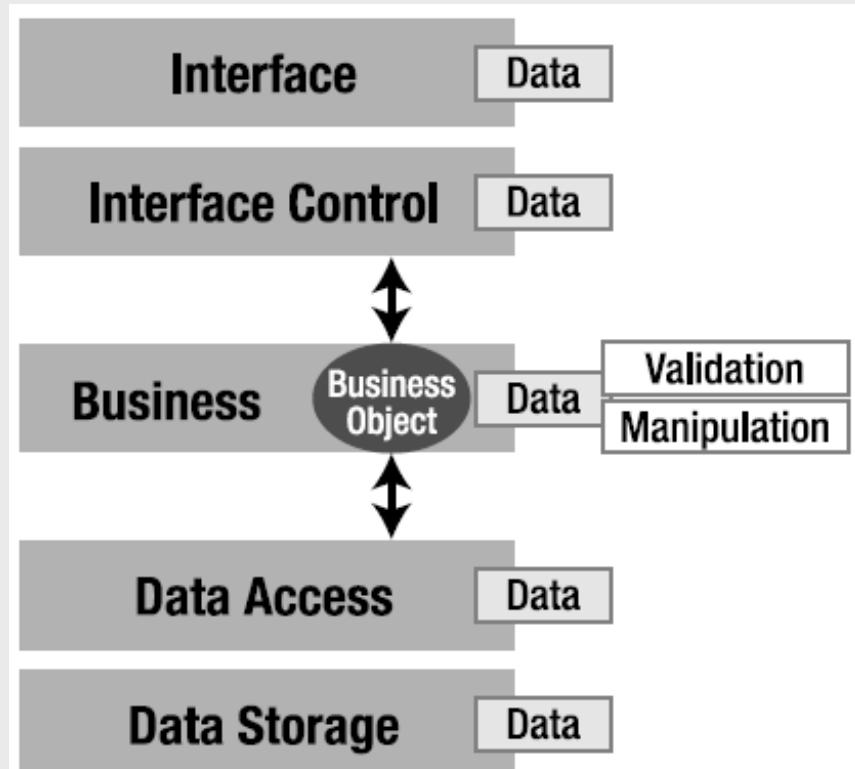
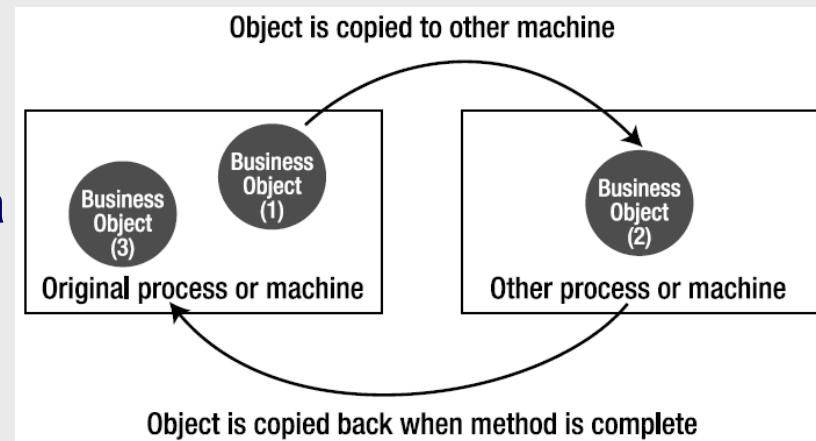
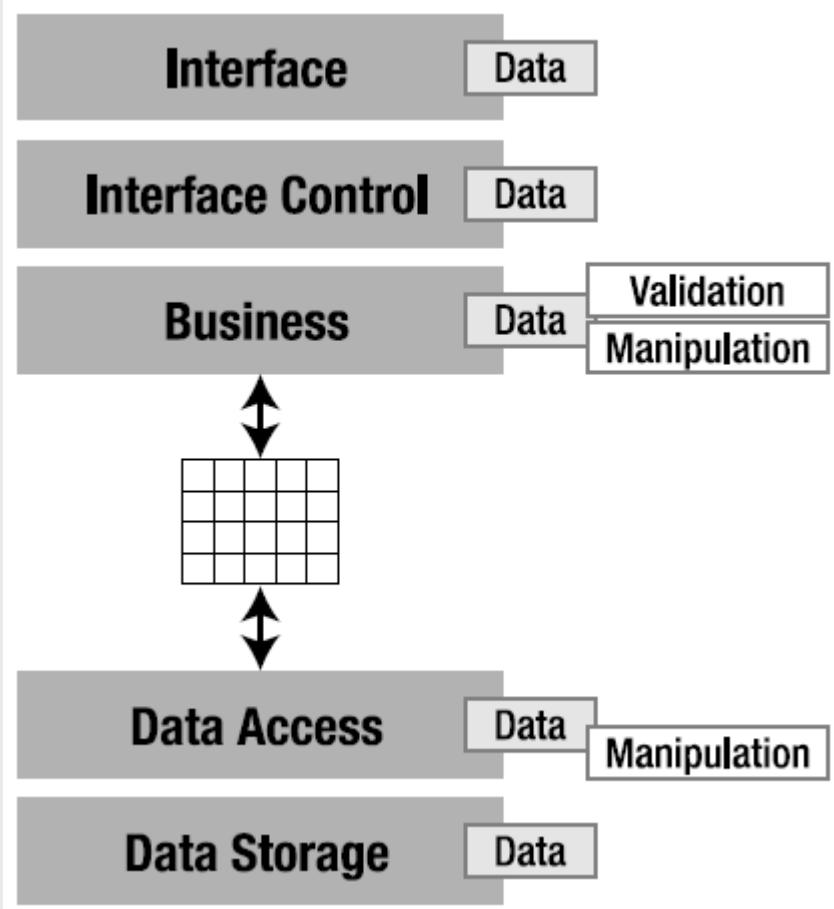
□ Podatkovni sloj

