

# Osnovni pojmovi

2022/23.01

# 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, Studentska, Zavodi, Financijska služba
- organizacija, interakcija, međuzavisnost, integriranost
  - mogu biti zamijenjeni bez potrebe za promjenom čitavog sustava (!?)

## □ Granica

- definira sadržaj i domašaj sustava

## □ Okolina

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

## □ Ulazi i izlazi

- sve što ulazi u sustav iz okoline ili ga napušta

## □ Sučelja

- veze između sustava i njegove okoline

## □ 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**

□ **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 ...

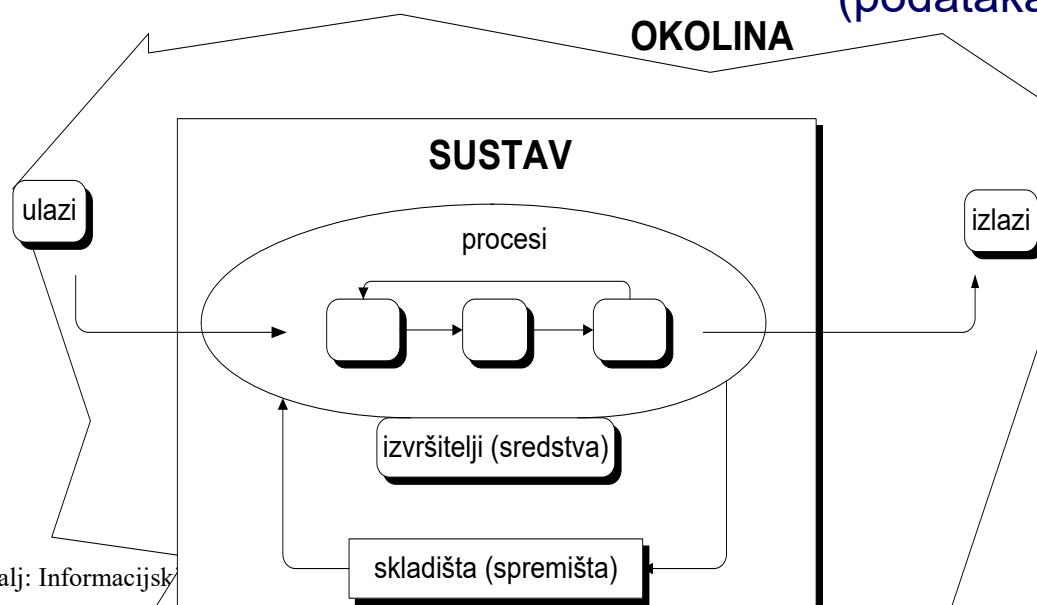
# Poslovni i informacijski sustav

## □ 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 o stanjima stvarnog sustava
- izvršitelji: osobe, programi, računala
- skladišta: spremišta informacija (podataka)



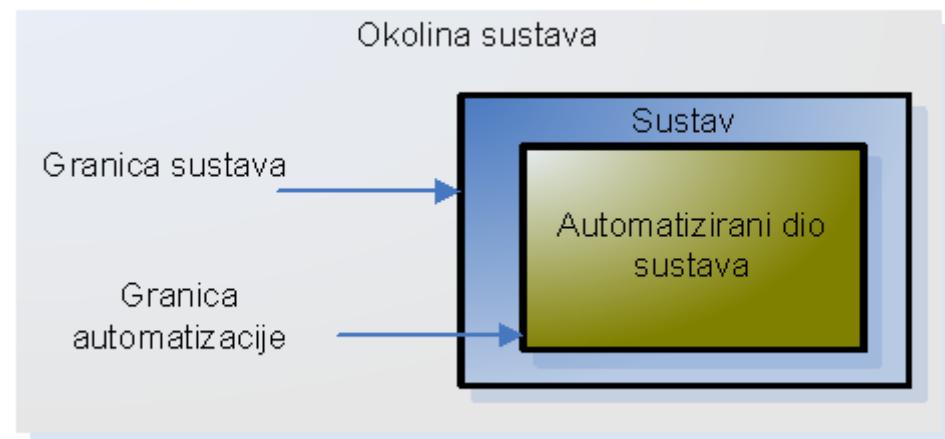
# Informacijski sustav podržan računalom

## □ 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 i razine IS

Volume of Information	Type of Information	Information Level	Management Level	System Support
Low Consensed	Unstructured	 <p>The diagram shows a triangle divided into three horizontal sections. The top section is labeled 'Strategic Information'. The middle section is labeled 'Management Control Information'. The bottom section is labeled 'Operational Information'.</p>	Upper	DSS
Medium Moderately Processed	Moderately Structured		Middle	MIS
Large Detail Reports	Highly Structured		Lower	DPS

[https://www.tutorialspoint.com/system\\_analysis\\_and\\_design/system\\_analysis\\_and\\_design\\_overview.htm](https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm)

# Ostale vrste IS

## □ Expert System (ES) – ekspertni sustav

- sustav pravila i mehanizam zaključivanja
- ugrađena stručna znanja za korištenje od strane manje stručnih korisnika

## □ Office Automation System (OAS), Office Support System (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

# Primjeri IS

## □ IS biljnih vrsta

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

## □ IS Subvencionirane Prehrane (ISSP)

- Centar za autorizaciju prava
- Restoran
- Visoko učilište

## □ IS personalnog upravljanja

- Osnovni podaci o osobama
- Ustroj
- Tijek službe
- Dodjela činova i redova, ocjenjivanje
- Dodjela odličja i medalja
- Školovanje
- Evidencija prekršaja, mjere
- Ostalo (naknade, doplatci, ...)

## □ ERP

- Financijsko računovodstvo
- Upravljačko računovodstvo (Kontroling)
- Prodaja i distribucija
- Nabava, skladištenje, logistika
- Upravljanje ljudskim potencijalom
- Proizvodnja
- Izvještaji

# 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 1994

- 16.2% uspješnih projekata
- 52.7% prekoračenje roka/troška ili reducirana funkcionalnost
- 31.1% prekinutih projekata

## □ Čuveni neuspjesi

- London Ambulance System (1992)
- Taurus – London Stock (1993)
- Denver Airport (1994)

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

# 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 razvojna pomagala i krivo korištenje, pa čak i svojevrsna zloporaba 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
- Upravljanje projektom (planiranje, izvedba, praćenje, kontrola)
- 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 oklijevati ako treba revidirati doseg ili otkazati projekt
- Projektirati za rast i promjene

# Programsko inženjerstvo

- **Programska potpora = softver (software)**
  - programi + podaci + dokumentacija
- **Primjenjena 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]
- **Računalom podržani informacijski sustav = sustav aplikacija za upravljanje ljudskim aktivnostima**

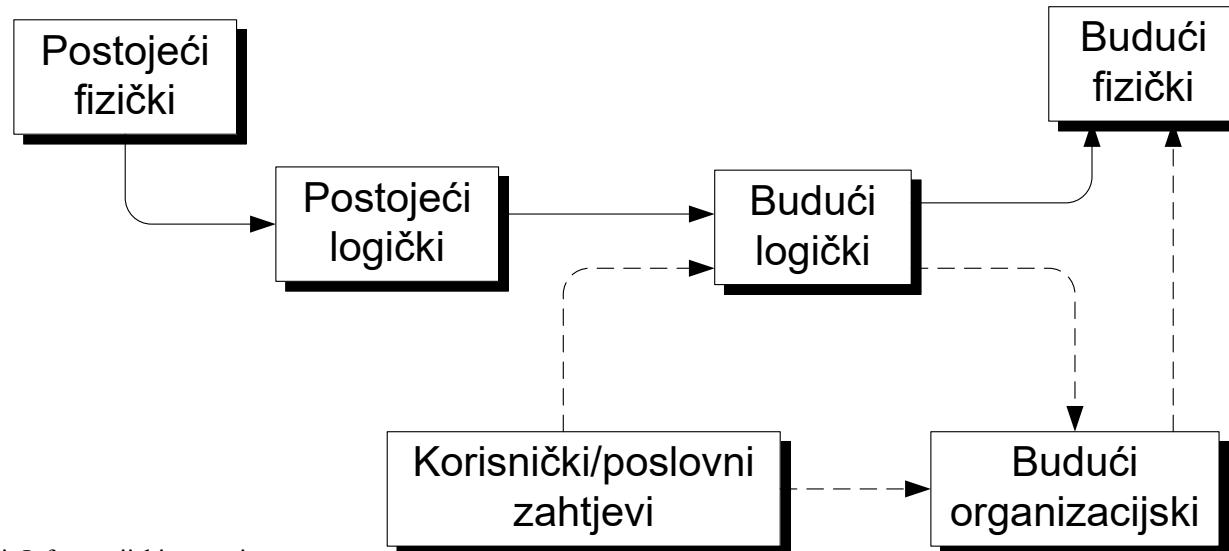
# Informacijsko inženjerstvo

- **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)**
  - smišljen 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

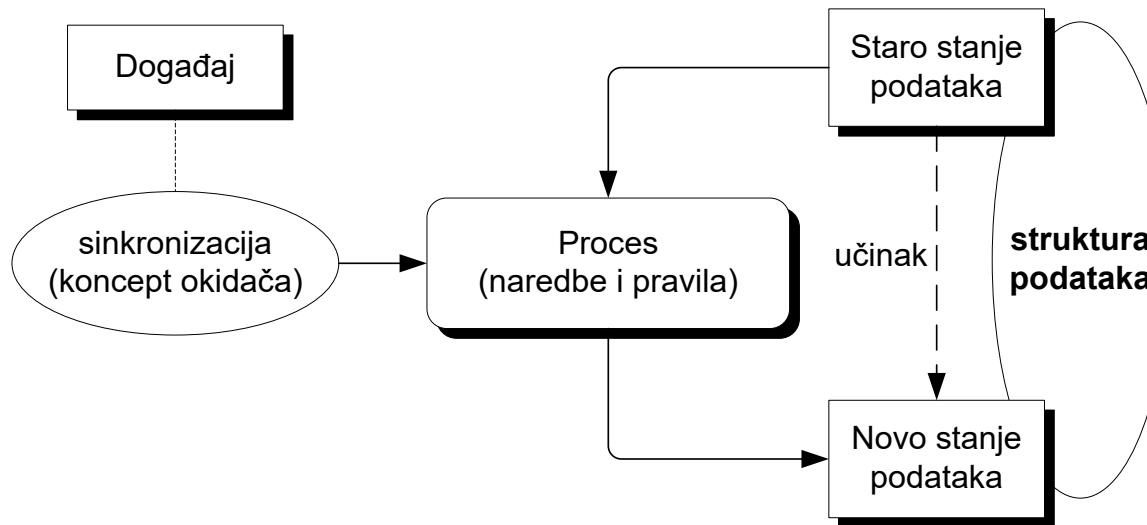
- fizički model (ugradbeni, implementacijski, tehnički)
  - kako je sustav fizički i tehnički ugrađen, te tko, gdje i kada nešto radi
- logički model (esencijalni, konceptualni, poslovni)
  - što je sustav, što radi, što su podaci
- operativni (budući fizički) sustav
  - što, tko i kada, ali ne i gdje
- 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 vlastiti sustav otprije ne postoji (!?) i/ili treba napraviti potpuno novi?
- "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

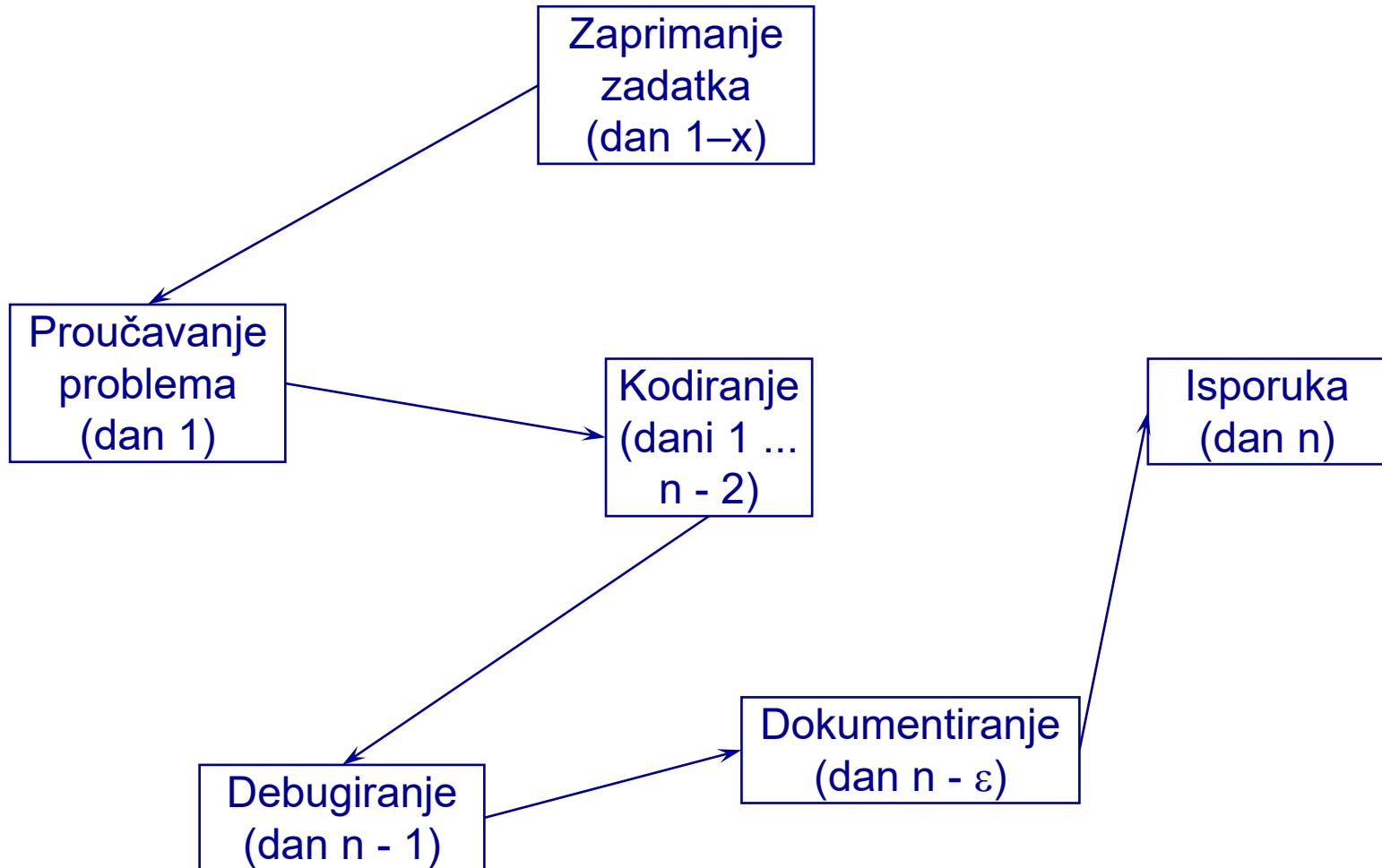
- planiranje, upravljanje ekipom (analitičara, razvojnika, korisnika), nadzor, ...

## □ 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

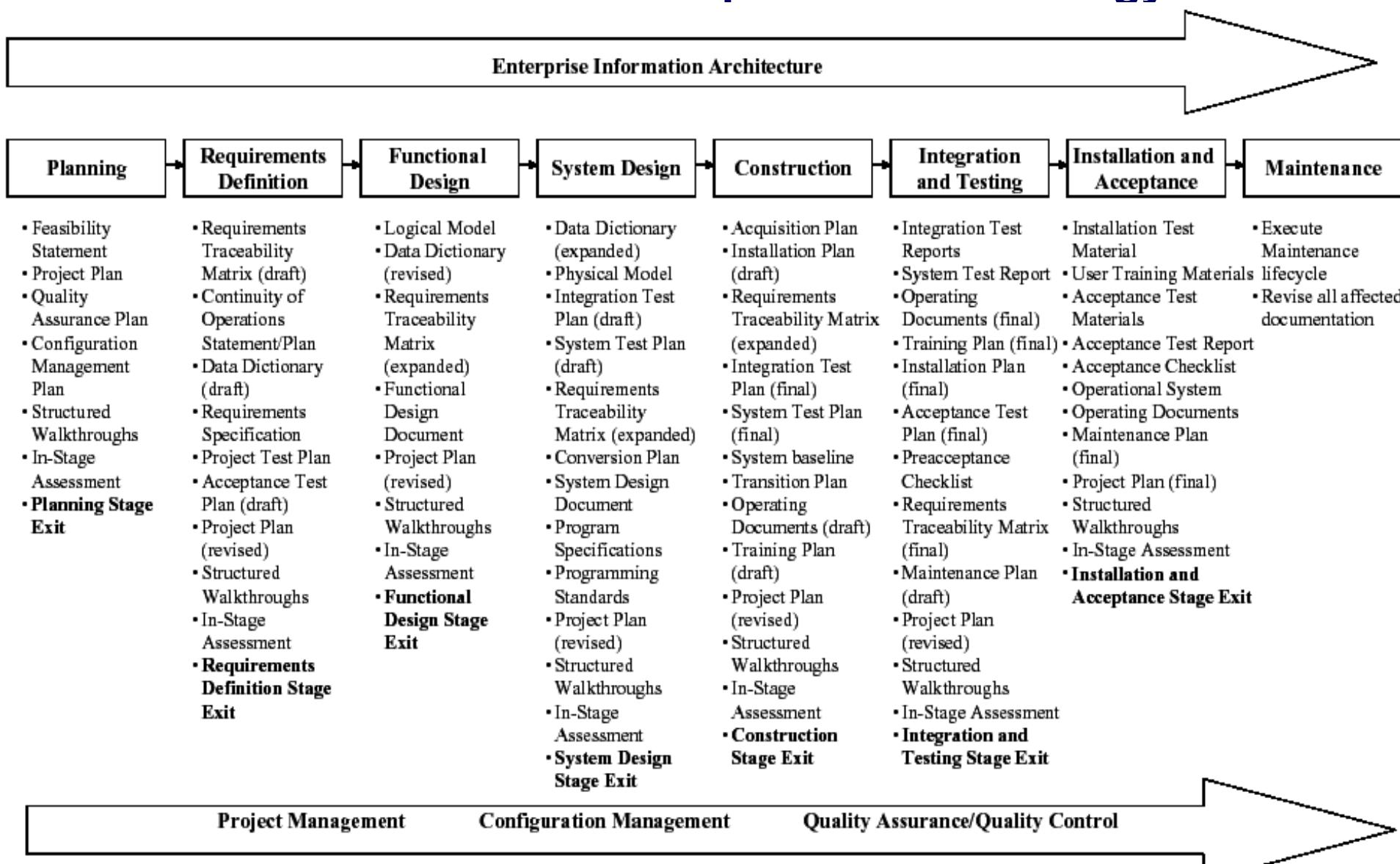
## □ Pregled (review)

- 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

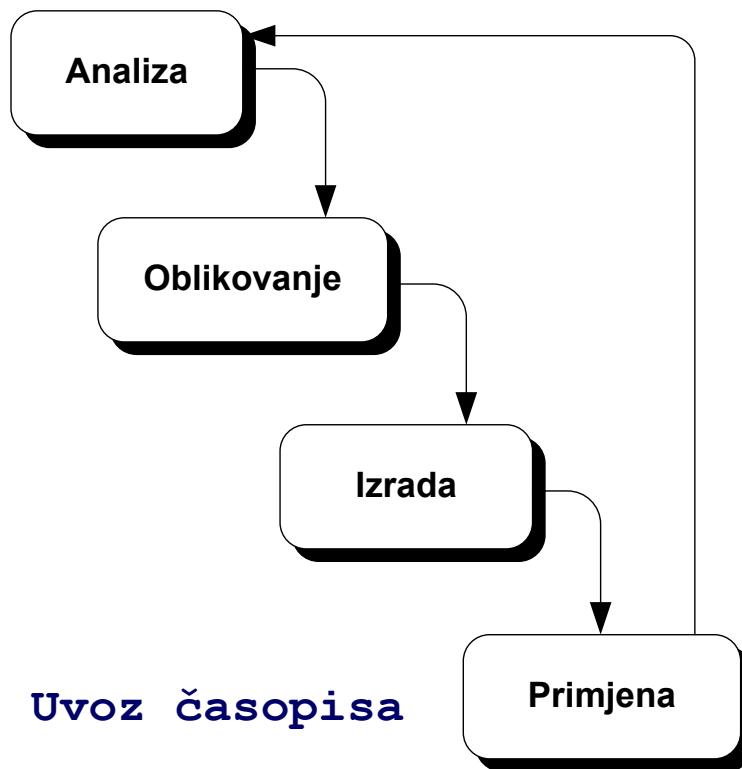


# **Modeli razvoja**

# 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

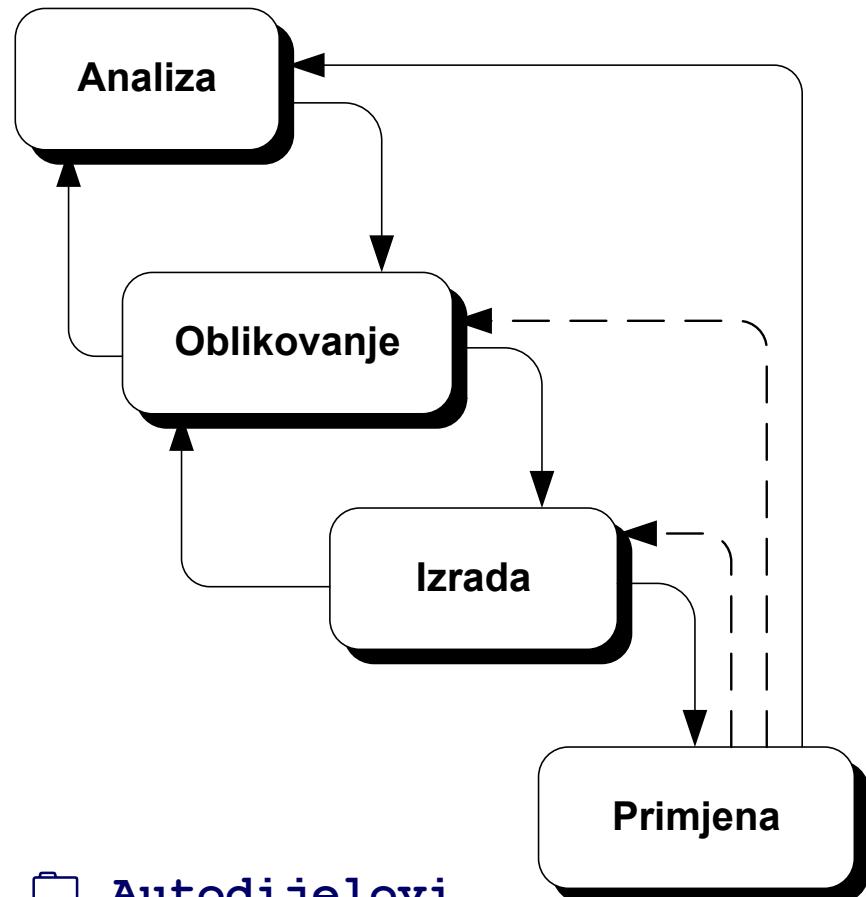
# Pseudostruktturni i radikalni vodopadni model

## □ Pseudostruktturni model

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

## □ Strukturni (radikalni)

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

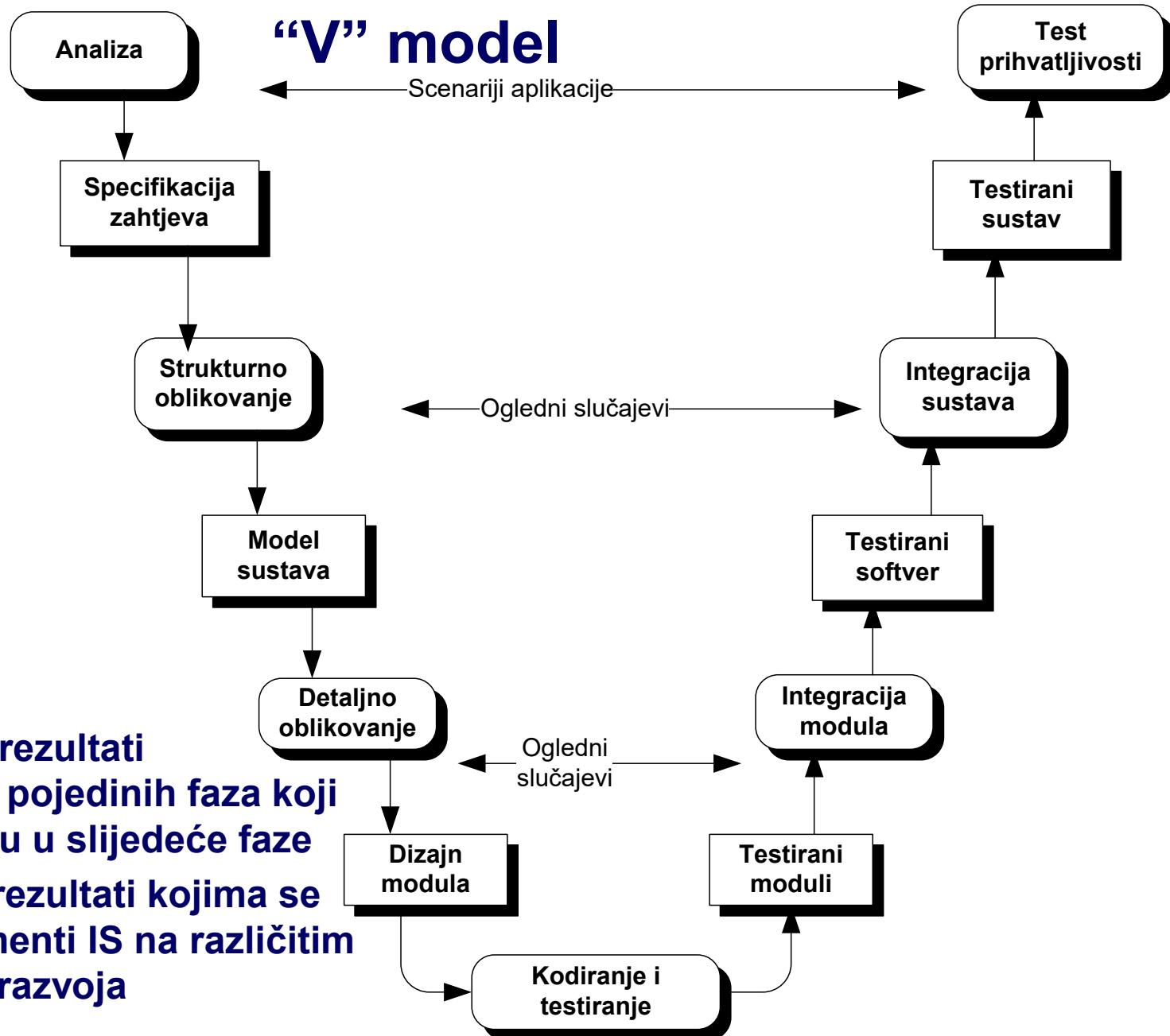


□ Autodijelovi

**Validacija**

**Verifikacija**

- definiraju se rezultati ("proizvodi") pojedinih faza koji se proslijeđuju 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

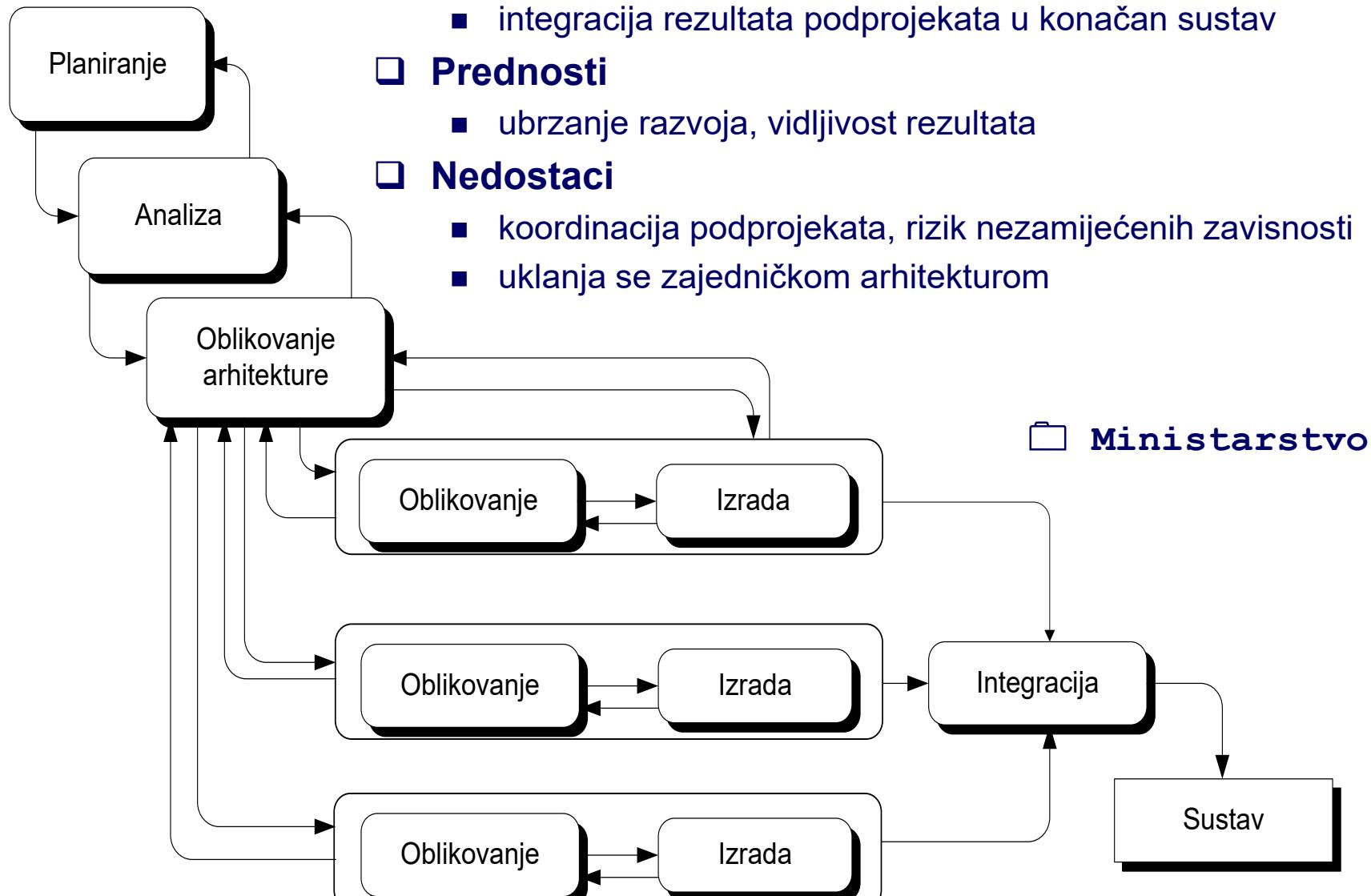
- integracija rezultata podprojekata u konačan sustav

## 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)

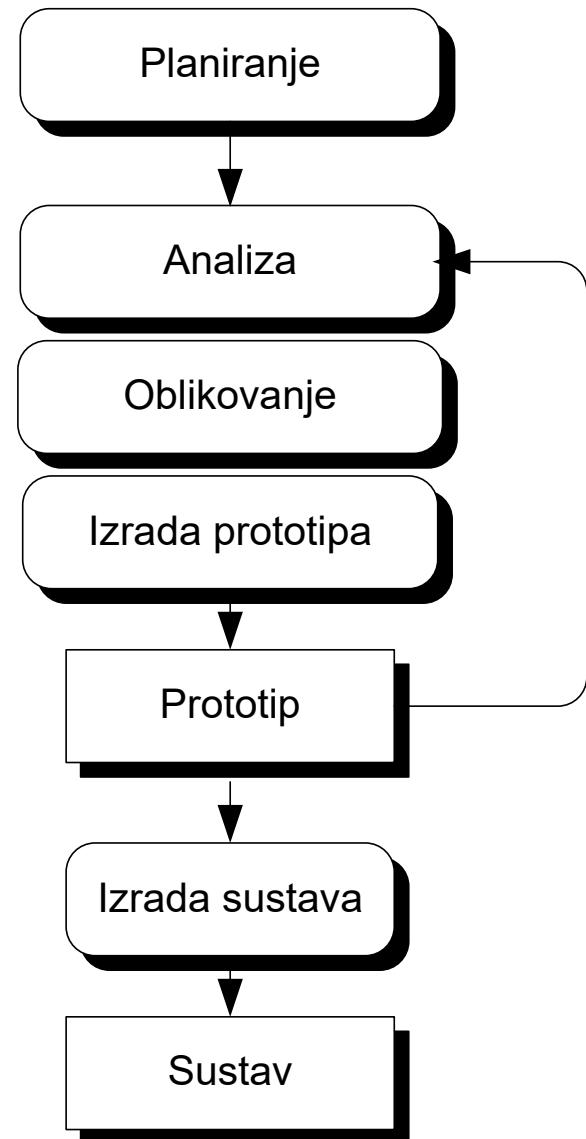
- funkcionalni prototip
- povratna informacija korisnika
- inkrementalna dorada (stepwise refinement)
- na kraju samostalna izrada (bez korisnika)
- mali (*one-man*) projekti s promjenjivim zahtjevima

## □ 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

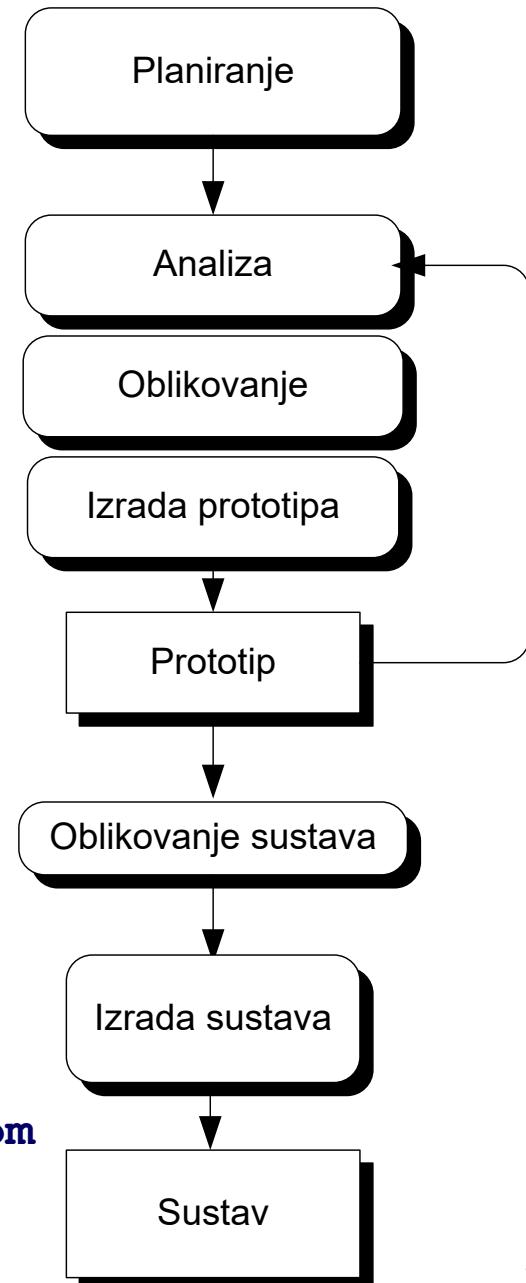
- smanjuje rizik neispunjениh očekivanja
- po određivanju zahtjeva prototip se odbacuje (throwaway prototyping)
- slijedi faza oblikovanja

## ❑ Prednosti

- postoji dokumentacija
- moguće je planiranje

## ❑ Nedostatak

- "bacanje" posla

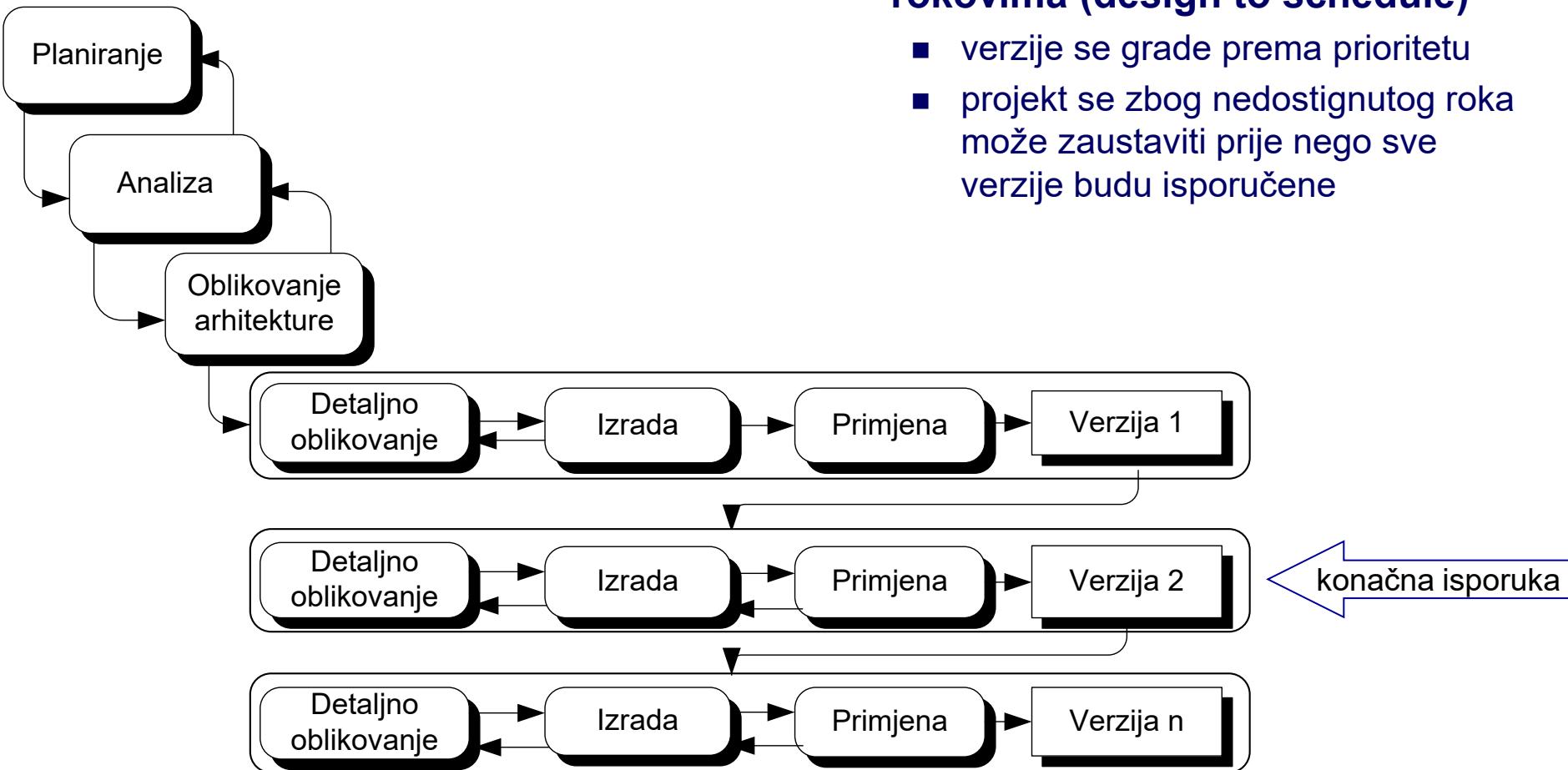


Izdavačka kuća, Telekom

# 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

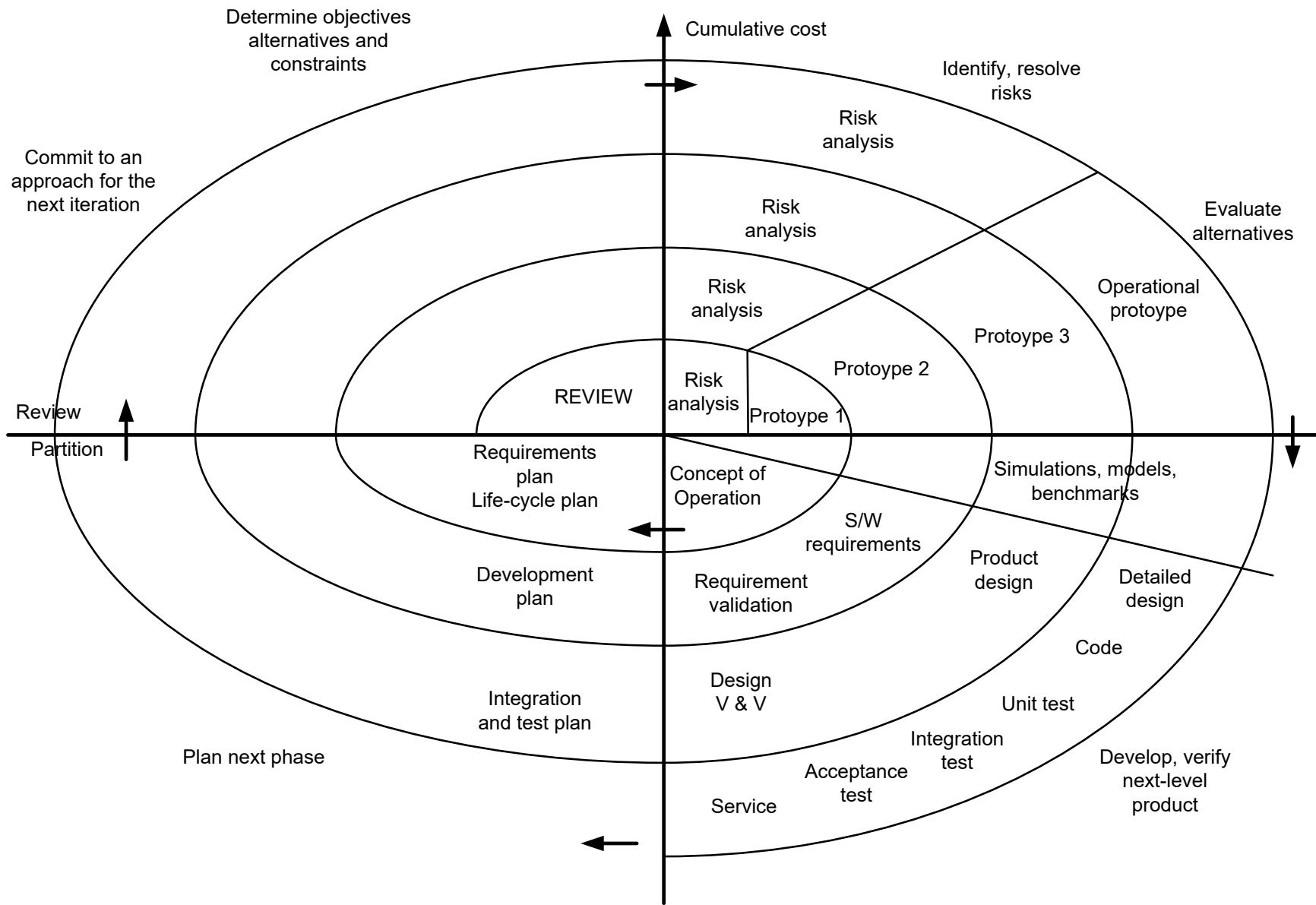


❑ Evidencija biljnih vrsta

# 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

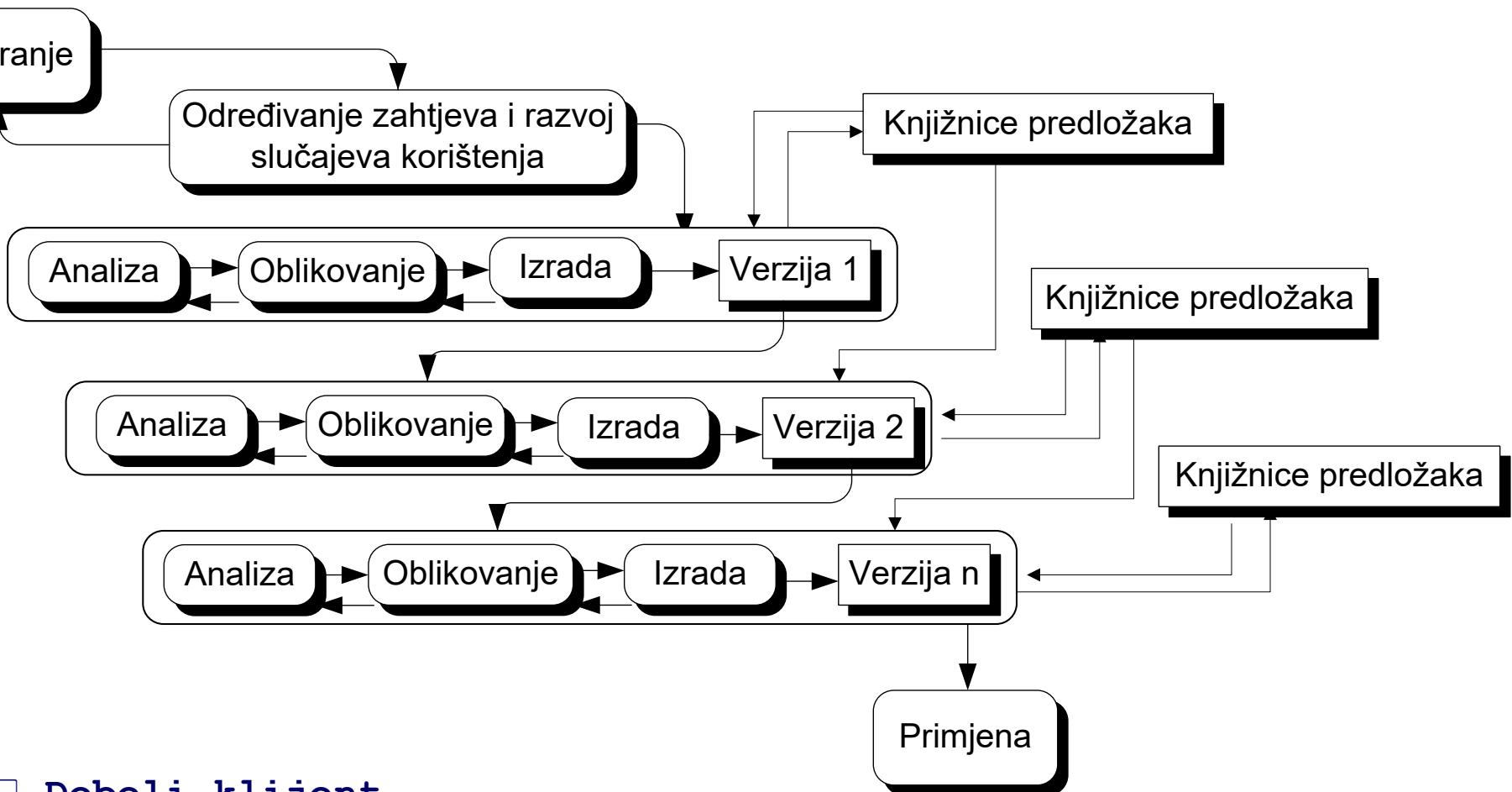
## □ Minimalist Object-Oriented Systems Analysis & Design (MOOSAD)

- 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)
  - intervali do 2 tjedna ili 1 mjesec ovisno o veličini i složenosti projekta

# MOOSAD pristup



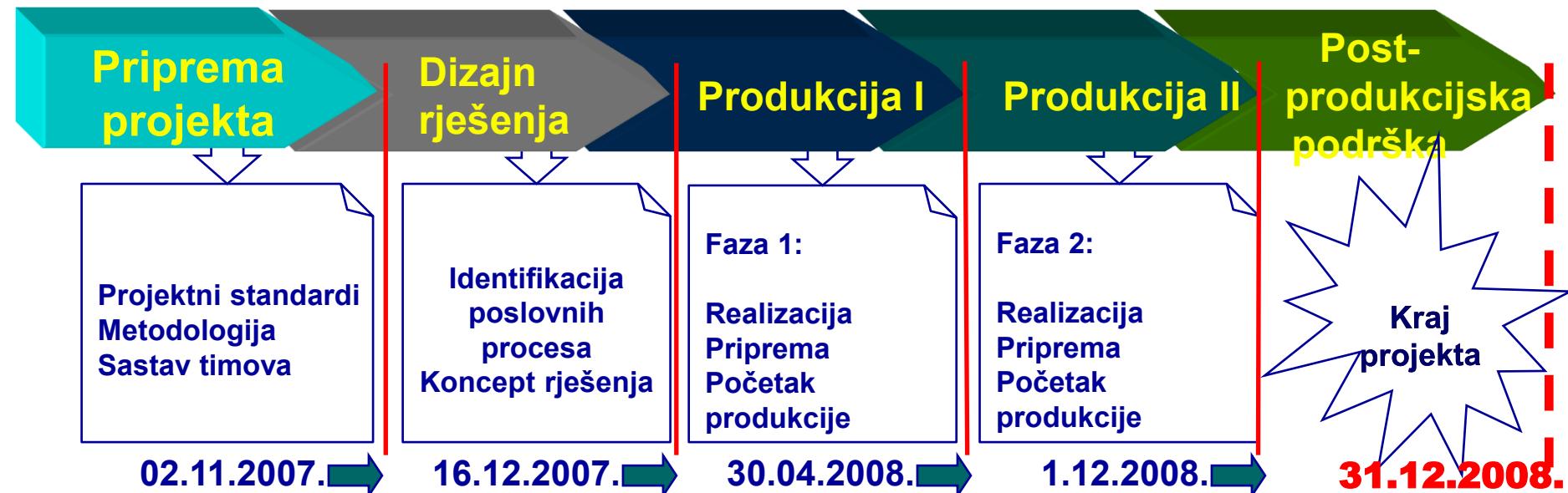
**Debeli klijent**

**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

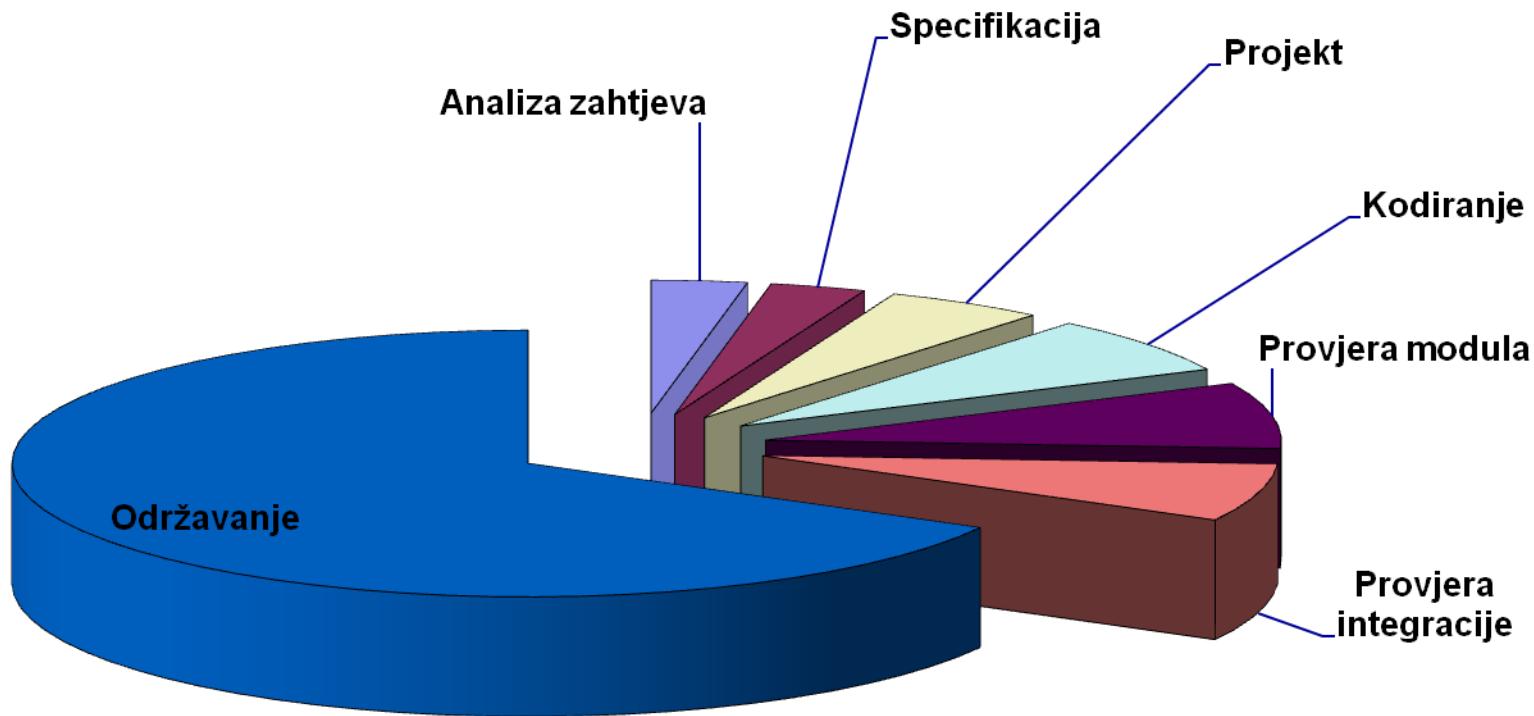




## Diskusija

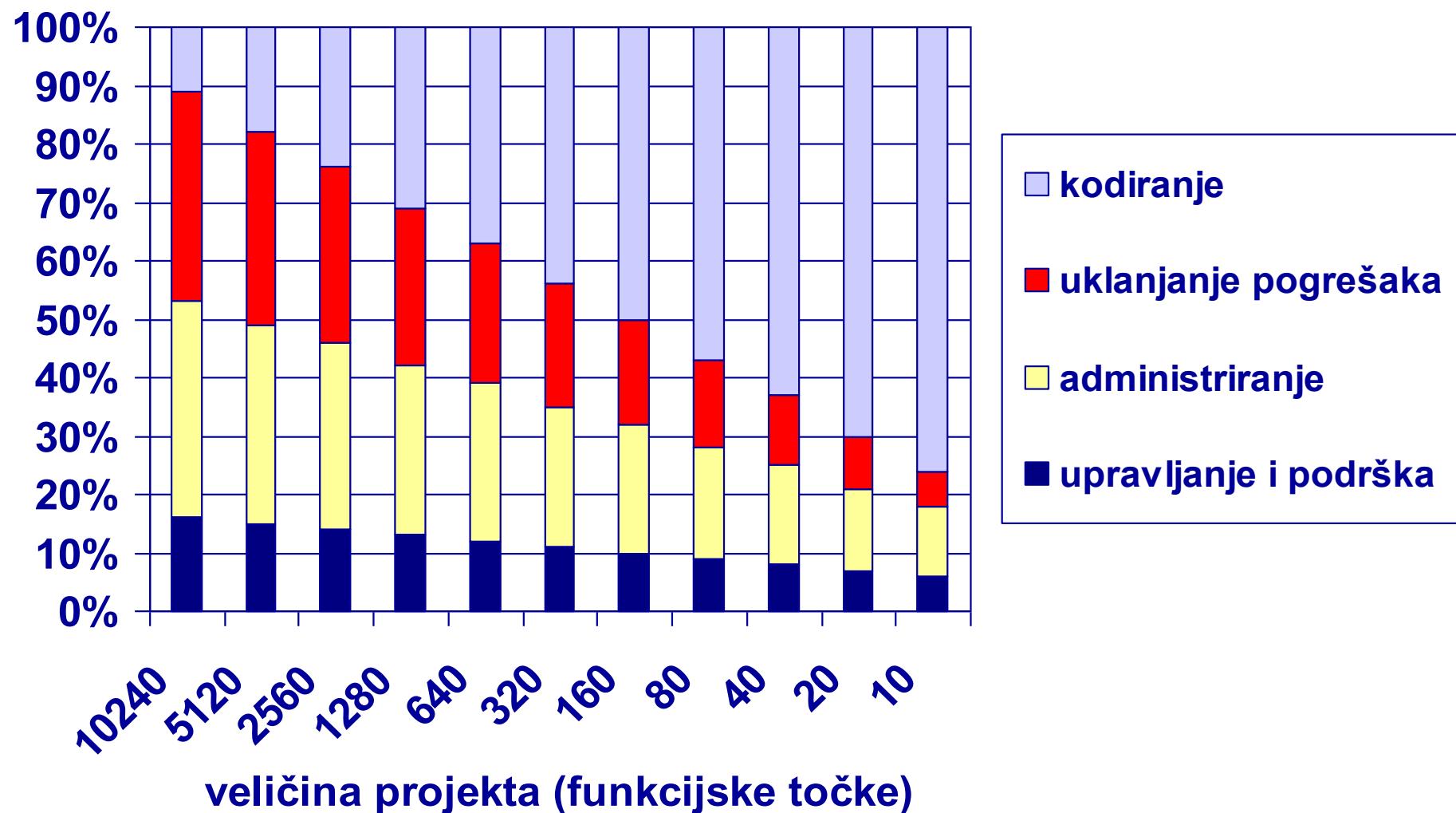
**Koji ćete model razvoja odabratи za svoj projekt i  
zašto?**

# Razdioba troškova u životnom ciklusu softvera



- Analiza:** 3%
- Specifikacija:** 3%
- Projektiranje:** 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://aisnet.org/>
- Kontrolne liste i predlošci projektne dokumentacije <http://www.construx.com>
- Prilagodljivi razvojni proces (Adaptable Process Model) <http://www.rspa.com>

## □ Literatura

- Modern Systems Analysis and Design. Joseph S. Valacich, Joey F. George, Pearson, 2017.
- Systems Analysis and Design - An Object Oriented Approach with UML. Alan Dennis, Barbara H. Wixom, David Tegarden, Wiley 2015.

# **Strategija organizacije**

## **Identifikacija i selekcija projekata**

## **Pokretanje i planiranje projekata**

**2022/23.02**

# 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, financijska, 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: FER misija, vizija, strategija 2019.-2023.

- [https://www.fer.unizg.hr/o\\_fakultetu/misija\\_i\\_vizija](https://www.fer.unizg.hr/o_fakultetu/misija_i_vizija)

# 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 (CROSBI, 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

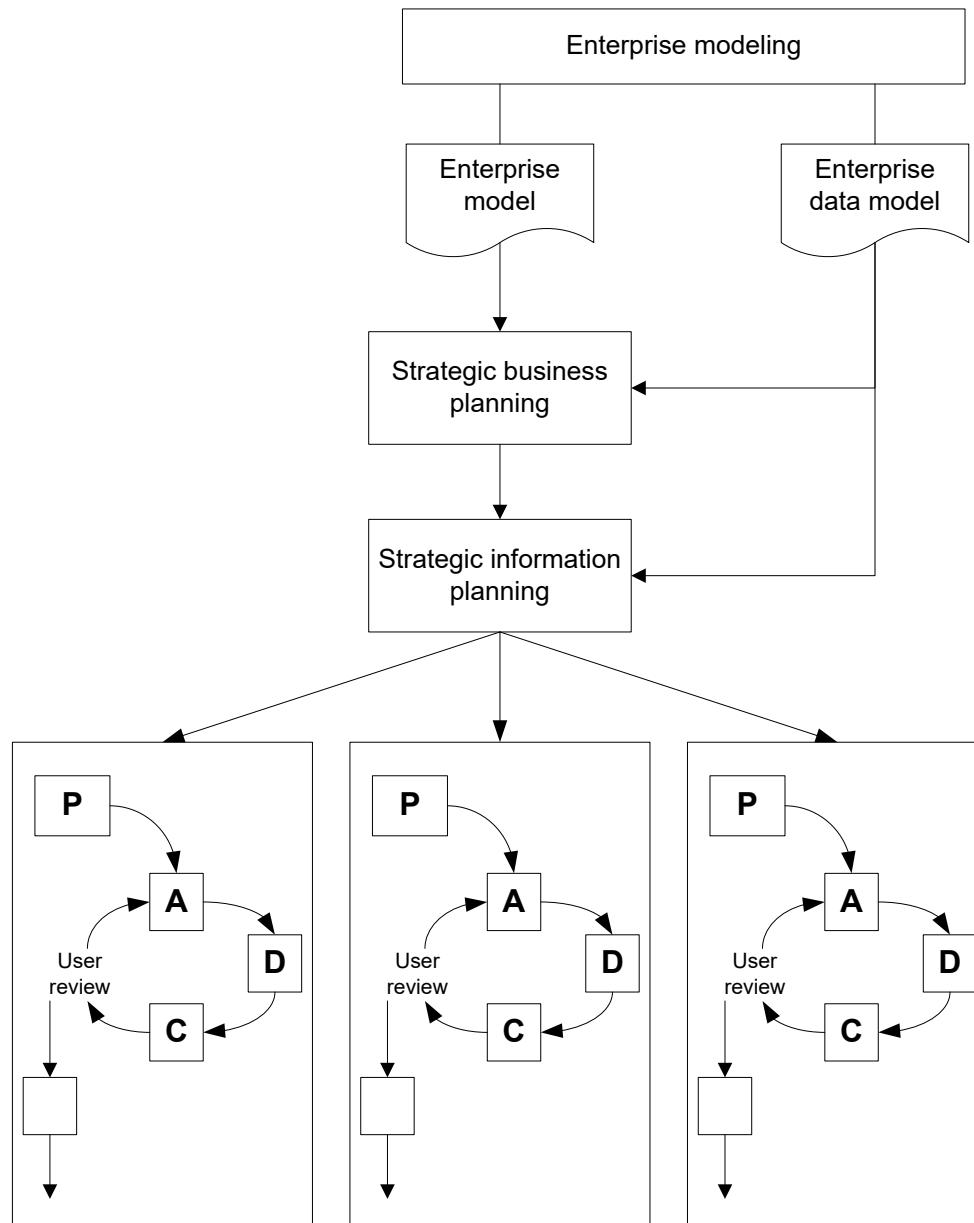
- planiranje IS sukladno strategiji razvoja organizacije
  - informatizacija kao potpora promjeni organizacije i poslovnih procesa
- *istraživanje poslovnog sustava s ciljem definiranja općeg (sveobuhvatnog) plana i arhitekture IS čiji razvoj slijedi*

## □ U praksi ...

- organizacija se "dovodi u red" tijekom informatizacije i s pomoću nje
- analiza poslovnog sustava – evidencija problema i slabosti poslovnih procesa
- dizajn informacijskog 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	F
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	Sistemski poslovi	800 days	Mon 17.4.06	Fri 8.5.09				
94	Osiguranje IT sustava na razini cijelokupne infrastrukture	17 days	Mon 19.6.06	Tue 11.7.06				
108	Izgradnja sigurnosnih politika/procedura/puta uskladenih	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				

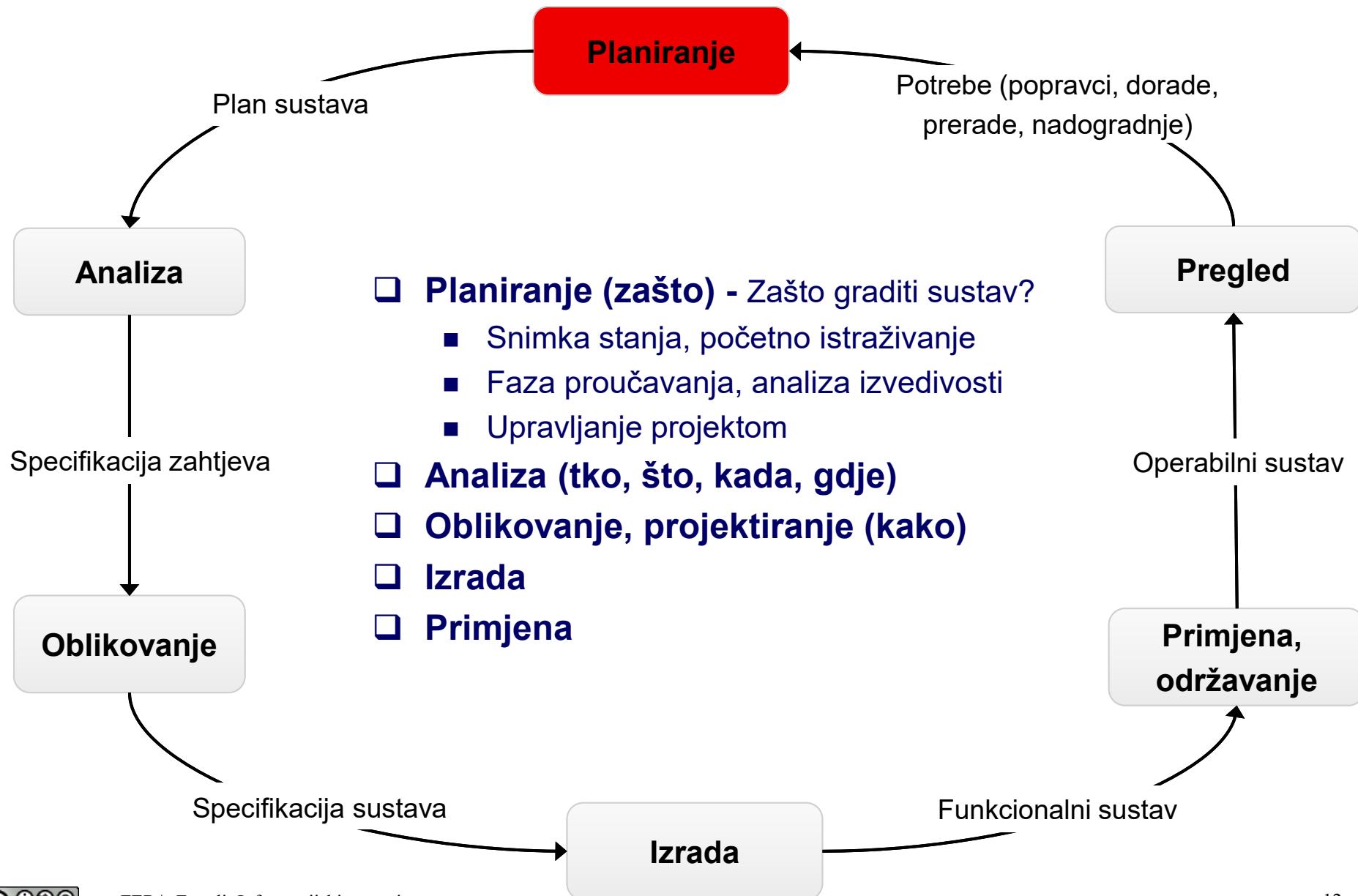
# Primjer strategije: Poduzeće (3)

## □ Ažurirani plan pripreme 2009.

ID		Task Name	Duration	Start	Finish	Qtr 1, Jan
1		<b>Faza pripreme</b>	<b>80 days</b>	<b>Mon 26.1.09</b>	<b>Fri 22.5.09</b>	
2		Uspostava infrastrukture projekta (TFS, SPs)	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



# 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, ... (npr. "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://hamagbicro.hr/bespovratne-potpore/programi-podrske-inovacijskom-procesu/ircro/>

## □ Odabir projekta (project selection)

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

# Zahtjev na sustav unutar organizacije

- **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

# Primjer: prijava na natječaj javne nabave

Narodne novine d.d. [HR] <https://eojn.nn.hr/SPIN/APPLICATION/IPN>

Aplikacije FER ZPR New CALLS eStudij ASP RIS Druge oznake

## Objava - Obavijest o dodatnim informacijama, poništenju ili ispravku

HTML PDF

Naručitelj HRVATSKA REGULATORNA AGENCIJA ZA MREŽNE DJELATNOSTI  
Oznaka/broj 2013/S 014-0014104  
Naziv Nabava programskih aplikacija iz natječaja programa potpora - faza 1  
Vrsta dokumenta\* Obavij Ovoj objavi pripadaju i slijedeće objave:  
Vrsta ugovora Isporu  
CPV 48900  
Vrsta postupka Otvore  
Proc.vrijed. 5.000.  
Rok za dostavu ponuda/zahtjeva za sudjelovanje 14.3.2  
Datum objave 18.2.2  
Datum slanja 15.2.2  
Zakon ZOJN  
Elektronička dostava ponuda Ne

Vrsta dokumenta Datum objave

Obavijest o sklopljenim ugovorima	14.01.2014.
Obavijest o sklopljenim ugovorima	07.08.2013.
Obavijest o dodatnim informacijama, poništenju ili ispravku	26.02.2013.
Poziv na nadmetanje	24.01.2013.

ZIP dokument. ZIP

### Dokumentacija za nadmetanje

25.2.2013 15:53:53

PDF Verzije IZ-GP-ND-PRO-nabava programskih aplikacija iz natječaja pr...

28.2.2013 17:15:30

ZIP Verzije PRIVITAK I-TEHNICKE SPECIFIKACIJE-20130228.zip

# Primjer: prijava projekta na natječaj fonda

**HAMAG BICRO** Hrvatska agencija za malo gospodarstvo,  
inovacije i investicije

Dobro došli [REDACTED] Odjava

Projekti Otvoreni natječaji [REDACTED]  

## Obrazac za prijavu za znanstvenike i istraživače

Šifra projekta: [REDACTED]

 Prikaži obrazac pretprijave  Komunikacija s Prepoznatim centrom

- ▶ A. INFORMACIJE O NATJECATELJU
- ▶ B. INFORMACIJE O PROJEKTU
- ▶ C. INOVATIVNOST
- ▶ D. TRŽIŠNI POTENCIJAL
- ▶ E. METODOLOGIJA IZVEDBE I OČEKIVANI UČINCI PROJEKTA
- ▶ F. PROJEKTNI TIM
- ▶ G. PRORAČUN I PROVEDBENI PLAN
- ▶ H. PROCJENA RIZIKA I IDUĆI KORACI
- ▶ I. PRETHODNA SURADNJA S BICRO-om, HIT-om, UKF-om, HAMAG-Investom i/ili HAMAG-BICRO-om
- ▶ J. IZJAVA NATJECATELJA
- ▶ K. POPIS DOKUMENATA

# 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 (npr. pravilnik) ili vanjski utjecaj (npr. 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\PISFER-Snimka-\*, jedan krug intervjeta, 30 SA, 1mj
- Primjer: Projekt „Marine”, 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 (iz <i>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 stanja
- 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

- Može teći usporedno i temeljem snimke stanja i dublje analize
- Započinjanje projekta (project initiation)
  - 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

# 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 ...*

## □ Ciljevi IS, moraju biti „pametni”

- **Specific** – određen
- **Measurable** – mjerljivi indikatori napretka
- **Action Oriented, Asignable** – usmjeren radu, ima nositelja
- **Realistic** – što se stvarno može napraviti s raspoloživim resursima
- **Time Oriented** – odredivo trajanje

# 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

- Gospodarstvo: Projekt se mora obaviti bez nabavke novog hardvera
- Znanost: Trošak opreme poželjno predstavlja barem XX% budžeta !

## □ Financijska sredstva

- Ukupni budžet projekta je XX HRK (uvijek manji od traženog!)
- Naeurade izvođačima ne smiju premašiti XX% ukupnog iznosa (teži 0!)

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

# Primjeri strategije, izvedivosti i plana

- Analiza postojećeg stanja i prijedlog konceptualnog rješenja za sustav CroTMCS, HKZP, 2022.
- Strategija informatizacije, Hrvatske autoceste d.o.o., 2019.
- Snimka stanja informacijskog sustava ACI, ACI, 2014.
- ...
- 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.



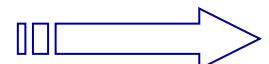
# Proučavanje problema, analiza izvedivosti (detaljnije)

# 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



- Analiza uzroka i posljedica (Root Cause Analysis)
- Analiza kritičnih faktora uspjeha (CSF - Critical Success Factors)
- Procjena troškova-koristi (Cost-Benefit Analysis, CBA)
- ...

# Analiza 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 ... ?

## □ Diskusija: pisarnica u koju je uvedeno novo programsko rješenje

- Prepolovljen broj djelatnica
- Ciklus obrade: 24h
- Dokumenti nastali u sustavu se urudžbiraju automatski, ostali ručno
  - Dokument se skenira
  - Lijepi se naljepnica s klasom i urudžbenim brojem
- Ostalo: skeniranje priloga, urudžbiranje vezanih dokumenata, ...

# Ključni faktori uspjeha

- čimbenici kojima naručitelj/korisnik posvećuje posebnu pažnju
- čimbenici koji razmjerno brzo i lako doprinose ostvarivanju ciljeva

## □ Primjer: Urudžbeni zapisnik

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

## □ Primjer: CIP

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

## □ 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?

» koncept PIECES

- **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

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

## □ Koji je stav korisnika prema rješenju? Hoće li se sustav koristiti?

- Podrška uprave – stvarna, praktična
- 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)

- 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 (80 kriterija, 5 alata)

# Vremenska izvedivost

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

## **Prihvatljivost rasporeda - opravdanost rokova s obzirom na stručnost, ograničenja,**

...

## **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 se 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 dosegнутом funkcionalnoшćу
6. Ponavljaju se koraci 3. do 5. s nadodanom funkcionalnoшćу

## **Koliko je razuman predloženi plan?**

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

# Postupak procjenjivanja

## 1. Procjena veličine projekta (metrika projekta)

- funkcije točke (function point – FP) i programski retci (LOC)
- slučajevi korištenja (UC point)
- objektne točke (object point)
- analogija s prethodnim projektima

## 2. Procjena potrebnog napora u osoba-mjesecima (om)

- 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 **radnim** mjesecima ili godinama
- npr. procijenjeni rad 36 om / 6 članova ekipe = 6 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 funkcijskim i UC točkama

## □ Funkcijske točke (FP)

1. Procjena broja ulaza, izlaza i datoteka
  - primjer: 1000 FP
  - uz 30 LOC / FP za C++
  - zahtjeva izradu 30000 LOC
2. Procjena napora iz produktivnosti
  - produktivnost za poslovni sustav: 500 LOC/om
  - napor =  $30000 / 500 = 60$  om

## □ Točke slučajeva korištenja (UC points)

1. Procjena broja slučajeva korištenja
  - primjer: 300 UCP
2. Procjena napora iz produktivnosti
  - produktivnost 5 od/UCP = 4 UCP/om
  - napor =  $300 / 4 = 75$  om

## □ U praksi se broj FP i UC množi

- faktorom težine - mali, srednji, veliki
- stupnjem utjecaja / faktorom tehničke složenosti - prenosivost, lakoća, ...

# Procjena objektnim točkama

## ❑ Objektne točke - aplikacijske točke, ne programski razredi!

1. Procjena broja zaslonskih maski, izvješća i modula za pristup podacima
  - primjer: 500 OT
2. Procjena napora iz produktivnosti
  - produktivnost: 25 OT/mj
  - napor =  $500 / 25 = 20$  om

## ❑ U praksi produktivnost varira

- 4-50 OT/mjesec, ovisno o alatu i vještini (statistike nema)
- Preciznije: baždarenje za različite veličine
- Primjer:
  - 2 h / šifrarnik,
  - 4 h / osnovna tablica
  - 8 h / master-detail,
  - 40 h / korisnička priča

# 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 projektu na planiranje utrošeno 40m, može se očekivati Analiza ( $4/15\% * 20\% = 5.33\text{om}$ ), Dizajn (9.33om), Izrada (8om)

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

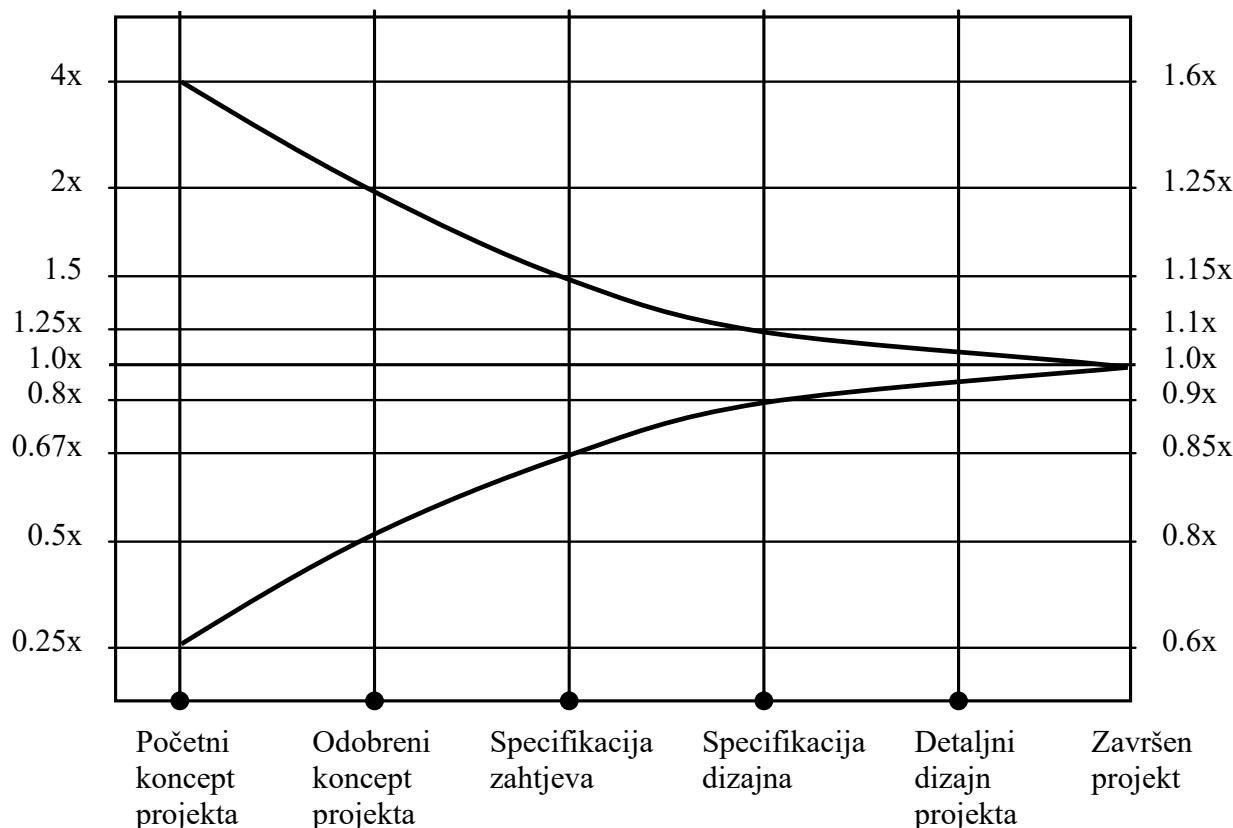
- Povijesni podaci: vrste projekata, evidencija utrošenog vremena, veličina i složenost baze podataka (broj tablica, broj veza, složenost ključeva), broj zaslonskih maski i izvješća
- Ako je projekt 80% sličan nekom drugom projektu, ...

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

# (ne)preciznost procjene – razlog za analizu i dizajn

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

Troškovi projekta  
(veličina i potrebni rad)



# Procjena trajanja projekta (3. korak)

- Iskustveno pravilo za optimalnu procjenu trajanja temeljem npora:
  - optimalno trajanje u mjesecima =  $3 * \sqrt[3]{om}$
- Primjer: procjena projekta koji zahtijeva napor 65 om,
  - optimalno trajanje =  $3.0 \cdot 65^{1/3} \approx 12$  mjeseci
  - slijedi optimalna veličina tima  $65/12 \approx 5\text{-}6$  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
  - **što može 1 programer za mjesec dana to mogu 2 za dva mjeseca!**
- I zato ... procjena ekonomiske izvedivosti
- Prije toga diskusija:
  - osoba-mjesec ?= osoba-dan ?= osoba-sat
  - koliko dana ima osoba-godina?



# Ekonomска извештавост

» Isplati li se graditi? Kada ће 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

	Troškovi	Koristi
<b>Mjerljivi (tangible)</b>  zna se točan iznos ili iznos može biti procijenjen	plaće, režije, licence ...	izražene kao godišnja ušteda ili ušteda po proizvedenom predmetu
<b>Nemjerljivi (intangible)</b>  postojanje ili "vrijednost" ne može se egzaktno mjeriti ili dokazati	pad morala, (nemjerljivi) pad produktivnosti ili gubitak tržišta	popravljaju dojam, npr. poboljšano zadovoljstvo kupaca, zaposlenika, ...

# Primjeri troškova i koristi

## □ Vrijednost novog kupca (\$ - novčana jedinica u bilo kojoj valuti)

- 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, 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  $4,5 \text{ min} * \$50/60 \text{ po min} = \$3,75 \text{ po zadatku}$

## □ Nemjerljive koristi nastojimo izraziti paušalnom procjenom iznosa

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

# Primjer: troškovi razvoja

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

<b>Edukacija:</b>			
Vrsta	Količina	Cijena	
„in-house“ poduke za programere	3 dana	7,000 eur	
„in-house“ poduka za korisnike	3 dana	10,000 eur	

<b>Materijal:</b>			
Vrsta	Količina	Cijena	
Kopiranje		500 eur	
Diskovi, trake, papir		650 eur	

<b>Sklopoljje i programska podrška:</b>			
Vrsta	Količina	Cijena	
Licence		5,000 eur	
Memorija za X klijenata		8,000 eur	
Periferni uređaji za Y klijenata		2,500 eur	
Mrežni programi		15,000 eur	
Office alati		20,000 eur	

# Primjer: godišnji troškovi rada

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

<b>Nadogradnja sklopoljia i programske podrške:</b>		
Vrsta	Količina	Cijena
Sklopolje		5,000 eur
Programska podrška		6,000 eur
Ostali troškovi		3,500 eur

## Komentar:

- Viđenje naručitelja
- Viđenje dobavljača
- Vlastiti razvoj



# Sadašnja vrijednost troškova i koristi

## □ Sadašnja vrijednost (Present value - PV)

- Današnja vrijednost onoga što će postati  $V$  nakon ' $n$ ' godina u budućnosti, ako uzmemo u obzir kamate (*interest*) ' $I$ ' iznosi:

$$PV = V / (1 + I)^n = V * (1 + I)^{-n}$$

- Razlika predstavlja kamatu koja se može zaraditi tim novcem

## □ Primjeri:

- troškovi razvoja od \$100.000 imaju sadašnju 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 = (ukupna korist – ukupni troškovi) preračunati na današnji dan

## □ Primjer: Koji je projekt isplativiji ?

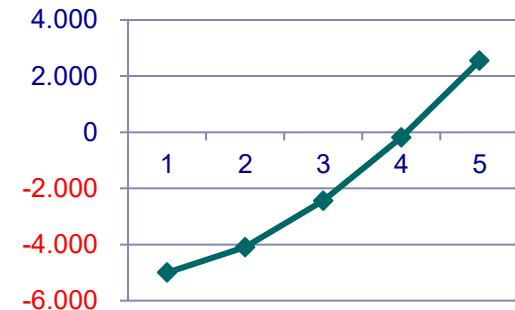
- Komentirati 0. i 1. godinu

$$NPV = \sum_{j=1}^n \frac{values_j}{(1 + rate)^j}$$

Projekt 1	0	1	2	3	4	Ukupno	NPV
Trošak	-5.000	-1.000	-1.000	-1.000	-1.000	-9.000	-8,170
Korist	0	2.000	3.000	4.000	5.000	14.000	10,718
Korist-trošak	-5.000	1.000	2.000	3.000	4.000	5.000	2,548
Projekt 2	0	1	2	3	4	Ukupno	
Trošak	-2.000	-2.000	-2.000	-2.000	-2.000	-10.000	-8,340
Korist	1.000	2.000	4.000	4.000	4.000	15.000	11,861
Korist-trošak	-1.000	0	2.000	2.000	2.000	5.000	3,522

# Povrat investicije

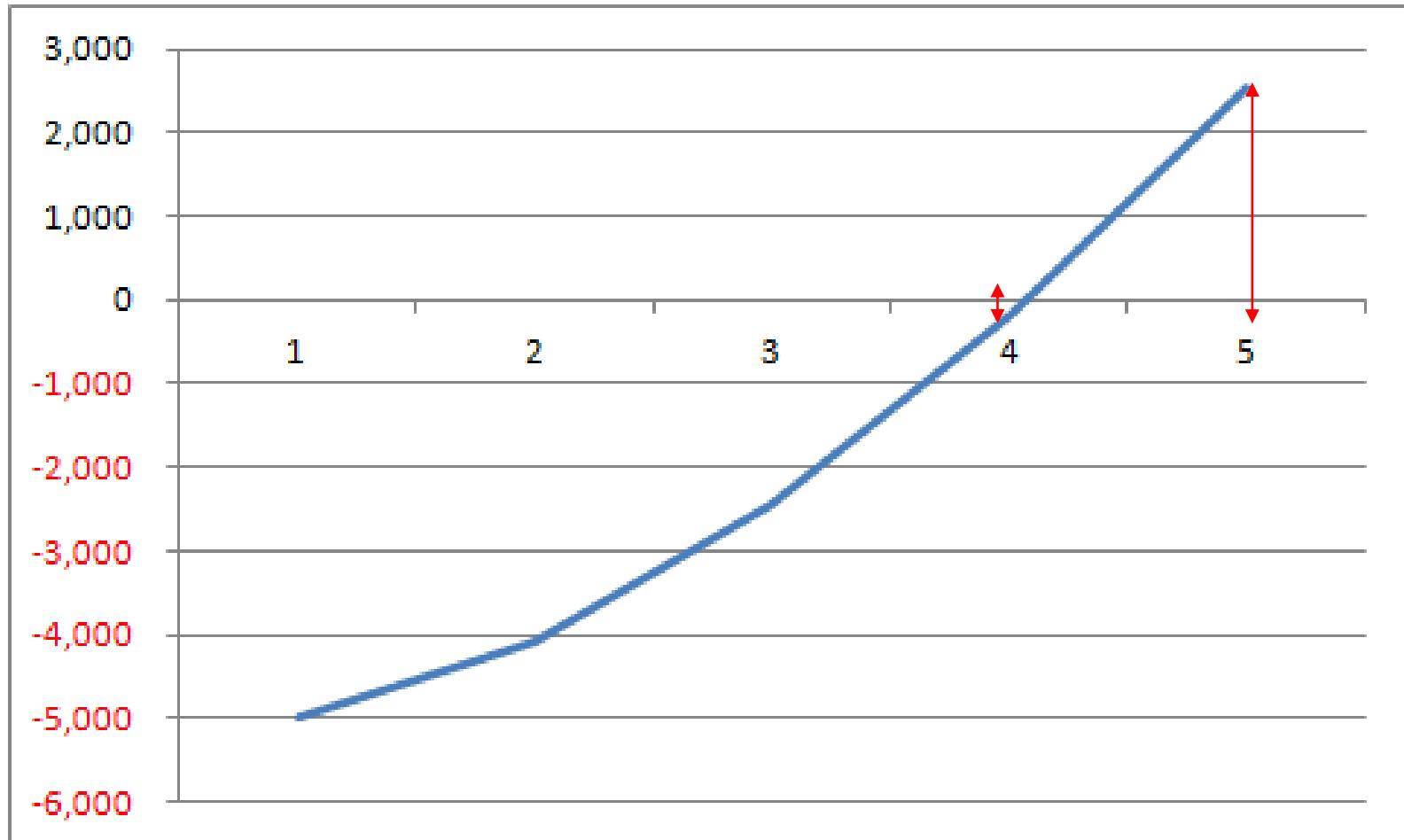
- **Povrat investicije (Return On Investment – ROI)**
  - Ulaganja donose korist koja s vremenom postaje sve veća
  - U jednom trenutku prihod dosegne rashod
- **Indeks profitabilnosti (Benefit to Cost Ratio - BCR)**
  - omjer sadašnje vrijednosti koristi i sadašnje vrijednosti troškova (**PVB / PVC**)
  - za prethodni primjer  $10.718/8.170 \sim 1,31$ , odnosno  $11.861/8.340 \sim 1,42$
  - 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
- **Interna stopa rentabilnosti - Internal Rate of Return (IRR)**
  - Kamatna stopa pri kojoj je  $NPV = 0$



# Primjer: Vrijeme povrata investicije

## □ Vrijeme povrata za Projekt1

$$= 4 \text{ god} + 184 / (184 + 2548) \sim 4.06 \text{ godina}$$



# Povrat investicije

## □ Postotak povrata investicije

- postotak relativne koristi projekta u odnosu na trošak
- $ROI\% = (\text{ukupna korist} - \text{ukupan trošak}) / \text{ukupan trošak}$
- preračunato u sadašnju vrijednost,  $ROI = NPV / PVC = (PVB-PVC) / PVC$
- Projekt1,  $ROI\% = 2.548 / 8.170 = 31,19\%$
- Projekt2,  $ROI\% = 3.522 / 8.340 = 42,23\%$

## □ Faktor obnavljanja kapitala

- $ROI\%$  se obično dijeli s dužinom projekta kako bi se dobio godišnji  $ROI \rightarrow$  faktor obnavljanja kapitala
- Nizak faktor ( $\sim$  manji od 10% godišnje) može pokazivati da je korist preniska da bi bila isplativa
- Projekt1,  $ROI\% = 6,24\%$  godišnje
- Projekt2,  $ROI\% = 8,45\%$  godišnje

# **Modaliteti izgradnje sustava**

**Dio studije izvedivosti ili međukorak analize i oblikovanja**

# 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 – crowd+outsourcing**

- 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.)*
  - Danas: Crowdsource, crowdsourcing platform
- Electrolux i Philips – razvoj novih proizvoda
- Procter & Gamble (P&G) – objavljuje zadatke i cijenu
- NASA – podrška za obradu velikih količina podataka svemirskih istraživanja
- HP – predviđanja prodaje
- Netflix, Amazon, Wikipedia, ...

# 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 !
- Novija varijanta: najam „gotovog“ rješenja (SaaS)

# Graditi ili kupiti ili unajmiti (**build vs buy vs rent**) ?

## Određivanje mogućih rješenja

- Identifikacija rješenja na temelju analize zahtjeva
- Ulazi: specifikacija HW i SW te odabrana tehnološka 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 izvedivost
- 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	KuhaSprema	Prvi uradak	SAAS	Zadovoljni korisnik
Operacijski sustav	Windows	Linux	Cloud	Android
Baza podataka	SQL Server	MySQL	DynamoDB	Firebase
Brzina obrade	velika	velika	mala	srednja
Programski jezik	C#	PHP	Python	Java
Raspoloživi izvorni kod	ne	ne	da	ne
Korisničko sučelje	GUI nakićen	GUI responzivan	GUI čudan	GUI skučen
Interaktivna pomoć (on-line help)	da	ne	ne	ne
Dokumentacija	dobra	ne	ne	dobra
Mogućnosti aplikacije	velike	vrlo male	male	velike
Integracija s drugim aplikacijama	dobra	srednja	ne	ne
Brzina ispisa	srednja	srednja	velika	velika
Rad s različitim pisačima	da	da	ne	ne
Višejezičnost	da	ne	ne	ne
Vrijeme obuke korisnika	1-2 dana	1 dan	1 dan	1-2 dana
Arhiviranje podataka	da	ne	da	da
Reference	jake	jake	nema	slabe
Minimalna konfiguracija	...	...	...	...
...	...	...	...	...
Cijena paketa	1500 eur	100 eur	500 eur	2000 eur
Trošak uvođenja	3500 eur	4000 eur	2200 eur	3500 eur

# 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 ?**

# Primjer: vrednovanje mogućih rješenja

Karakteristike:	Težinski faktor	Kuha-Sprema		Prvi uredak		SAAS		Sve po malo	
		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 obrade	4	5	20	4	16	1	4	4	16
Programski jezik	1	4	4	5	5	2	2	2	2
Raspoloživi izvorni kod	1	0	0	0	0	5	5	0	0
Korisničko sučelje	2	5	10	5	10	3	6	3	6
Interaktivna pomoć (on-line help)	2	5	10	0	0	0	0	0	0
Dokumentacija	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	4	2	8	3	12	5	20	5	20
Rad s različitim pisačima	3	5	15	5	15	0	0	0	0
Višejezičnost	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
Reference	3	5	15	5	15	0	0	3	9
Minimalna konfiguracija	1	3	3	2	2	5	5	5	5
...	1	3	3	3	3	5	5	5	5
Cijena paketa	2	2	4	5	10	4	8	2	4
Trošak uvođenja	3	3	9	2	6	5	15	3	9
<b>Ukupno bodova:</b>			<b>171</b>		<b>124</b>		<b>97</b>		<b>124</b>

**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

- IT firma živi od aplikacije zasnovane na staroj tehnologiji
- Umorni poduzetnik 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 stari sustav novim ?

## □ Koji je bolji u razdoblju od 5 godina

- stari projekt:
  - stara tehnologija i održavanje 10 keur / mjesecno
- novi projekt:
  - nakon 3 godine besplatnog razvoja i prilagodbe
  - nabava licenci za 300 keur uz održavanje licenci 20% / god
  - dodatni trošak usluge održavanja 100 keur/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 eur 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.
- Na tržištu se može **kupiti** gotovi sustav inicijalnog troška 150000 eur za čiju prilagodbu i implementaciju treba 2 mjeseca i 6 konzultanta uz trošak od 25 000 eur mjesечно po konzultantu

## Izračunajte što se više isplati, graditi ili kupiti gotovo rješenje.

- Pretpostaviti da je isplata ljudskog rada na kraju godine
- A kamata/inflacija od 3% godišnje



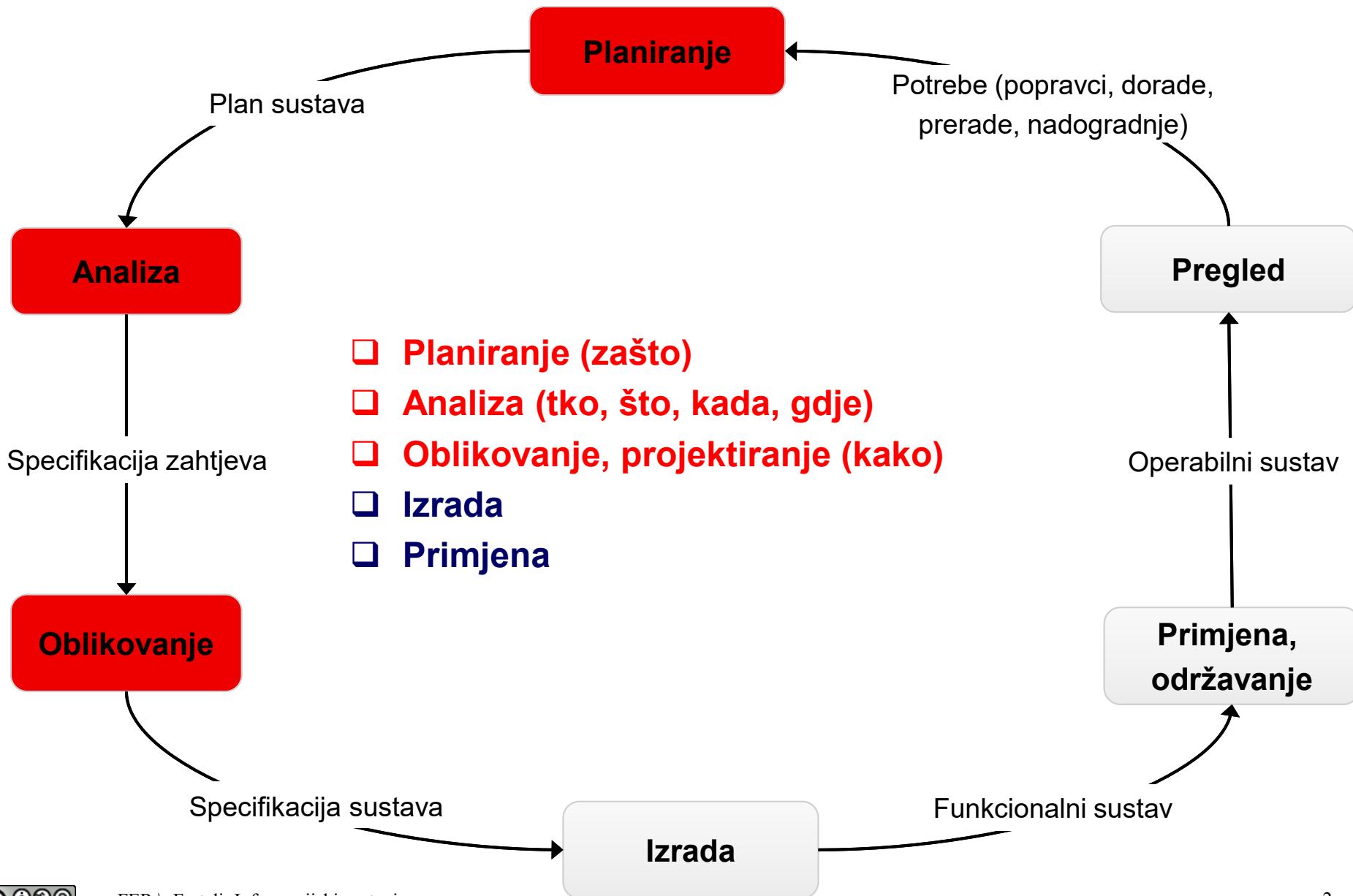
# Reference

- **Modern Empirical Cost and Schedule Estimation Tools**
  - <https://csiac.org/state-of-the-art-reports/modern-empirical-cost-and-schedule-estimation-tools>
- **International Functional Point Users Group (IPUG)**
  - <http://www.ifpug.org/>
- **Use Case Points**
  - [https://en.wikipedia.org/wiki/Use\\_Case\\_Points](https://en.wikipedia.org/wiki/Use_Case_Points)
- **COCOMO**
  - <http://sunset.usc.edu/csse/research/COCOMOII>
  - <http://softwarecost.org/tools/COCOMO/>
- **NASA Cost Estimating Handbook**
  - [https://www.nasa.gov/pdf/263676main\\_2008-NASA-Cost-Handbook-FINAL\\_v6.pdf](https://www.nasa.gov/pdf/263676main_2008-NASA-Cost-Handbook-FINAL_v6.pdf)

# Prikupljanje informacija i utvrđivanje zahtjeva

**2022/23.03**

# Kontekst prikupljanja informacija



# Postupci prikupljanja informacija



## □ Intervjui s ključnim korisnicima i radne sjednice

- Bistrenje, sučeljavanje, rješavanje proturječja

## □ Upitnici i ankete

- Nadomjestak za intervju ili prikupljanje informacija o resursima.

## □ Analiza dokumentacije

- Prikupiti dokumente značajne za poslovanje (pravila, strukture podataka)

## □ Promatranje

- Neposredni uvid u poslovne procese promatranjem radnih sredina

## □ Ostale tehnike

- Prosudba postojećih računalnih aplikacija ili evidencija
  - analiza funkcionalnosti, struktura i zapisa 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 teško unaprijed točno odrediti - prilagoditi situaciji

# Postupak 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

# Postupak 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 (prikljupljanje 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

# Postupak 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 daju 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**
  - sugovornici 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 ...?, Kako se ... ?*
- Pitanja otvorenog tipa: *Što mislite o ... ?, Koji su najveći problemi ... ?*
- “Probna” pitanja: *Zašto?, Primjer?, Objašnjenje za ...*

## □ Analiza odgovora

- razlučivanje činjenica od mišljenja
- provjera činjenica, provjera kontradikcija

## □ Preporučljivo ponašanje

- iskrenost i nepristranost – cilj je rješenje, a ne podupiranje nekog proizvoda
- izbjegavanje sugestivnih pitanja, nemametanje zaključaka
- pažnja (“brzo misli, jasno govori”) i praćenje reakcija

# Radne sjednice (združeni razvoj)

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

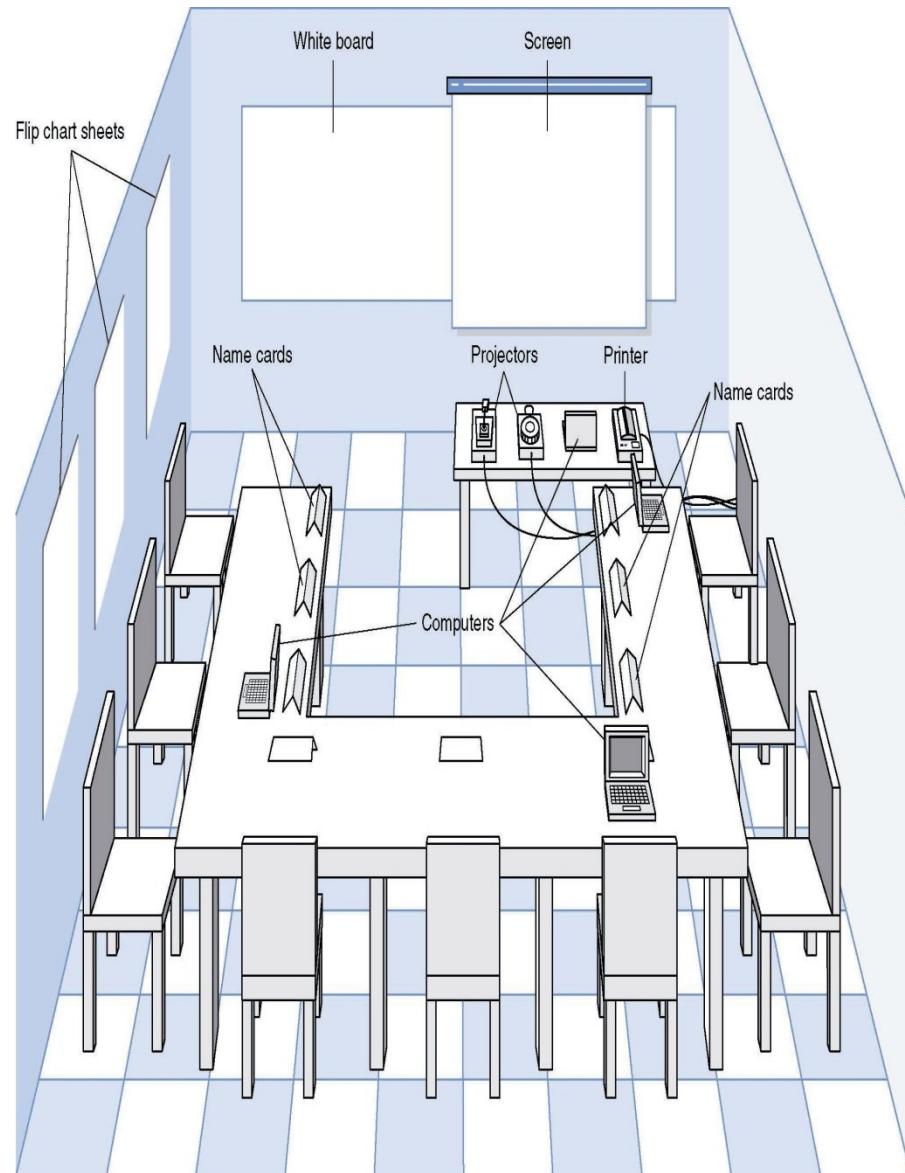
## □ Radni sastanci, sjednice (workshop)

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

## □ Brainstorming – mozganje

- aktivno i kreativno sudjelovanje
- nekriticnost, maštanje, kvantiteta, kombiniranje i poboljšavanje ideja
- svakog sudionika se stimulira da definira po njemu idealno rješenje
- odabire se najbolji, izvediv prijedlog
- tamo gdje korisnici dobro poznaju sustav ali teško prihvataju nove ideje

## □ Brainwriting, Speedstorming, Nominal group technique (NGT)



# Radne sjednice (2)

## □ Prednosti

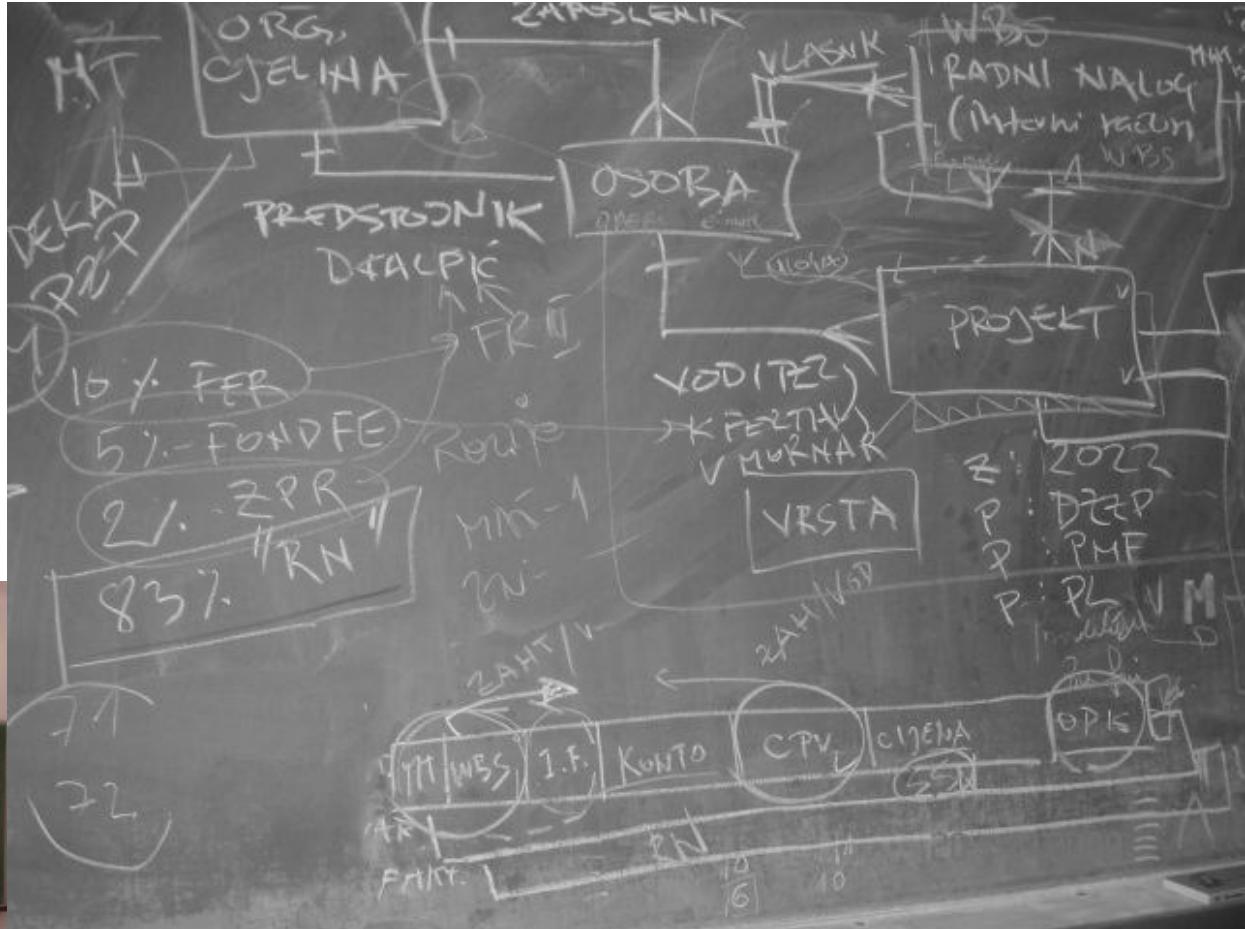
- rješavanje problema važnih za čitavu organizaciju ili veći dio poslovanja
- izbjegavanje specifičnih (egzotičnih) i nejasnih zahtjeva
- preciznije ustanovljavanje dosega projekta
- bolje uočavanje proturječnih zahtjeva

## □ Nedostaci

- sadržaj i dinamika
  - pasivnost sudionika, "usitnjavanje" razgovora, udaljavanje od tema
- trajanje
  - više dana (5-10) u nekoliko tjedana, nalaženje termina zbog obaveza 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 info. dobivenih analizom (samo) dokumenata je niska

- praksa može odudarati od pravilnika i obrazaca
- treba shvatiti zašto i kada dokumenti nastaju, kako se popunjavaju, ...
- poželjno je da budu reprezentativni, tj. popunjeni na tipičan način, ali ...
  - reprezentativni ne ukazuju na izuzetke - rjeđe bilježene podatke
  - stalno bilježenje nekih podataka ne mora značiti da su stvarno potrebni
- treba prikupiti više uzoraka iste vrste dokumenta (uzorkovanje) !