- bazne klase za pristup podacima
- automatizacija čitanja, spremanja i brisanja podataka
- upravljanje transakcijama



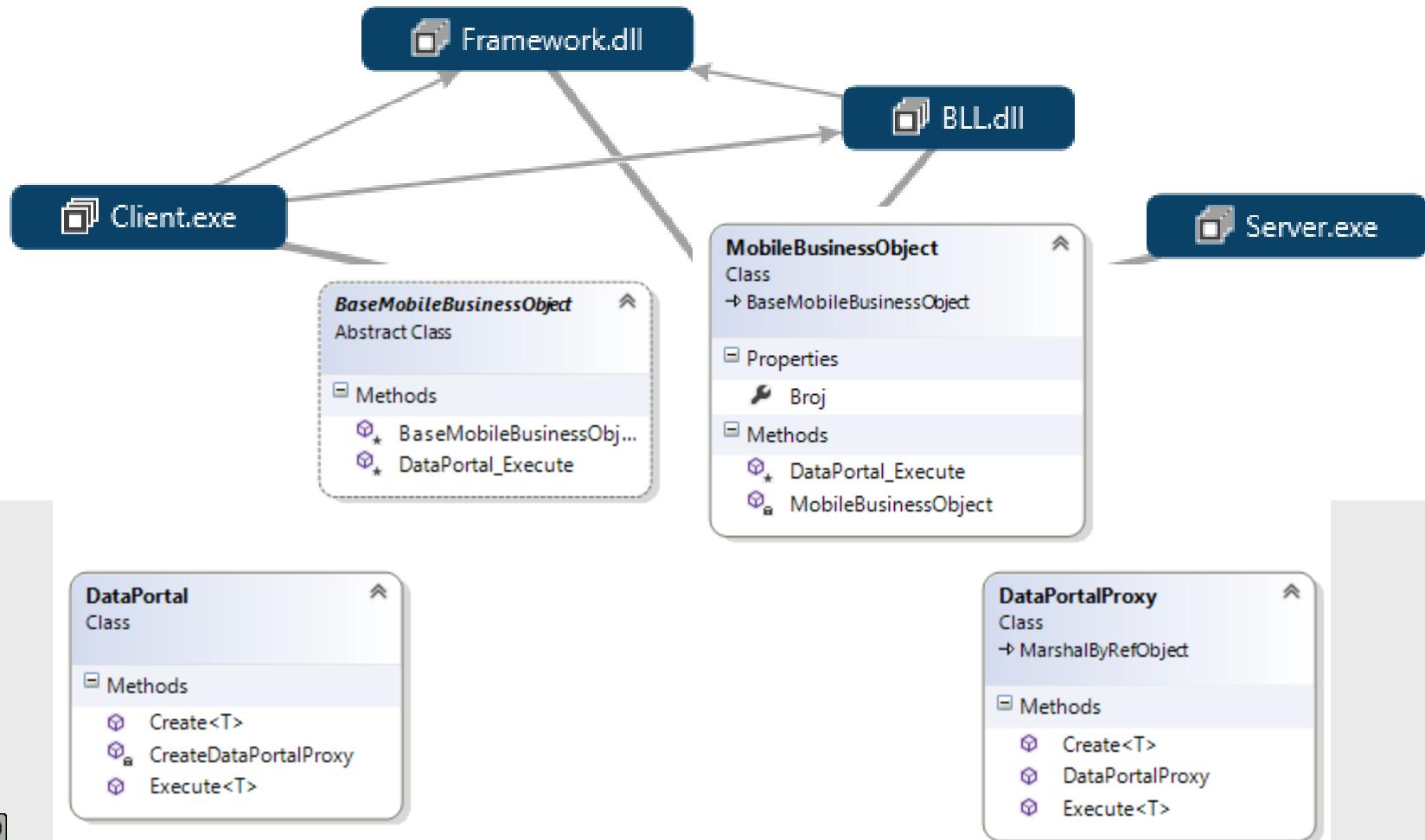
Pametan i mobilan objekt

- umjesto pasivnih skupova podataka
- poslovni objekt centralizira logiku

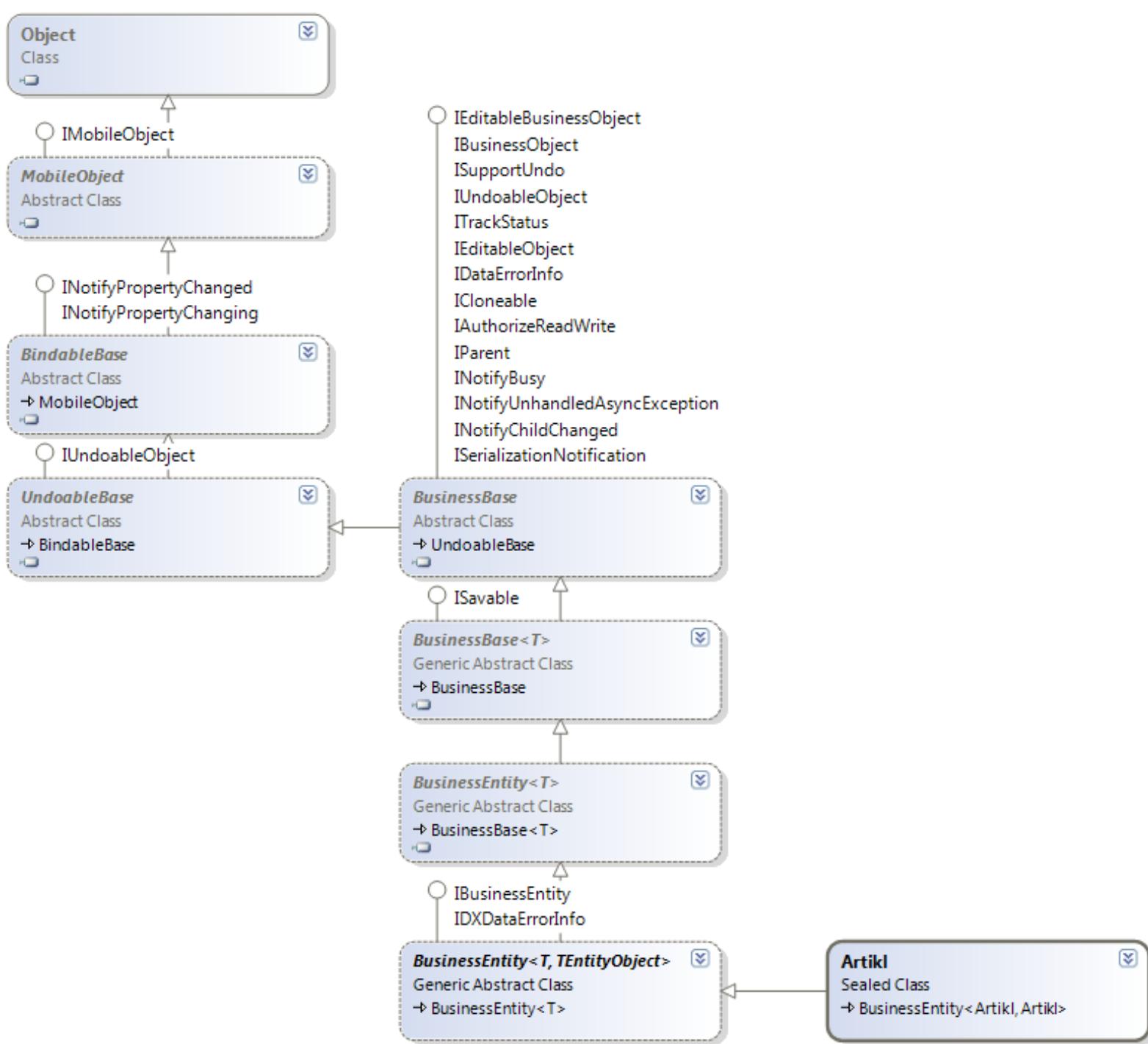


Ilustracija mobilnog objekta

- Primjer: DataPortal (client: Step into, server: Start new)



Primjer: poslovni entitet



Dijagram paketa

Package Diagram

Paketi

□ Paket (Package) = opći mehanizam grupiranja elemenata

- logički povezana grupa elemenata modela

□ Elementi dijagrama

- podsustavi
- drugi (manji) paketi
- realizacija slučajeva korištenja
- sučelja (interfaces)

□ Svojstva

- paketi tvore imenike (*namespace, domain*)
- pojedini element sadržan je samo u jednom paketu → jedinstveni nazivi
- paketi mogu referencirati druge pakete
- UML podrazumijeva da postoji anonimni paket-korijen (root package)

□ Primjena - u fazi razvoja

- grubi pregled zahtjeva (high-level overview)
- konceptualno projektiranje – dizajn sustava
- organizacija izvornog koda, modularizacija složenih dijagrama

Dijagram paketa

□ Notacija, varijante prikaza i alternative prikaza

- paket, zavisnost, ugniježđeni razredi
- moderni alati preferiraju dijagrame povezane u hijerarhiju (primjeri slijede)



□ Naziv paketa koristi se za tvorbu punog naziva razreda

- paket definira prostore imena - *namespace* u C# i C++
- vanjski paketi ponekad se nazivaju *domene*.
- npr. puni naziv razreda Racun u paketu Dokumenti je Dokumenti.Racun

Veze između paketa

- **Zavisnost (Dependency)** – jedan treba drugog radi specifikacije ili implementacije
 - nije tranzitivna !
- **Import** – elementi modela koji se uvoze iz drugog
 - omogućuje referenciranje bez kvalifikacije
- **Access** – elementi modela koji mogu biti korišteni ali ne uvezeni
- **Realizacija (realization)** – jedan specificira, drugi implementira
- **Merge** – sadržaji koji se jednako zovu i predstavljaju isti koncept
 - npr. parcijalne klase