## □ Primjeri: Primjeri dokumenata

# Promatranje poslovnog sustava

## □ Uvid u poslovne procese promatranjem radnih sredina

- promatranje lokacija, kretanja ljudi, kolanja dokumenata te tijeka posla
- **najteža metoda za pronalaženje činjenica**

## □ Prednosti

- analitičar je u stanju realno sagledati poslovni proces
- učinkovitost, ako se dobro provede
- pouzdanost prikupljenih informacija

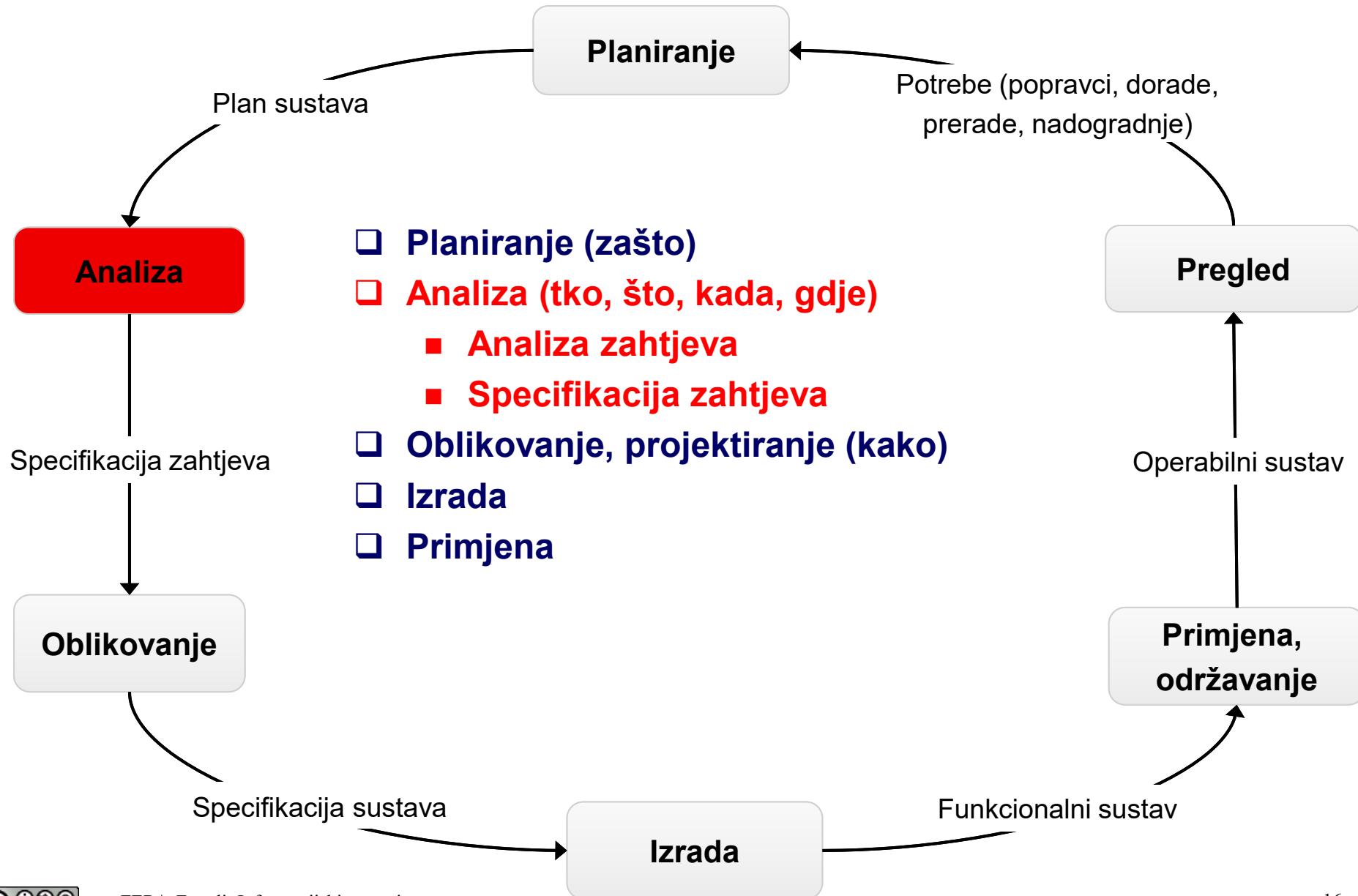
## □ Nedostaci

- utrošak vremena
- ometanje i neugoda promatralih osoba
- mogućnost manipulacije - prikrivanjem uobičajenog kršenja radnih procedura
- problem određivanja trajanja i učestalosti
- nepotpunost premalog broja i/ili prekratkih promatranja

## □ Primjer: Ministarstvo

# **Analiza sustava**

# Kontekst analize sustava



# Analiza sustava

## □ Analiza sustava (sistemska analiza) 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, ciljevi s različitim tehnikama i dubinom 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 utvrdi 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 – intranet (podnositelj)
- automatizacija odobravanja – intranet (voditelj, predstojnik, dekan)
- isplata predujmova putem tekućeg računa
- isplata povrata putem tekućeg računa
- uklanjanje deviznih isplata

# 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 za obavljenu nastavu u svakom semestru i za svaki predmet
  - nezavisni ugovori za vođenje (mentorstvo) i obrane doktorata
  - sve isplate temeljem ugovora provodi Računovodstvo
- Poboljšanje:
  - izrađuje se autorski ugovor s općim uvjetima – potpisuje se jednom
  - primjerak potписанog ugovora dostavlja se u Studentsku službu
  - Studentska služba po obavljenom poslu radi obračun nastave te generira nalog za isplatu s pozivom na broj autorskog ugovora
  - Kombiniranje s BPA: ovjera obračuna putem intraneta (default, unos postotaka, potvrda)

# 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 urudžbiran

# Preustroj, preoblikovanje poslovnih procesa

## □ Business Process Reengineering/Redesign (BPR)

- radikalni redizajn analizom posljedica, procjenom alternativnih tehnologija, ukidanjem ili zamjenom aktivnosti, analizom troškova-koristi i analizom rizika

## □ Tehnike

### ■ Analiza ishoda (outcome analysis)

- Razumijevanje rezultata koji predstavljaju vrijednost za korisnika
- procjena što bi se moglo ponuditi krajnjim korisnicima

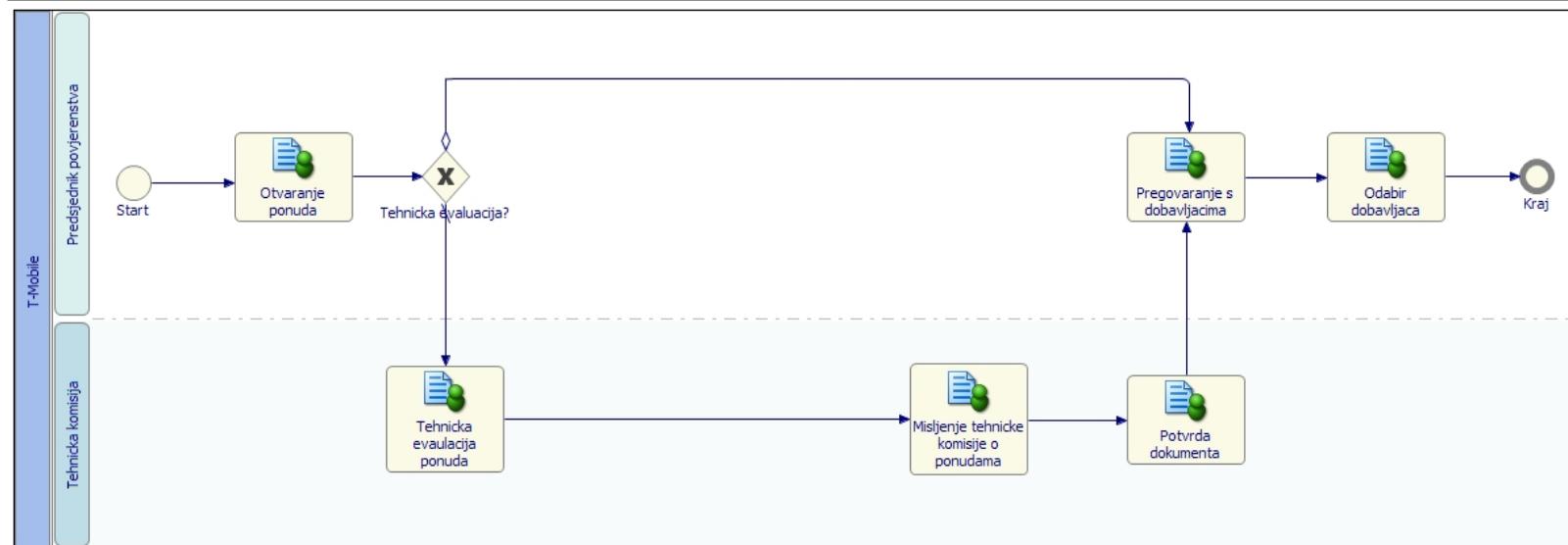
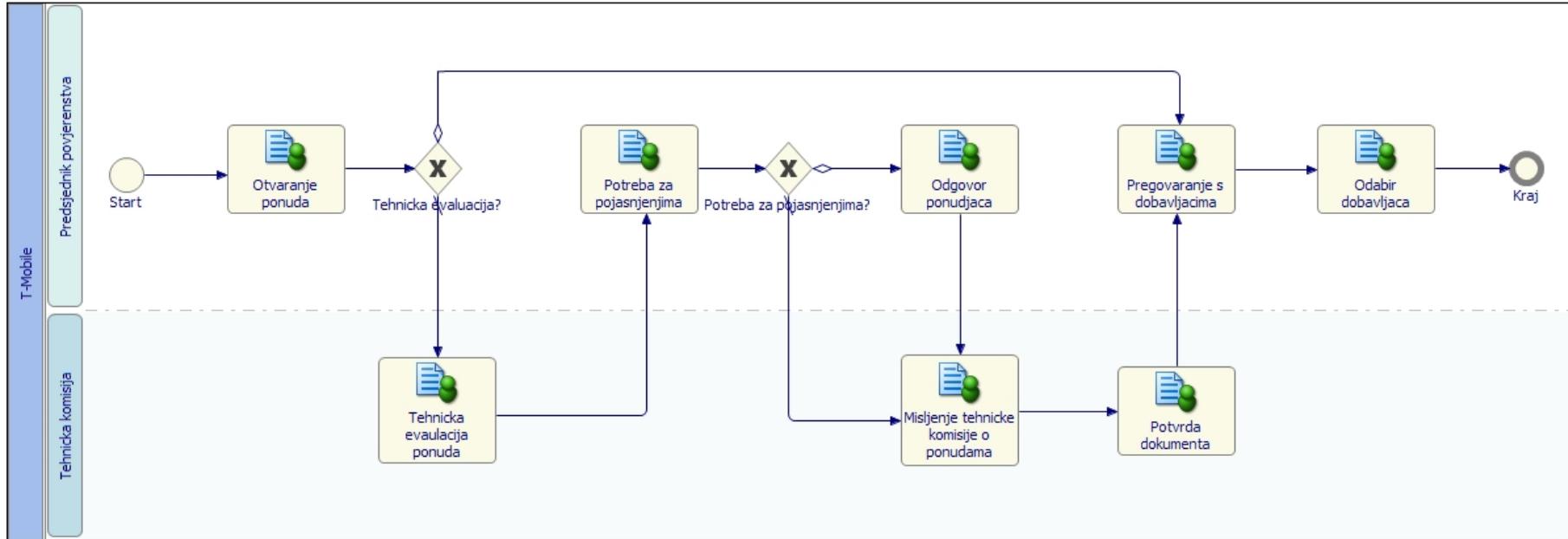
### ■ Analiza tehnologije (technology analysis)

- 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: optimizacija simulacijom koristeći GA





# Preporuke za analizu

Slika: Awad, 1985;  
preveo Anonymous  
(WWW)

## □ Ispravni pristup

- analiza mora biti prilagođena korisniku
- ključno je razumjeti suštinu organizacije i način poslovanja

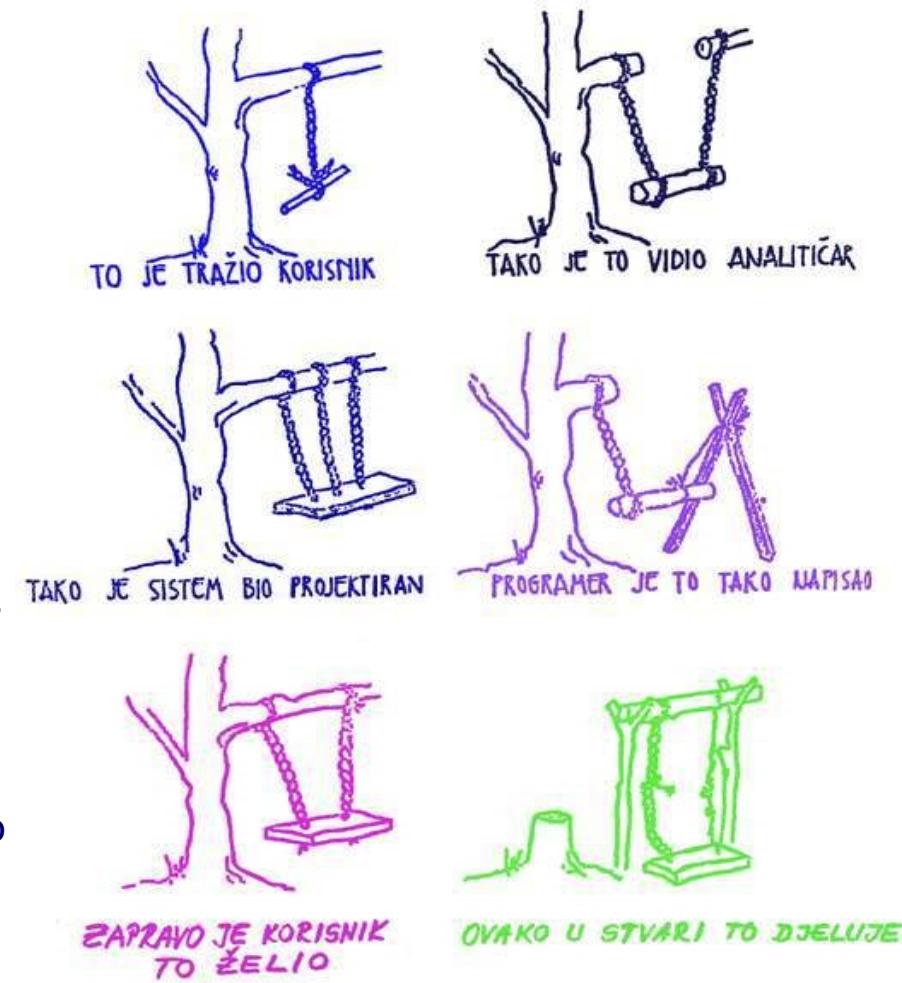
## □ korisnik želi <> korisnik treba

- 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

## TRADICIONALNA PRIČA O RAZVOJU



# **Zahtjevi na sustav, softver**

**System/Software Requirements**

# Zahtjevi

## □ Zahtjev [ISO/IEC/IEEE 24765]

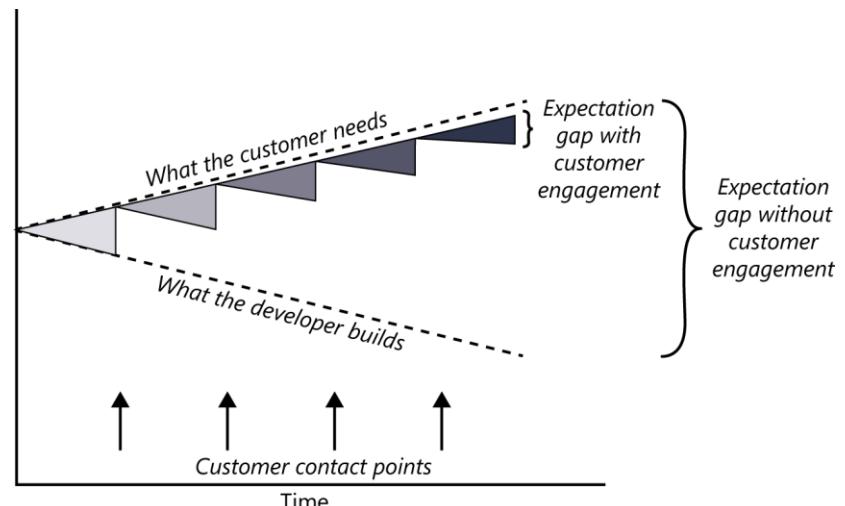
- (1) Uvjet ili sposobnost koje korisnik treba da bi riješio neki problem ili ostvario neki cilj.
- (2) Uvjet ili sposobnost koji mora posjedovati sustav ili komponenta da bi zadovoljili ugovor, standard, specifikaciju ili drugi formalni dokument.

## □ Zahtjevi ne smiju sadržavati detalje dizajna ili implementacije

- ali ako su nepotpuni onemogućuju planiranje projekta i praćenje promjena

## □ Ključni problemi

- 40-60% pogrešaka uzrokovano je pogreškama postavljanja zahtjeva
- što se pogreška kasnije otkrije to je veći napor/trošak popravka
- "neispunjena očekivanja"  
(expectation gap)



# 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 internetsku 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

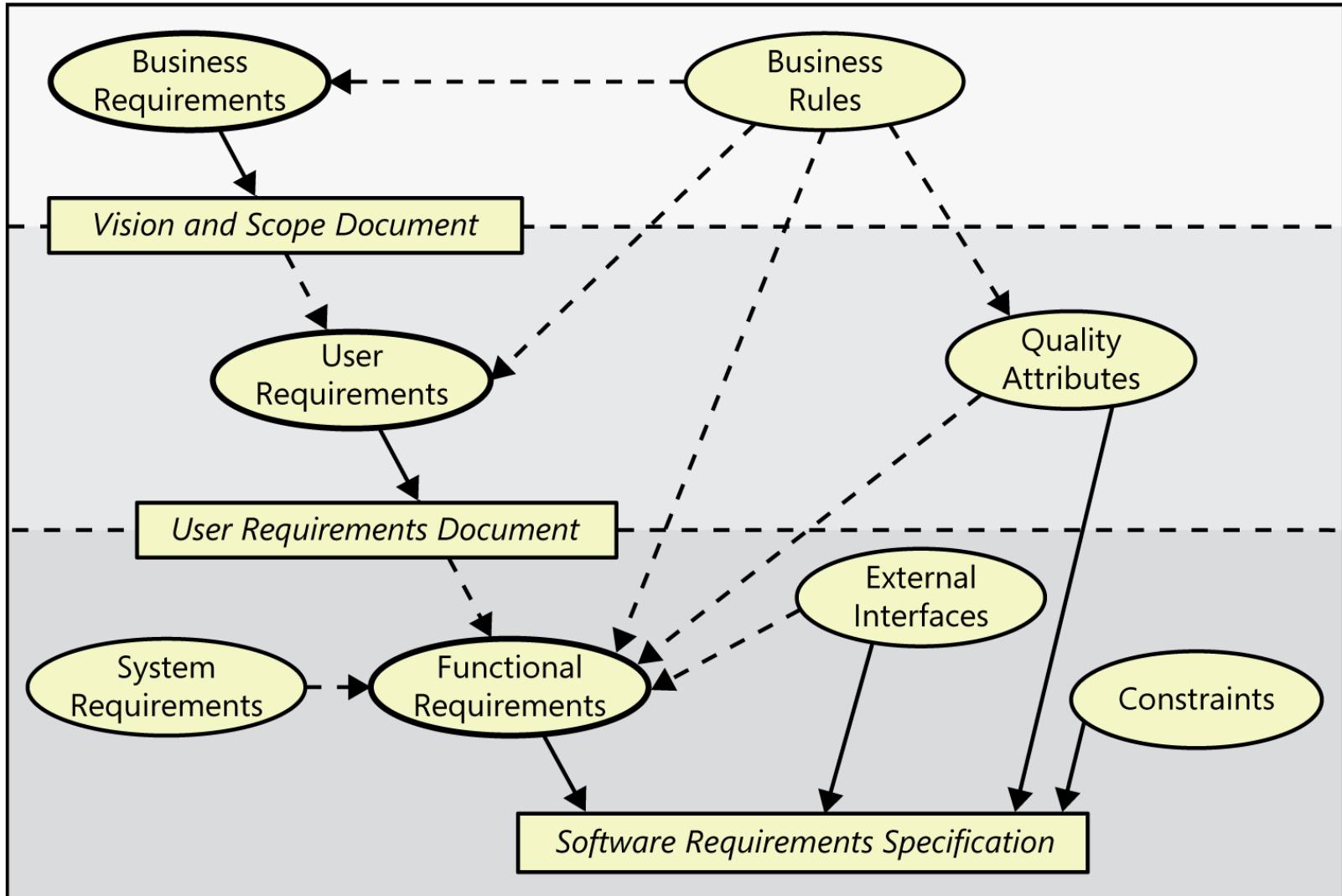
## □ Funkcionalni zahtjevi (što)

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

# Kontekst postavljanja zahtjeva



[Wiegers & Beatty, 2013}

# 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 matičnoj 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.*

## □ Zahtjevi korisnika definiraju ograničenja, doseg

# 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 *format* ili web servis
- Prenosivost – rad na različitim operacijskim sustavima, npr. Linux i 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: sistemski i ostali nefunkcionalni zahtjevi

## □ Primjer: PISFER-Podprojekt

- sve informacije treba moći unijeti na mjestu nastanka
- omogućiti da sve org. 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 evidentirat 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 (IEEE)

❑ Neki su međusobno proturječni – riješiti postavljanjem prioriteta

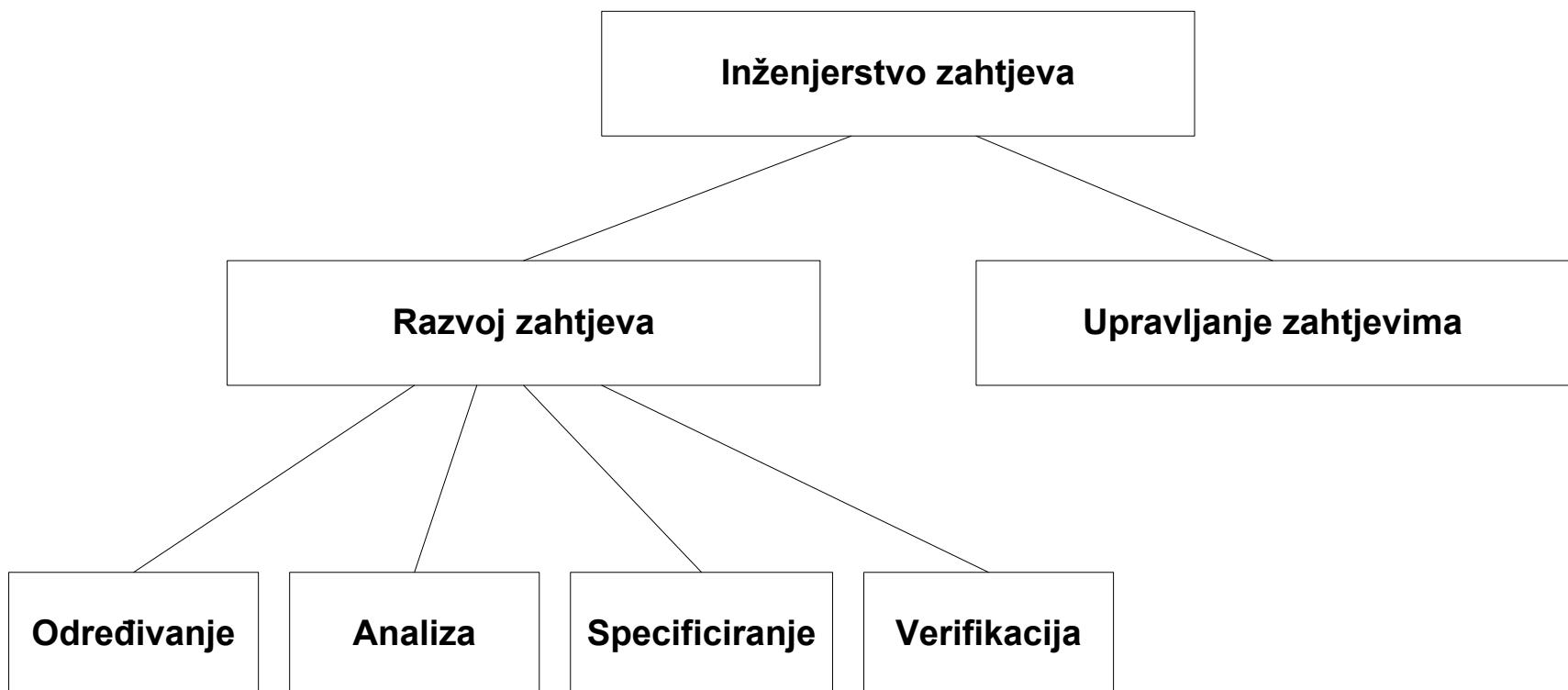
	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability									+	+						
Efficiency	+				-	-	+	-		-	+	-				
Installability	+								+				+			
Integrity		-		-			-		-	+		+	-	-		
Interoperability	+	-	-				-	+	+	+	-		-			
Modifiability	+	-					-		+	+		+				+
Performance		+		-	-			-		-		-				-
Portability		-		+	-	-			+				-	-		+
Reliability	+	-	+		+	-				+	+		+	+	+	
Reusability		-	-	+	+	-	+						-			+
Robustness	+	-	+	+	+		-	+			+	+	+	+	+	
Safety		-	+	+			-			+		+		-	-	
Scalability	+	+	+				+	+	+							
Security	+		+	+			-	-	+	+	+			-	-	
Usability		-	+				-	-	+	+	+					-
FE	Verifiability	+		+	+	+			+	+	+	+	+	+	+	

# Inženjerstvo zahtjeva

# Inženjerstvo zahtjeva

## □ Inženjerstvo zahtjeva (requirements engineering)

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



# Aktivnosti razvoja zahtjeva i upravljanja zahtjevima

## □ Razvoj zahtjeva

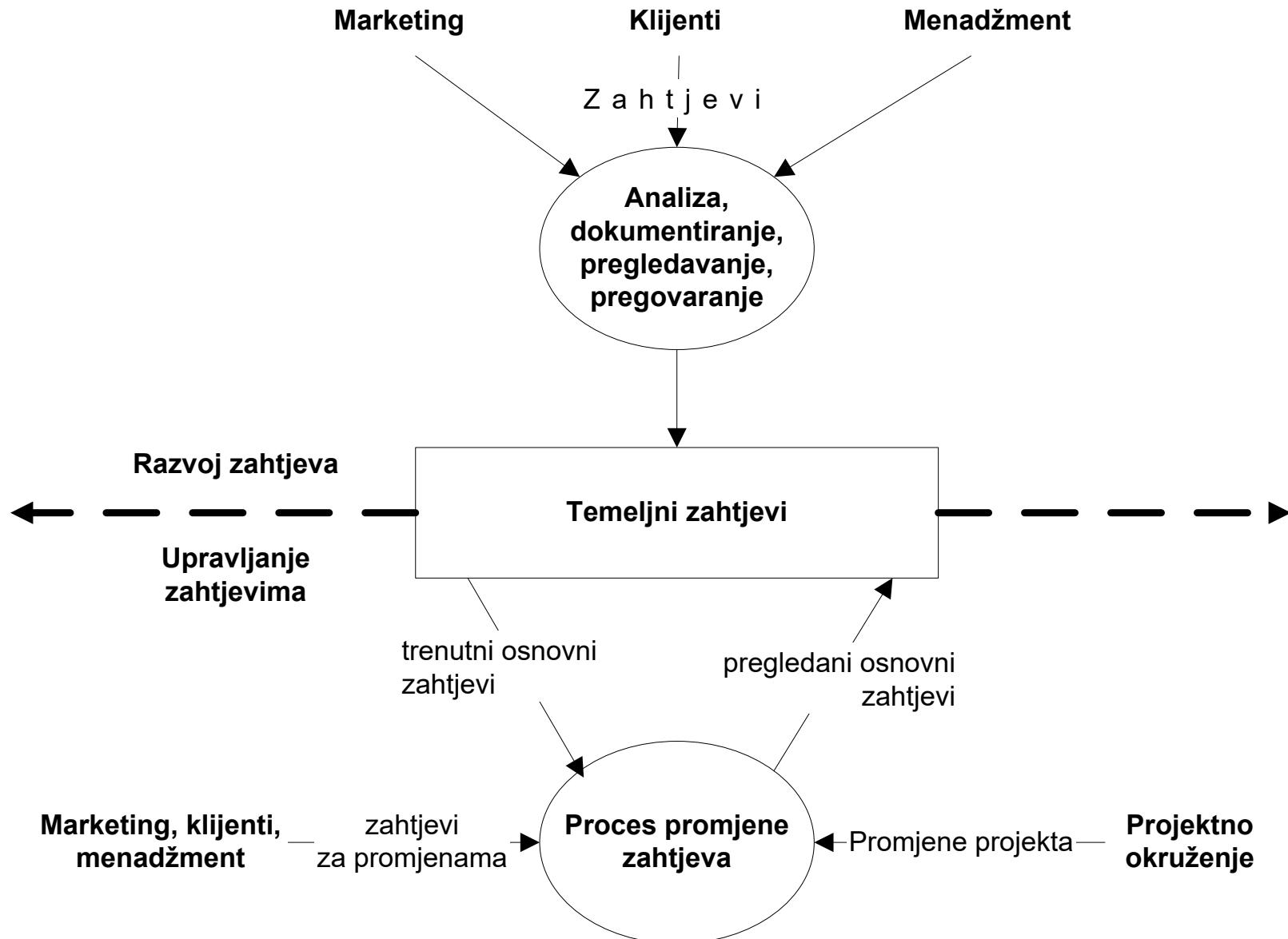
- Razredi korisnika ("najbolji", ključni)
- Poslovne potrebe, zadaci i ciljevi
- Analiza informacija, kategorizacija
- Razdioba s razine sustava na podsustave i udio u komponentama
- Razumijevanje relativne važnosti (težine) svojstava kvalitete
- Određivanje prioriteta ugradnje
- Modeliranje zahtjeva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

## □ Upravljanje zahtjevima

- Rokovi prihvaćanja zahtjeva
- Pregled promjena zahtjeva, procjena utjecaja, odluka o (ne)prihvaćanju
- 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
- „brzinu, jednostavnost, intuitivnost, ...“ - razjasniti i precizirati (kvaliteta PP)

## ❑ 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 (portabilnost) 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, otvorи se *menu*, ... a tu mi ugradи *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, ...
  - neiskustvo ili nedostatak 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
  - „to nije moj posao“
- 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 svi sudionici razumiju što zahtjevi znače - ujedno provjera 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**
  - prioritet ugradnje za slučaj korištenja, svojstvo proizvoda ili drugi zahtjev
- **Modeliranje zahtjeva**
  - izrada modela strukture, ponašanja i stanja – ERD, DFD, STD, UCD, AD, ...
- **Izrada rječnika podataka**
  - središnji repozitorij za definiranje struktura podataka i 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

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, UCD, AD) •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 (**MoSCoW**)

## 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!

## Potrebno 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

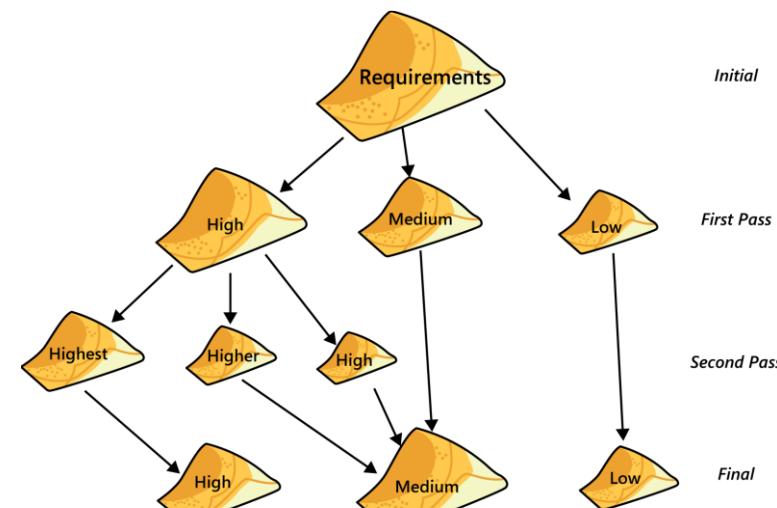
- "bilo bi lijepo, ali se može bez"
- nisu pravi zahtjevi, ali mogu biti rangirani i odrediti kvalitetu

## Nepotrebna svojstva (**Won't have [this time]**)

- možda kasnije (drugi put)

# Rangiranje 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

	Hitno	Manje hitno
Važno	visok	srednji
Manje važno	delegirati	nizak / odbaciti

# 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 (strukturni 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 Pretpostavke i ovisnosti

## 3. Zahtjevi za sučeljem

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

## 4. Svojstva sustava

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

## 5. Ostali nefunkcionalni zahtjevi

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

## 6. Ostali zahtjevi

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

# 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

# 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 (1)

## □ 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, CCB)

- 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 pronaže 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, materijali, diskusija

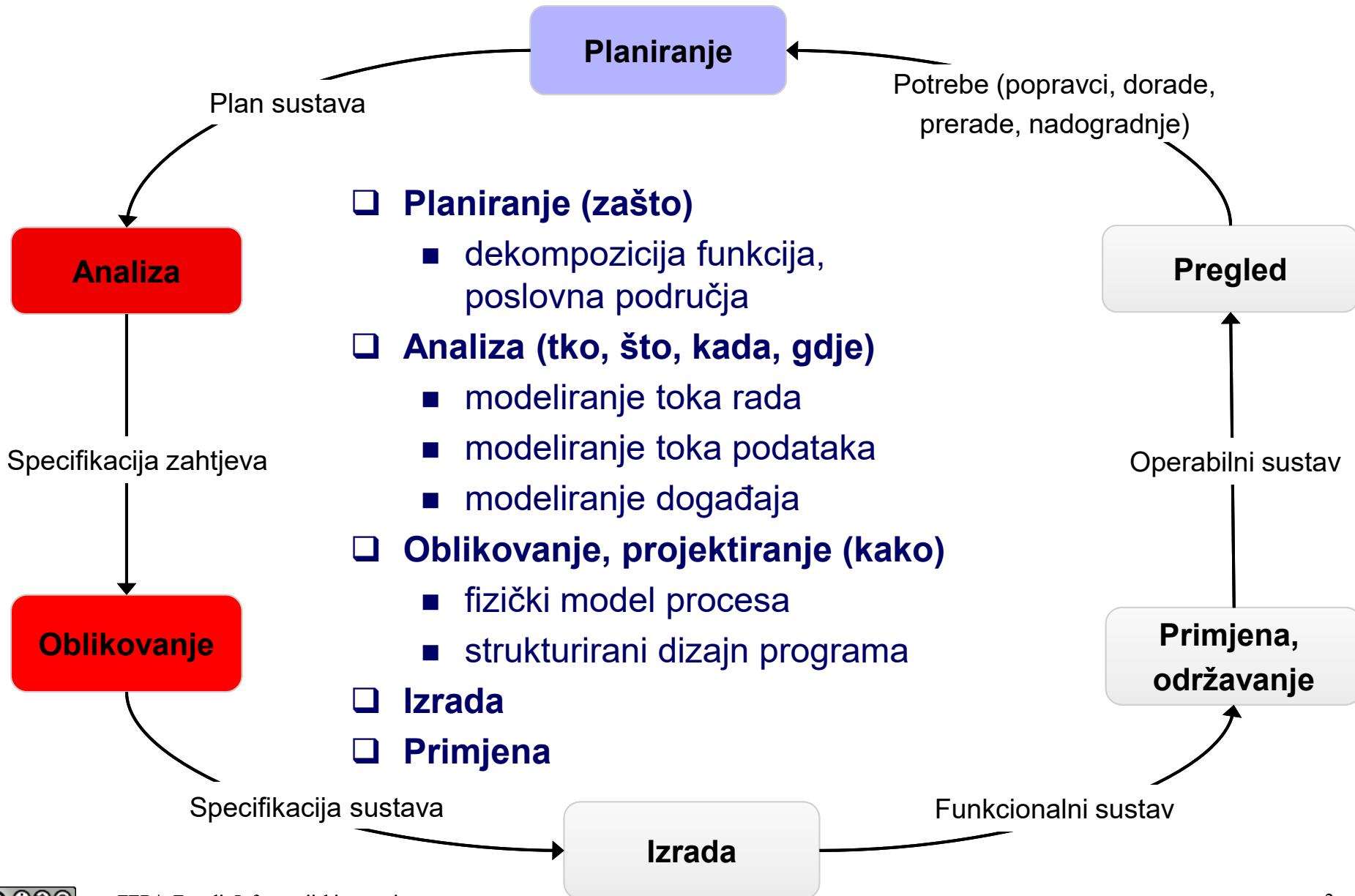
- Karl Wiegers and Joy Beatty. Software Requirements, Microsoft Press 2013.
- Dean Leffingwell. Agile Software Requirements, Pearson, 2011.
  
- Primjeri: DOO-Specifikacija, TK-upute
  - dvije iteracije teksta, prototip, nekoliko? promjena pred isporuku



# **Strukturirana analiza i dizajn**

**2022/23.04**

# Strukturirana analiza i strukturirani dizajn



# 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

- **Moderna strukturirana analiza = logički dizajn (česti sinonim)**
  - tehnika modeliranja zahtjeva na sustav
  - usmjereni procesima, ali obuhvaća i podatke
- **Strukturirani proces i strukturirani rezultati analize**
  - logički modeli – određuju 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

- Hijerarhija (poslovnih) funkcija, događaja i njihove razrade do elementarnih procesa

## □ 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

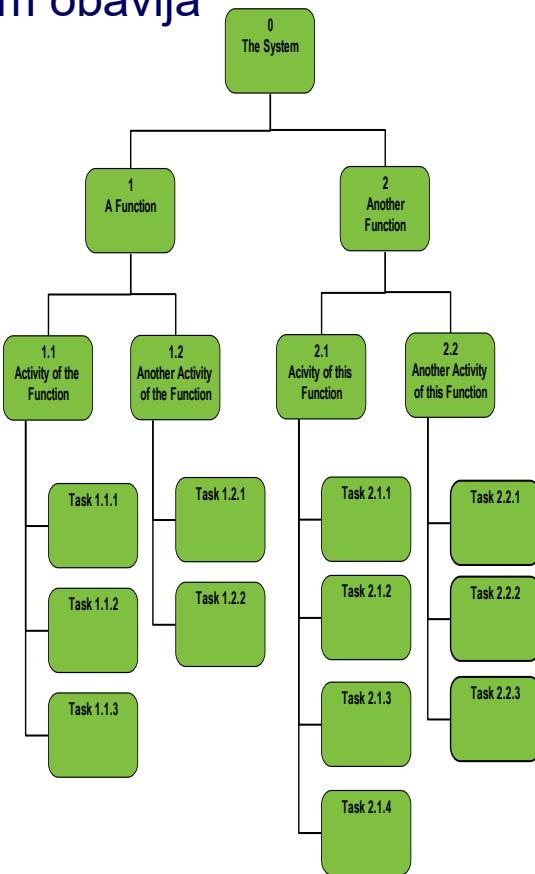
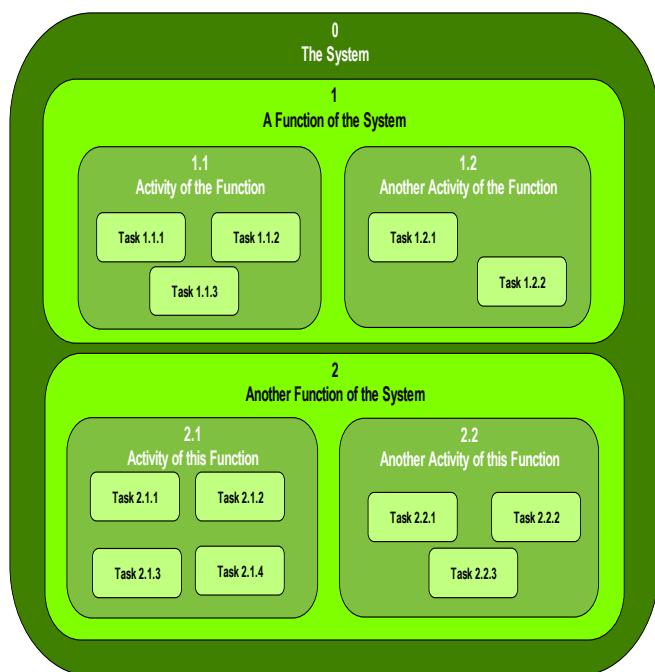
# Poslovna pravila i poslovna politika

- **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



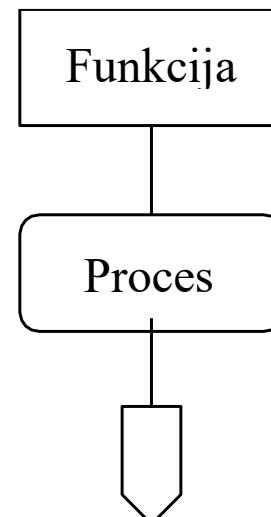
# Dijagram dekompozicije funkcija

## □ 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**

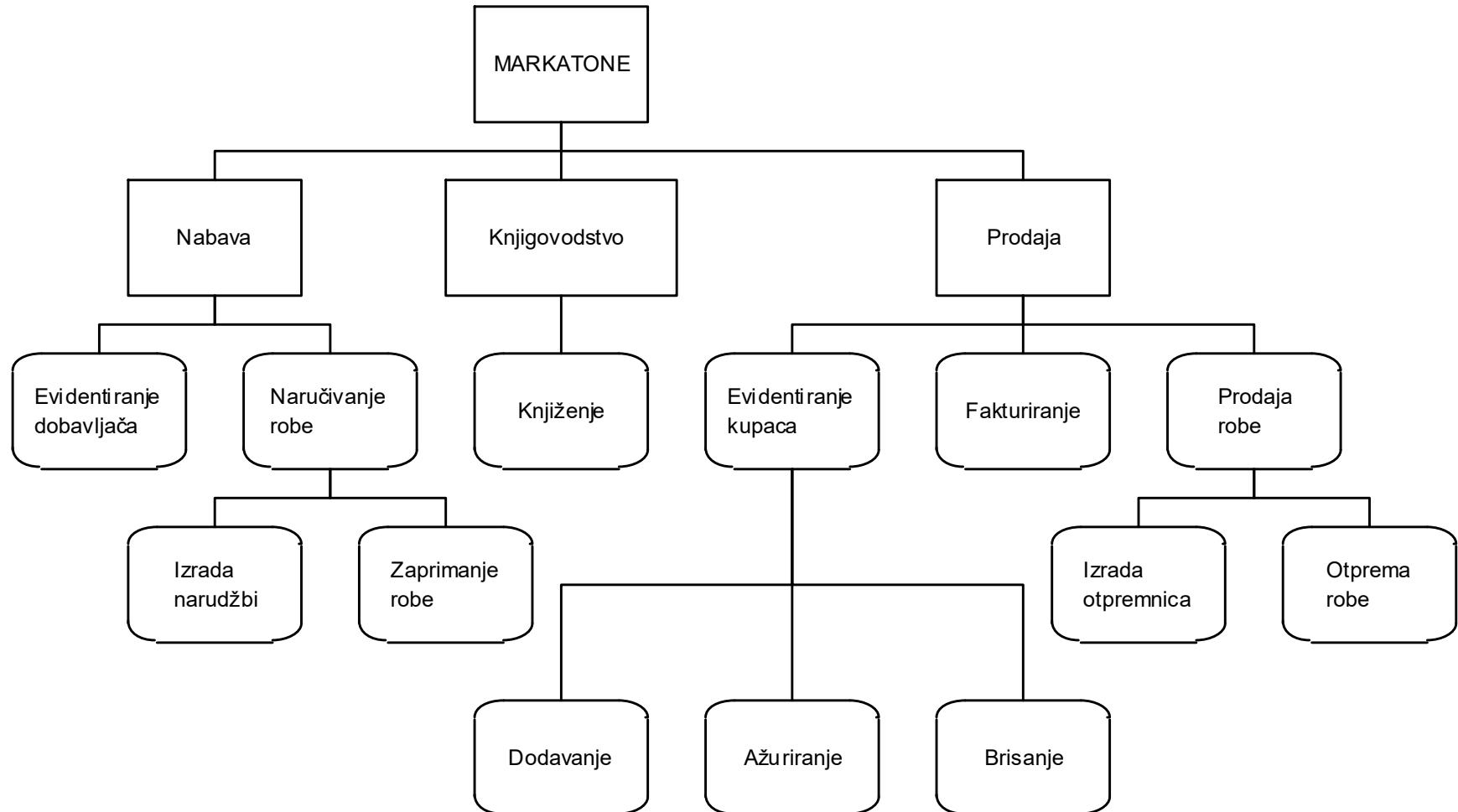
## □ Elementi

- funkcije - označavaju se (glagolskom) imenicom, npr. Prodaja, Proizvodnja
- procesi - označavaju se glagolskom imenicom ili izrazom *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:



# 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!

# Hijerarhijski prikaz funkcija/procesa

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

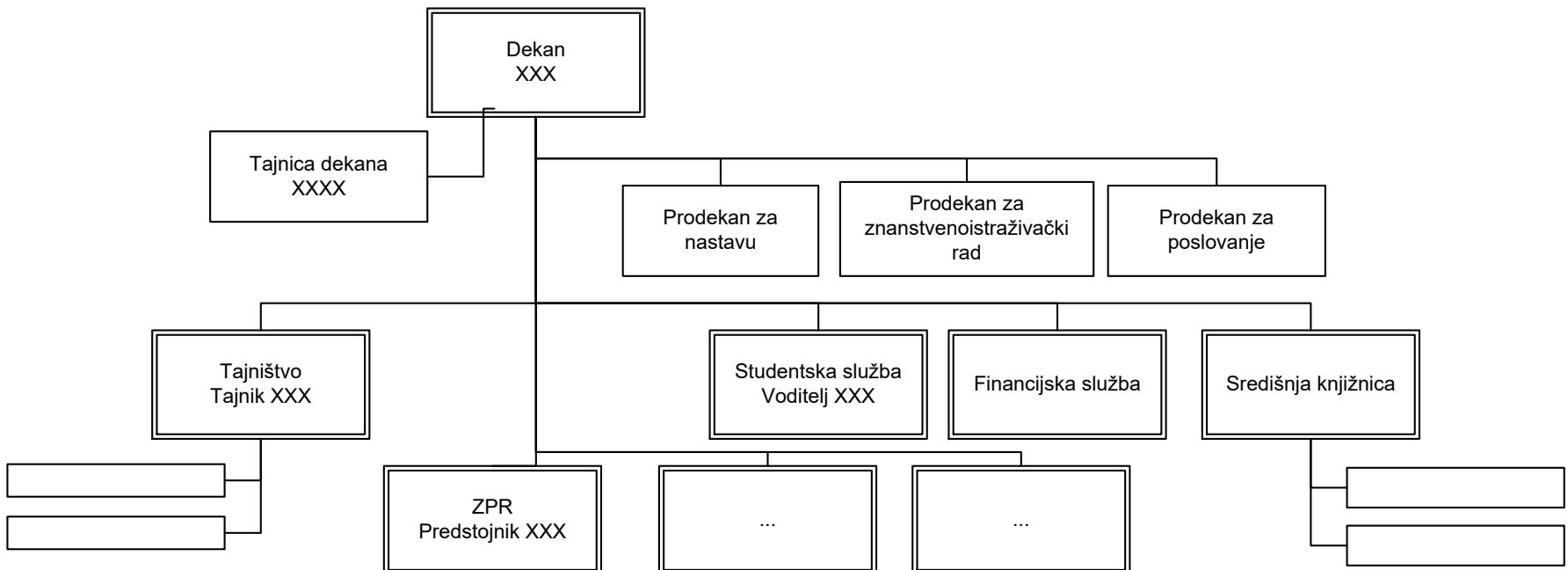
## □ Primjer:

- 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

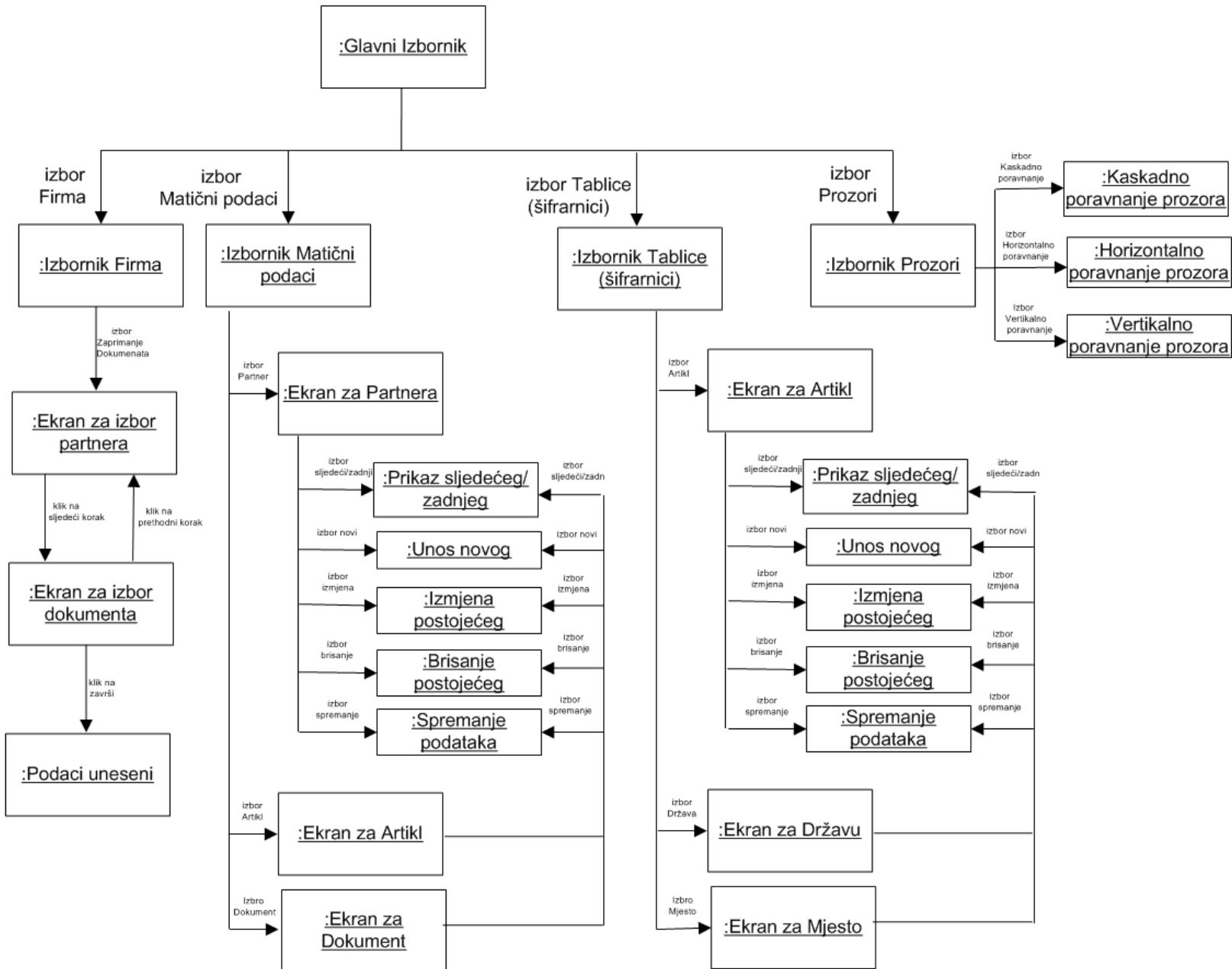
# Dijagram organizacije

## □ 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



# Hijerarhija funkcija aplikacije



# Razrada poslovnih procesa

## □ Tehnike modeliranja DFD

- 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 ?)