Formiranje paketa

□ Kriteriji za grupiranje slučajeva korištenja u pakete

- potpora određene poslovne funkcije
- podrška određenom sudioniku

□ Kriteriji za grupiranje razreda u pakete

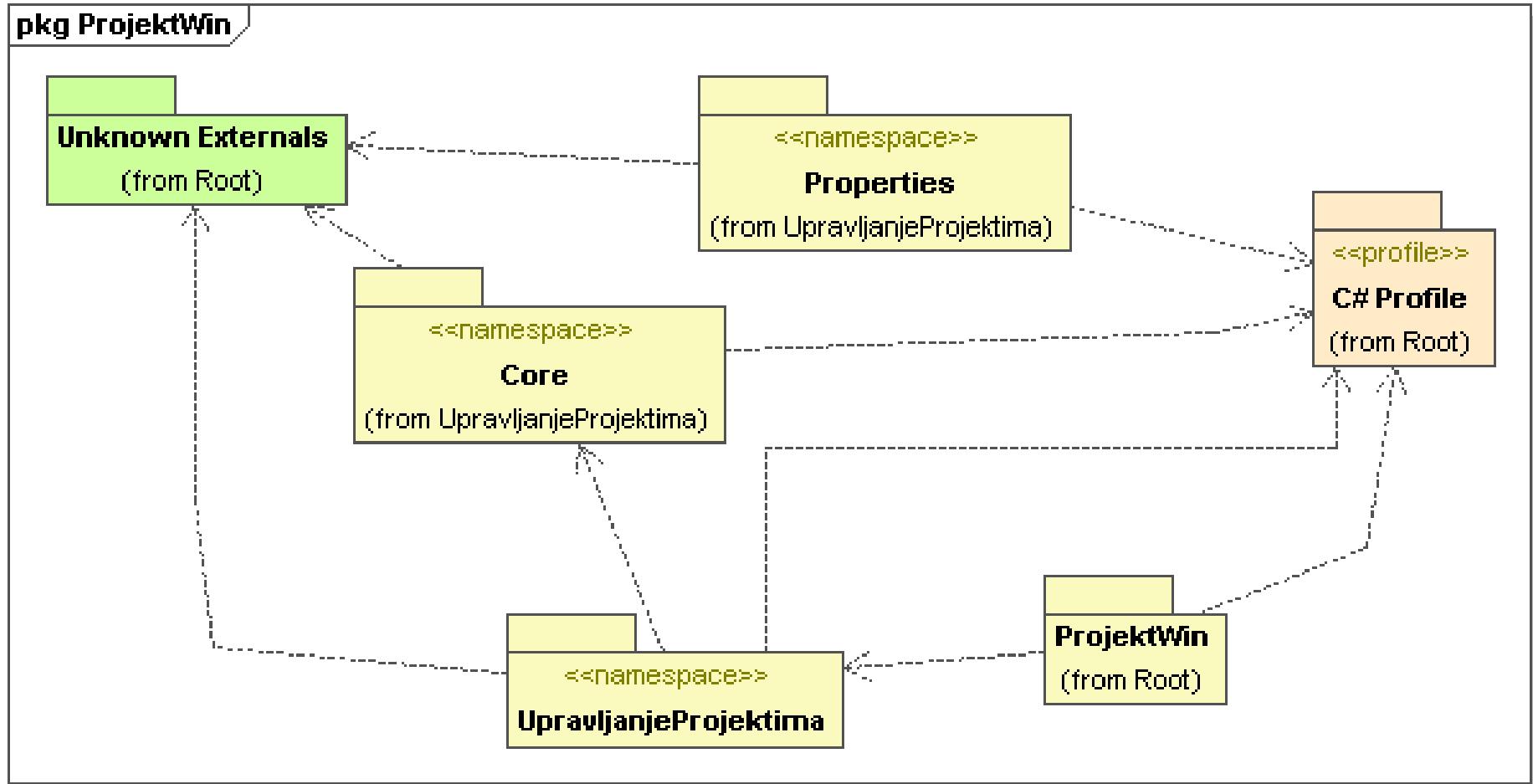
- razredi okvira pripadaju u isti paket
- razredi u zajedničkoj hijerarhiji nasljeđivanja
- razredi povezani agregacijom ili kompozicijom
- razredi koji međusobno često surađuju

□ Smanjenje zavisnosti postiže se

- smanjenjem broja javnih razreda i njihovih operacija
- delegiranjem ponašanja odgovarajućim razredima
- pročeljima (facades) - dodatni javni razredi za komunikaciju s okolinom
 - izdvajanjem javnih podskupova operacija za kolaboraciju
 - ostali razredi paketa postaju privatni (vidljivi unutar istog paketa)

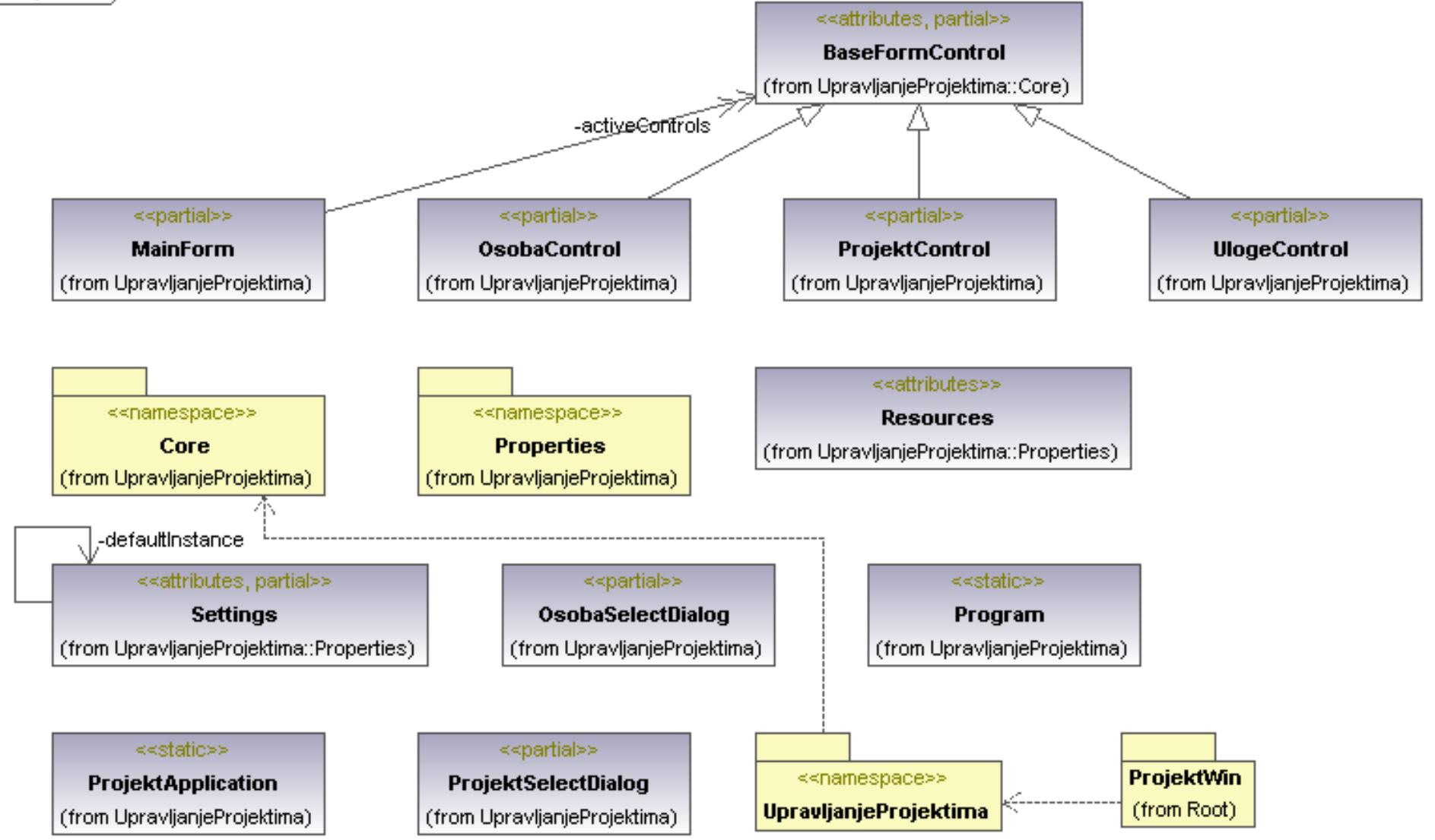
Primjer: Zavisnost paketa *ProjektWin*

- Primjer: ProjektCSLA \ ProjektWin
 - alt.: VS - Architecure – Generate Dependancy Graph