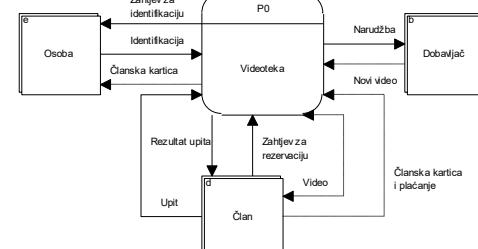
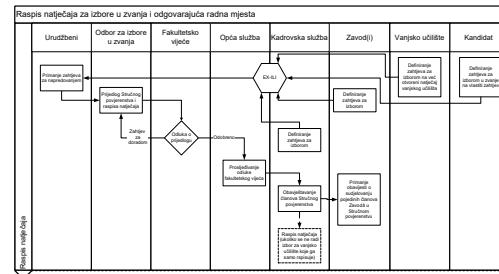
- do dubine dovoljne za razumijevanje modela (!?)
- do stanja u kojem prevladaju ulazi/izlazi - razina iznad listova u stablu prikaza

## □ Nastavak se može provesti

- modeliranjem toka rada (dijagram aktivnosti, Cross-Functional FlowChart, ...)

Ili

- modeliranjem toka podataka (Data Flow Diagram - DFD )



# **Modeliranje toka rada i toka podataka**

# Modeliranje toka rada

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

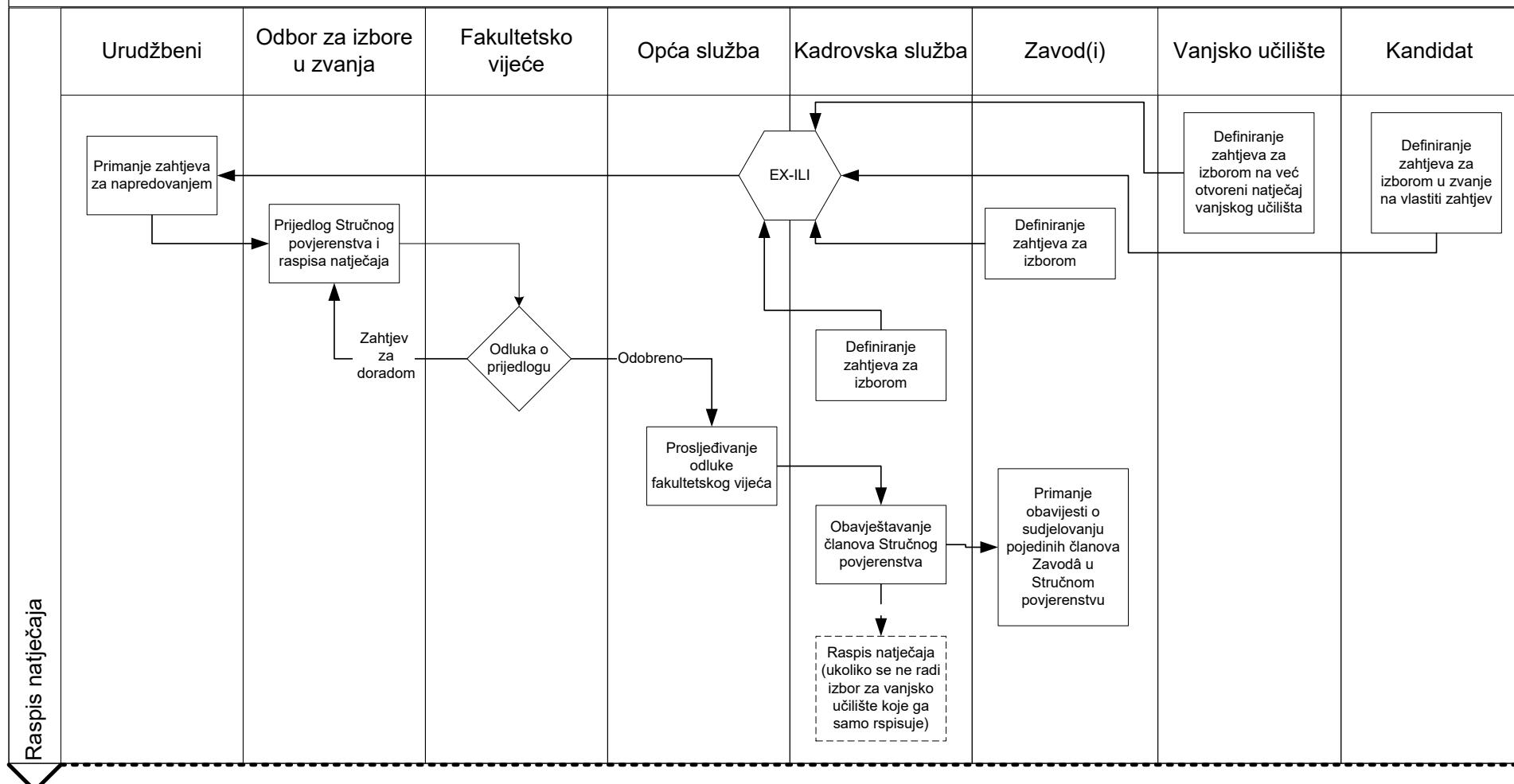
- slijed koraka obrade koji obrađuje jednu poslovnu transakciju
- definira logiku obrade i precizira nositelja
- može imati više varijanti (scenarija)

## □ 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



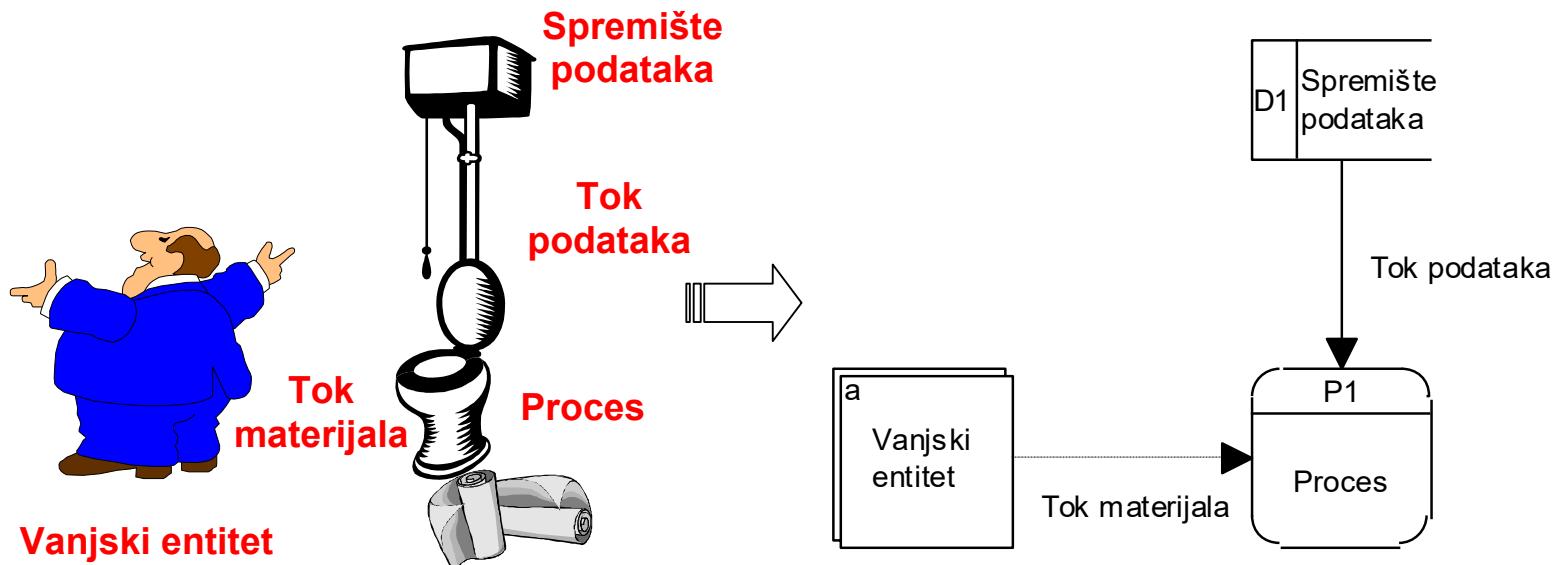
## Slika vrijedi 1024 riječi

- 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 ?

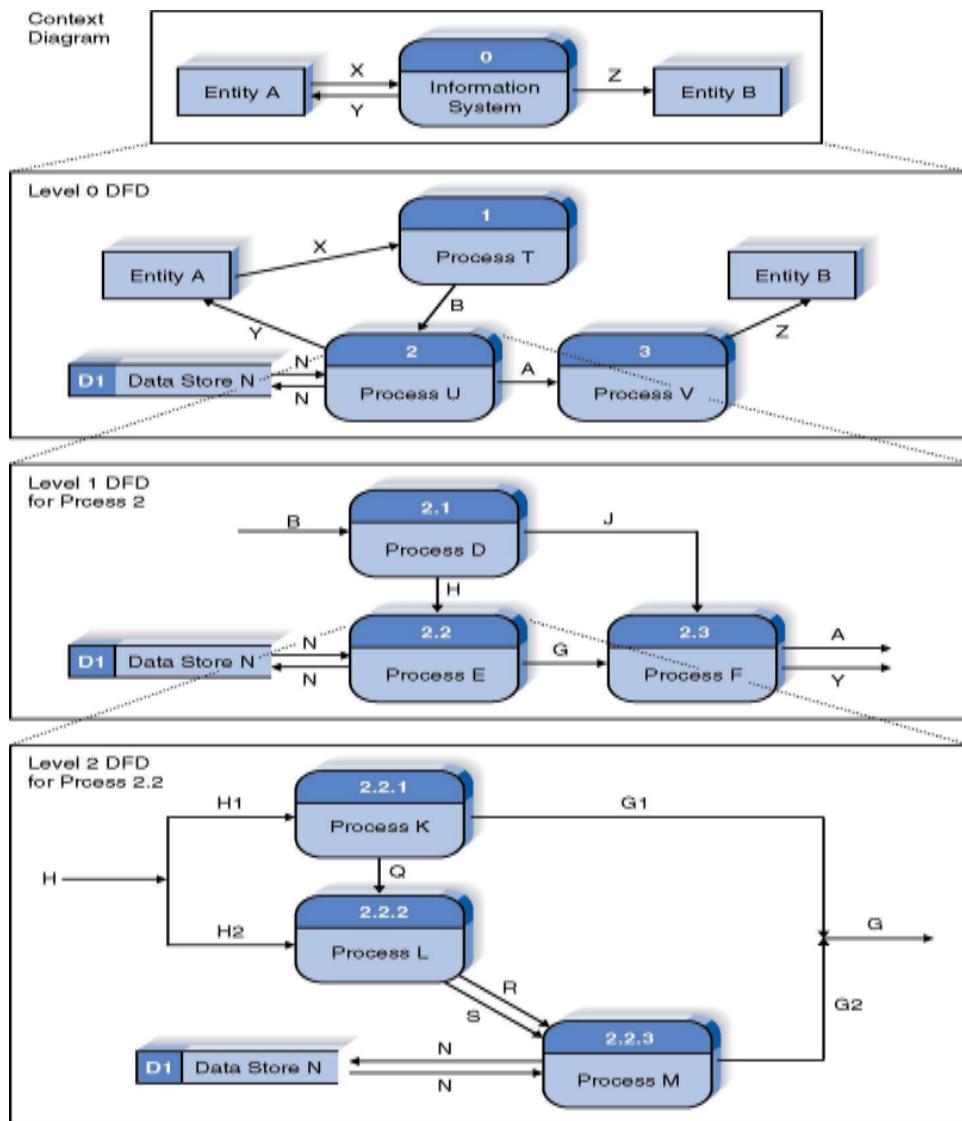
# Izrada dijagrama toka podataka

## □ 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  $7 \pm 2$ "
- zaustavljanje kada postane očigledna ugradnja procesa na najnižoj razini

## □ Preporuke za označavanje elemenata

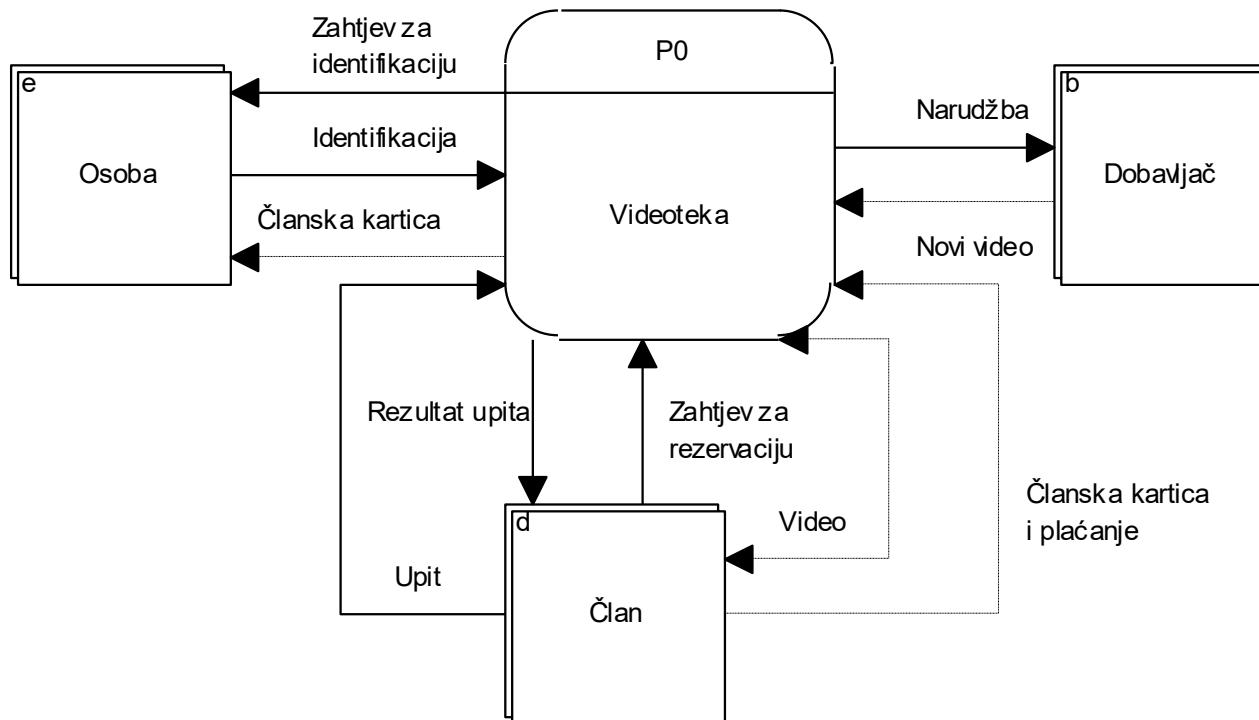
- hijerarhija, razina konteksta = 0
- spremišta, izvori i odredišta – nazivlje velikim slovima, oznake oblika *slово* ili *slovo+broj* (pr.D1)
- procesi i tokovi - malim slovima



# 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



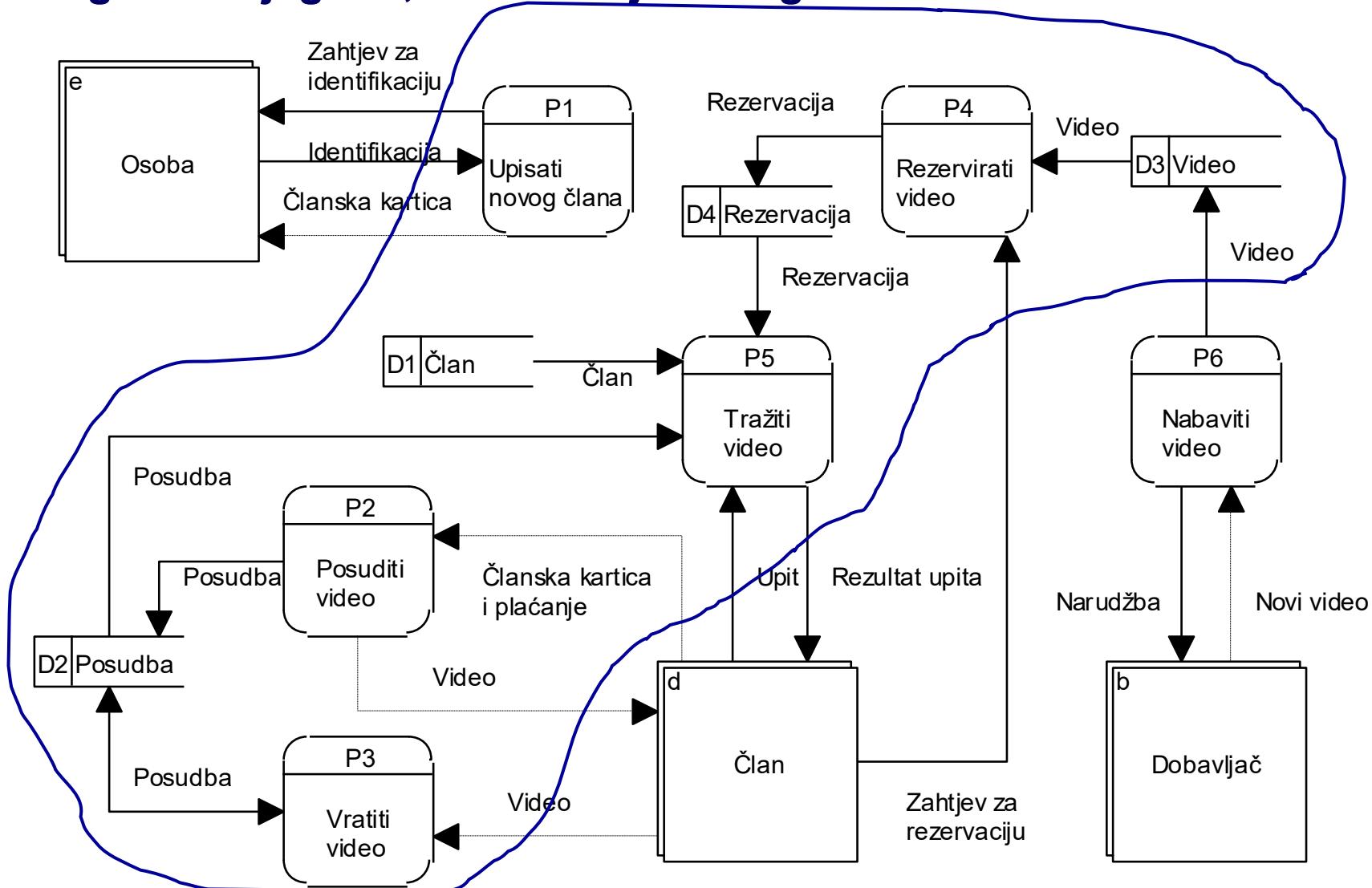
# Izrada dijagrama toka podataka (3)

## □ 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)

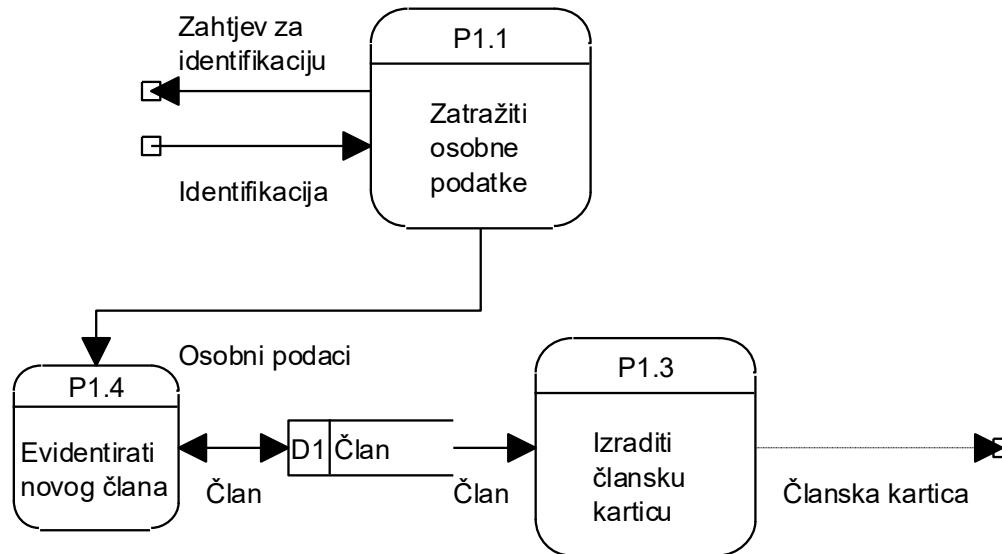
## □ Pregledni dijagram, određivanje dosega



# Izrada dijagrama toka podataka (5)

## □ Razrada

- za svaki proces u dosegu 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  $7\pm 2$ "
- 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)

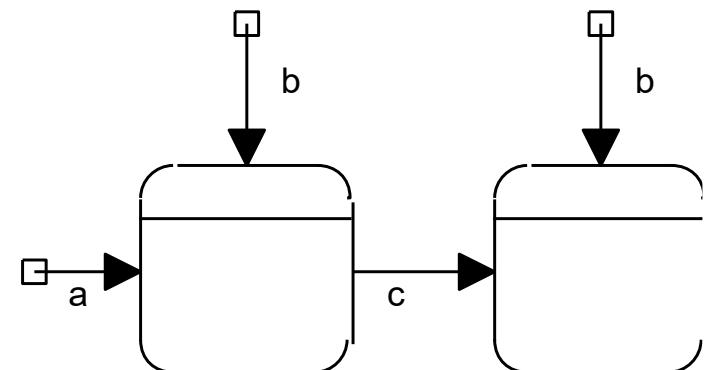
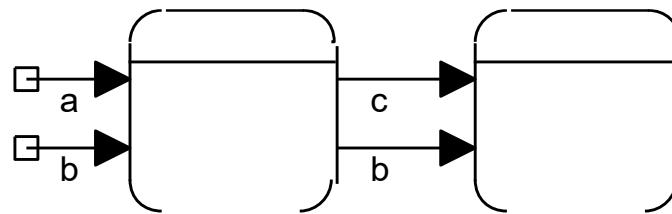
## □ Ograničenja i posebni slučajevi

- 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:



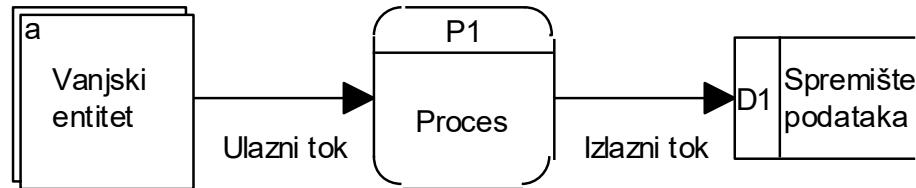
## □ Procesi se mogu zbivati istovremeno

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

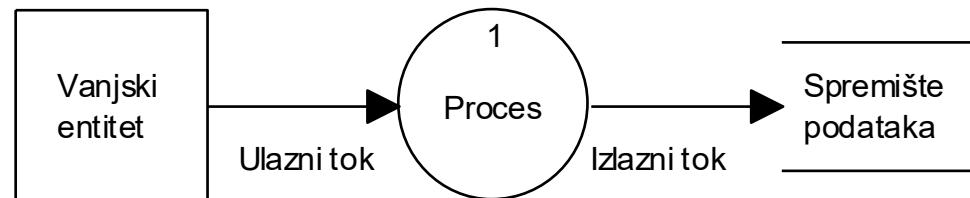
# Metode koje koriste DTP

## □ Notacije

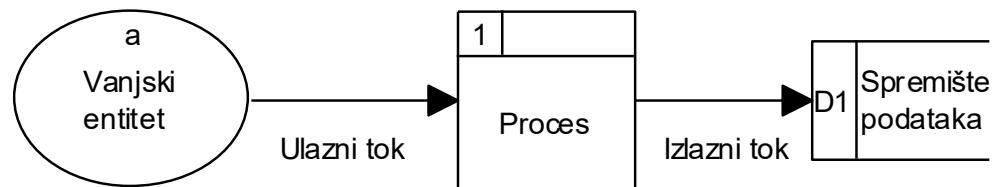
- Gane/Sarson (korištena u primjerima)



- Yourdon/DeMarco



- SSADM

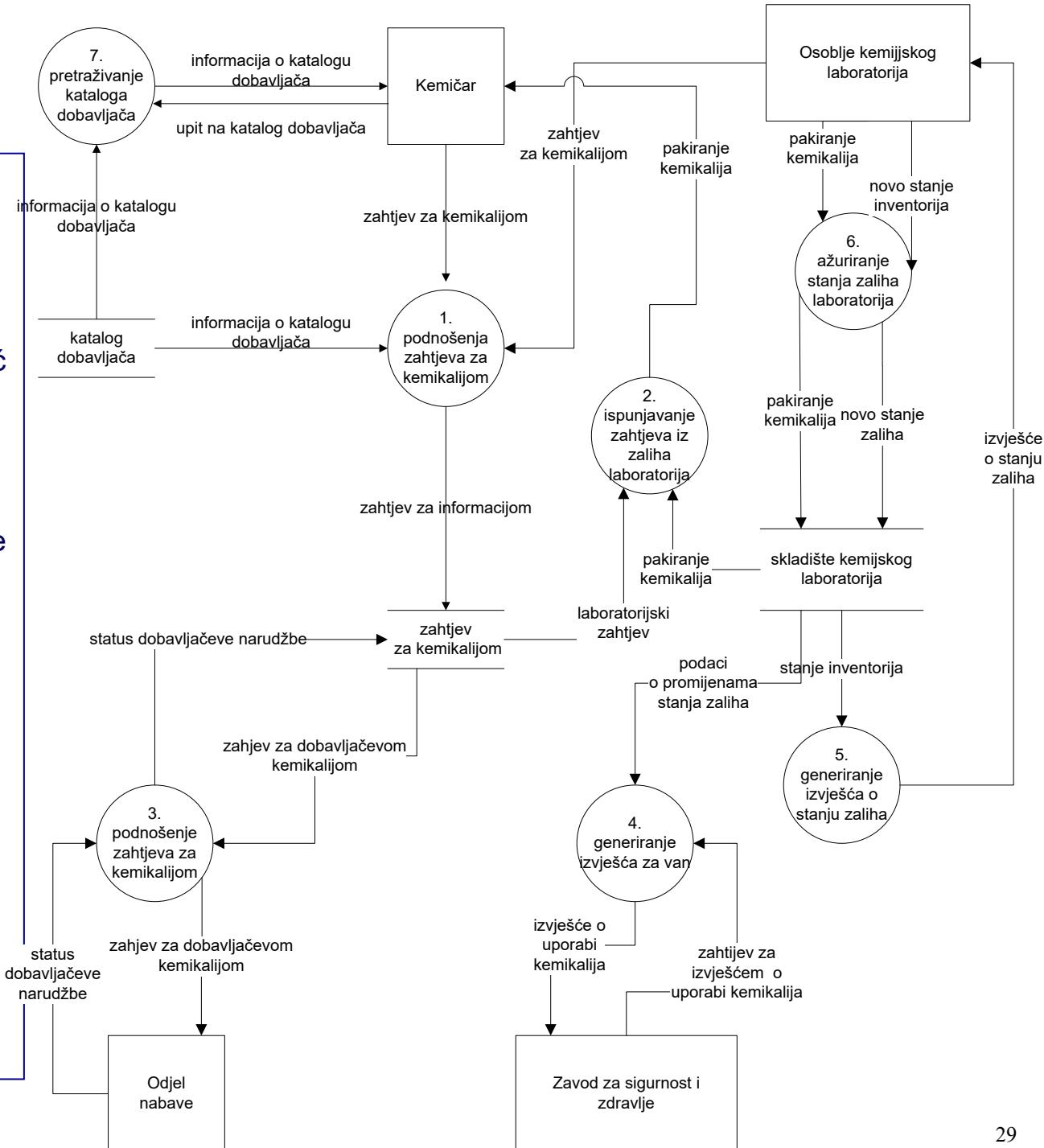


## □ Proširenja modela

- 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:

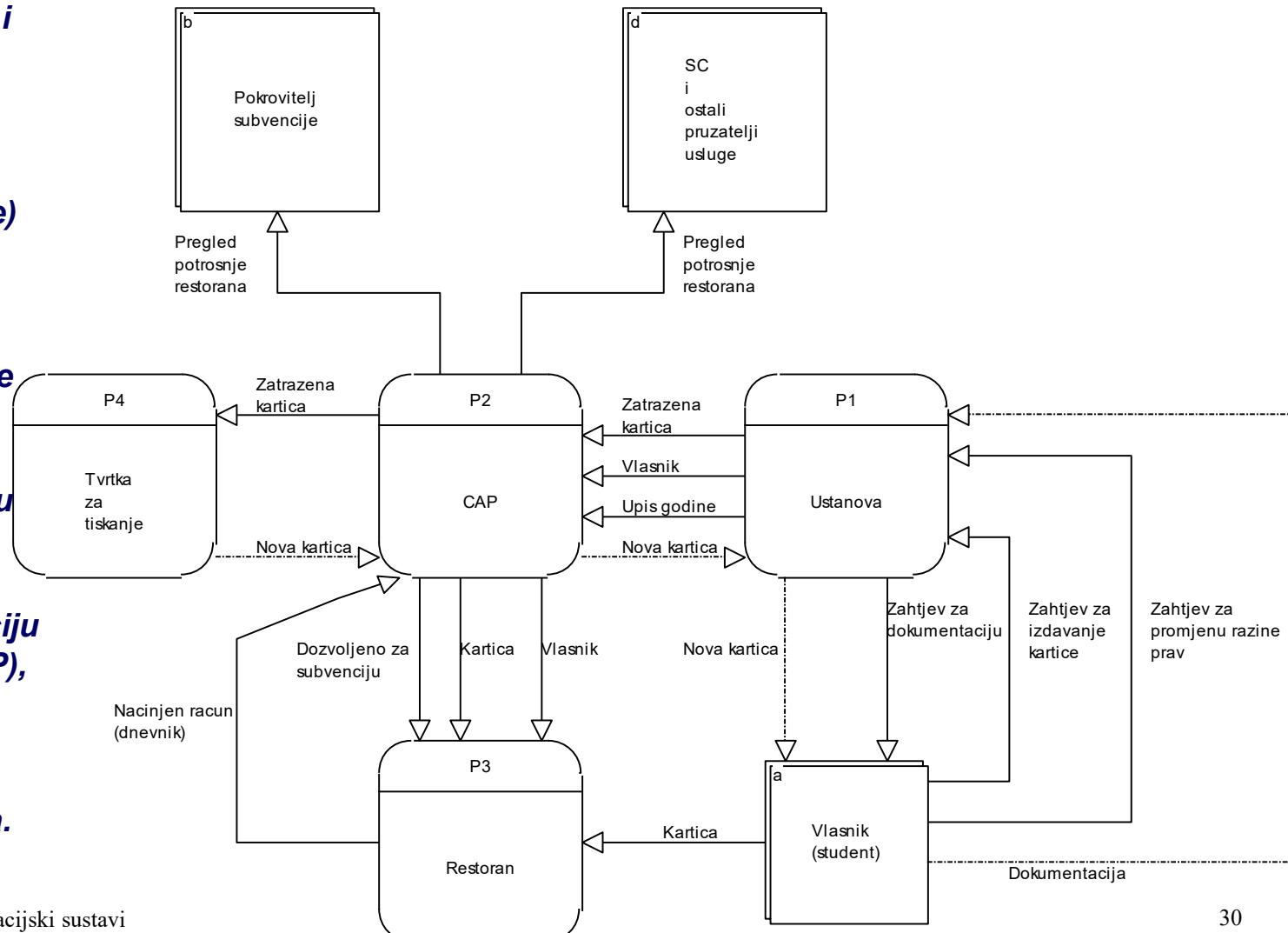
- Članice sveučilišta i veleučilišta, čiji studenti ostvaruju pravo na subvencioniranu prehranu (ustanove)

- 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 ...



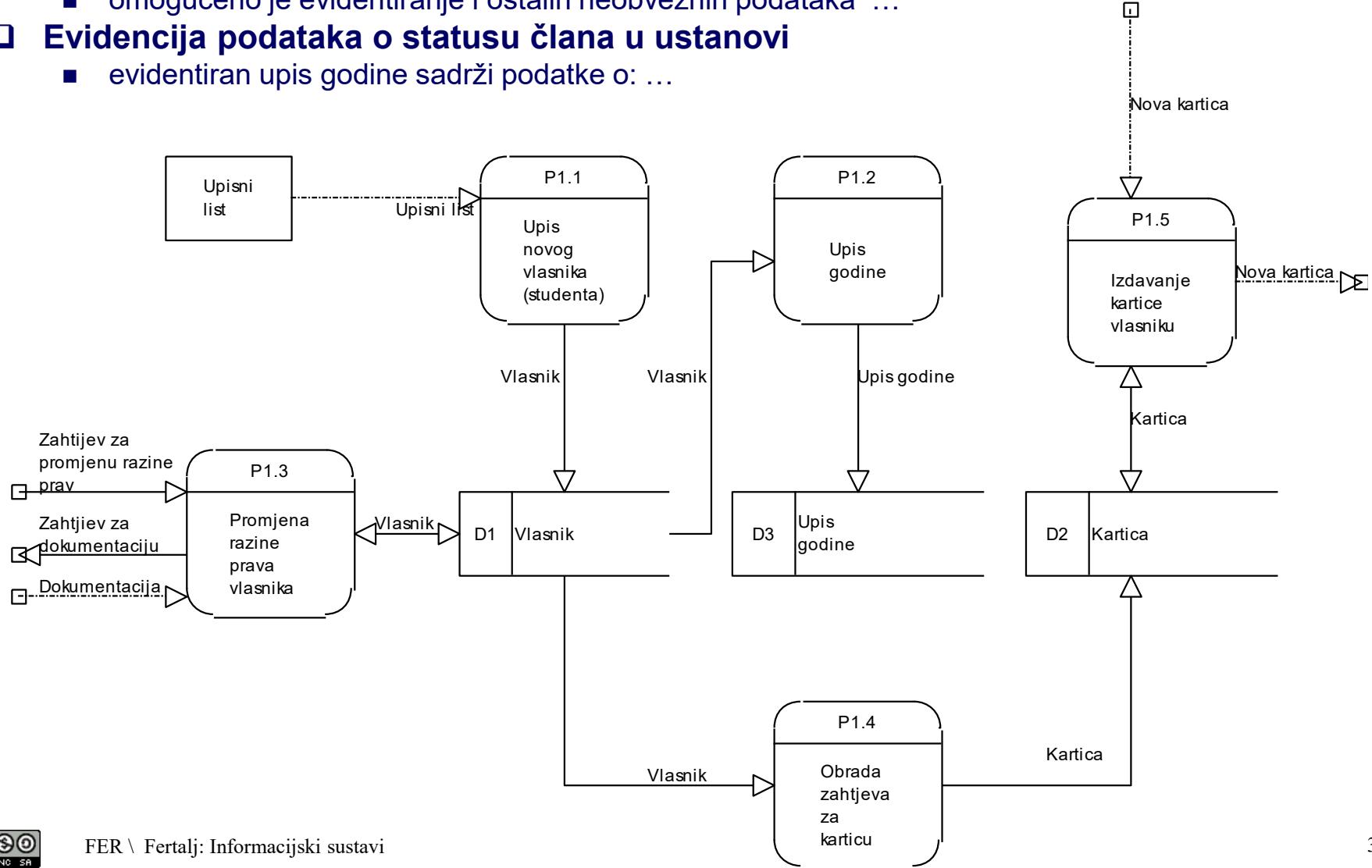
# Primjer: razrada procesa ISSP

## □ Evidencija matičnih podataka članova

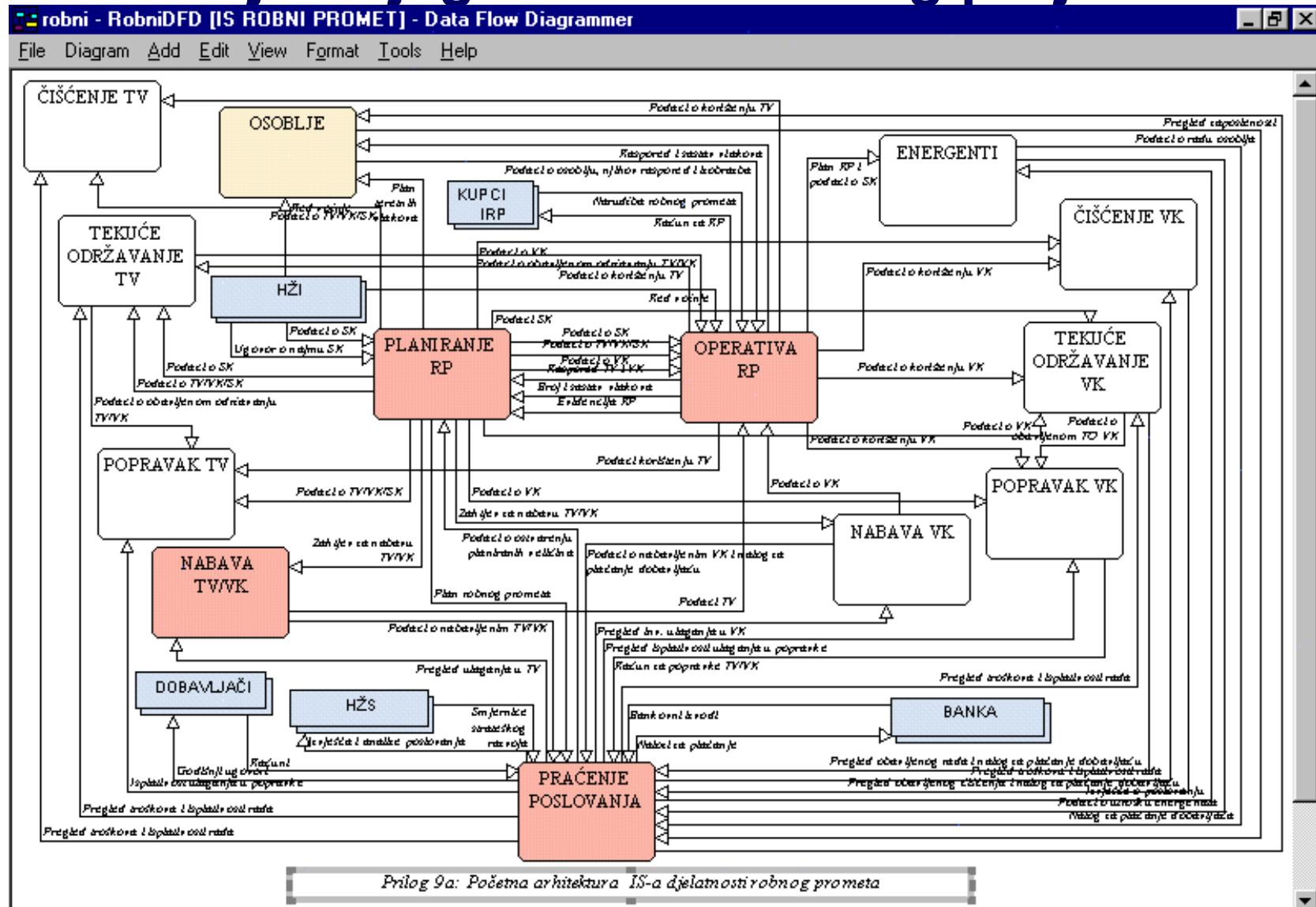
- matični podaci studenta : Ime, Prezime, JMBG, Matični broj (broj indeksa), Razina prava, JMBAG
- omogućeno je evidentiranje i ostalih neobveznih podataka ...

## □ Evidencija podataka o statusu člana u ustanovi

- evidentiran upis godine sadrži podatke o: ...



# Primjer dijagrama iz stvarnog projekta



Prilog 9a: Početna arhitektura IS-a djelatnosti robnog prometa

# 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

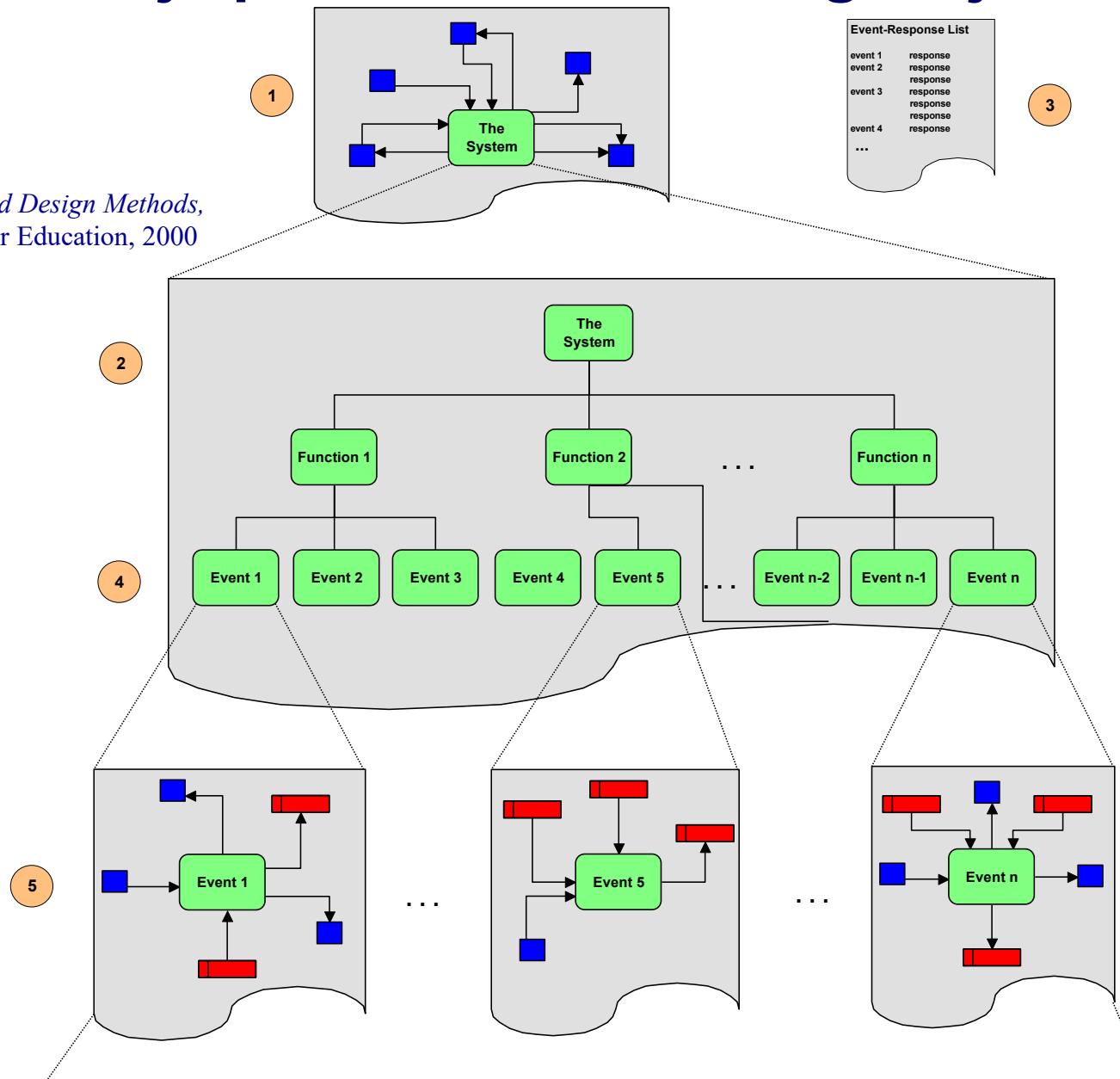
# Modeliranje procesa vođeno događajima

## □ Raspodjela događaja (event partitioning)

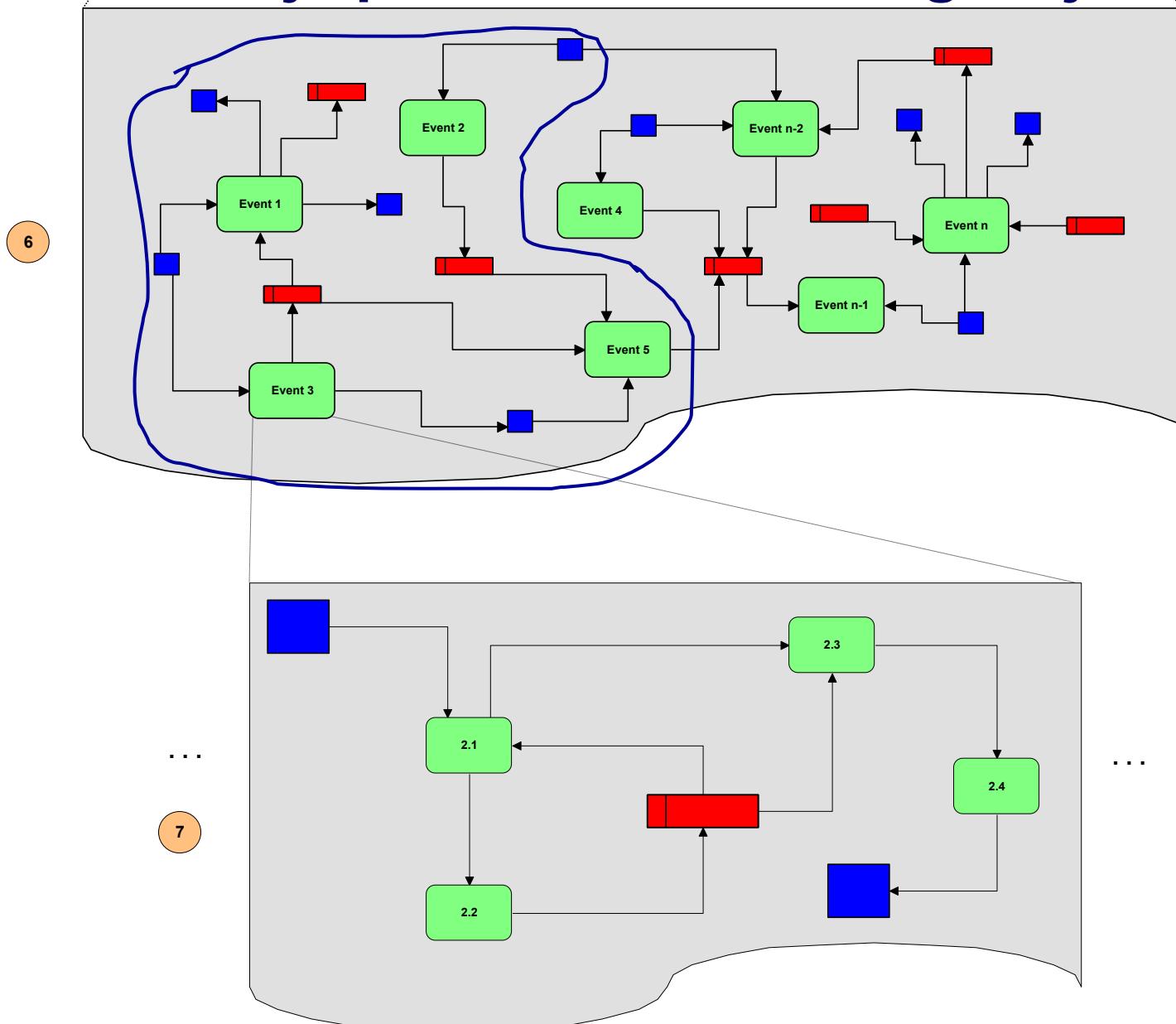
1. Izrada dijagrama konteksta sustava
  - postavljanje početnog dosega projekta
2. Izrada dijagrama funkcionalne dekompozicije
  - podjela sustava u logičke podsustave i/ili funkcije
3. Izrada popisa događaja i odziva
  - utvrđivanje poslovnih događaja na koje sustav mora odgovoriti
  - element popisa = 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 vodeno događajima (3)



# Tablica događaja u sustavu

## □ Sadržaj tablice

- 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 - trenutak koji aktivira djelovanje sustava
- Izvor - vanjski uzročnik, izvor podataka koji ulaze u sustav (za vanjske dog!)
- Aktivnosti - što sustav radi u slučaju događaja?
- Odgovor - koji/kakav izlaz izlazi iz sustava (ako postoji)
- Odredište – kamo ide odgovor?