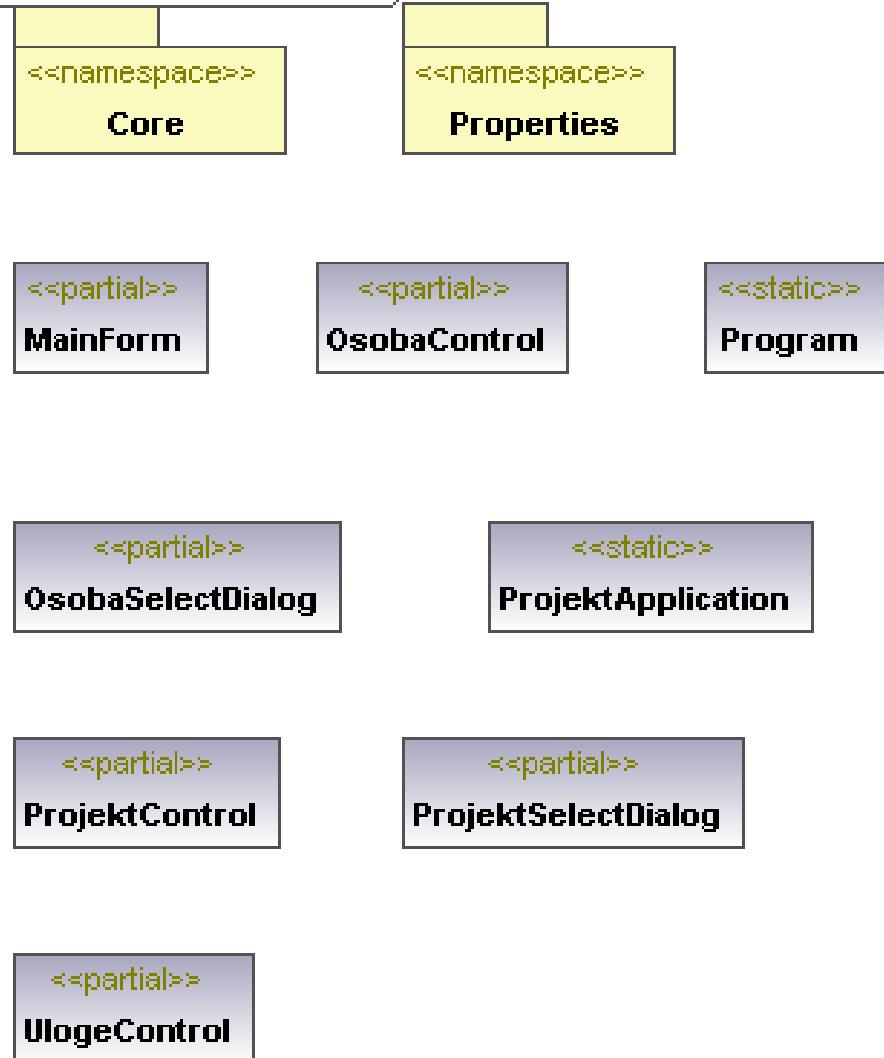
Paket ProjektWin

ProjektWin



Sadržaj paketa *UpravljanjeProjektima*

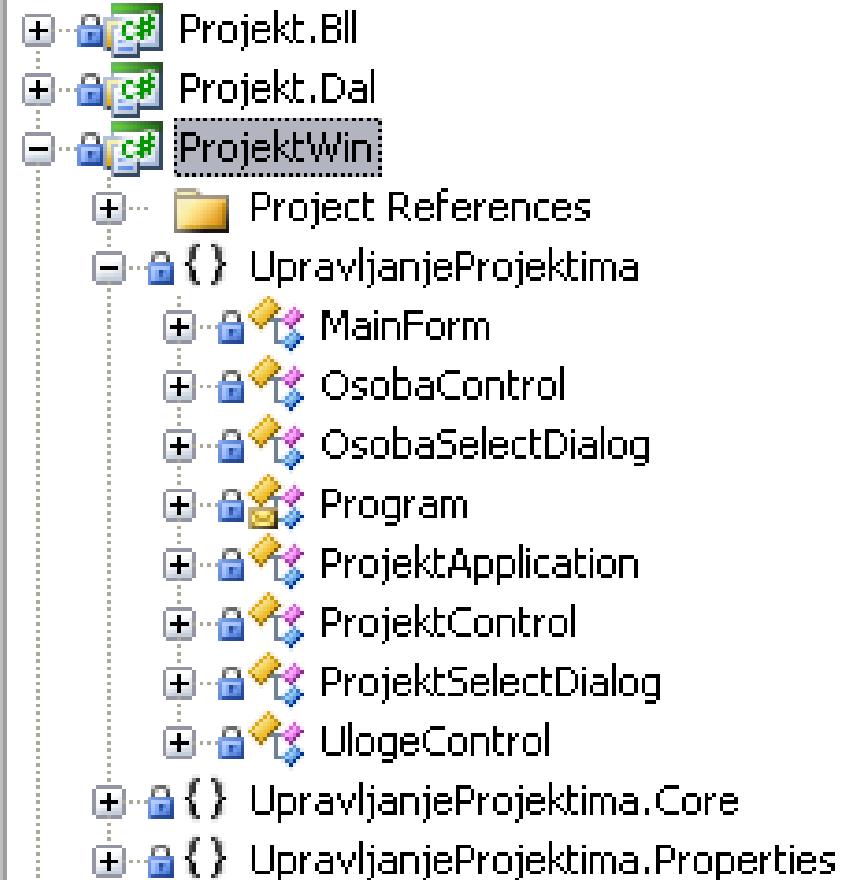
pkg UpravljanjeProjektima



Class View



<Search>



Fizički dijagrami

Physical Diagrams

Dijagrami komponenti

□ Dijagram komponenti (Component diagram)

- prikaz organizacije i zavisnosti softverskih komponenti

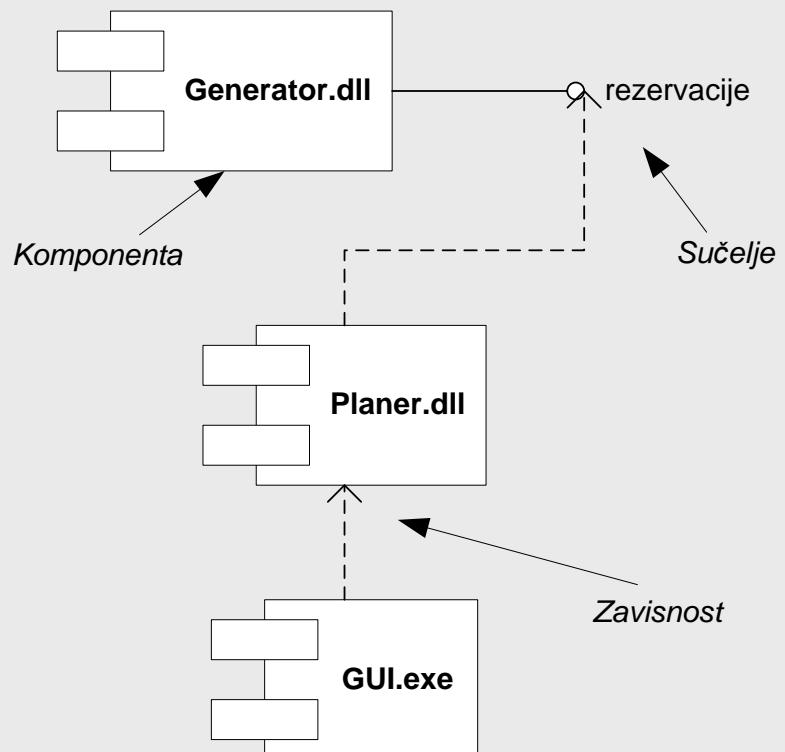
□ Komponenta: fizički modul programskog koda

- izvorna datoteka, pogonska komponenta (run time) ili izvedbeni program (executable)

□ Zavisnost komponenti

- iskazuje na koji način promjena jedne komponente može utjecati na promjenu drugih komponenti
- primjeri: zavisnost pri komunikaciji, zavisnost pri prevođenju