## □ Kako nastane tablica?

# 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

Klasa 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	R	R	R	R	R	R	R	R		
P6	CRUD	R*	*	CRUD	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
P10	CRUD	R*	R*	R*	CRUD	CRUD	*	*	*	*	*	*	*	*	CRUD	*	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	R	R	R	R	R	R	R	R	
P11	CRUD	R*	R*	R*	*	*	*	*	*	*	*	*	*	*	CRUD	*	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	R	R	R	R	R	R	R	R	
P12	CRUD	R*	R*	R*	*	*	*	*	*	*	*	*	*	*	CRUD	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*	R*	R*	CRUD	*	*	*	*		
P4																					R*	R*	R*	R*	CRUD	*	*	*	*		
P3																					R*	R*	R*	R*	CRUD	*	*	*	*		
P36																					R	R									
P48																															
P45																					R	R									
P18																															
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		
P30	R			R		R														R	R	R							RU		
P25	R																			R	R	R	R	R	R	R	R	R	R		
P29	R																			R				R	R	R	R	R	RU		
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			
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	RU	R			
P50																				R				R	R	R	R	R	RU		
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	RU			
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	RU			
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	RU			
P46																				R	R	R	R	R	R	R	R	R	RU		
P31																				R	R	R	R	R	R	R	R	R	RU		
P15																				R									R		
P49																															
P26																				R	R	R	R	R	R	R	R	R	R		
P13	R																			R	R	R	R	R	R	R	R	R	R		
P27																				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		
P17	R																			R									R		
P2																				R	R	R	R	R	R	R	R	R	R		
P5																				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			
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			
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

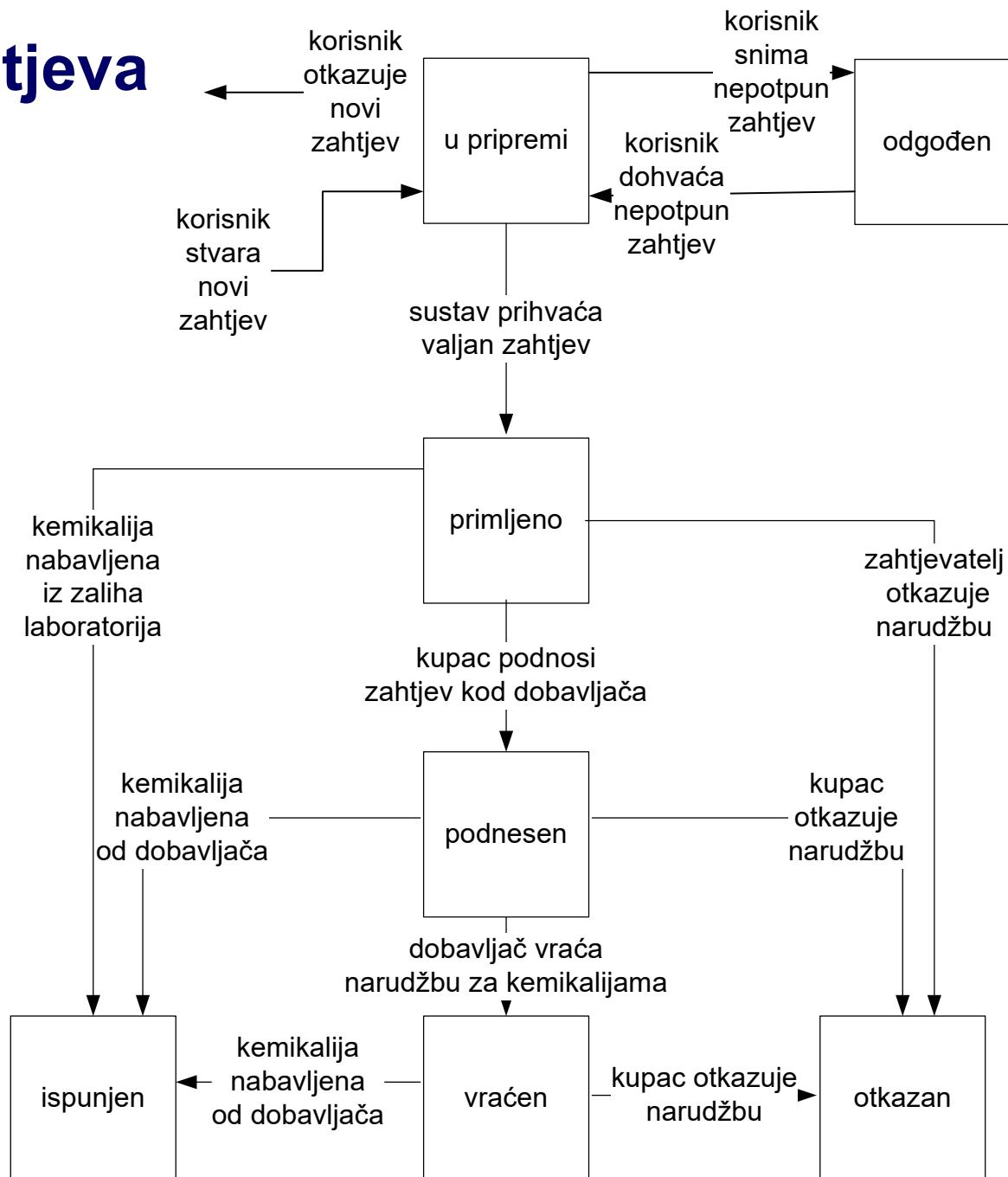
- manji sustavi mogu se prikazati jednim dijagramom
- veći sustavi razlažu se slično DTP

## □ Primjena

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

# Primjer: Stanja zahtjeva

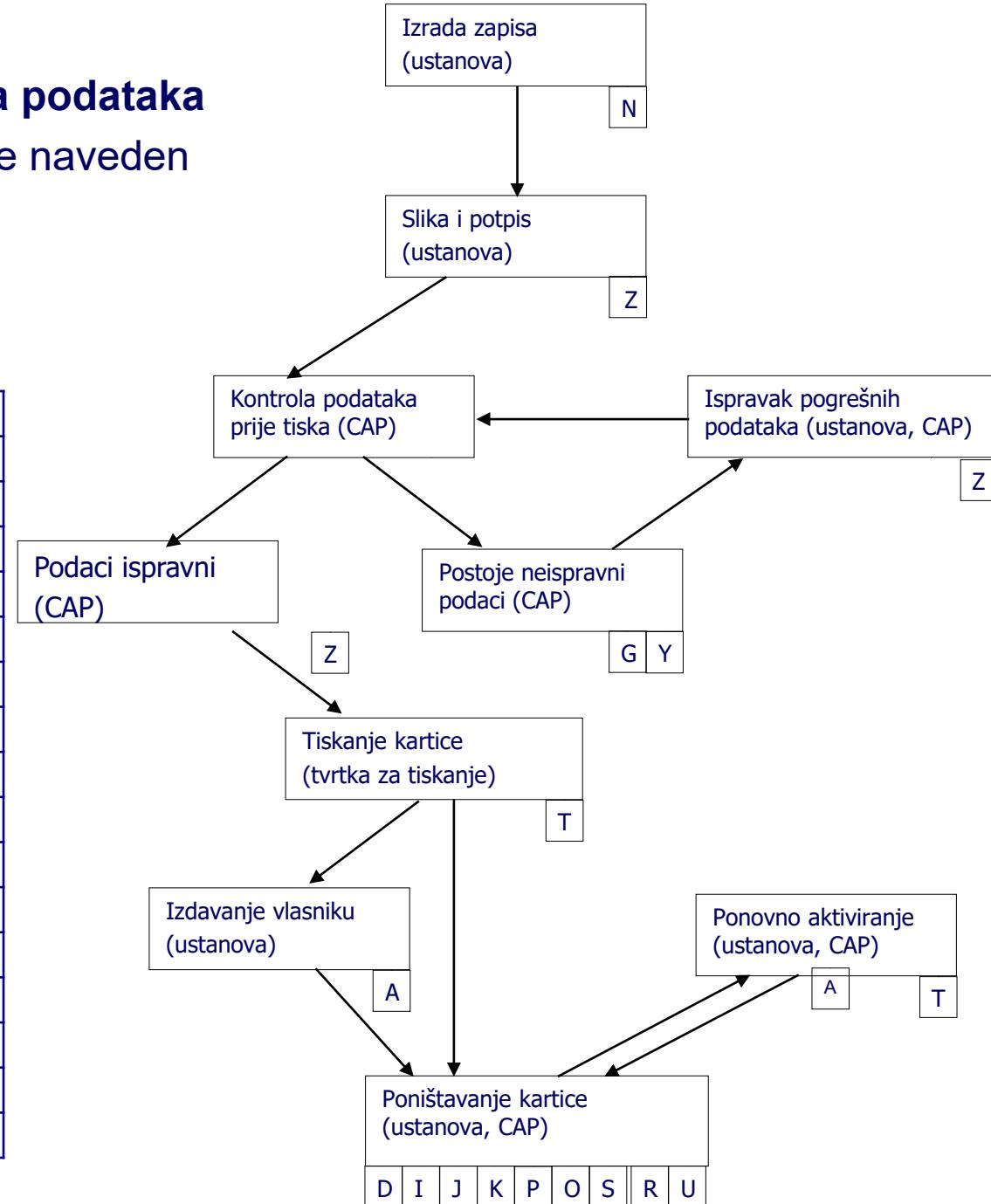
- **U pripremi:** Zahtjevatelj stvara novi zahtjev, pokretanjem f-je sustava
- **Odgoden:** Zahtjevatelj je snimio svoj rad kako bi dovršio kasnije
- **Zaprmljen:** Zahtjevatelj je predao valjan zahtjev
- **Podnesen:** dobavljač je sposoban isporučiti i kupac podnosi narudžbu kod dobavljača
- **Ispunjen:** zahtjev je zadovoljen bilo da je dostavljen iz zaliha ili nabave
- **Vraćen:** dobavljač ne može zadovoljiti narudžbu i vraća narudžbu uz mogućnost kasnije dostave
- **Otkazan:** zahtjevatelj je otkazao primljen zahtjev



## ❑ Primjer: prijelazi statusa podataka

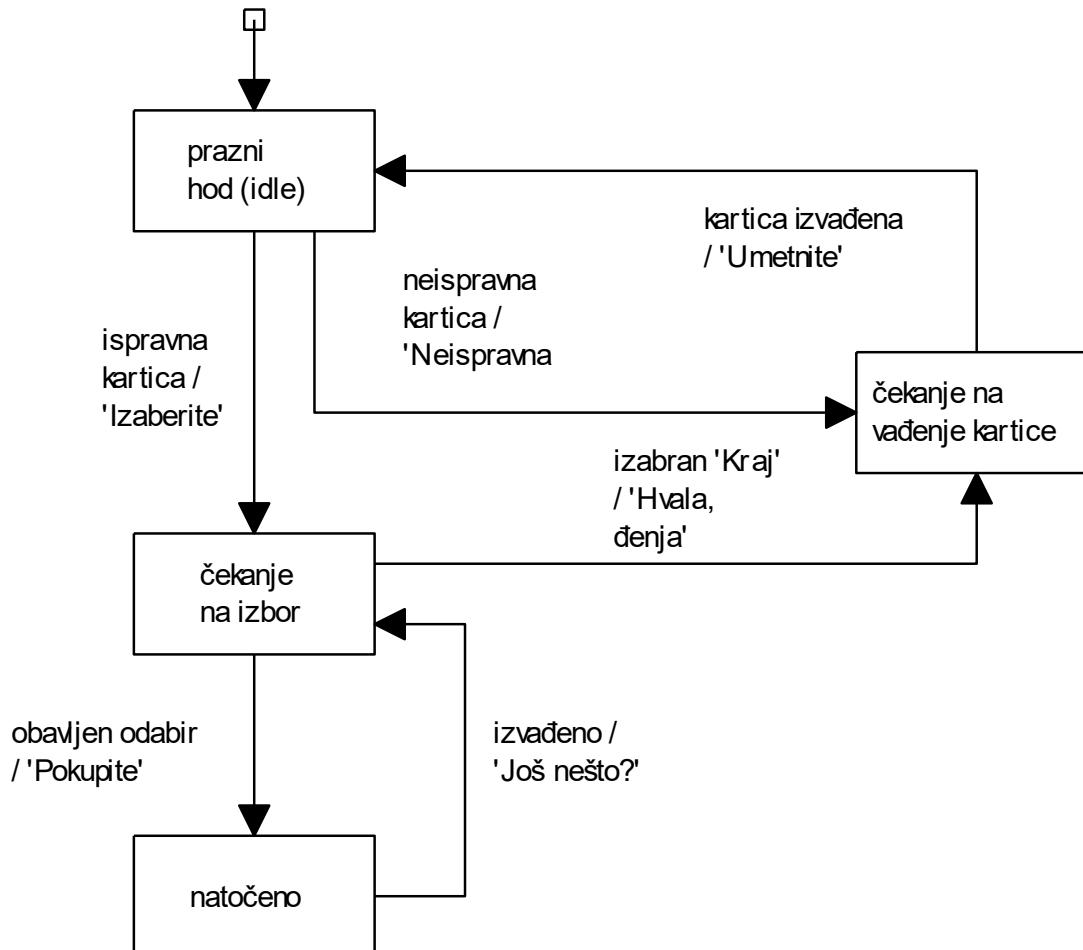
- X-ice : u zagradama je naveden nositelj promjene

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



# 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*



# Mape dijaloga

- Varijanta DPS – stanja aplikacije ovisno o akcijama korisnika
- **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 tehniku**
  - 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*

## □ Mape dijaloga mogu prikazati

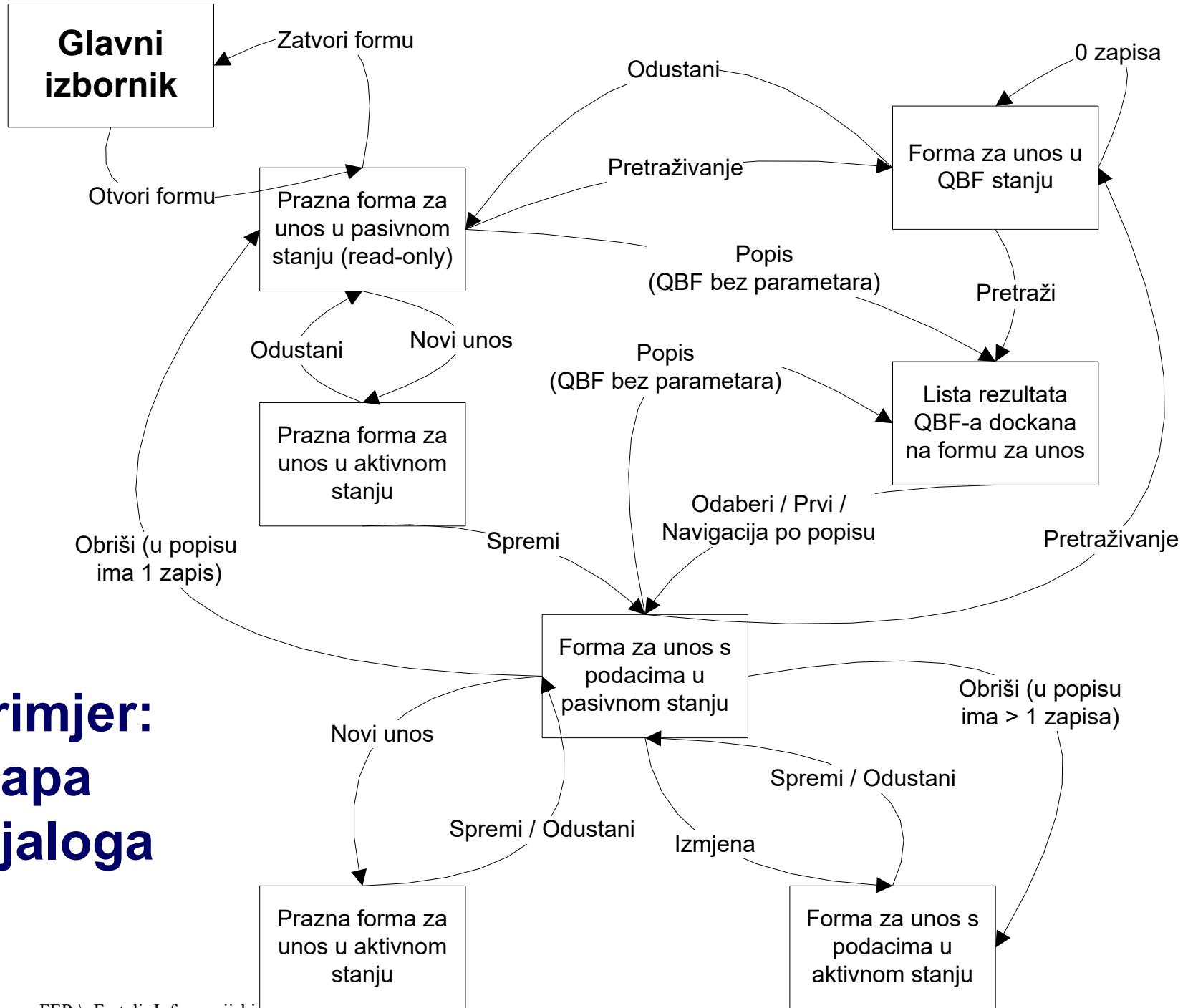
- Alternative korištenja ili pristupa istom dijelu sučelja,
- optionalnu funkcionalnost,
- posebna stanja (npr. Prikaz poruke o ...)

## □ 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 (ovisno o tehnologiji).

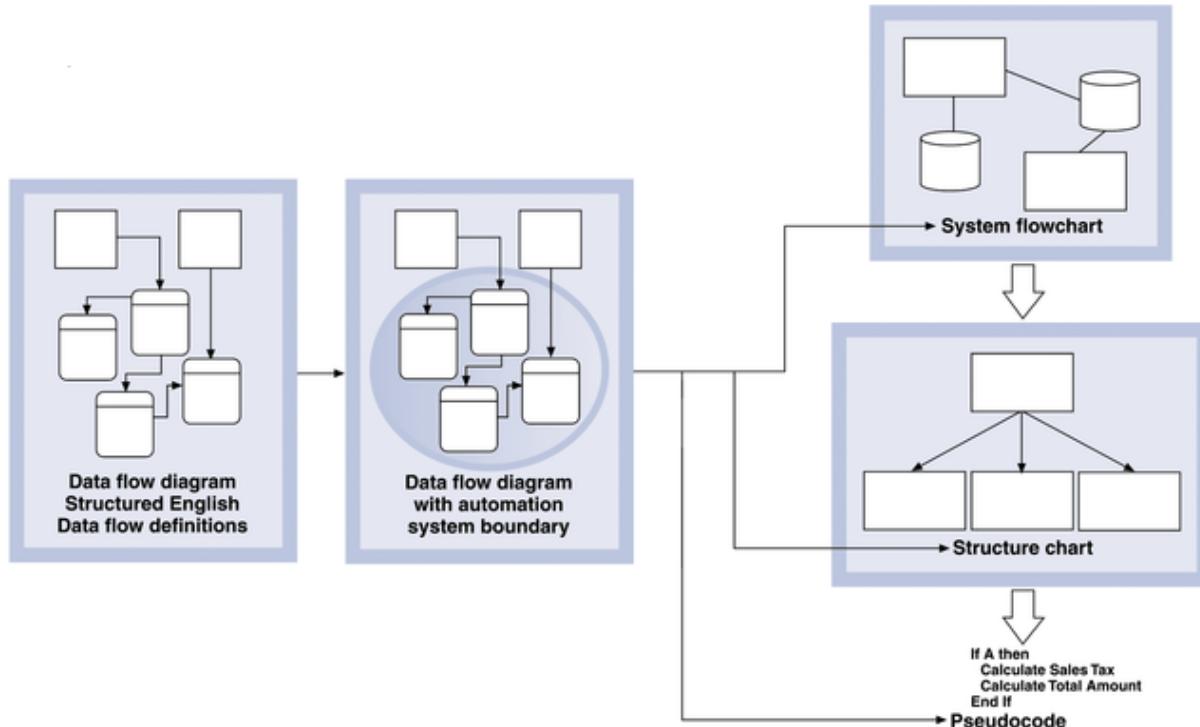


## Primjer: mapa dijaloga

# Dodaci i primjeri

## 📁\Primjeri\IS04-StrukturiranaAnalizaDizajn.pdf

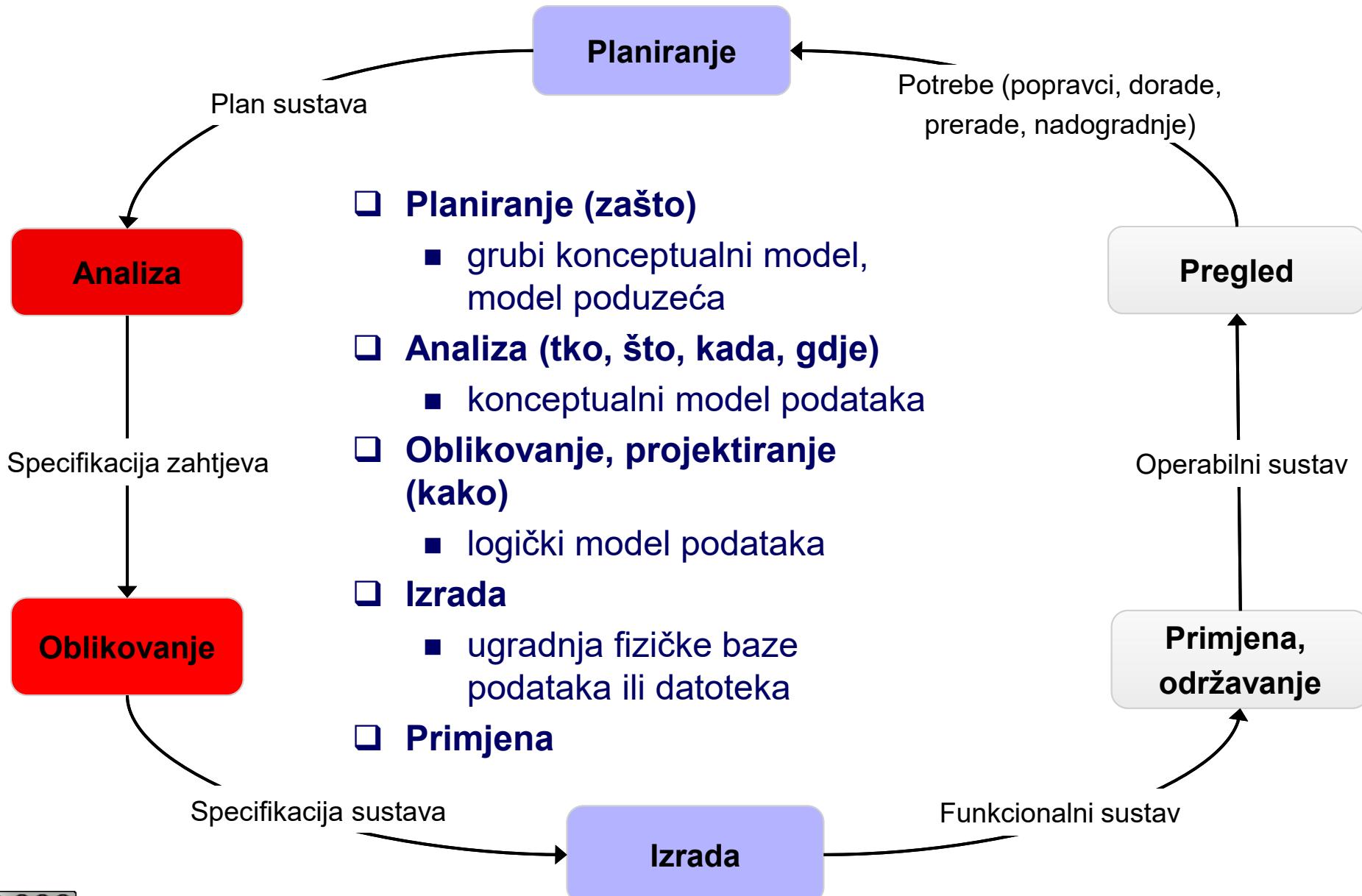
- primjer strukturiranog oblikovanja
- primjer analize izvedivosti
- primjer strukturiranog dizajna programa



# **(konceptualno) Modeliranje podataka**

**2022/23.05**

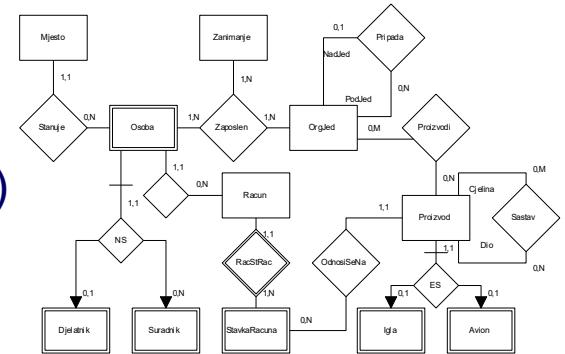
# Kontekst modeliranja podataka



# Vrste modela podataka

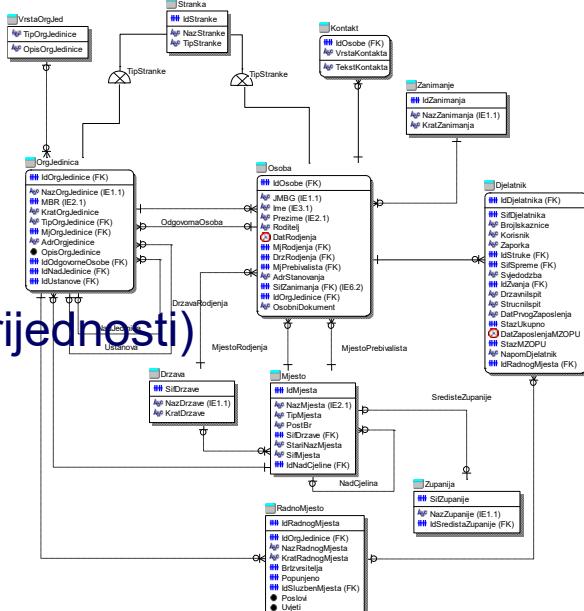
## ❑ Konceptualni

- najčešće model entiteti-veze (MEV, prikazan s ERD)
- poslovni pogled na podatke
- zanemaruje nekritične detalje
- uglavnom sadrži značajne entitete i veze "više prema više"
- može sadržavati važne atribute i ključeve



## ❑ Logički

- relacijski, postrelacijski, objektno-relacijski
- potpuno definirani atributi i domene (logički tipovi vrijednosti)
- potpuno definirani ključevi
- potpuno normalizirani entiteti (samo veze 1:N)

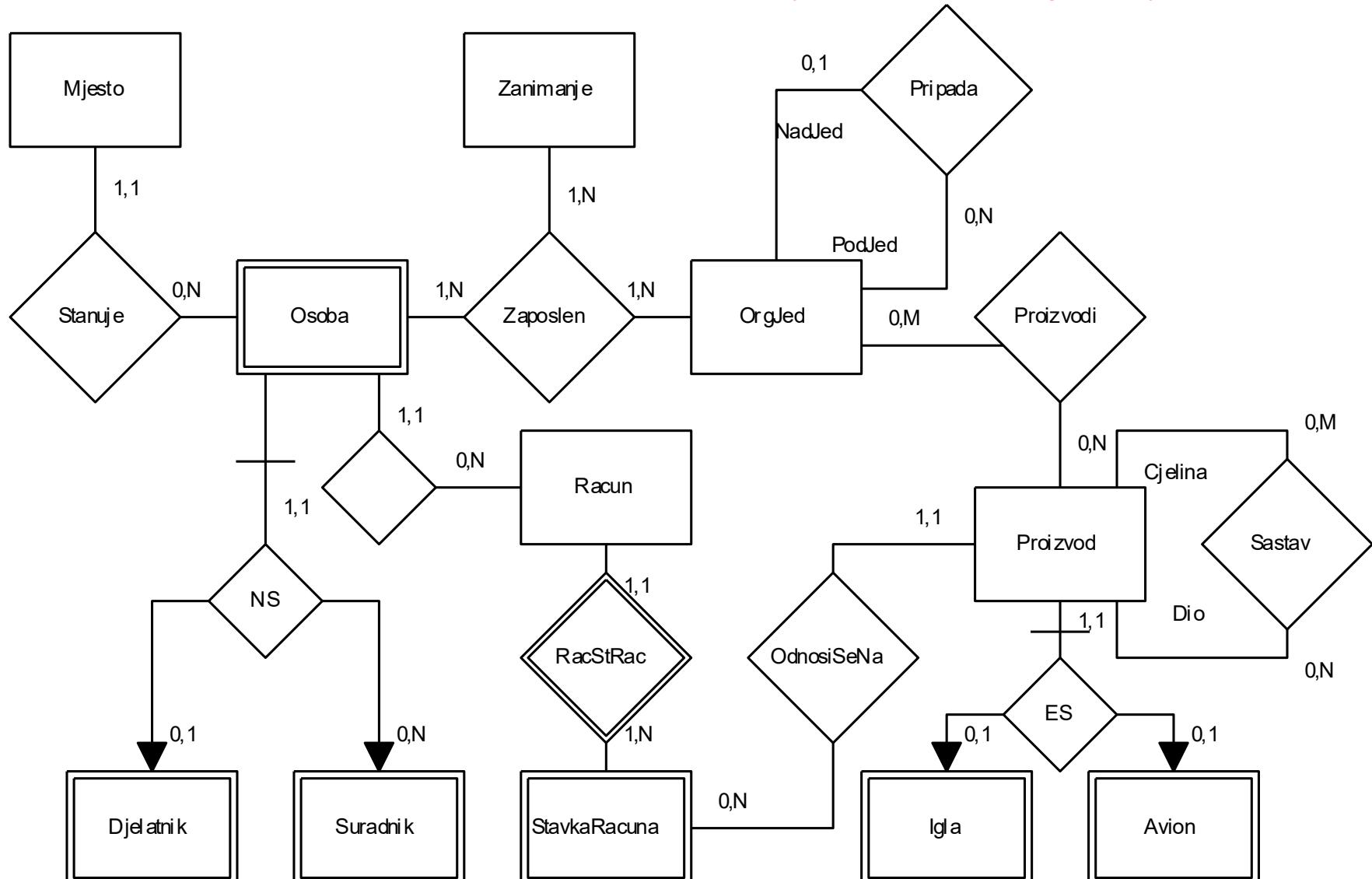


## ❑ Fizički

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

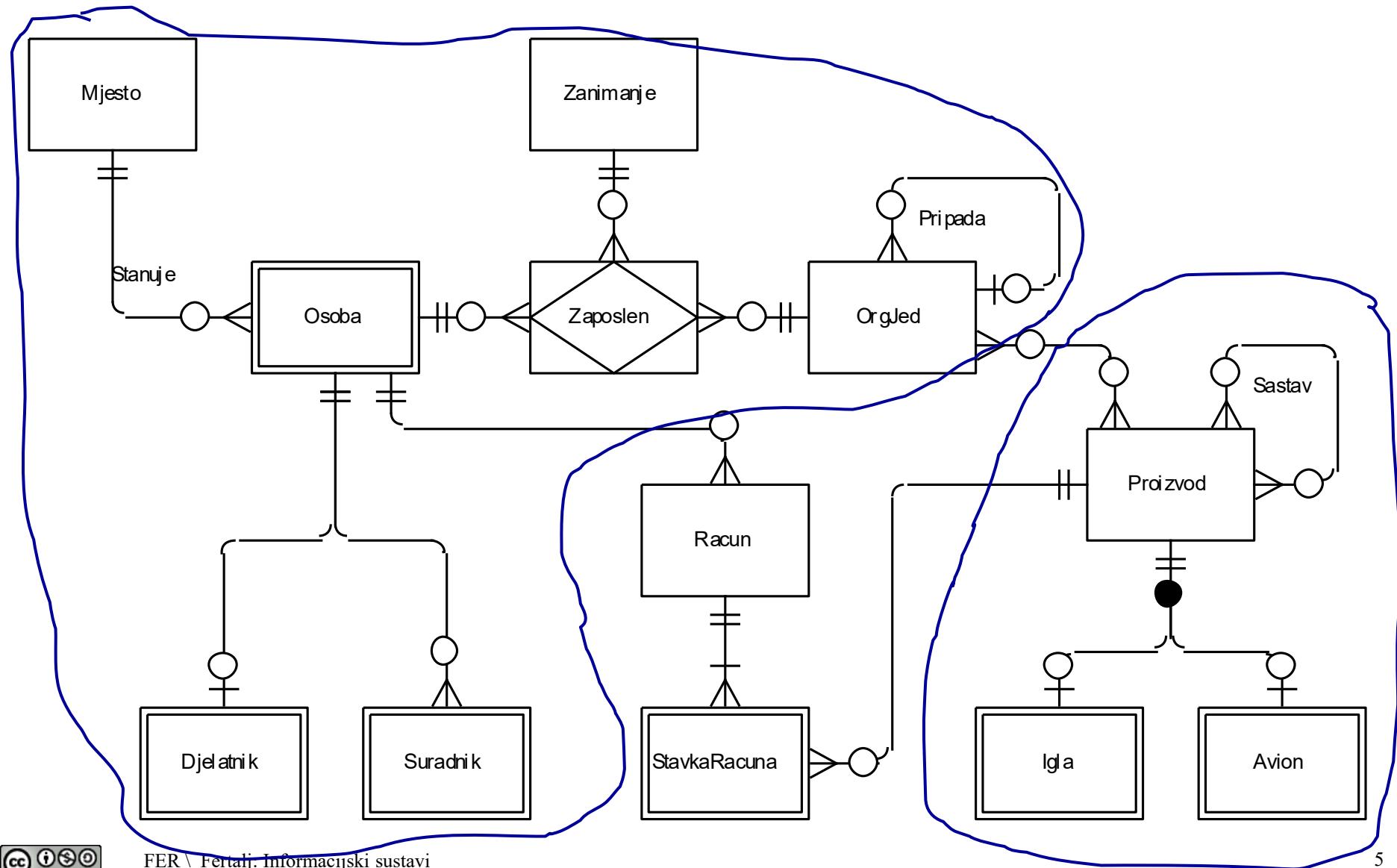
# Primjer, dijagram modela poduzeća (Chen)

<http://www.conceptdraw.com/How-To-Guide/erd-entity-relationship-diagram-symbols>



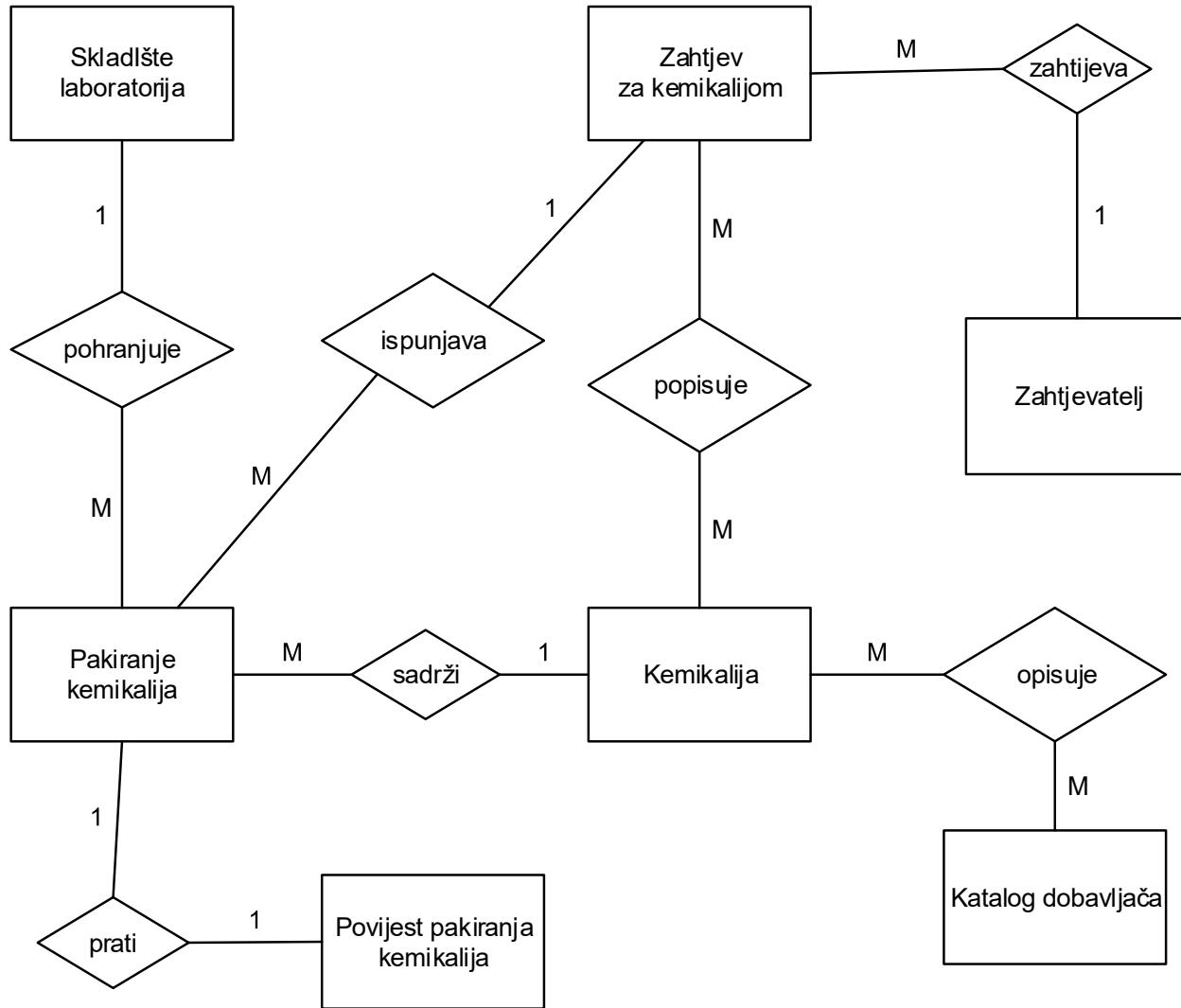
# Model poduzeća (Martin), doseg, podsustavi

<http://www.conceptdraw.com/How-To-Guide/erd-entity-relationship-diagram-symbols>



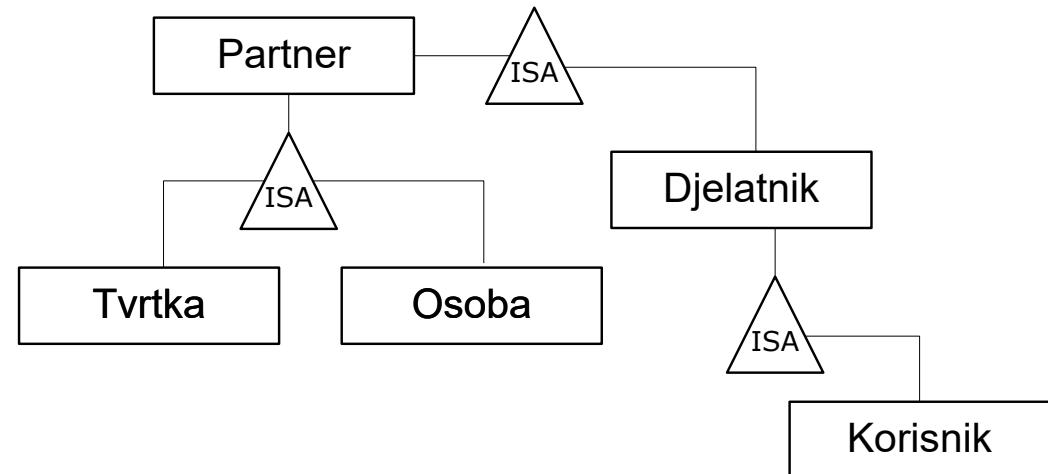
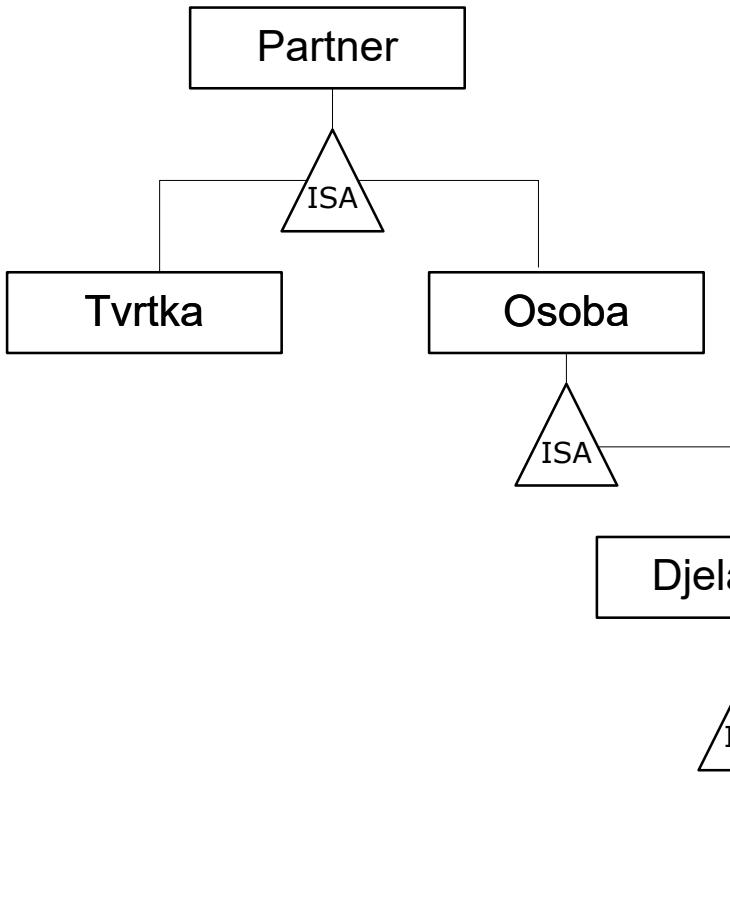
# Izrada ERD analizom izjava korisnika

"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

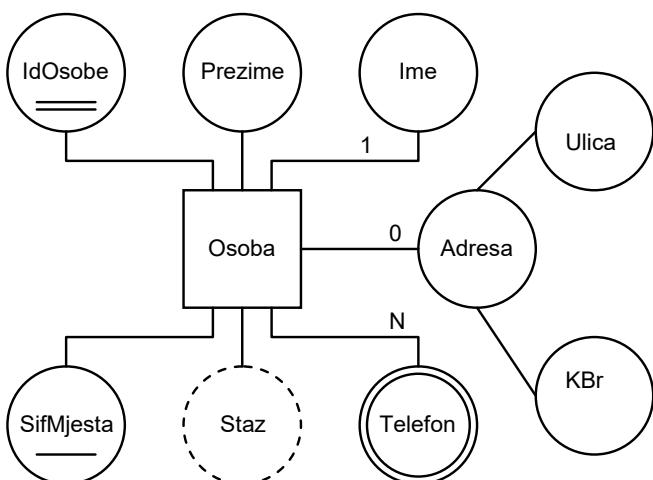
# Pretvorba modela E-V u relacijski model

## □ Entiteti

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

## □ Atributi

- kardinalnost 0 → opcionalna vrijednost (null)
- kardinalnost 1 → zahtijevana vrijednost (not null)
- kardinalnost N → više značni 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



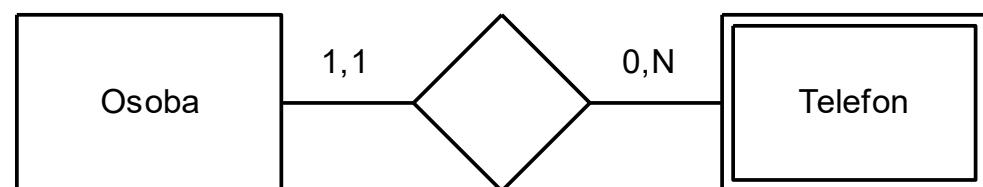
Osoba
IdOsobe
Prezime
Ime
Adresa
DatRod
Staz

# 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

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

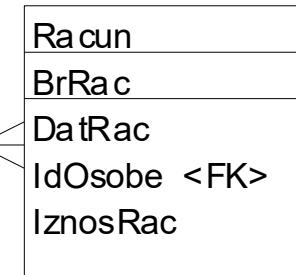
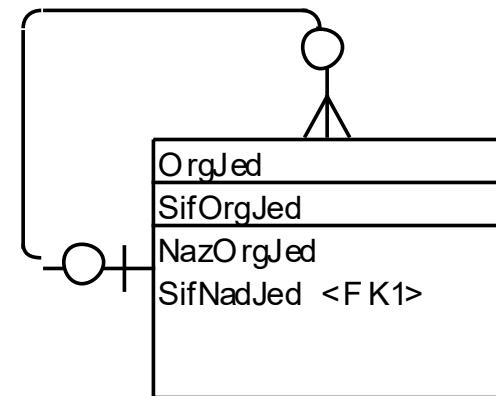
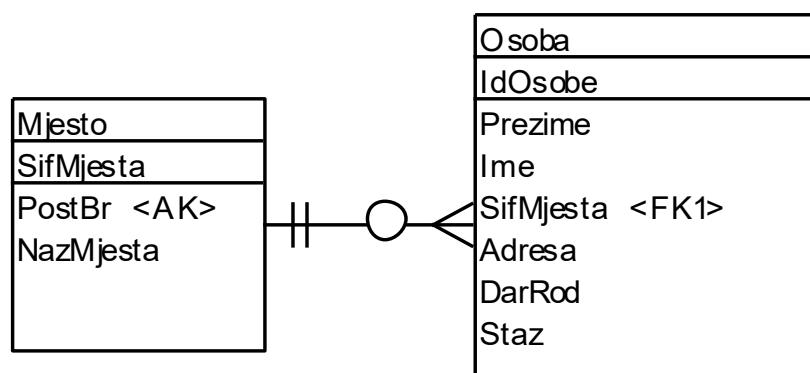
OsobaSKljuce m
IdOsobe
Prezime
Ime
Adresa
DatRod
Staz

Mjesto
SifMjesta
PostBr <AK>
NazMjesta

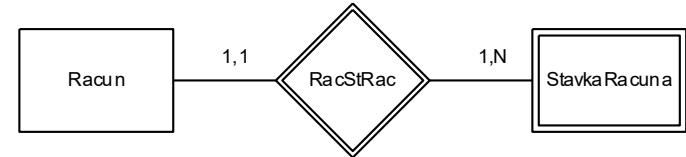
# Pretvorba binarnih veza

## □ Binarna veza 1:N → strani ključ

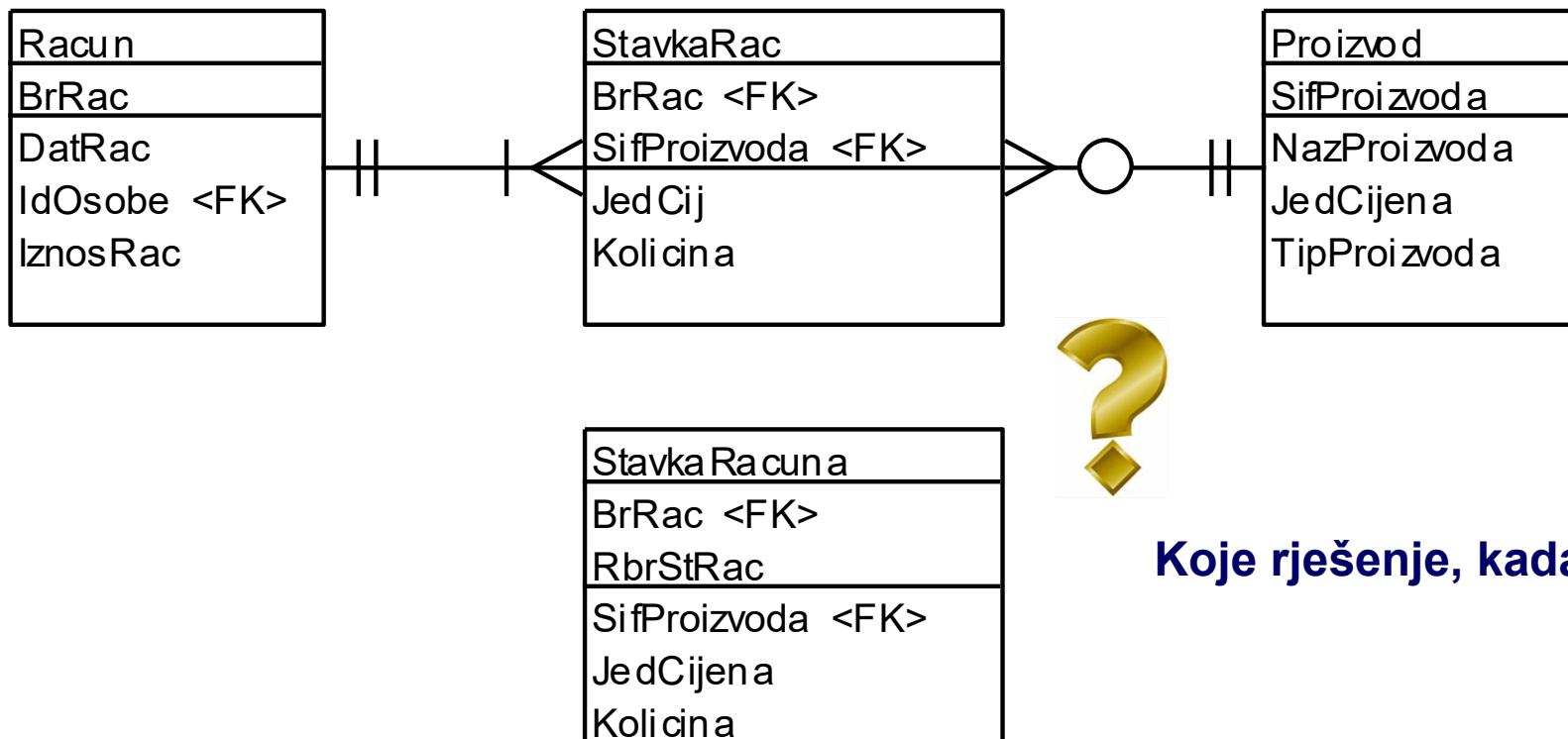
- egzistencijalno slabi entitet → obični strani ključ
  - npr. Stanuje → Osoba.SifMjesta, Pripada → OrgJed.SifNadJed, RacunOsoba → Racun.SifOsobe



# Pretvorba identifikacijskih veza

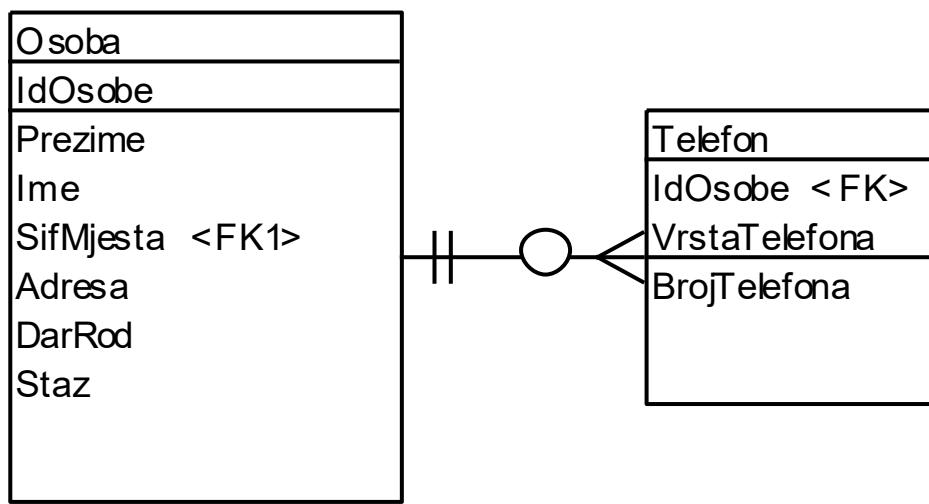
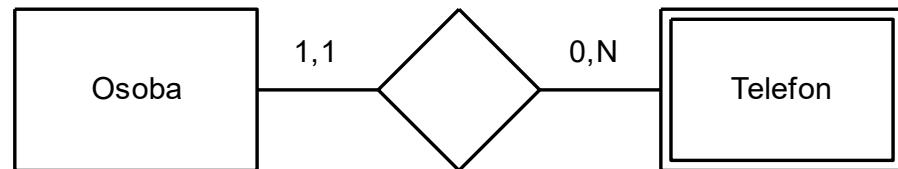


- identifikacijski slabi entitet → nasljeđuje ključ jakog entiteta
  - spojni ključ (compound key), npr. StavkaRacuna (BrRacuna, SifProizvoda, JedCijena, Kolicina)  
» ili
  - kompozitni ključ (composite key), npr. StavkaRacuna (BrRacuna, RbrStRac, SifProizvoda, JedCij, Kolicina)



# Pretvorba identifikacijskih veza (2)

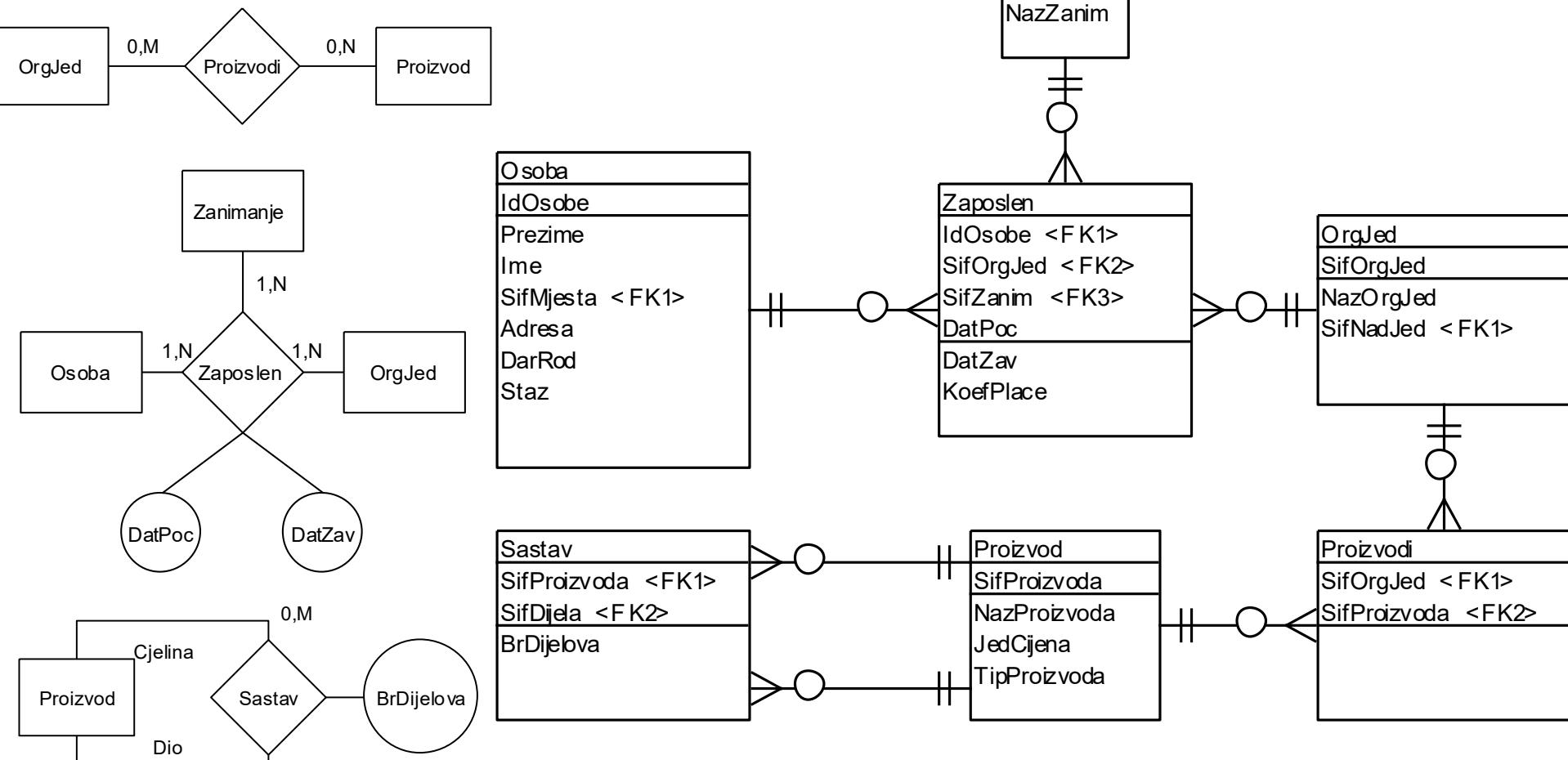
- ❑ Koliko i kakvih telefona osoba može imati ?
- ❑ Uvesti kompozitni ključ (pogledati prethodni slajd) ?
- ❑ Elegantnije rješenje ?



# Pretvorba nespecifičnih veza

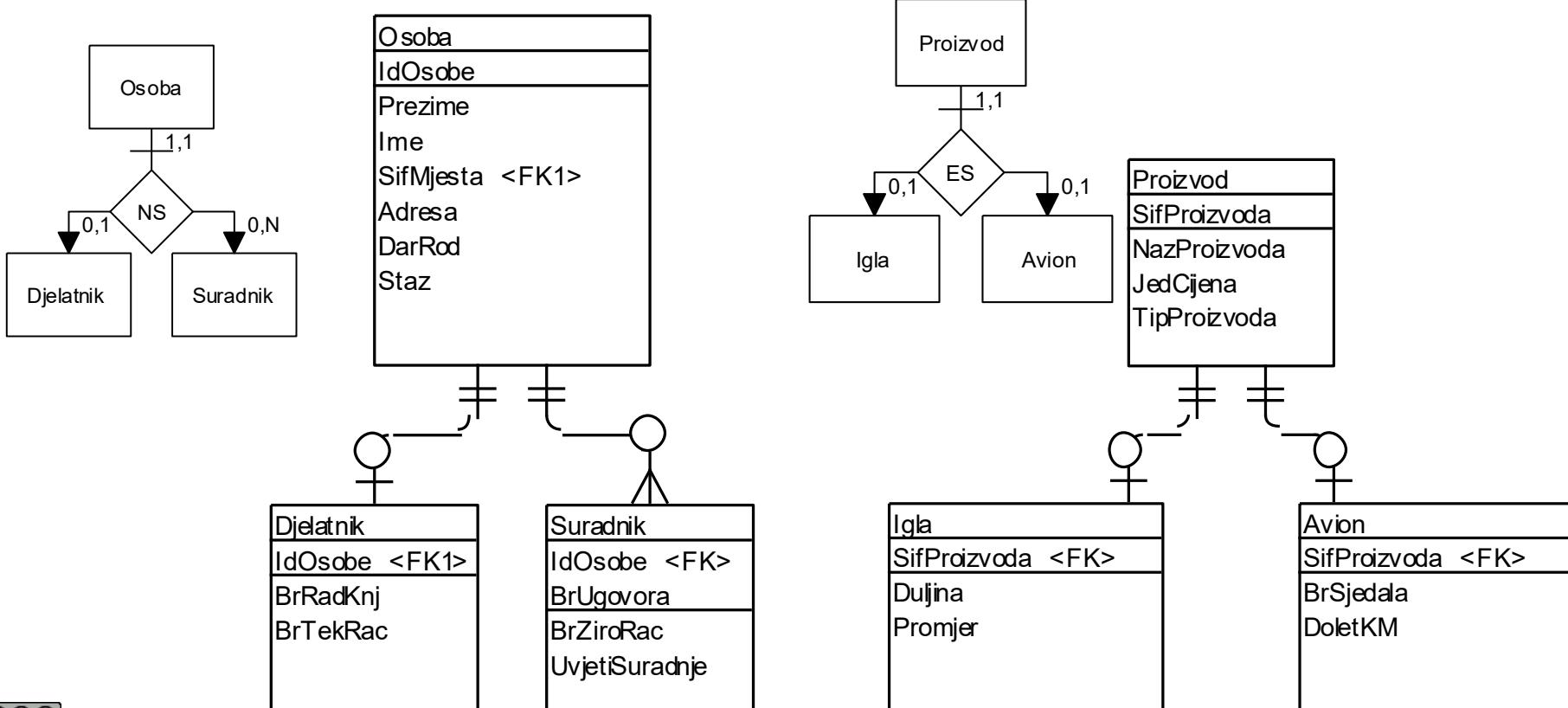
## □ Nespecifične veze → asocijativni entitet + $n$ binarnih veza 1:N

- ključ asocijativnog entiteta = unija ključeva entiteta spojenih vezom
  - određivanje ostalih ključnih atributa, ako nije bilo obavljeno ranije
- primjer: Zaposlen, Proizvodi, Sastav



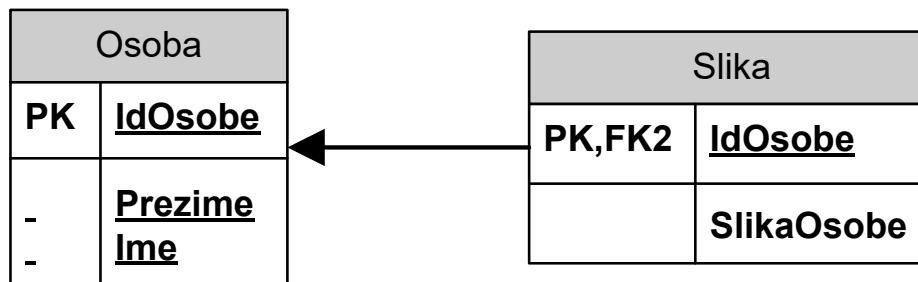
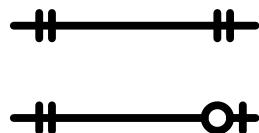
# Pretvorba specijalizacija

- 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) ?

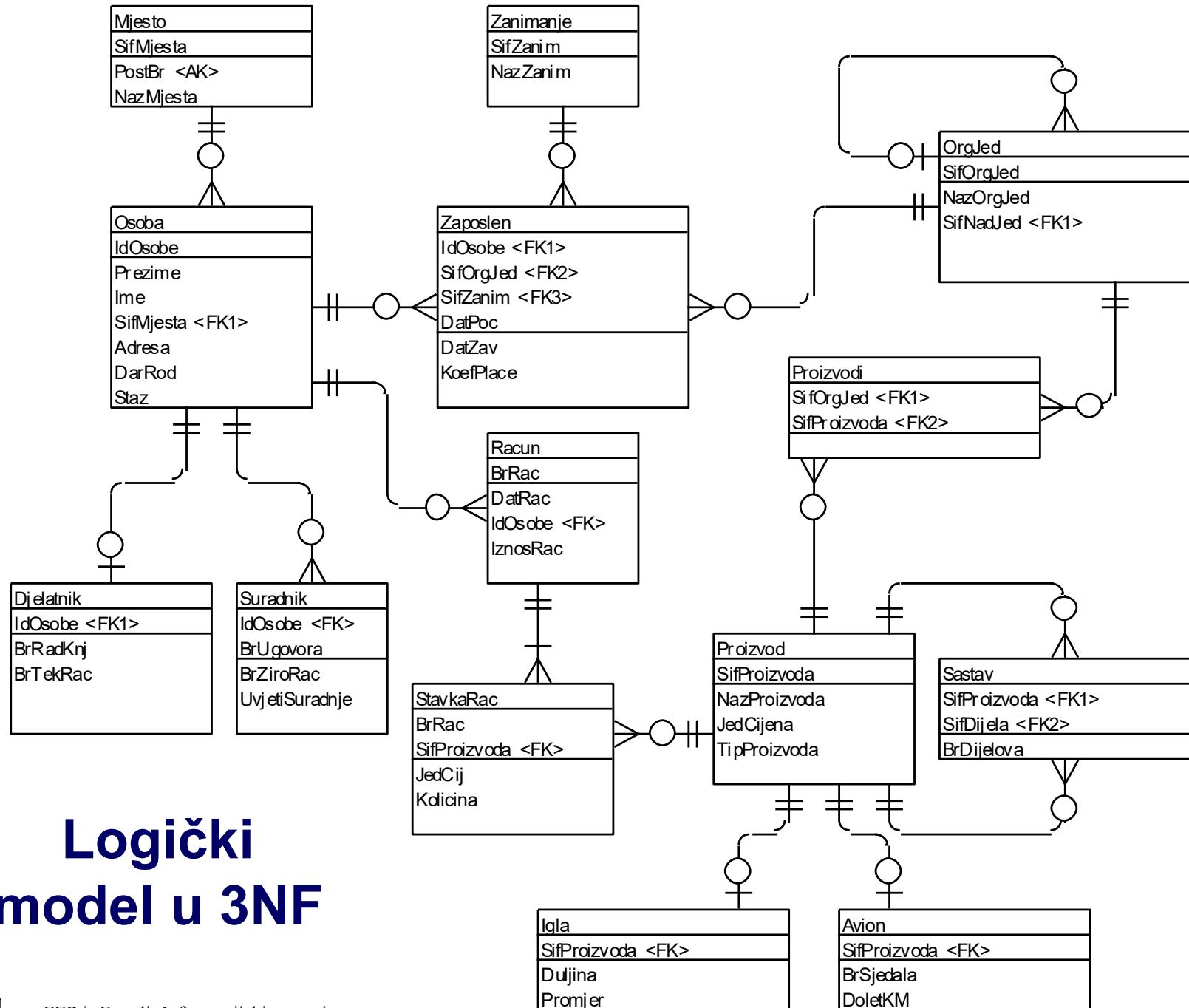


# Pretvorba ostalih binarnih veza

- 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)

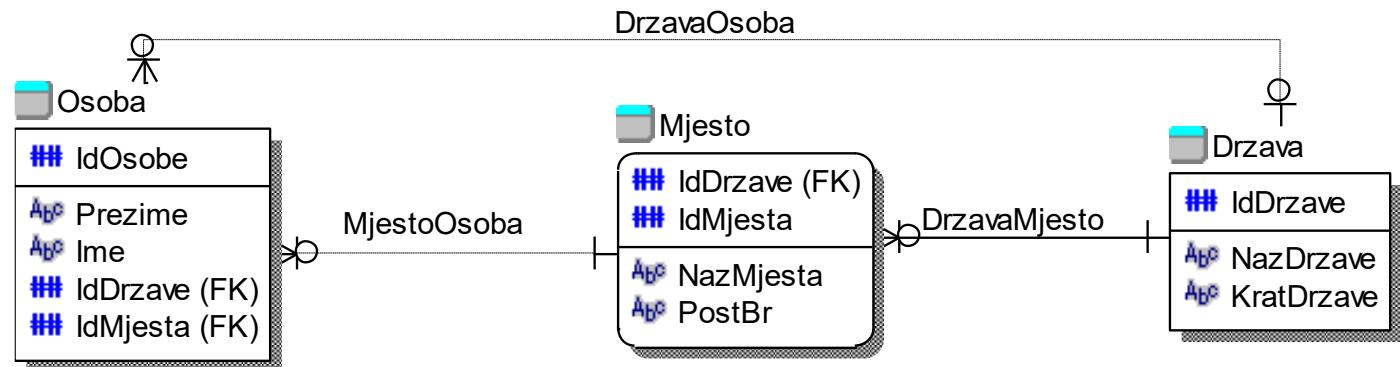
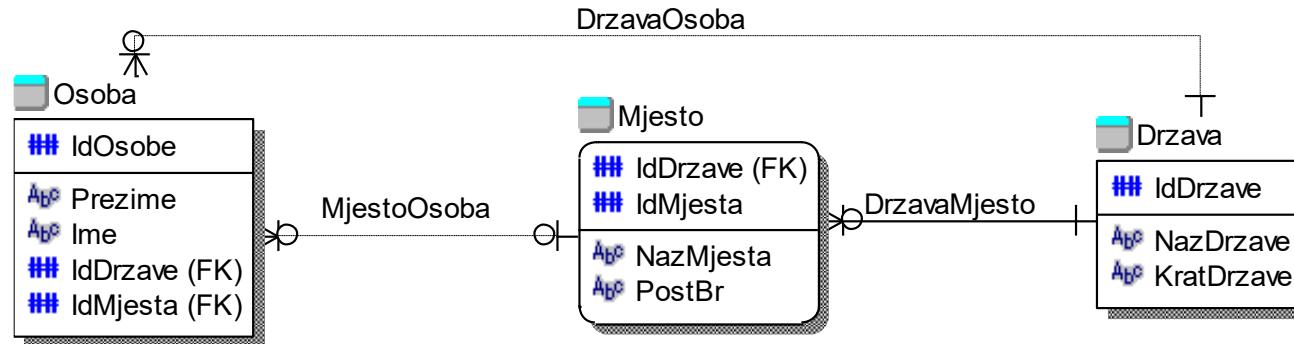


# Logički model u 3NF

# Preporuke za izradu modela podataka

- **Sinkronizacija s modelom procesa (DFD – ERD)**
  - 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)
    - ukloniti balansirane veze 1,1:1,1 (ako je moguće!)
    - pripaziti na nebalansirane veze 0,1:1,1 (zavisnost entiteta)
    - cirkularne reference
  - izbaciti izvedene atributе
    - mogući gubitak informacije o potrebnim izvedenim/izračunatim poljima

# Uklanjanje veza koje se daju izvesti iz drugih ?

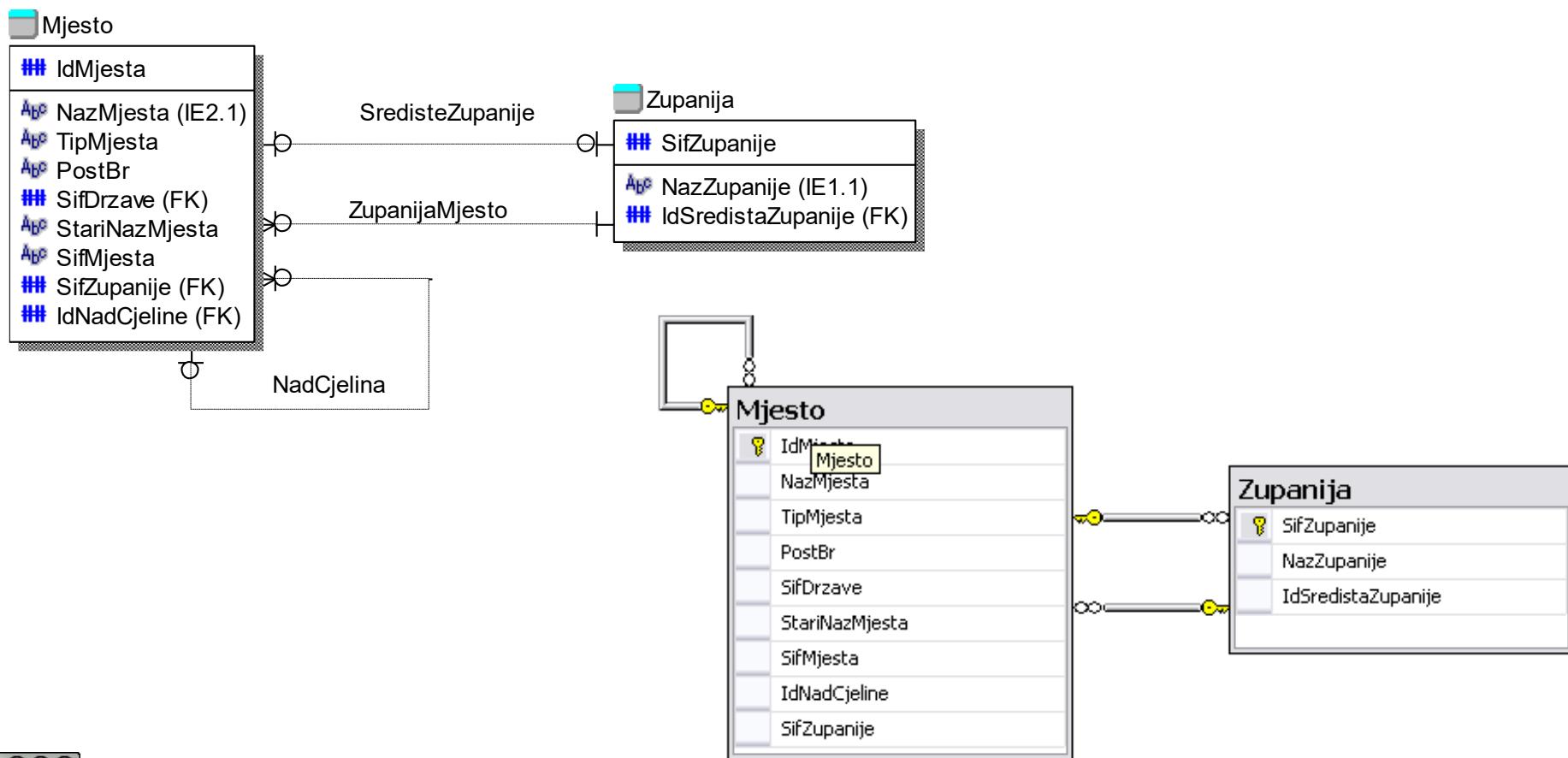


Osoba form fields:

OID	Enter Text	
Ime prezime	Enter Text	
Država	Id. države	Naziv države
Mjesto	Id. mjesta	Naziv mjesta

# Problem paralelnih i cirkularnih veza - ukloniti?

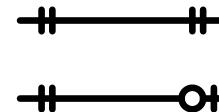
- primjer: Zupanija 1:N Mjesto i Zupanija 1:1 MjestoSrediste
- slično: OrgJedinica 1:N Djelatnik i OrgJedinica 1:1 DjelatnikUpravitelj
- problem: zahtijevanost vrijednosti stranih ključeva
- problem: realizacija veze 1:1, redundantna središta županija



# Balansirane i nebalansirane binarne veze

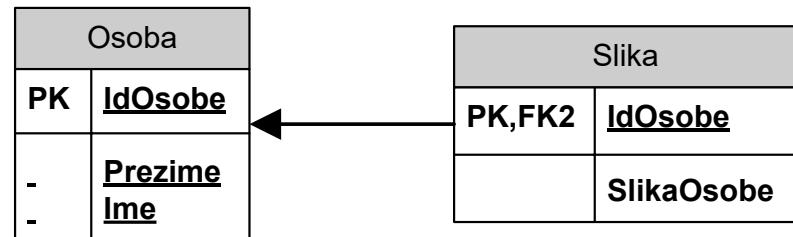
## □ Udruživanje u zajedničku tablicu

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



## □ Razlozi za udruživanje

- jednostavniji dohvat podataka
- ubrzanje upita



## □ Razlozi za razdvajanje

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

# 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 - DDMMGGGRNNNK
- Primjer: IBAN - HRcc AAAA AAAB BBBB BBBB B
- Oznake proizvoda, pr. auto gume www/aa Rnn Iss

## □ Hijerarhijski kodovi

- Podjela u grupe, podgrupe itd.

## □ Šifre s kontrolnom znamenkom

- Zadnja znamenka se izračunava iz svih ostalih
- Detektira se pogreška u znamenci ili zamjena mjesta dvije znamenke
- JMBG, "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 individuumu	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),



# 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

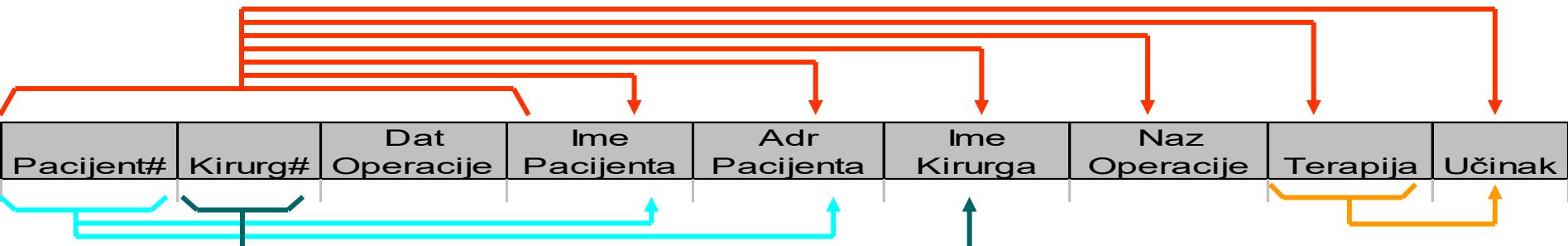
# Normalizacija

## □ Normalizacija

- postupak strukturiranja sheme RBP 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 performans, 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 zahtjeva aplikacijsko upravljanje integritetom podataka koje je "skupo,"
  - proučiti vlastitu BP, denormalizirati mjestimično i oprezno
  - probati i drugo: izračunata polja, pohranjene funkcije, indekse, optimizaciju upita, forsiranje plana izvođenja, fizičko skaliranje

# Podešavanje (umjesto denormalizacije)

## □ Postavljanje indeksa

- primarni ključevi implicitno indeksirani
  - ali im se može kreirati sortirani (*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

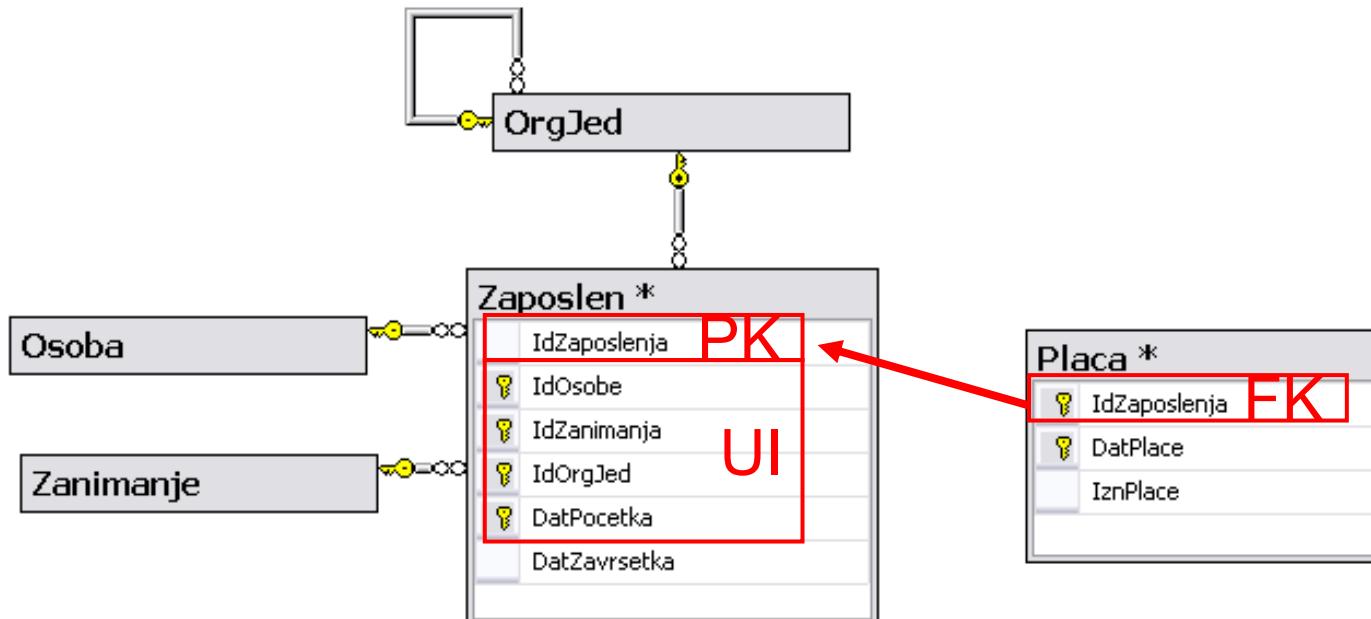
## □ NULL strani ključevi - vanjska spajanja koja ne koriste indekse

- 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.  $\geq 3$ ) 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 a Windows application interface. At the top, there is a navigation bar with icons for back, forward, search, and other functions. Below the navigation bar, there is a section labeled "Prethodni:" followed by a dropdown menu. To the right of the dropdown is a "ComboBox Tasks" panel with the following settings:

- Use data bound items
- Data Binding Mode**
- Data Source: prethDokumentBindingSource
- Display Member: NazDokumenta
- Value Member: IdDokumenta
- Selected Value: dokumentBindingSource - I

Below the dropdown, there is a toolbar with buttons for navigating through the list. The status bar at the bottom displays the text "ileDbDataAdapterArtikl", "oleDbConnection1", and "oleDbDat".

# "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 ap. Firma)

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

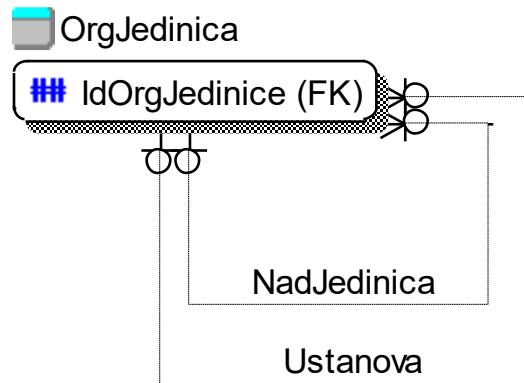
- pohranjenom procedurom, po potrebi uz IdDokumenta kao argument uz:
  - WHERE Dokument.IdDokumenta = @IdDokumenta
- okidačem tablice Stavka uz:
  - WHERE Dokument.IdDokumenta IN Select (IdDokumenta FROM inserted)

# 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.  $\geq$  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



# 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'

# Preopterećeni tipovi podataka (overloaded datatypes)

## □ Udruživanje šifrarnika

- Općenito: Sifrarnik<i> = @Sifra<i> <datatype>, Naziv<i>
  - pr. Mjesto = @PostBr INT, NazMjesta VARCHAR
  - pr. Valuta = @KratValute VARCHAR, NazValute VARCHAR
  - pr. Koeficijent = @SifraKoef, VrijKoef 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**

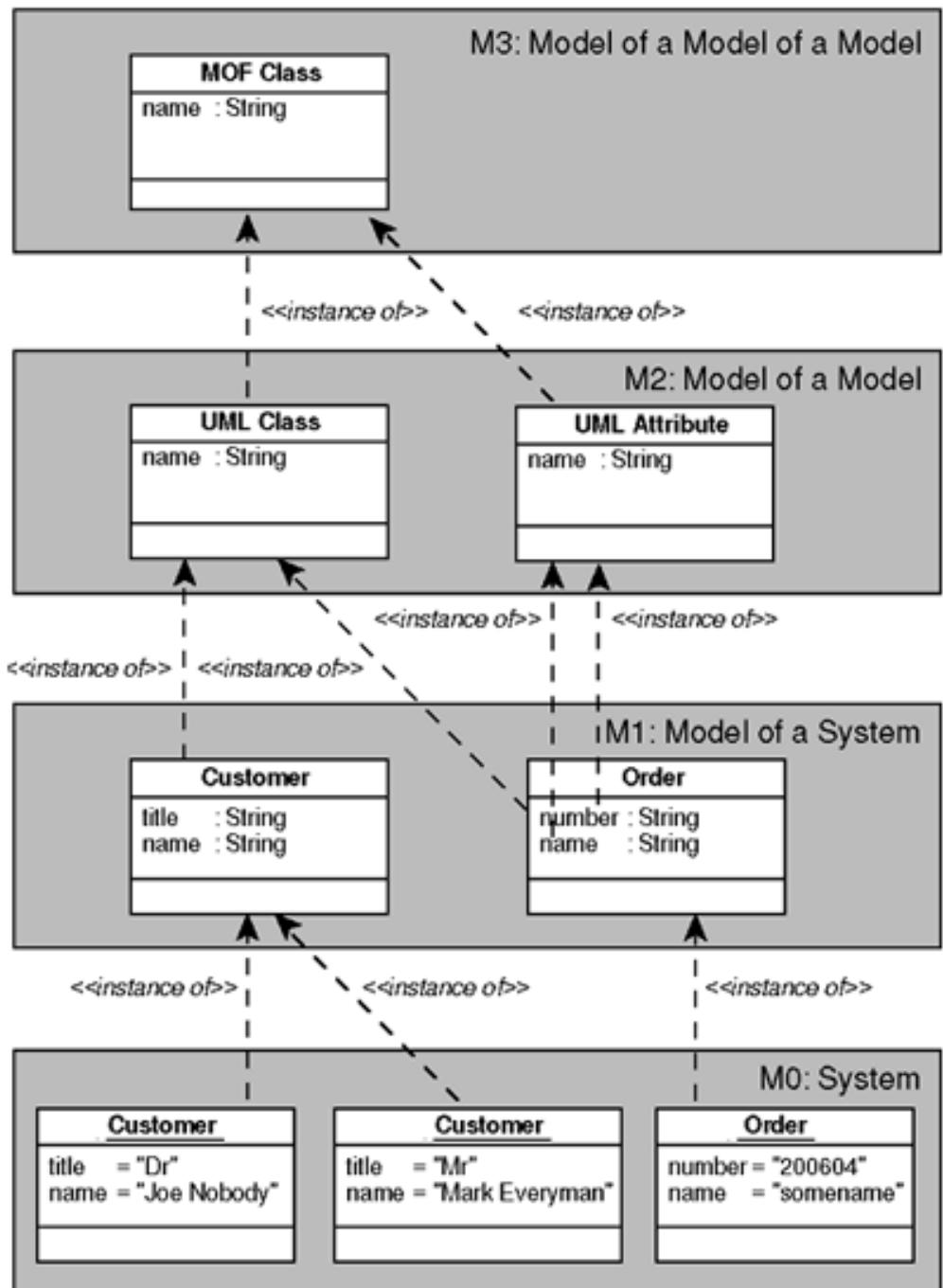
# Meta-modeliranje

## □ Meta-modeliranje = modeliranje podataka o podacima

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

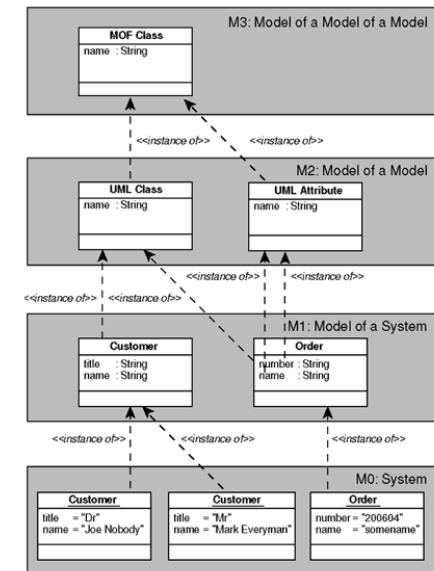
## □ Primjene

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



# Opisivanje i pohrana (meta)modela

- Metamodel – model (za opisivanje drugih) modela
- Metamodeliranje – definicija i upotreba metametashema
  - Metashema – instanca metametasheme
  - Shema – instanca metasheme
- Primjer
  - 3 MOF – MetaMetaShema (shema metaModela)
  - 2 UML – MetaShema (shema modela)
  - 1 Model – Shema (shema podatka)
  - 0 System – Objekt (podatak)
- 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, ...



# Modelom vođeno ...

## □ Model-Based Engineering/Development (MBE)

- „mekša” verzija MDE, u kojoj su modeli važni ali ne ključni / automatizirani

## □ Model-Driven Engineering (MDE)

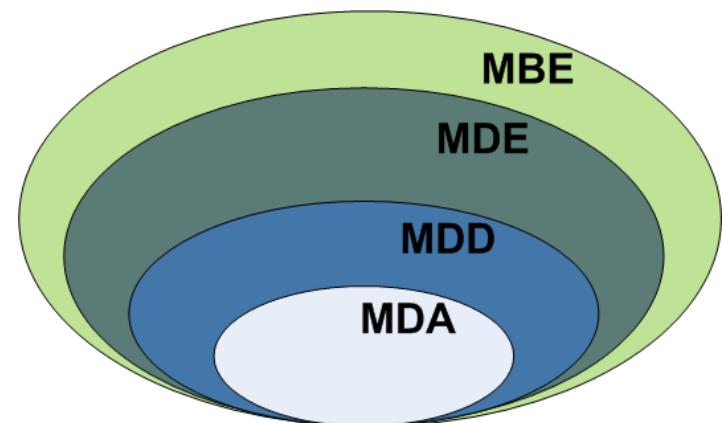
- Nadilazi razvojni proces, uključuje i druge na modelu zasnovane aktivnosti, npr. model-based evolution, model-driven reverse engineering

## □ Model-Driven Development (MDD)

- Modeli ključni artefakt - „vode” proces, forward engineering (no/low code)

## □ Model-Driven Architecture (MDA)

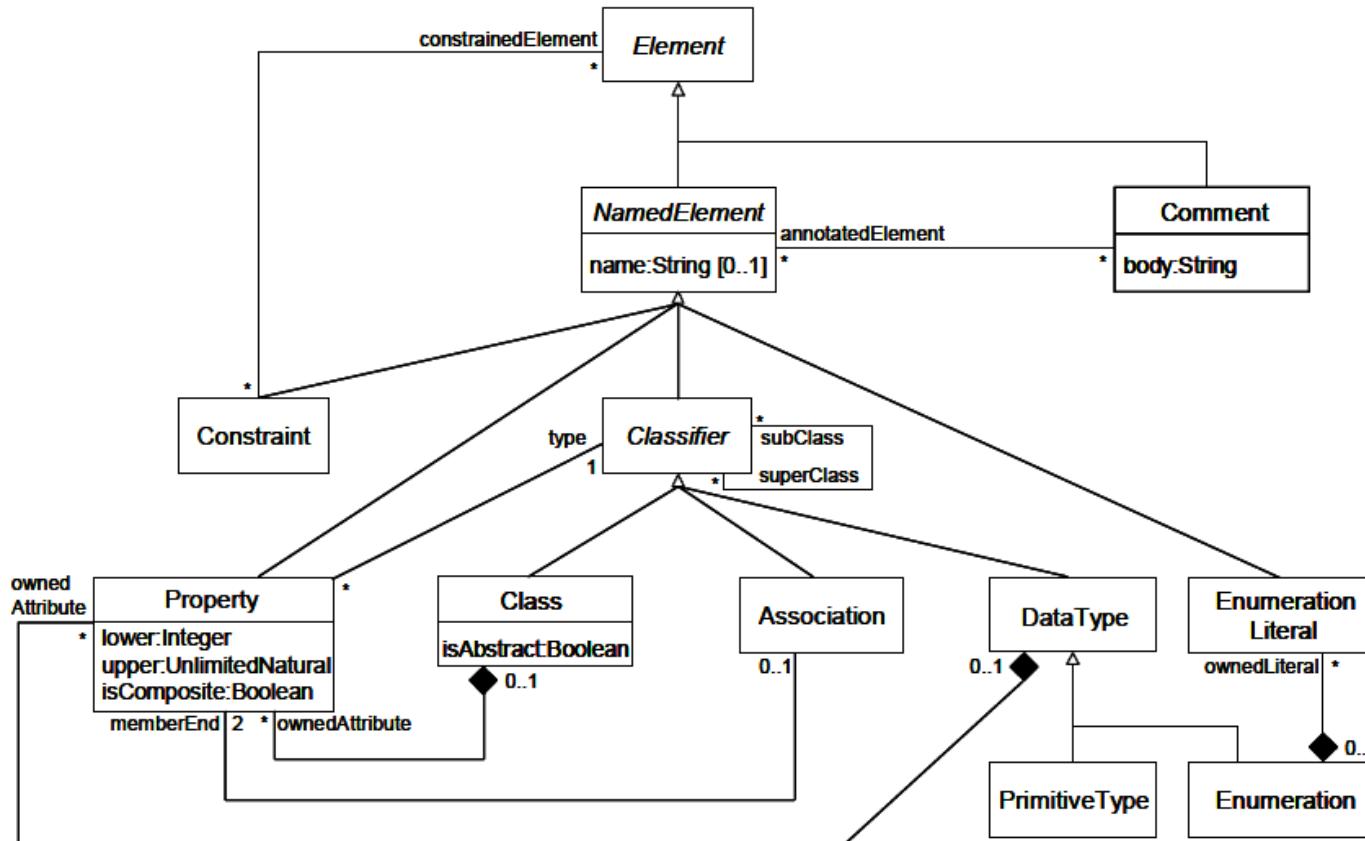
- OMG-ova vizija ili standard MDD



<http://modeling-languages.com/clarifying-concepts-mbe-vs-mde-vs-mdd-vs-mdal/>

# [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, pojednostavnjena verzija MOF



# Životni ciklus MDA

## □ CIM – računalno neovisan

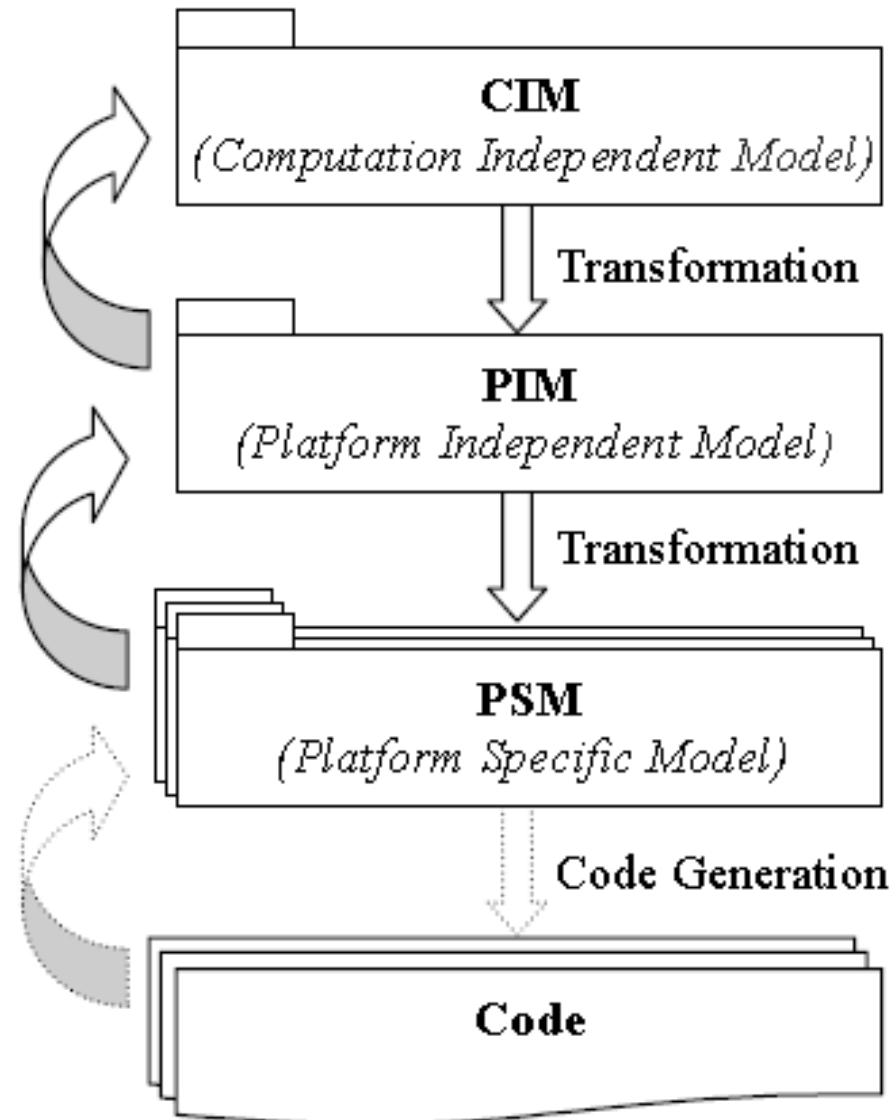
- Model domene, poslovni model

## □ PIM – platformski neovisan

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

## □ PSM - platformski određen

- implementacija



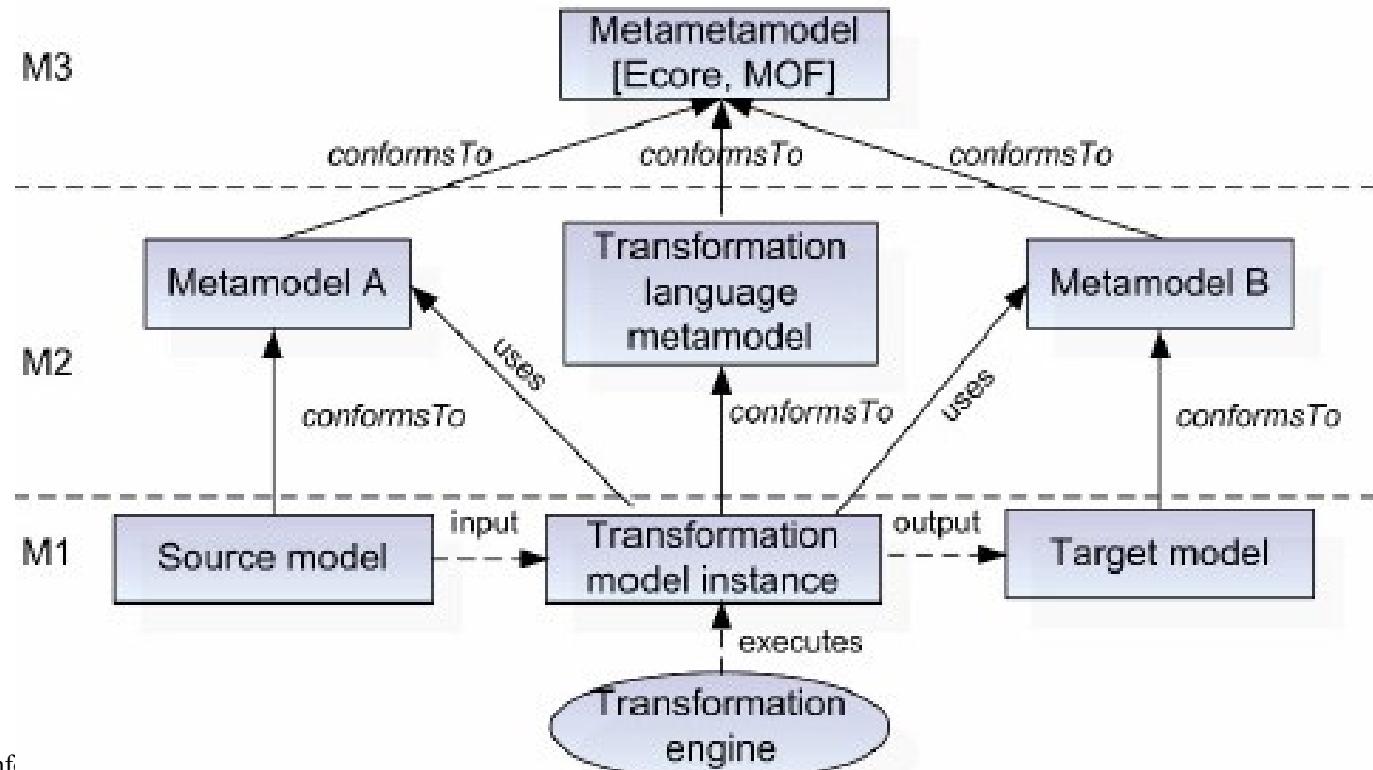
# Transformacija modela

## □ Transformacija

- jednog modela koji je sukladan s nekim metamodelom u drugi model sukladan nekom drugom metamodelu (npr. UML u Javu, SQL i obratno)

## □ Jezici za transformaciju

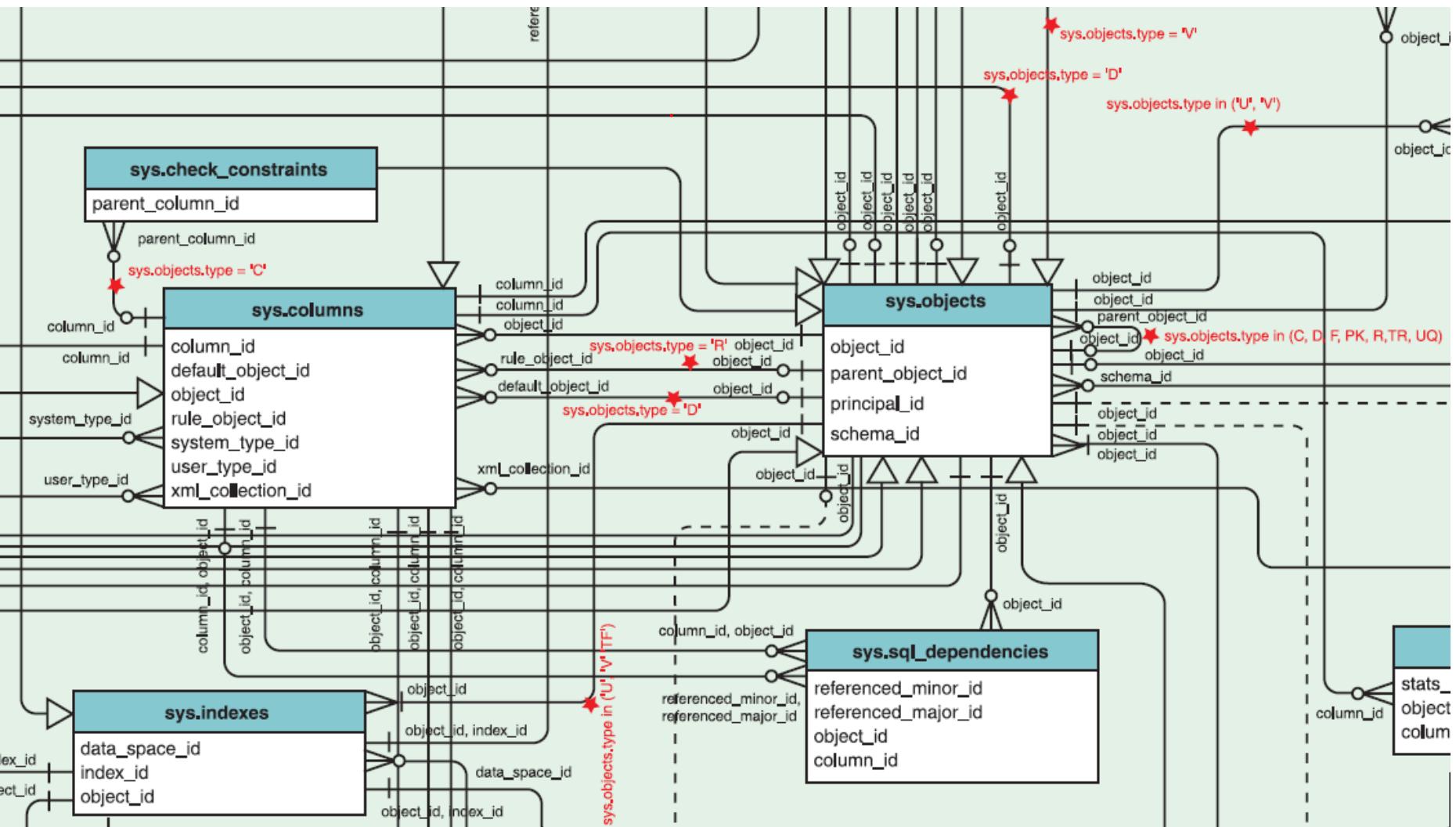
- QVT (Query/View/Transformation) – OMG standardiziran, opisan u MOF
- Atlas Transformation Language (ATL) – hibridni
- ...



# Primjene meta-modeliranja

# Shema rječnika SUBP

## □ SQL Server System Views Map



# Sistemske tablice i objekti

## □ Sistemske tablice

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

## □ Pohranjene procedure za rad s rječnikom

- sp\_help [ [ @objname = ] 'name' ]
- ...