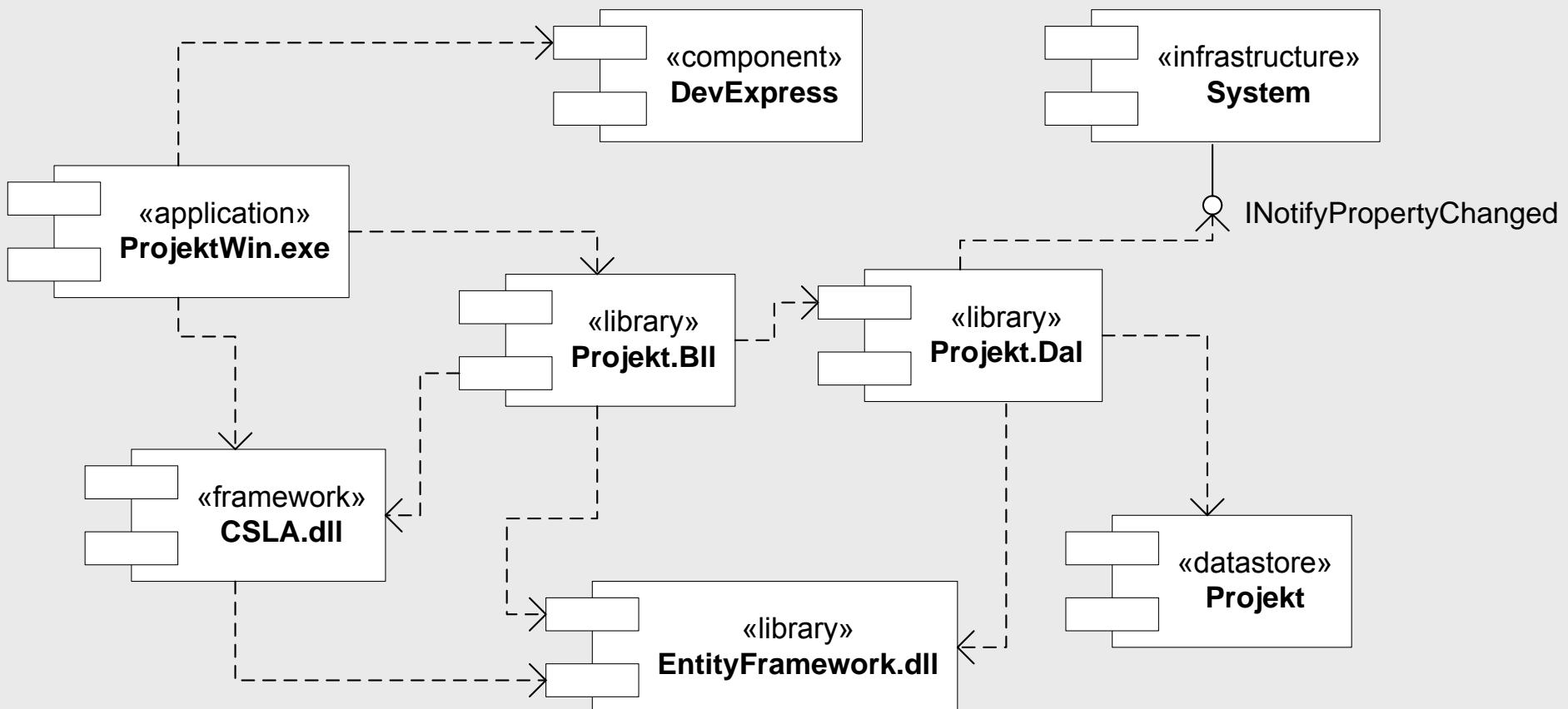
- Naziv tipa komponente je oblika: *tip_komponente*.
- Naziv instance komponente je oblika: *ime_komponente : tip_komponente*.
 - Ako se naziv tipa izostavi izostavlja se i dvotočka.



Primjena dijagrama komponenti

- **Komponenta sliči paketu, ali podliježe drugim pravilima**
 - reprezentira fizičko pakiranje programskog koda
 - pojedini razred može se koristiti u različitim komponentama, ali je definiran u samo jednom paketu
- **Primjena**
 - detaljni dizajn konfiguracije sustava
 - modeliranje tehničke infrastrukture
 - modeliranje poslovne/domenske arhitekture organizacije
- **Označavanje komponenti stereotipovima (pr. Visio – UML – Stereotypes)**
 - <<application>> komponenta prednjeg plana, npr. Web stranice ili Win GUI
 - <<datastore>> trajna pohrana podataka
 - <<document>> papirnat ili elektronički dokument
 - <<executable>> komponenta izvodljiva na fizičkom čvoru
 - <<file>> datoteka (s podacima)
 - <<infrastructure>> tehnička komponenta, npr. servis pohrane ili *logger*
 - <<library>> knjižnica funkcija
 - <<source code>> datoteka izvornog koda
 - <<table>> tablica u bazi podataka
 - <<web service>> jedan ili više web servisa
 - ...

Primjer dijagrama komponenti *ProjektWin*



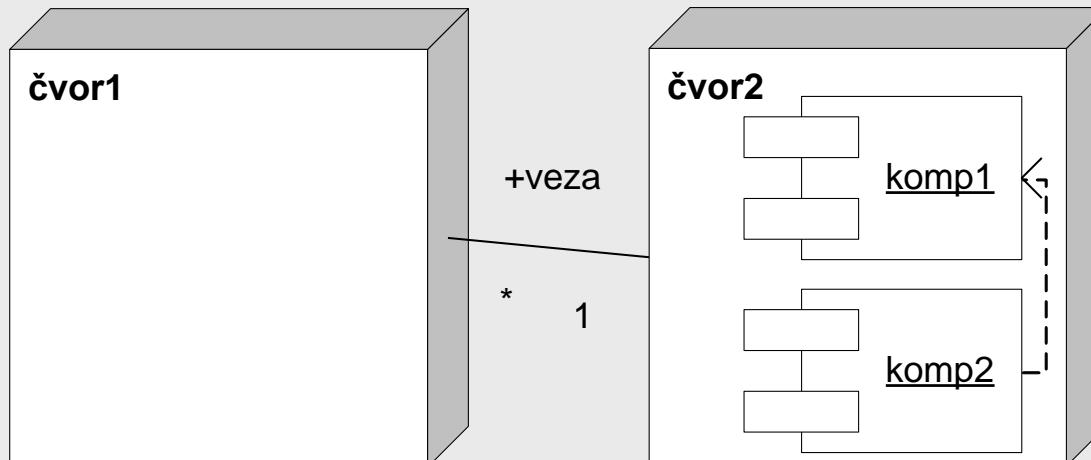
Dijagram ugradnje

□ Dijagram ugradnje (Deployment Diagram)