## ■ 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	Tvrđka	491
1	sysdiagrams	1
1	Dokument	857
1	Artikl	1531
1	Država	239

# INFORMATION\_SCHEMA

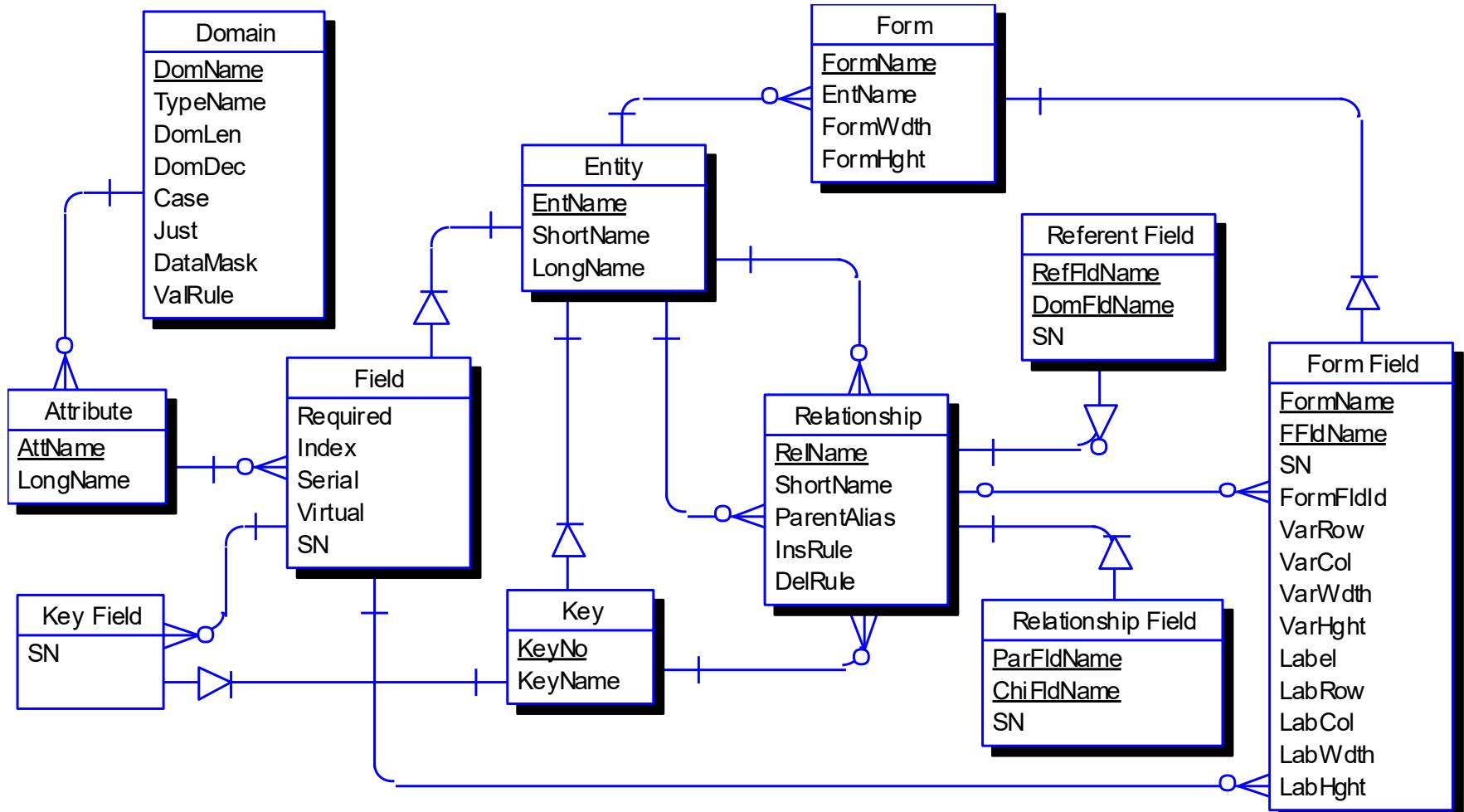
## □ Sistemski nezavisni pogled na metapodatke - ISO standard

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

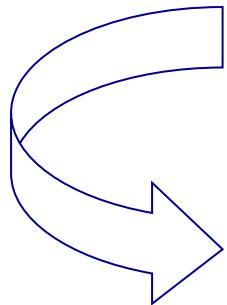


# Primjer generiranja programskog koda (MDD!)

```
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 *Račun + StavkaRac* na slajdu  
„Prevorba identifikacijskih veza”
- Zamijenimo ih s Dokument = VrstaDokumenta + BrojDokumenta
  - Vrsta Dokumenta  $\in \{ Z, U, P, N, O, T, R \}$

## □ Prednosti

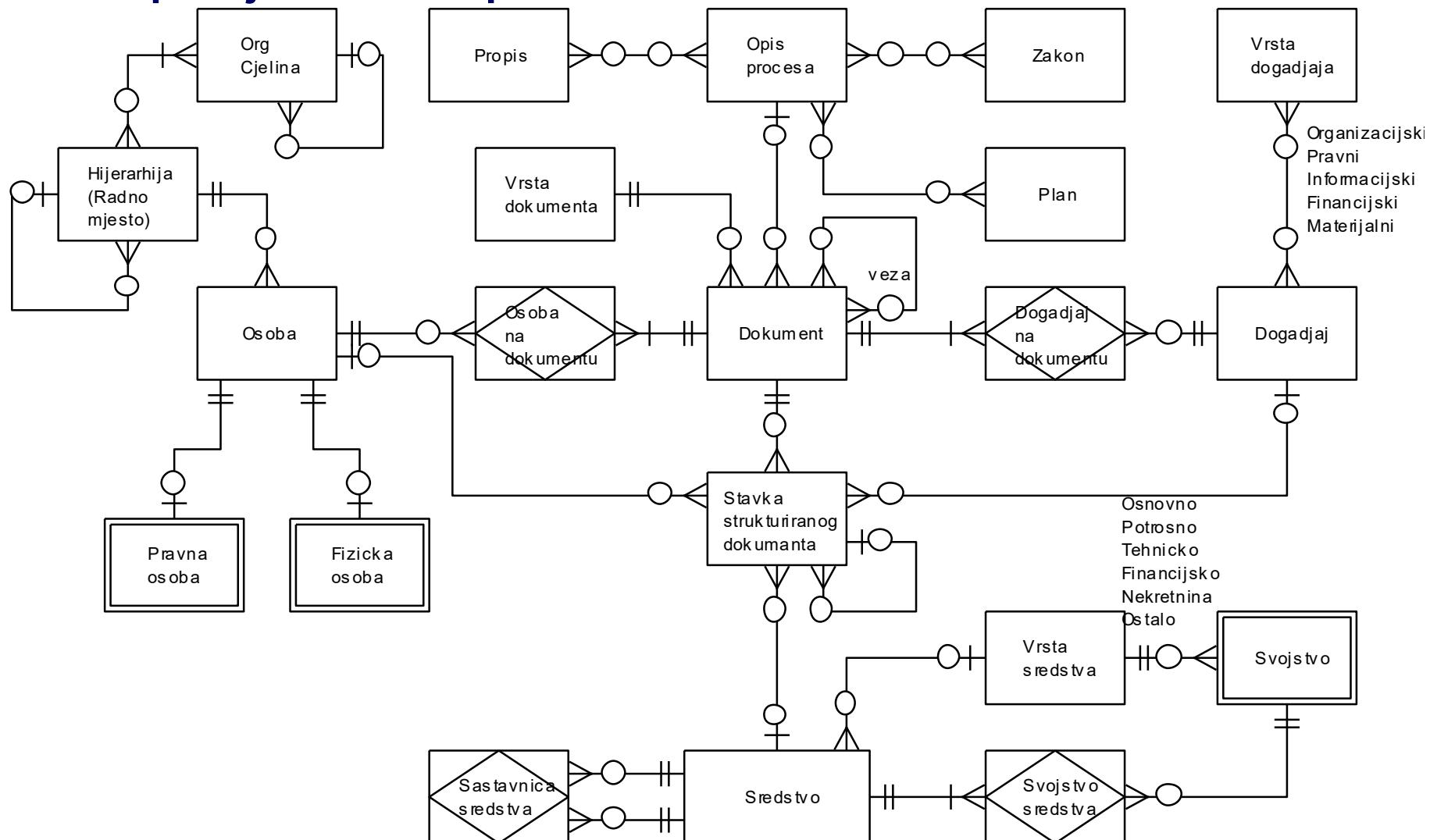
- Pojednostavljanje modela
- Smanjenje broja objekata

## □ 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

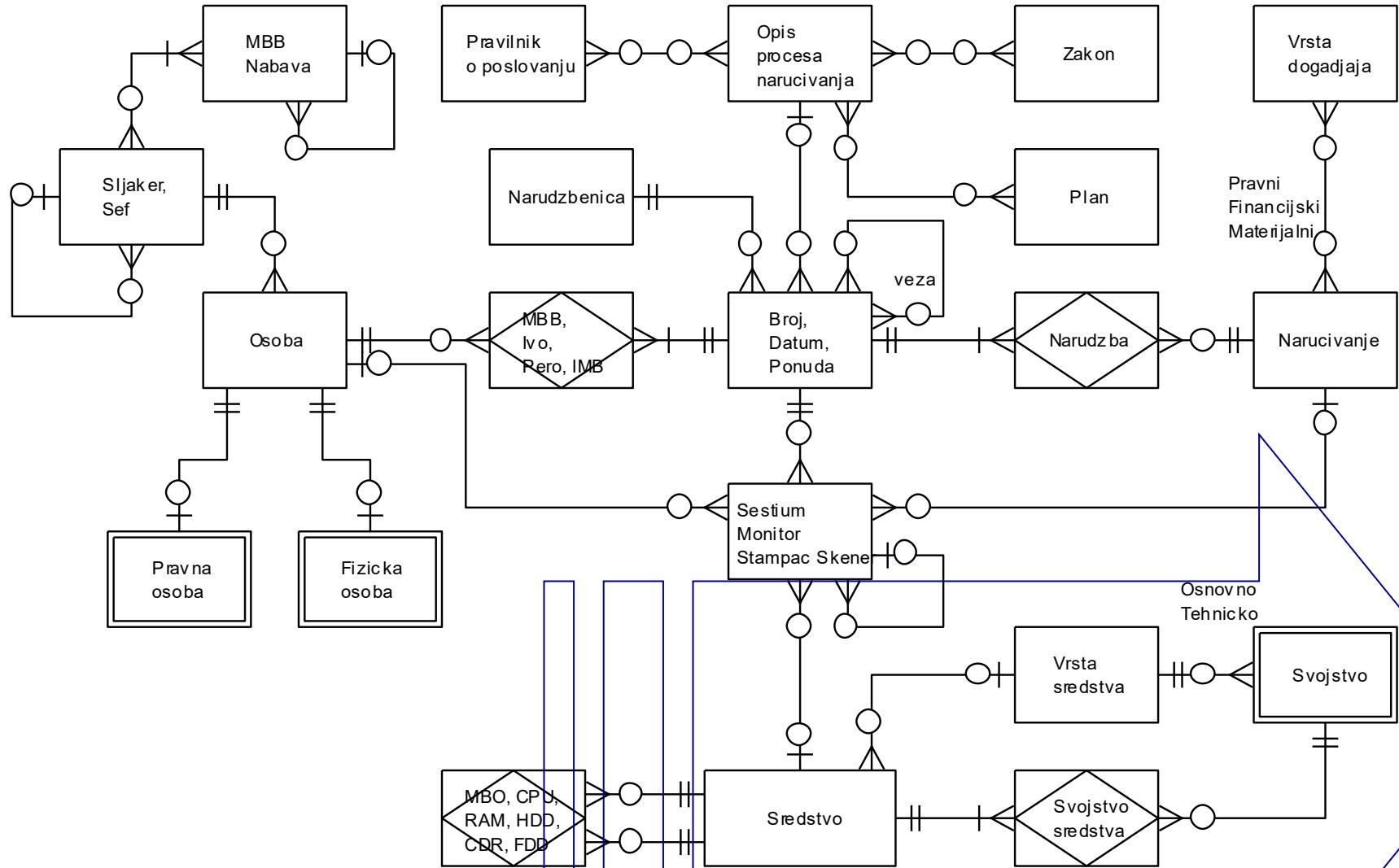
Odobrio njegov šef: Ivo

Stavke:

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

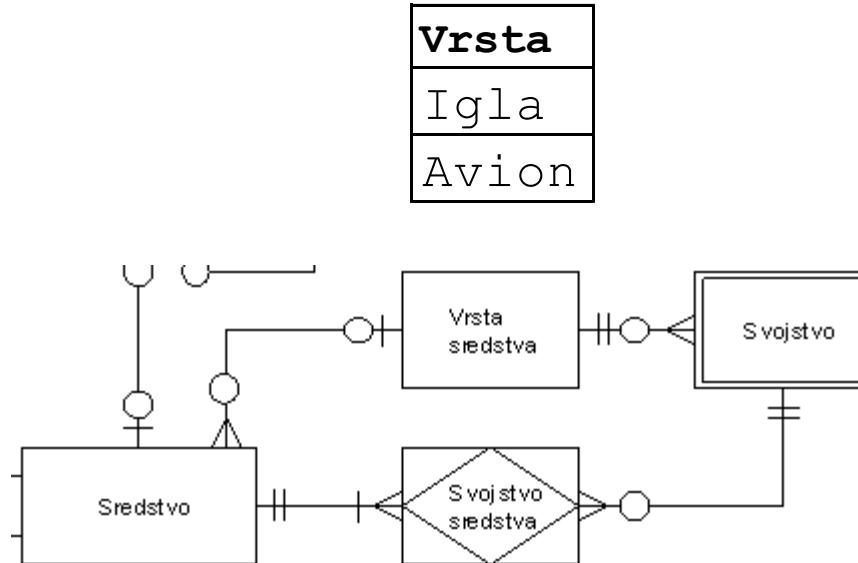
# Ideja aplikacijskog meta-modeliranja (3)

## □ 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

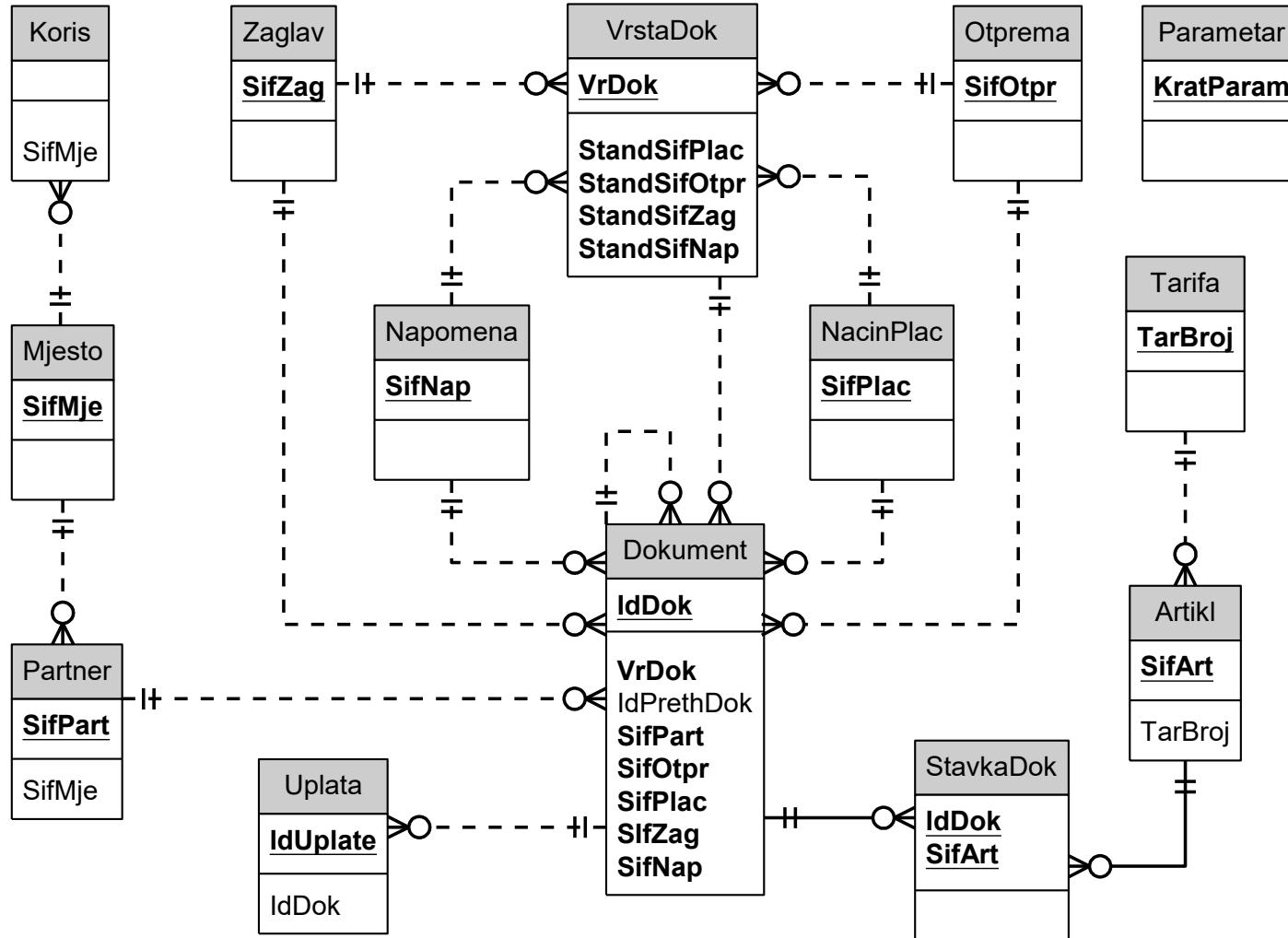
## □ Primjer: unos podataka

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 :						
*		oNumber)						

# Primjer aplikacije nad meta-modelom

## □ Primjer, prilagodba funkcionalnosti

Dokument

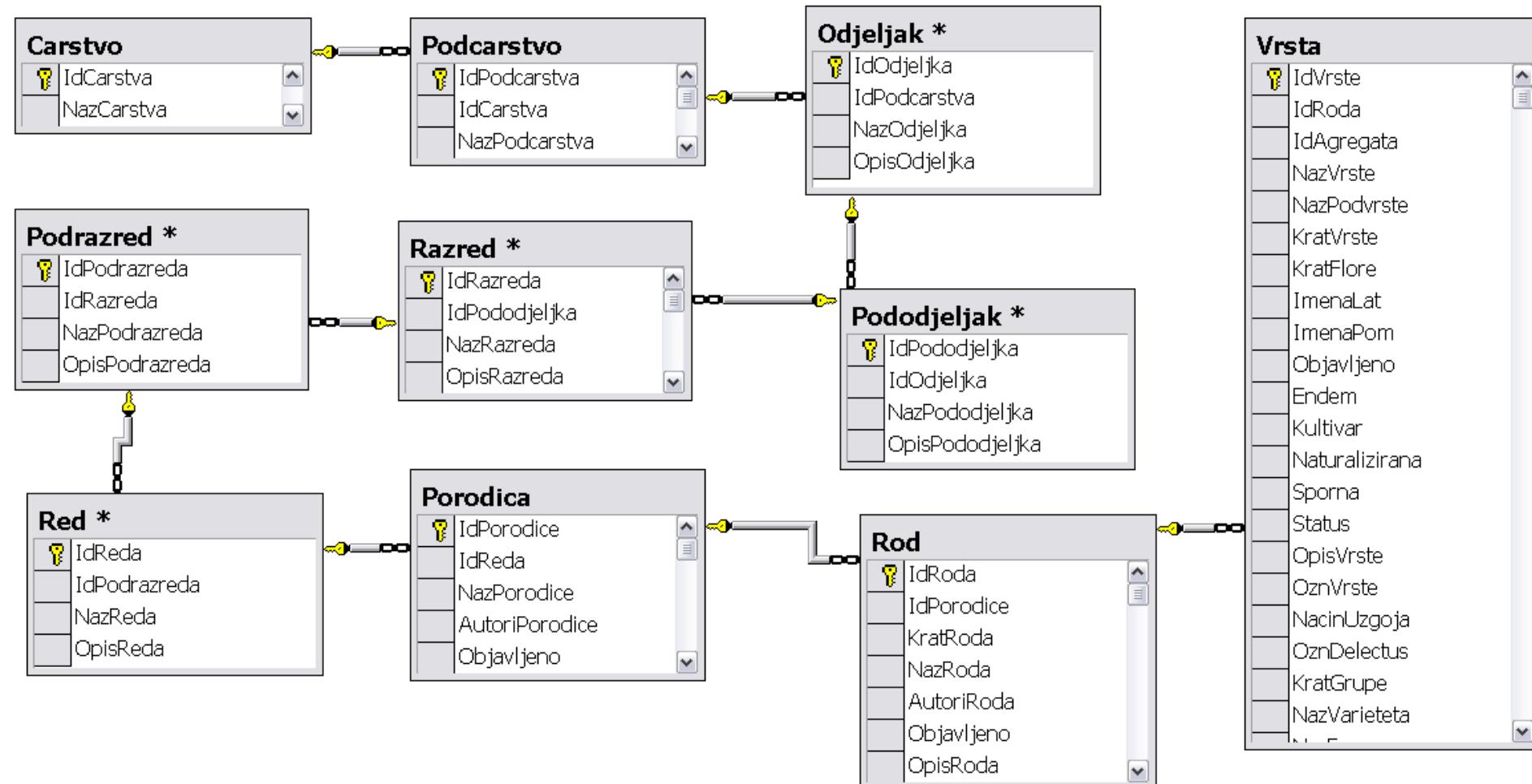
Vrsta	Predračun	Broj	102	Datum	09.09.96	Prethodni		Broj	
Partner	Dječji vrtić "Vrapčići"								
Datum valute	20.09.96	Dana valute	12	10 % Rabat	Račun PP	<input type="checkbox"/>	Br. ulaznog dokumenta		
Otprema	HPT	Plaćanje	Virman	Ukupno	432.00	Datum storna		Datum ažuriranja	
Ostali troškovi	manipulacija + poštarnina Gotovina 18.00								
Slovima	četiristopedeset kuna Kreditno pismo 450.00								
Zagлавje	Napomena	Virman	Ukupno	0.00	Datum ispisa	09.09.96	Datum uplate		
ESC									
Šifra	Naziv	Količina	Cijena	Iznos	%Rab	Rabat	%PP	Porez	Ukupno
1	Pokaži što znaš	8.00	20.00	160.00	10	16.00	0	0.00	144.00
2	Pokušaj nešto novo	8.00	20.00	160.00	10	16.00	0	0.00	144.00
3	Učimo opažati	8.00	20.00	160.00	10	16.00	0	0.00	144.00
Stavka [◀◀] 1 od 3 [▶▶] 480.00 48.00 0.00									
Dokument [◀◀] 1 od 6 [▶▶] Brisanje 0 Ažuriranje Uplate Storno									
F3 - Brisanje		F5 - Tablica	F7 - Ispis	F8 - Unos	F9 - Dohvat	F10 - Spremi	ESC - Kraj		

## □ 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
```

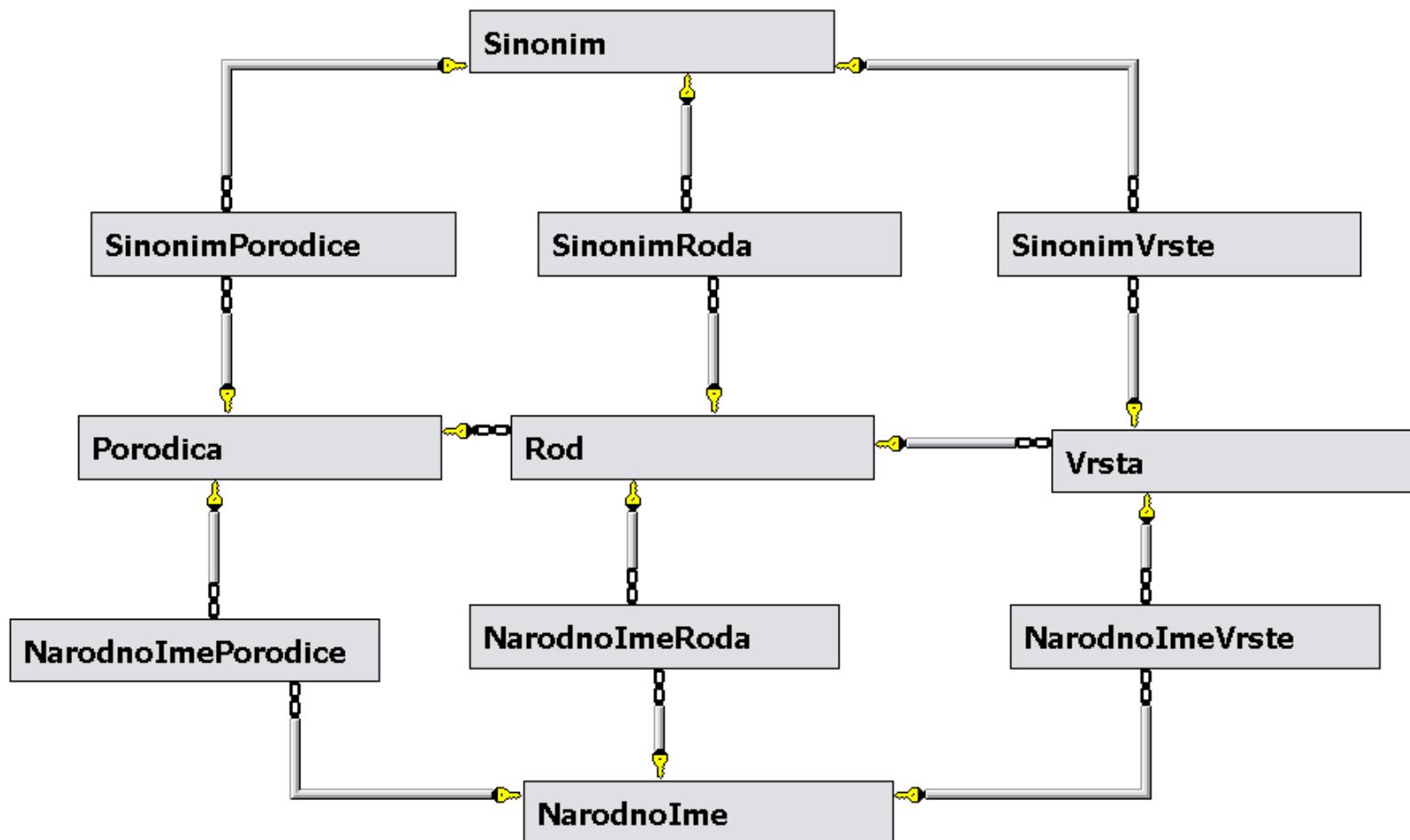
# Primjer: kaskada sličnih entiteta

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



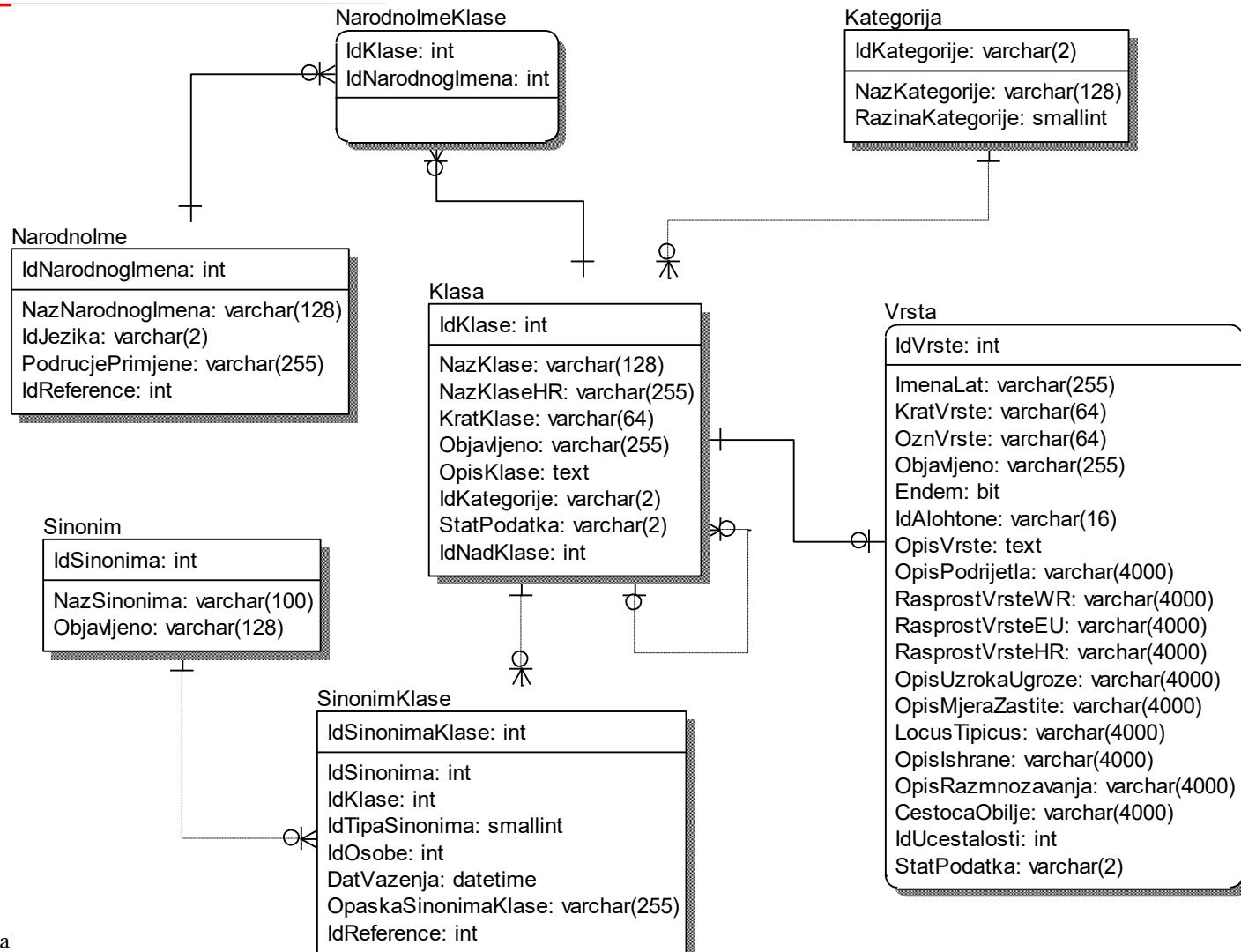
# Primjer: kaskada sličnih entiteta (2)

- *Porodica, Rod i Vrsta imaju još spojne tablice na narodna imena i sinonime, približno jednake strukture*
  - Rješenje: još 2\*3 specijalizacije ?

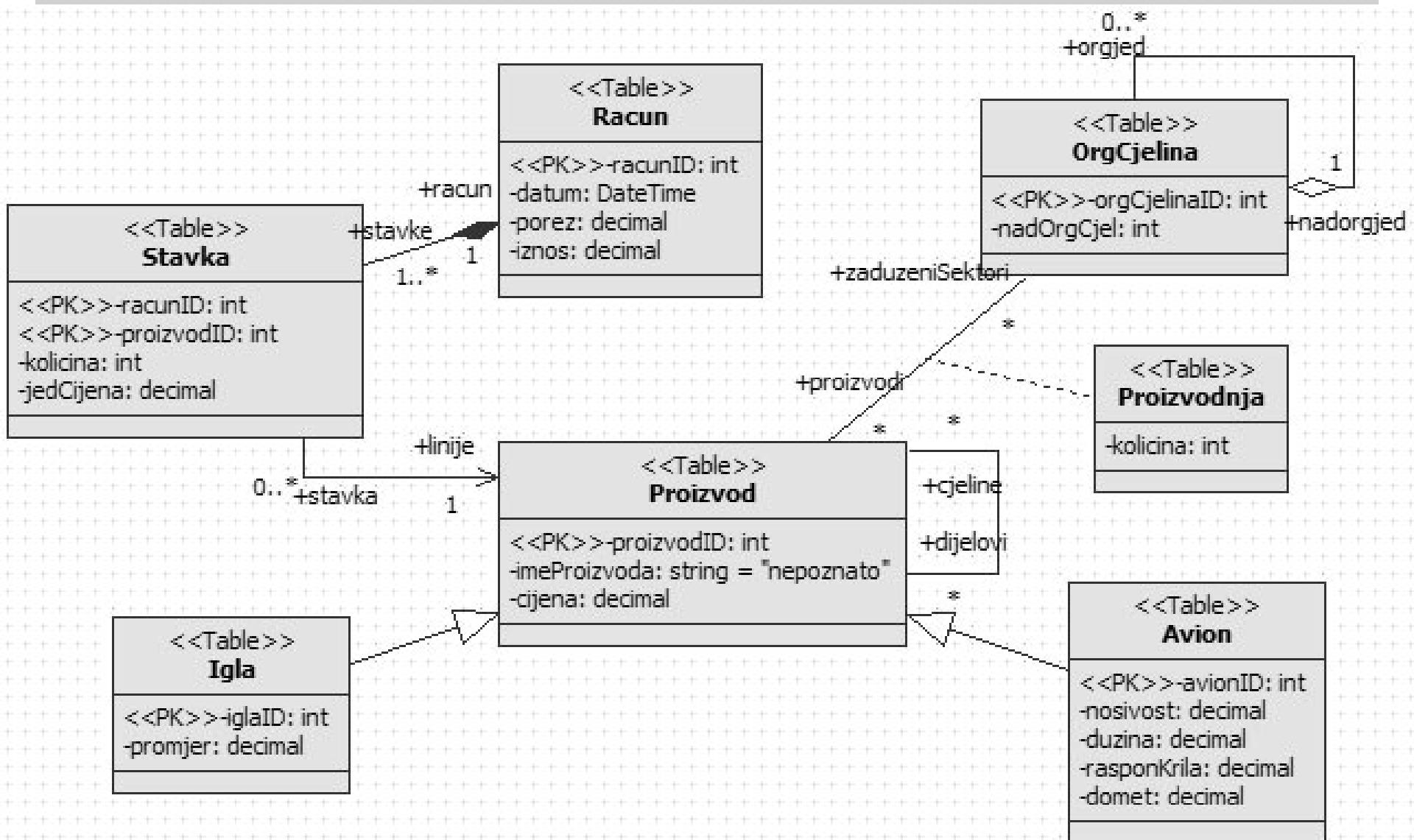


# Primjer: kaskada nakon modeliranja

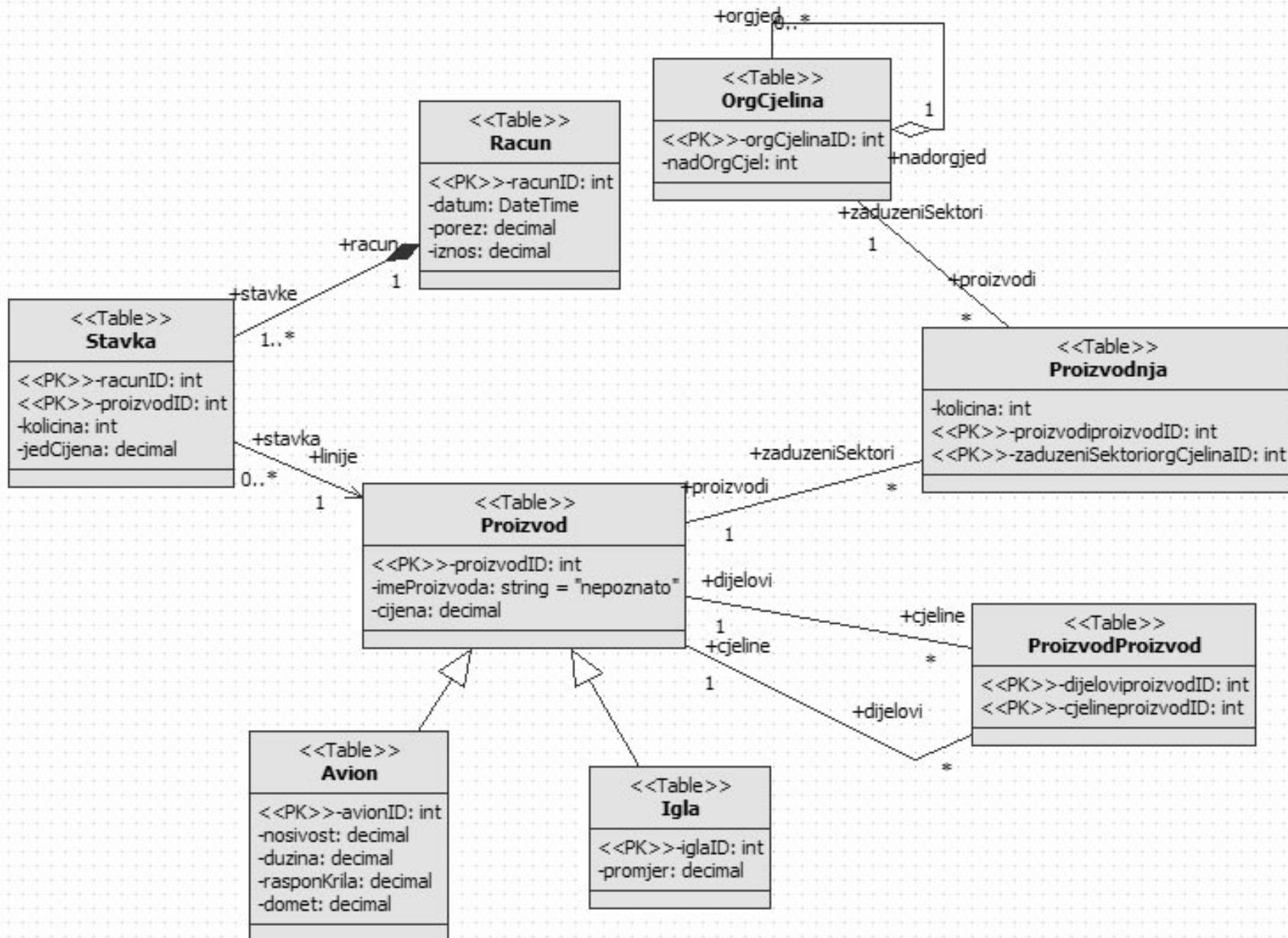
- rekurzivna hijerarhija, izjednačene strukture, jedna specijalizacija te spojni entiteti na narodna imena i sinonime (samo za entitete koji ih stvarno imaju)
- <https://hirc.botanic.hr/fcd/>



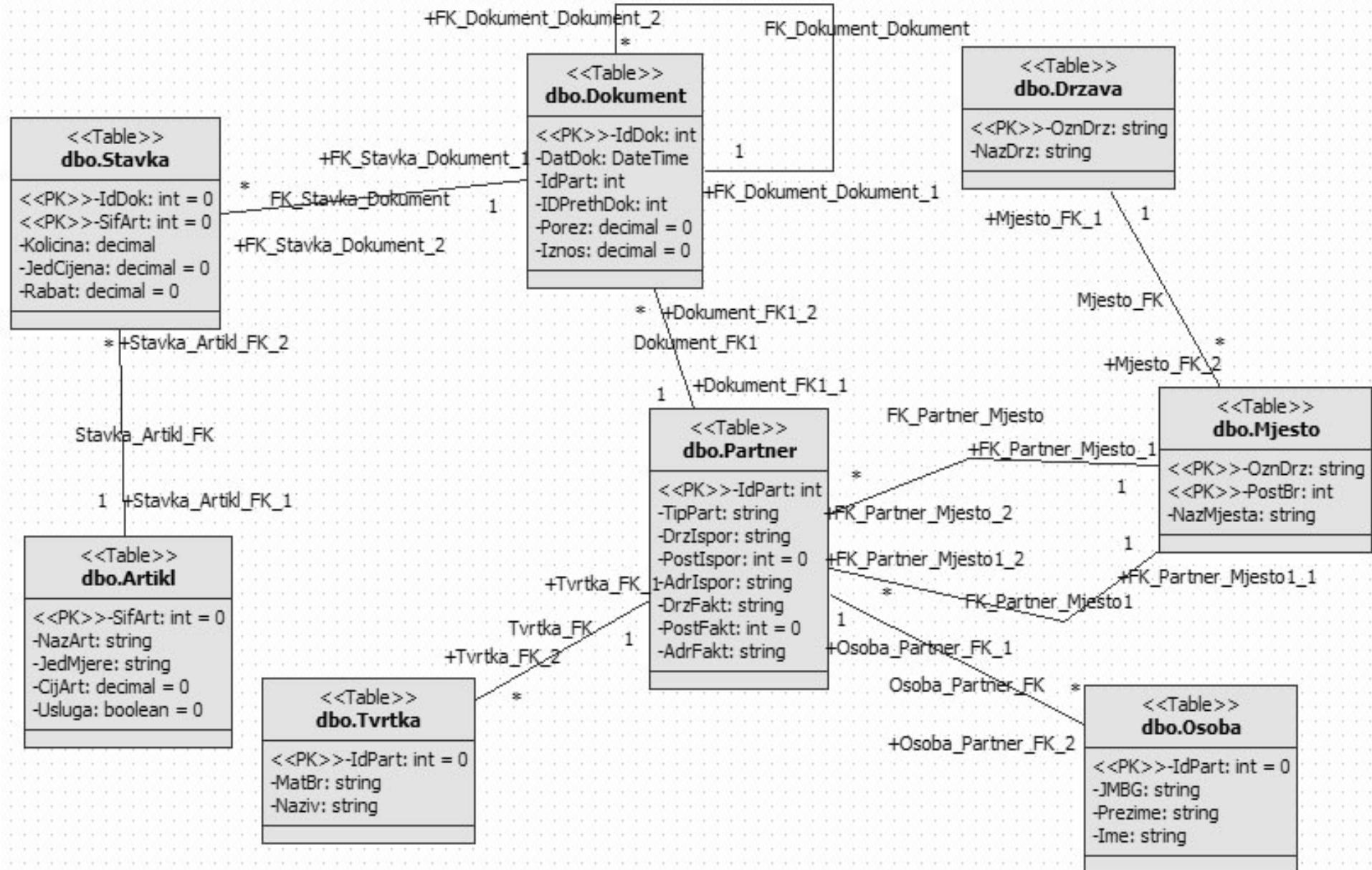
# Za sladokusce: konceptualni UML model ...



# ... model nakon transformacije ...



# ... i klasični relacijski model



# Definiranje podataka BNF notacijom

## □ Notacija

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

## □ Primjer, račun i stavke računa

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

## □ Može se napisati i kao:

```
Racun = @BrRac + DatRac + BrKupca  
      + { StavkaRac }  
      + (IznosRac)
```

StavkaRac = @SifArt, NazArt, Cijena, Kol, Vrijednost

# Primjeri i zadaci



## □ Primjer: Primjeri \ IS05-BazePodataka.zip

- Poduzeće – temeljem ER modela s predavanja
- Organizacija – model iz jednog projekta
- FirmaRIS – baza podataka debelog klijenta
- Projekt – baza podataka višeslojne aplikacije

## □ 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 ?

- <https://www.holistics.io/blog/top-5-free-database-diagram-design-tools/>

# Reference

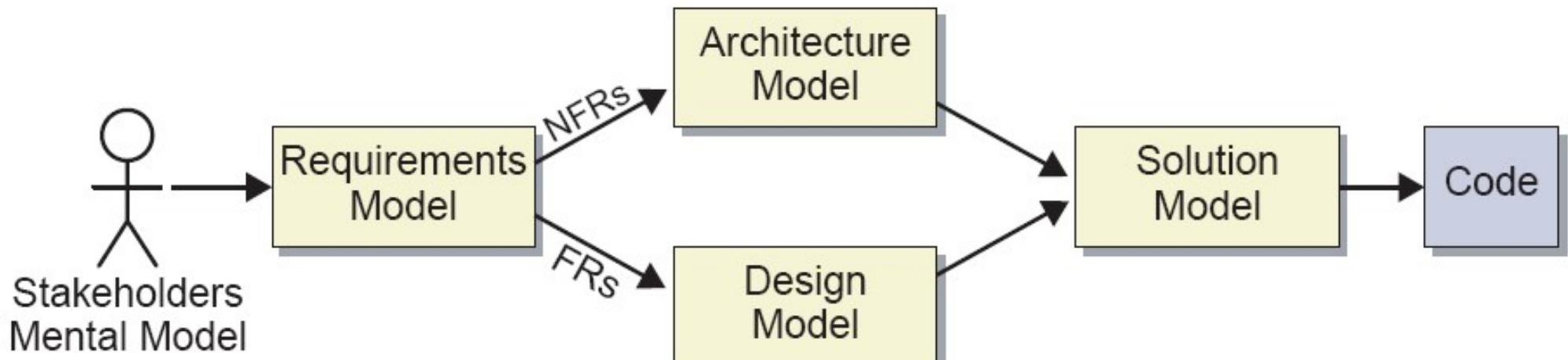
- "The entity-relationship model: toward a unified view of data", P.P. Chen, ACM Transactions on Database Systems 1:1 pp 9-36, 1976.
- ERD notacija
  - <http://www.conceptdraw.com/How-To-Guide/erd-entity-relationship-diagram-symbols>
- A UML Profile for Data Modeling
  - <http://www.agiledata.org/essays/umlDataModelingProfile.html>
- OMG's MetaObject Facility
  - <http://www.omg.org/mof>
- MD\*
  - <https://modeling-languages.com/clarifying-concepts-mbe-vs-mde-vs-mdd-vs-md/>
  - <https://modeling-languages.com/low-code-vs-model-driven/>

# **Objektno orijentirani razvoj**

**2022/23.06**

# 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.

## Prikupljanje zahtjeva

## Analiza zahtjeva / Modeliranje funkcija

## Modeliranje arhitekture

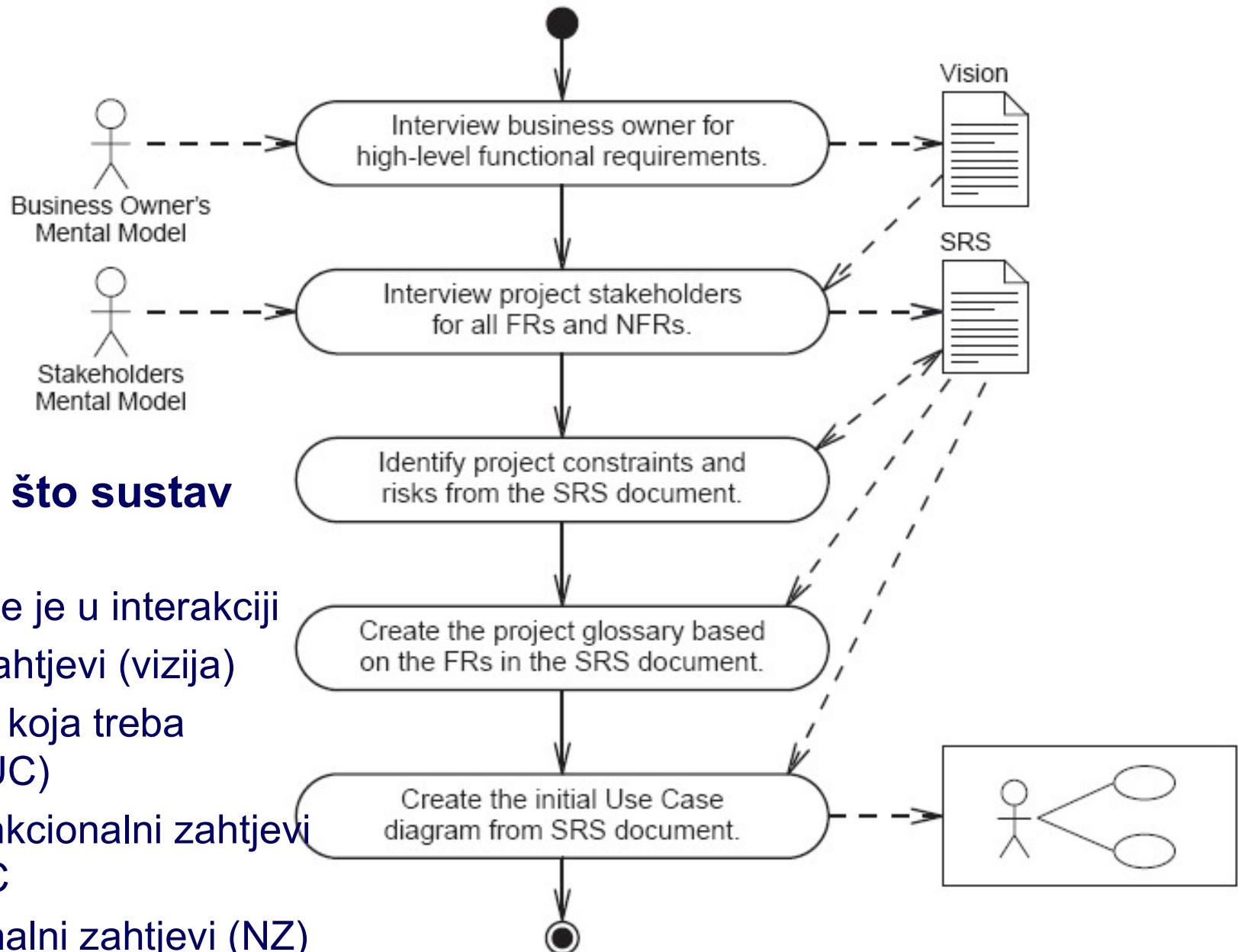
## Dizajn

## Konstrukcija

## Određivanje što sustav treba činiti

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

# Radni tok prikupljanja zahtjeva



Prikupljanje zahtjeva

Analiza zahtjeva /  
Modeliranje funkcija

Modeliranje arhitekture

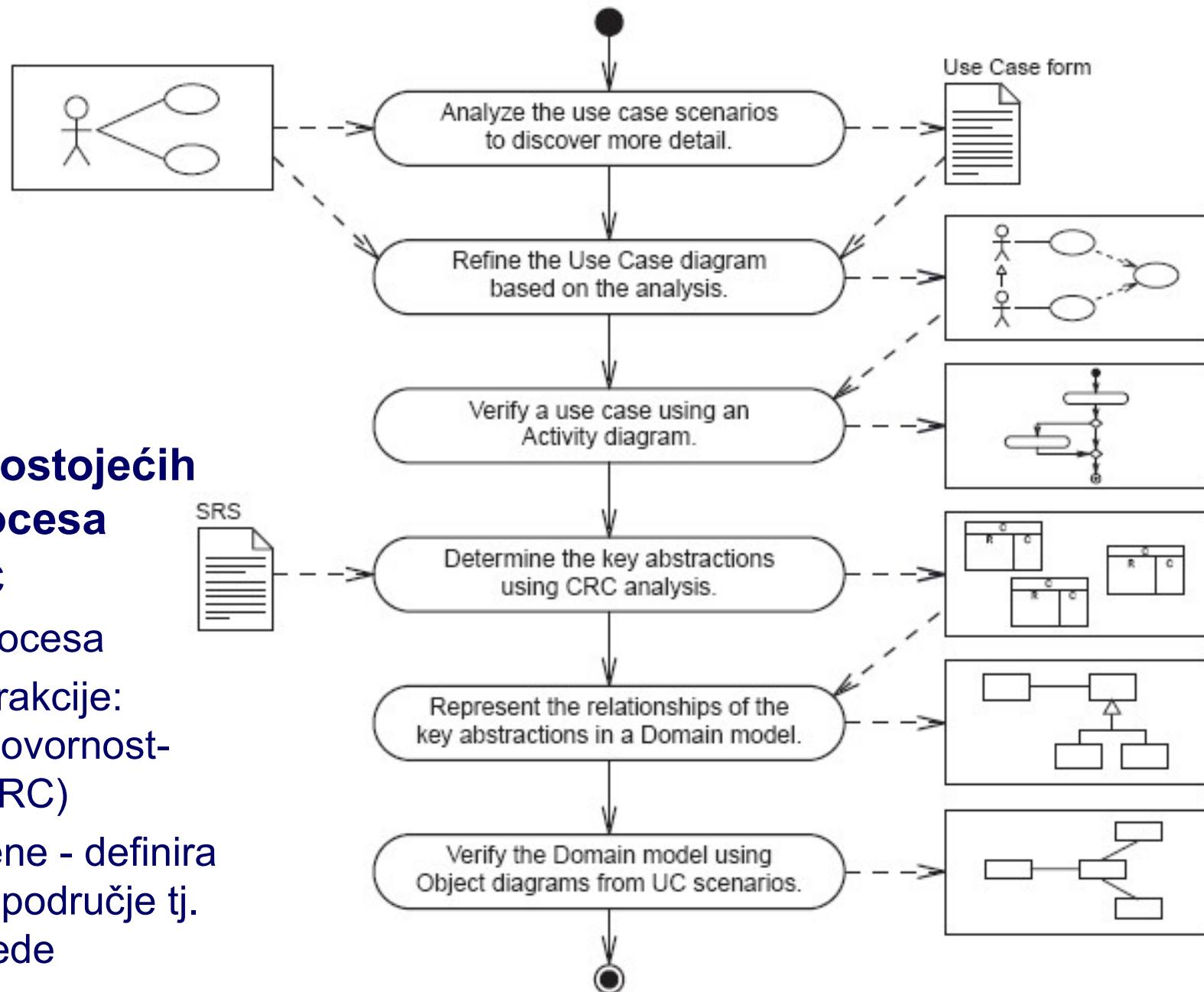
Dizajn

Konstrukcija

## □ Modeliranje postojećih poslovnih procesa

- rafinirani UC
- aktivnosti procesa
- ključne apstrakcije:  
Razred-Odgovornost-Suradnja (CRC)
- model domene - definira  
problemsko područje tj.  
ključne razrede