- prikaz konfiguracije pogonskih elemenata i pripadnih softverskih komponenti
- čvorovi (Nodes): izvršni resursi - najčešće sklopovlje
- spojevi (Connections) - komunikacijski putovi, pr. TCP/IP
- nazivlje: `ime_čvora : tip_čvora` (opcionalno)

□ Kombiniranje dijagrama komponenti i dijagraama ugradnje

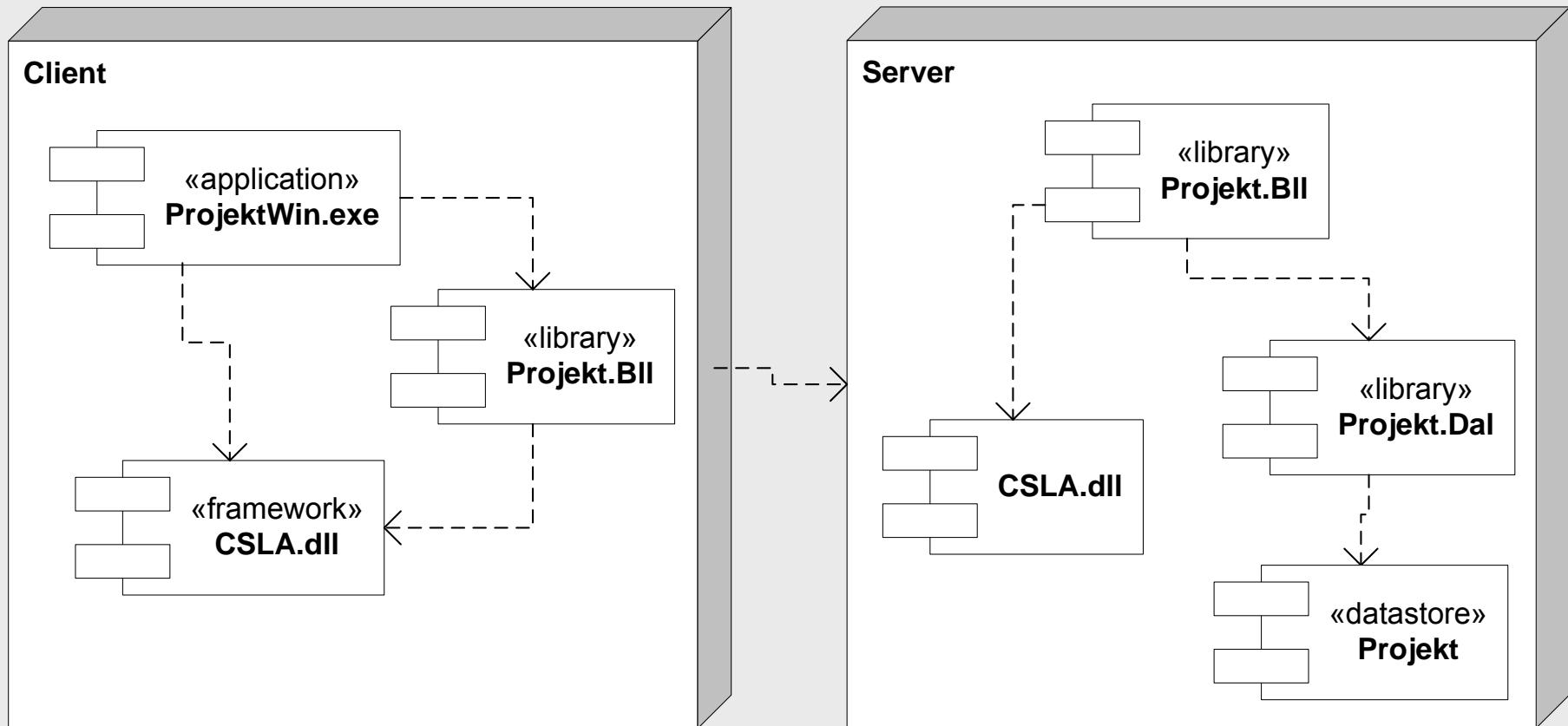
- prikaz fizičke zavisnosti između softvera i hardvera
- lokacije komponenti unutar (distribuiranog) sustava



Primjer dijagrama ugradnje

□ Distribuirani klijent-poslužitelj (tzv. smart client)

- odvajanjem baze *Projekt* na treći poslužitelj BP → troslojno rješenje
- stavljanjem svih komponenti osim BP na klijenta → debeli klijent





Programski primjeri aplikacija s različitim GUI nad istom osnovicom

Primjeri: Prilozi \ RIS07-ProjektCSLA.zip

ProjektWin

ProjektMVC

ProjektWeb

ProjektWPF

Developer Express komponente:

<http://web.zpr.fer.hr/DevExpressComponents-13.2.5.exe>

Reference

Resursi

- Rockford Lhotka: Expert C# 2008 Business Objects, Apress, 2009
- Pro ASP.NET MVC 3 Framework

Aplikacijski okviri i obrasci arhitekture

- CSLA: <http://www.lhotka.net/> , <http://www.cslanet.com/>
- MS EntLib: <http://msdn.microsoft.com/en-us/library/cc467894.aspx>

Web

- <http://www.asp.net/mvc>
- <http://www.asp.net/ajax/>
- <http://www.asp.net/ajaxlibrary/AjaxControlToolkitSampleSite/>

WPF

- http://en.wikipedia.org/wiki/Windows_Presentation_Foundation