# Radni tok analize zahtjeva (→ modeliranje funkcija)



Prikupljanje zahtjeva

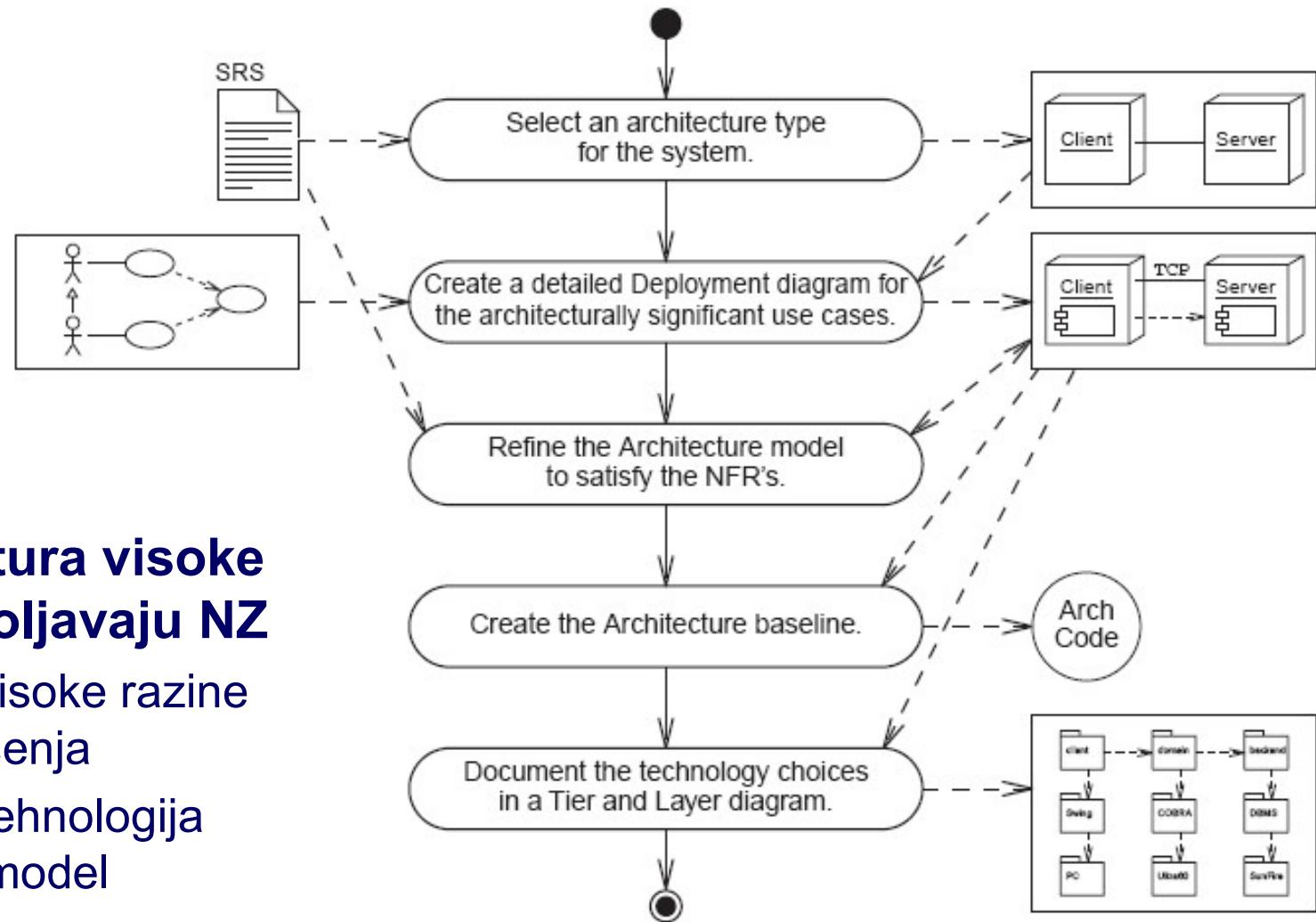
Analiza zahtjeva /  
Modeliranje funkcija

**Modeliranje arhitekture**

Dizajn

Konstrukcija

# Radni tok modeliranja arhitekture



## □ Modeliranje struktura visoke razine koje zadovoljavaju NZ

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

# Radni tok dizajna

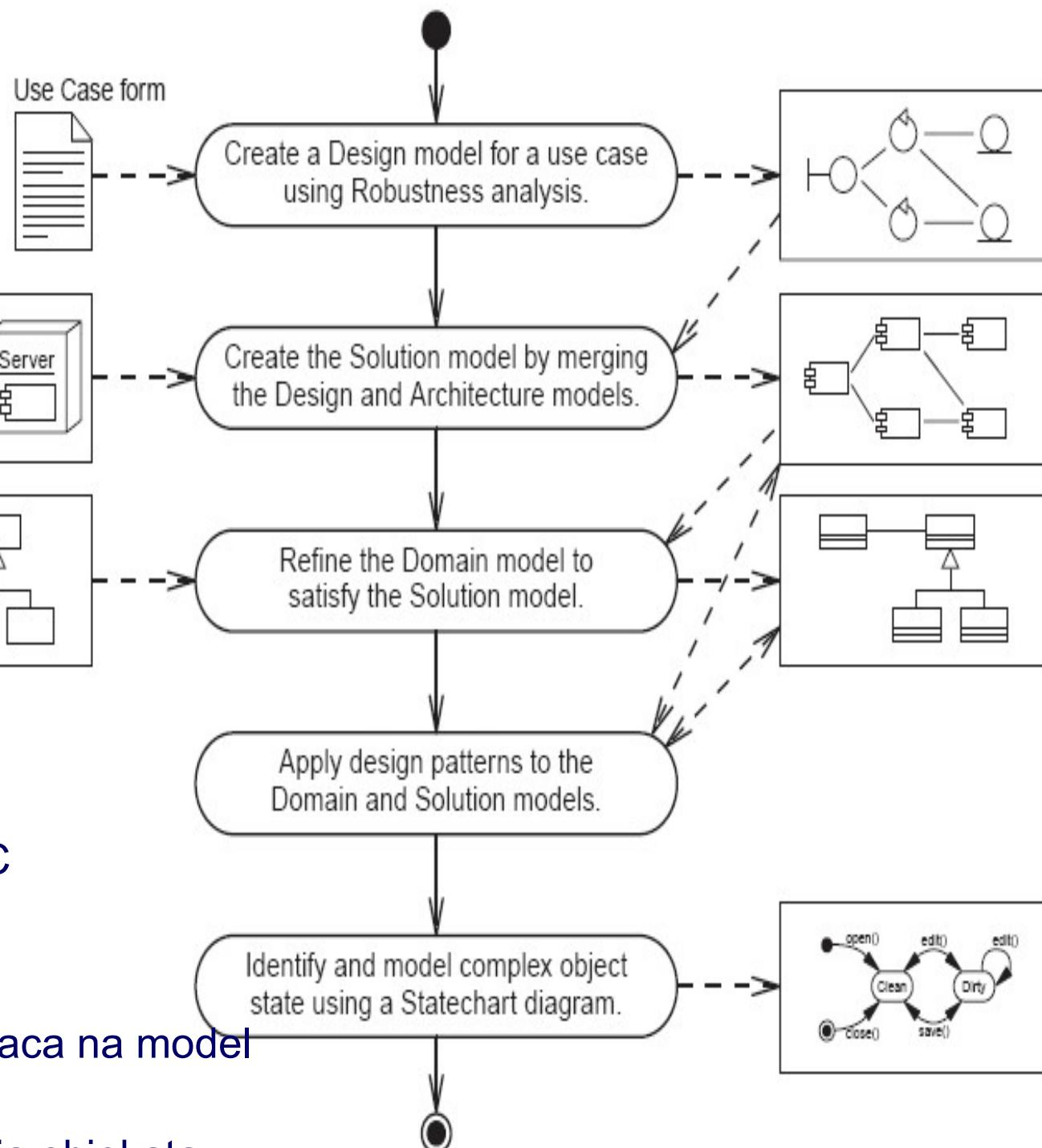
Prikupljanje zahtjeva

Analiza zahtjeva / Modeliranje funkcija

Modeliranje arhitekture

Dizajn

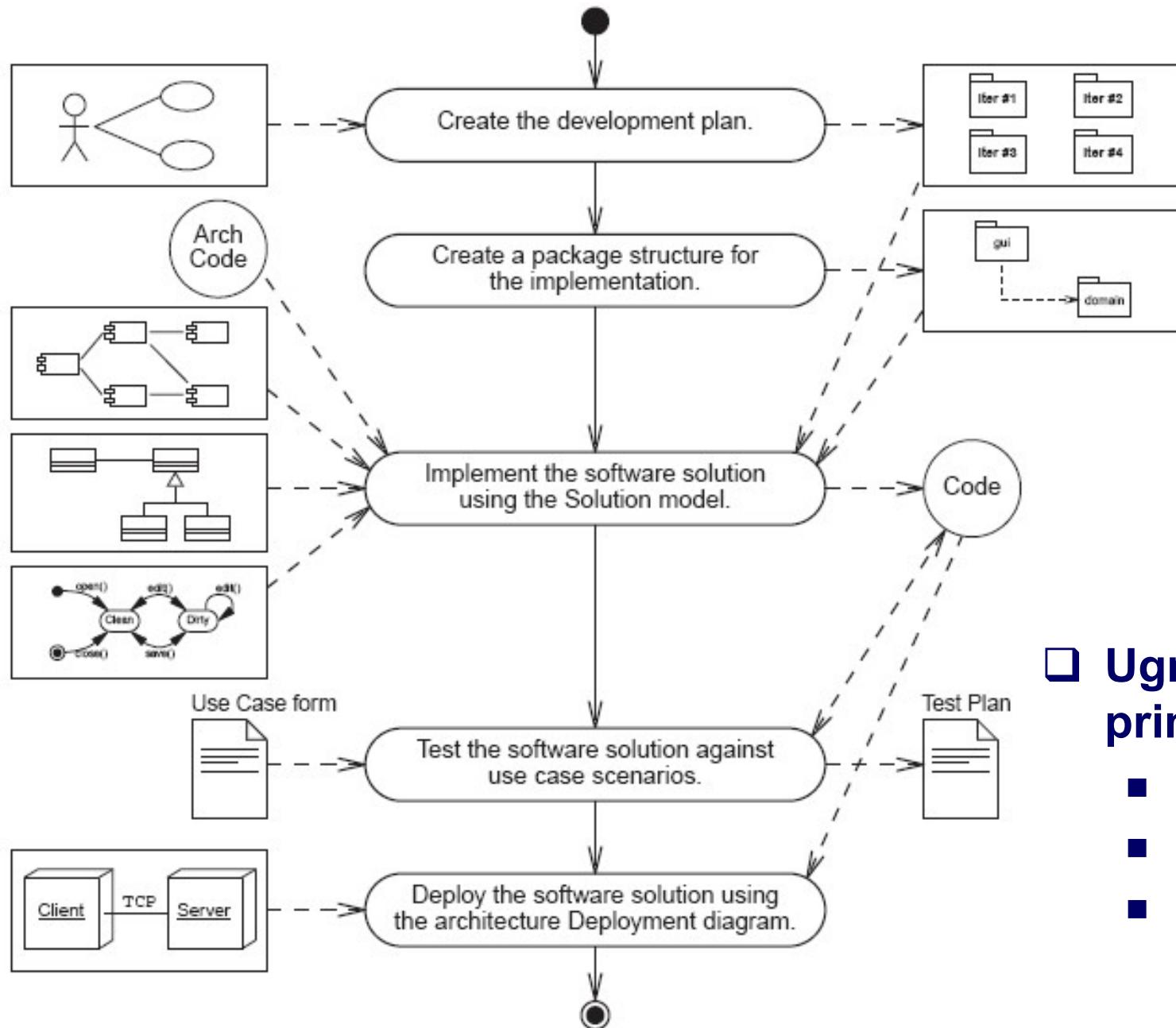
Konstrukcija



## □ 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



Prikupljanje zahtjeva

Analiza zahtjeva /  
Modeliranje funkcija

Modeliranje arhitekture

Dizajn

Konstrukcija

## ❑ Ugradnja, provjera i primjena sustava

- implementacija rješenja
- testiranje
- primjena u radnom okruženju

# **Modeliranje procesa i funkcionalnosti**

**Business process and functional modeling**

# Modeliranje funkcionalnosti

## □ Functional modeling

- Opisuje poslovne procese i interakciju IS s okolinom
- Dokumentiranje zahtjeva i vanjskog ponašanja sustava
- Određivanje potencijalnih (logičkih) sučelja sustava (prema okolini)

## □ Logički, modeli problemskog područja (problem domain models)

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

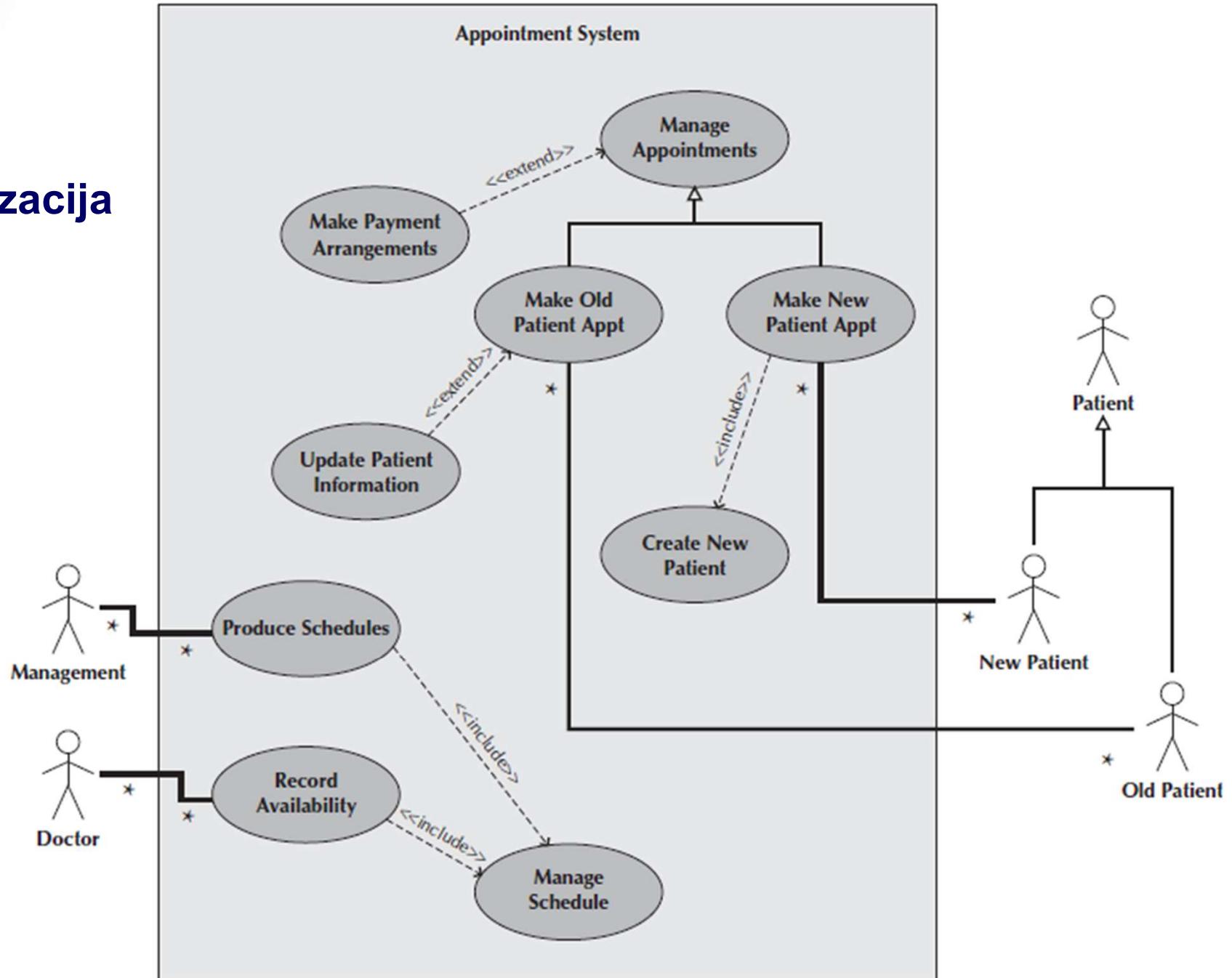
## □ Fizički modeli

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



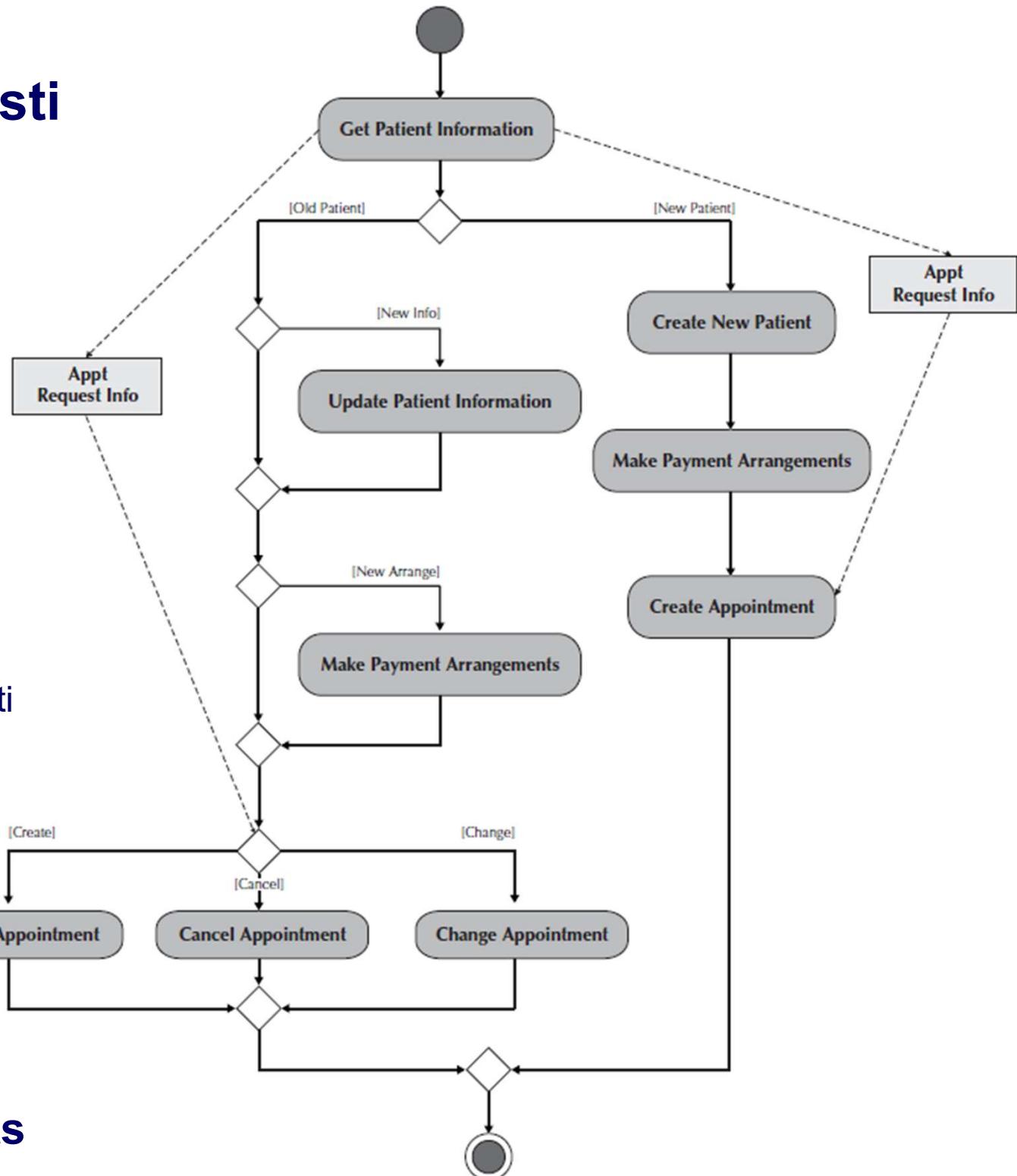
# Primjer: UC dijagram

- Include**
- Extend**
- Generalizacija**
- Doseg**



# Modeliranje procesa dijagramima aktivnosti

- "sofisticirani DTP"
  - dodatni elementi AD
- Objekt (object node)
  - klasa objekata u toku
- Upravljački tok (control flow)
  - puna strelica, slijed izvršenja
  - povezuje akcije/aktivnosti
- Podatkovni tok (object flow)
  - protok objekata
  - isprekidana strelica
  - veza objekt i aktivnost



## □ Primjer: Manage Appts



# 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 opisa slučaja korištenja 1/3

Naziv slučaja korištenja: Zakazati pregled	ID: 2	Prioritet: Visok
Glavni sudionik: Pacijent	Tip slučaja korištenja:	Detaljni, osnovni
<b>Sudionici i interesi:</b> Pacijent – želi zakazati, promijeniti ili otkazati pregled Liječnik – želi obraditi pacijenta u prihvatljivom vremenu		
<b>Kratki opis:</b> slučaj korištenja opisuje kako zakazati, promijeniti ili otkazati pregled.		
<b>Okidač:</b> Pacijent pozivom zakazuje novi pregled te mijenja ili otkazuje postojeći pregled. <b>Tip:</b> Vanjski		
<b>Veze:</b> Asocijacija (association): Pacijent Uključivanje (include): Izrada računa Proširenje (extend): Unos novog pacijenta Generalizacija (generalization):		
<b>Tok događaja:</b> 1. Pacijent kontaktira audijenciju radi rezerviranja		

# 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 računa

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.

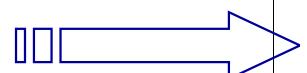
Ako pacijent želi promijeniti postojeći pregled,  
izvodi se podtok T-3: promjeni pregled.

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 koraka unutar 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, "zaboravljeni" je:

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

# Primjer opisa slučaja korištenja 3/3



## □ Primjer: [Dizajn\UCprimjer](#)

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.

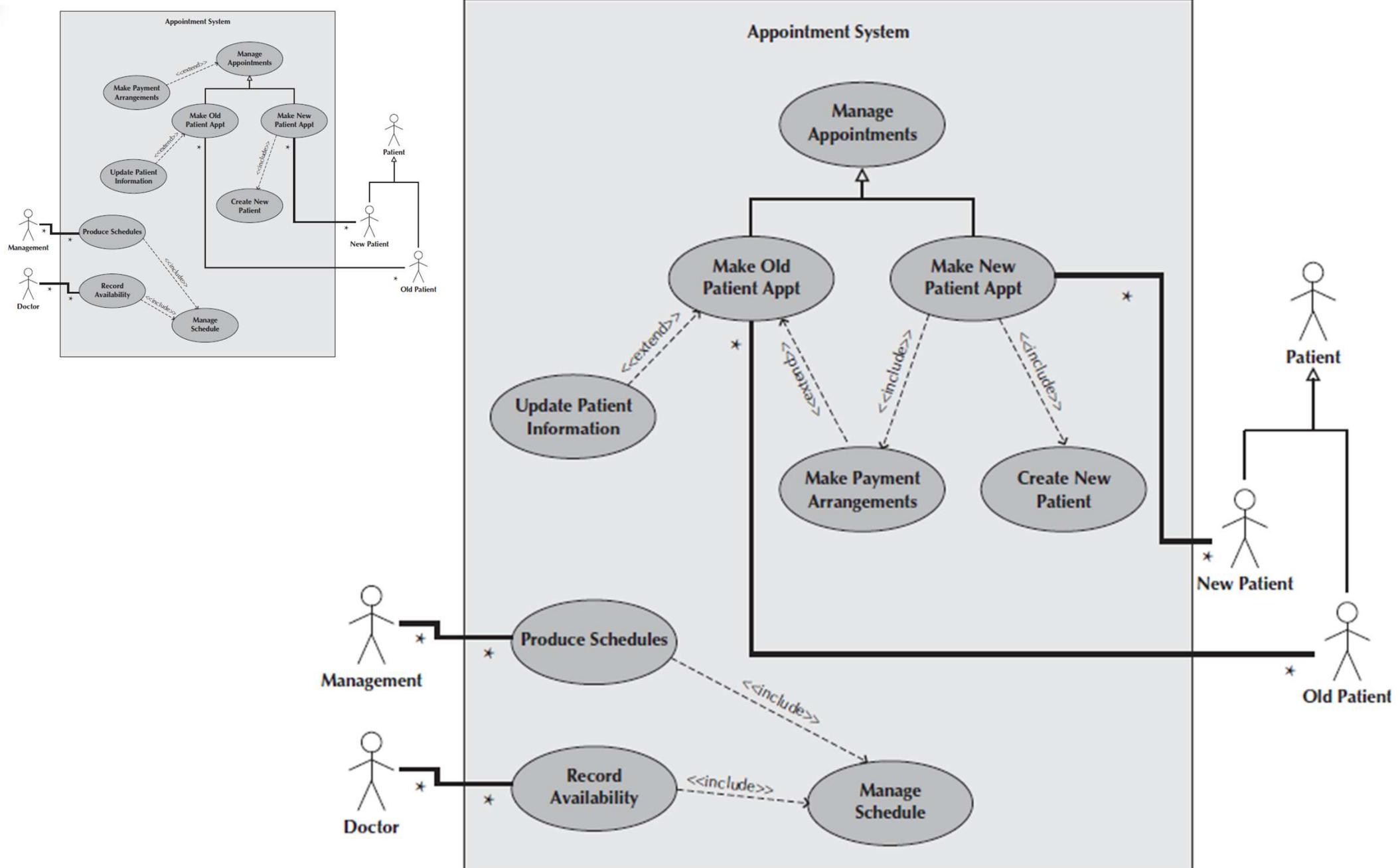
### Alternativni/izuzetni tokovi:

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.

# Primjer: UC dijagram revidirani



# Dijagrami aktivnosti naspram slučajeva korištenja

## □ Slučaj korištenja

- opisuje samo jednu funkciju interakcije korisnika i sustava
- može uključivati različite scenarije
  - Npr. scenariji za isti UC: pretraživanje knjiga po autoru, po naslovu, ...
- UC smije biti povezan s jednom i samo jednom ulogom
  - Npr. liječnička "sestra" dogovara pregledе, ispisuje uputnice i recepte, naplaćuje participaciju ...

## □ Dijagram aktivnosti

- analiza jednog slučaja korištenja za proces povezan s jednom ulogom
- slijed akcija za nekoliko objekata i nekoliko UC (posljedično više uloga)
- razumijevanje radnog, poslovnog procesa
- opisivanje složenih slijednih algoritama (obrade podataka)
- opisivanje višenitnih aplikacija

# Komentar dijagrama i daljnji koraci

## □ Granica sustava (subject boundary) – doseg projekta

- razdvaja vanjske i unutarnje dijelove (pravokutnik oko slučajeva)
- na prethodnoj slici nedostaje *Službenik*, jer je dio sustava (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 ulazeњa 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
<b>Opis:</b> Pojedinac koji treba ili prima medicinsku pomoć.		<b>Slučajevi korištenja:</b> 2
<b>Odgovornosti</b> Zakazati pregled _____ Odrediti zadnji posjet _____ Promijeniti status _____ Dohvatiti povijest bolesti _____ _____ _____		<b>Suradnici</b> Pregled _____ _____ Povijest bolesti _____ _____

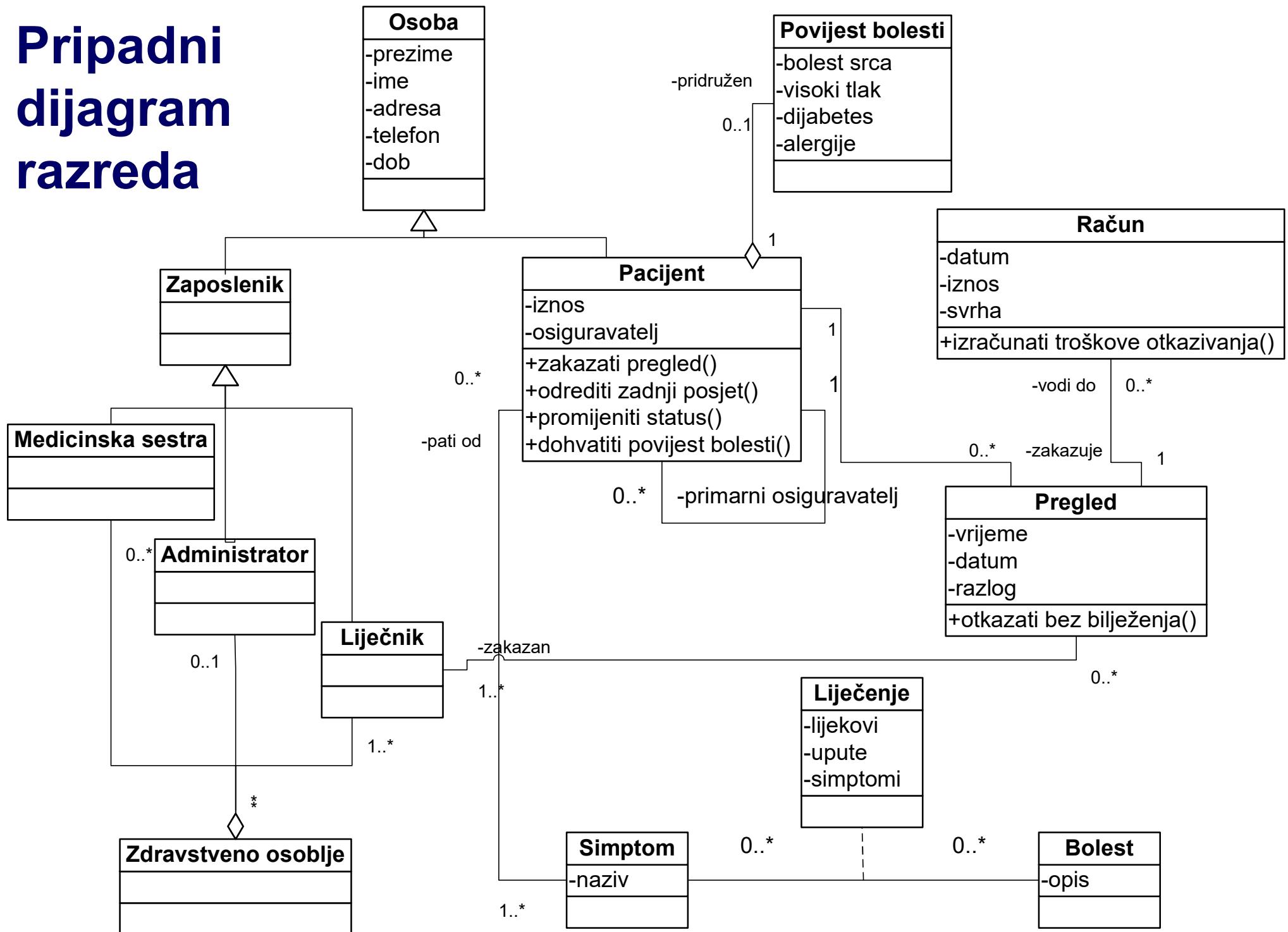
## Stražnja stranica:

Atributi:	Iznos (double) Nositelj osiguranja (text)	_____ _____
Veze:	Generalizacija (a-kind-of): Agregacija (has-parts): Ostale veze :	Osoba _____ Povijest bolesti _____ 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-Artikl*
  - “kako realizirati”, “kako ponovno iskoristiti”
    - oblikovni obrasci, aplikacijski okviri, knjižnice razreda

# Pripadni dijagram razreda



# Dizajn vođen odgovornostima

# Dizajn vođen odgovornostima

## □ Responsibility-Driven Design (RDD)

- 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 podataka .... (kodiranje, refaktoriranje)

## □ Dizajn jednostrukе odgovornosti

- Objekti postoje da bi podržali UC/scenarij
- Pojedini objekt treba imati jednu jednostruku odgovornost unutar UC
- Objekt može imati više ponašanja (svojstava, postupaka), ali takvih da objekt podupire jedinstvenu odgovornost
- Pažnja se usmjerava na to što objekt radi a ne na podatke !

# 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

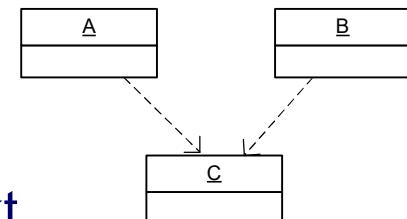
## □ Npr. Objekt A i Objekt B podupiru svaki svoj UC

- A treba neko ponašanje od B (kopčanje, eng. 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**

# Primjer RDD: Slučajevi sustava evidencije projekata

## Dodavanje projekta (provodi upravitelj projekta)

- podaci o projektu sadrže naziv, opis, datum početka i datum završetka
- 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)



## 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 (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, odabirom uloge iz liste uz automatski datum
  - treba dozvoliti promjenu uloge osobe
  - osoba može sudjelovati u više projekata
  - treba omogućiti brisanje zaduženja
- 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

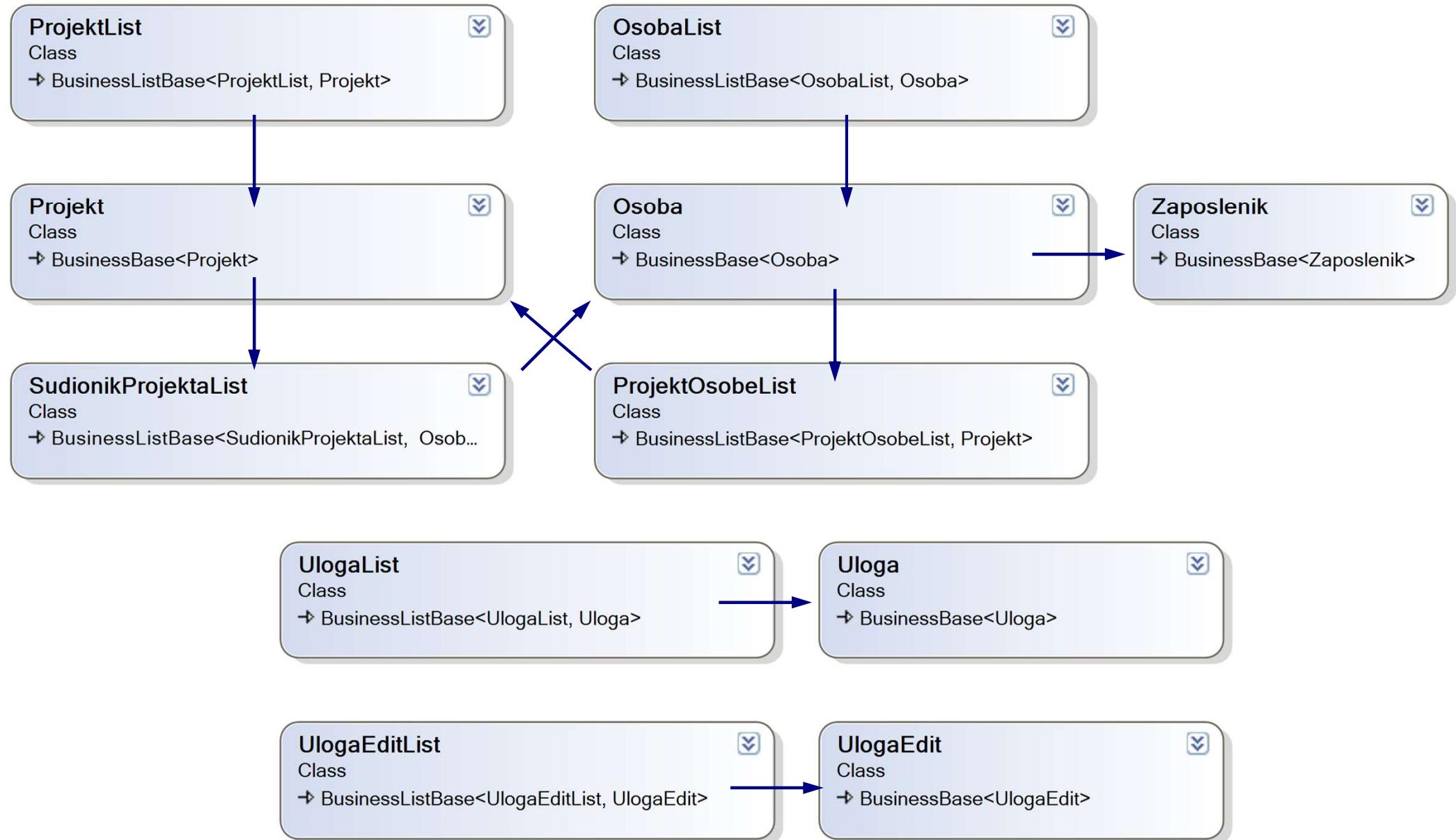
# Razredi potencijalnih objekata

## □ 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

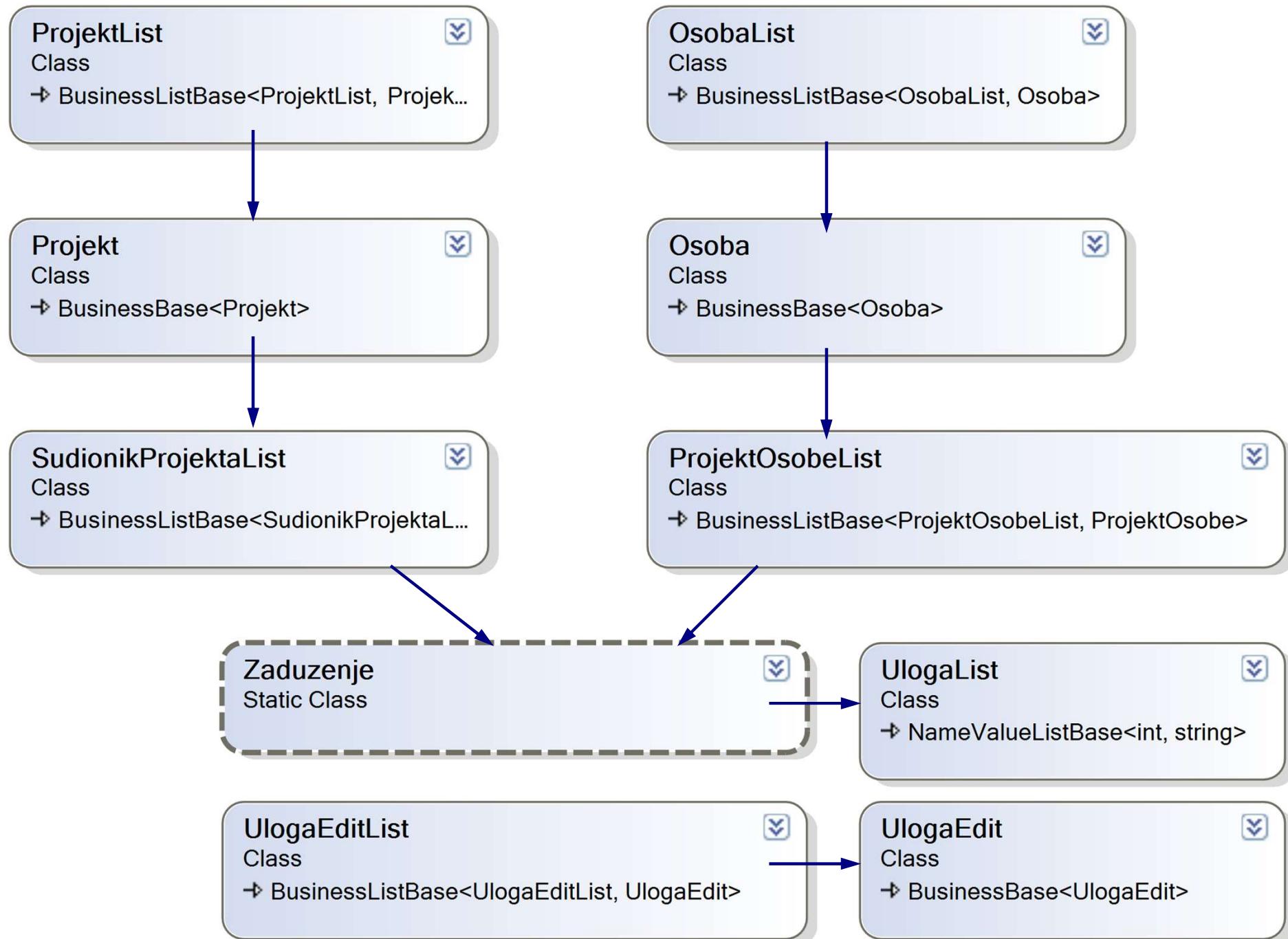
## □ UlogaList i Uloga

- 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 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 osoba projekta
  - podrazumijeva da **Zaduzenje** sadrži informaciju o osobi
- Osoba koristi ProjektOsobeList za dohvat liste projekata osobe
  - podrazumijeva da **Zaduzenje** sadrži informaciju o projektu

## □ Očigledno postoji konflikt ponašanja ali i konflikt podataka

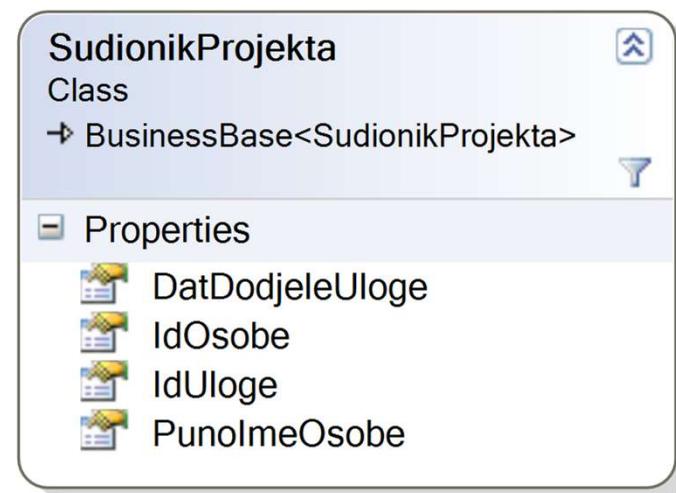
## □ Moguća rješenja

- 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



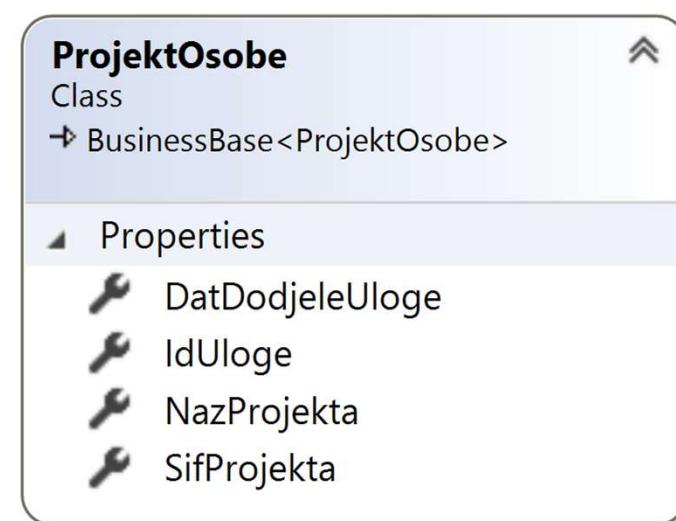
# Alternativa (1.dio) : Dodjela osobe projektu

- 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



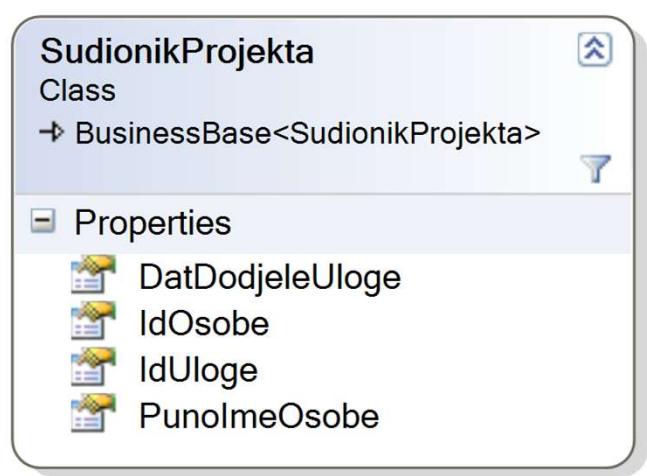
# Alternativa (2.dio) : Dodjela projekta osobi

- 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 DodjeliProjekt (SifProjekta)
  - davanje podataka za pregled i uređivanje

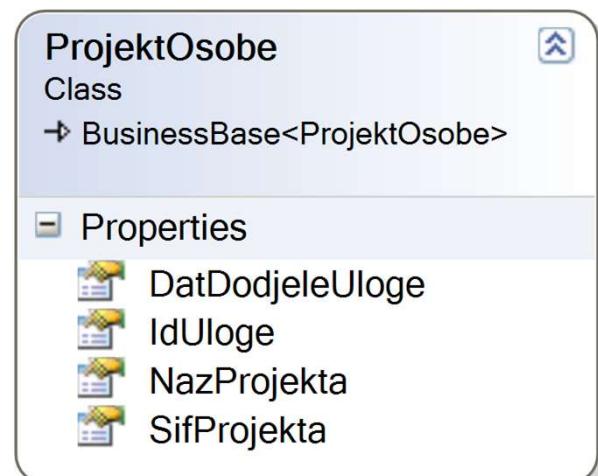


# Što dalje: Udruživanje sličnih razreda ?

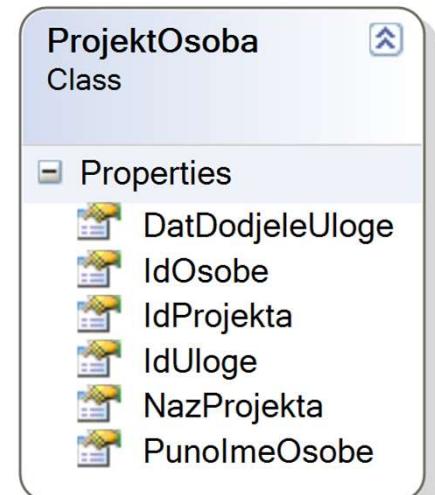
- SudionikProjekta i ProjektOsobe sadrže slične podatke, ali im je ponašanje različito
- Smislenost udruživanja u zajednički razred ProjektOsoba ?
  - Naziv projekta nema smisla kada se objekt referencira iz konteksta projekta
  - slično vrijedi za neka druga svojstva, ovisno o kontekstu
- Rješenje?
  - poslovna logika koja će baciti iznimku ako 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)



+



=

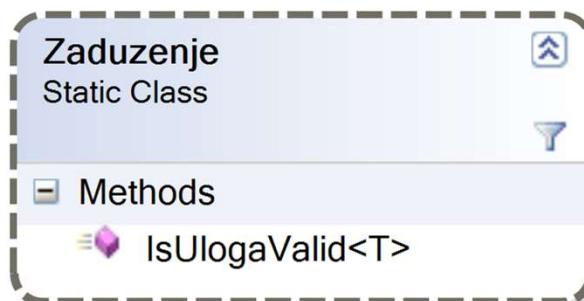


# Vladanje zajedničkim ponašanjem i informacijama

- **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

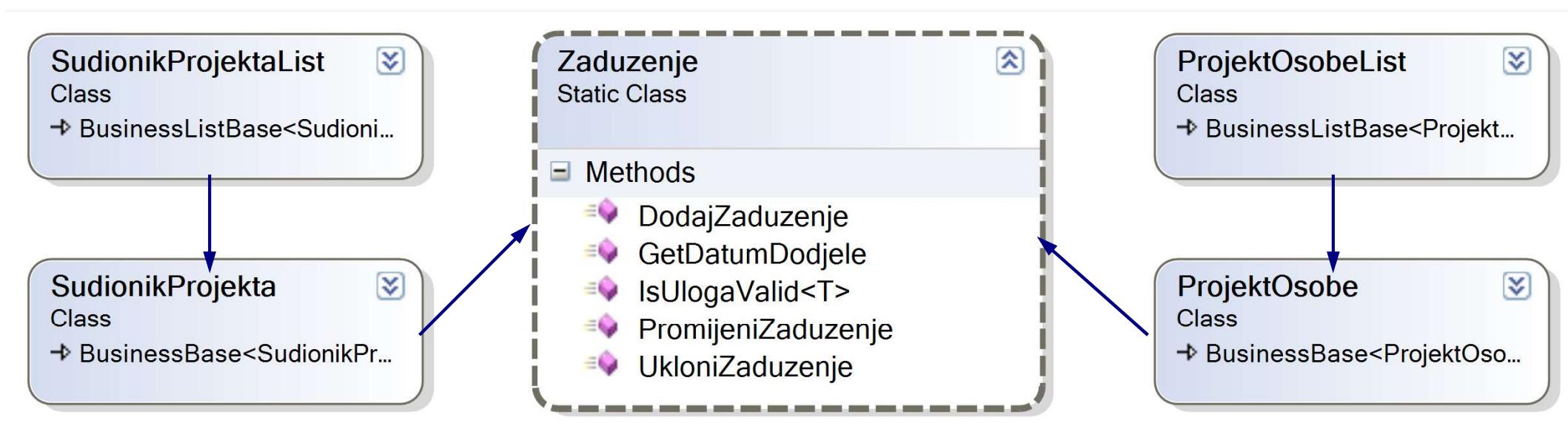
- **DatDodjeleUloge i IdUloge ne treba smještati u zajednički objekt**
  - jer ne razmišljamo relacijski, ne normaliziramo podatke
- **Pitanje je imaju li ti atributi zajedničko ponašanje (poslovno pravilo ili poslovnu logiku) koje se može staviti u zajednički objekt**
  - Primjer: validacija dodijeljene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



# 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
  - Koji znaju šifru projekta, odnosno osobe
- kolekcijski razredi samo će dodavati i uklanjati ove objekte
- naravno, koristiti i druge postupke razreda Zaduzenje



# Poboljšanje performansi

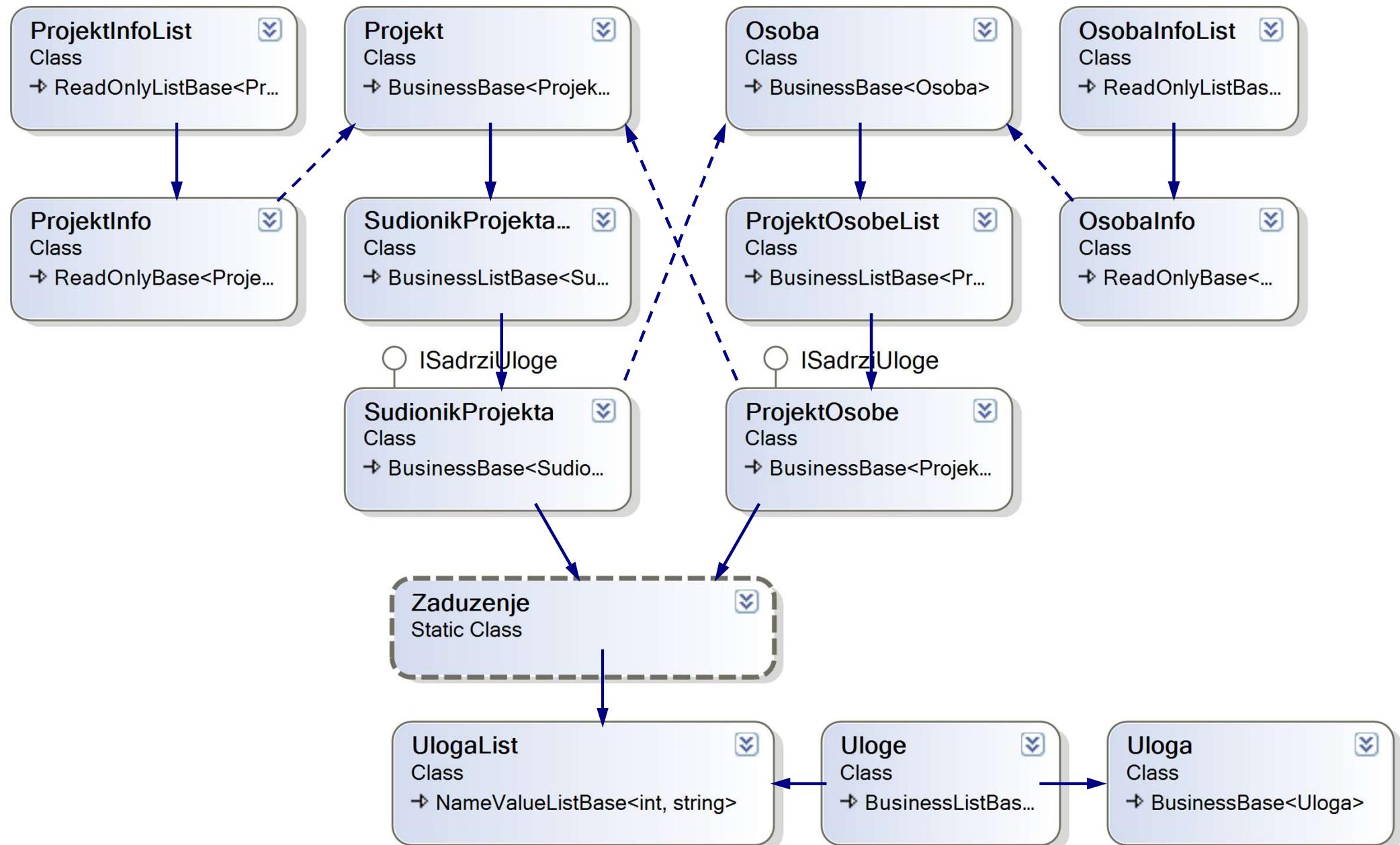
- SudionikProjektaList i ProjektOsobeList dohvaćaju kolekcije poslovnih objekata Projekt i Osoba samo da bi ih prikazali u listi
  - učitavanje objekta Projekt ili Osoba povlači učitavanje lista zaduženja ... (kaskadno)
- Umjesto učitavanja hijerarhija - formirati **read-only** liste
  - Uvode se potrebni 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`
- **Iza 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	

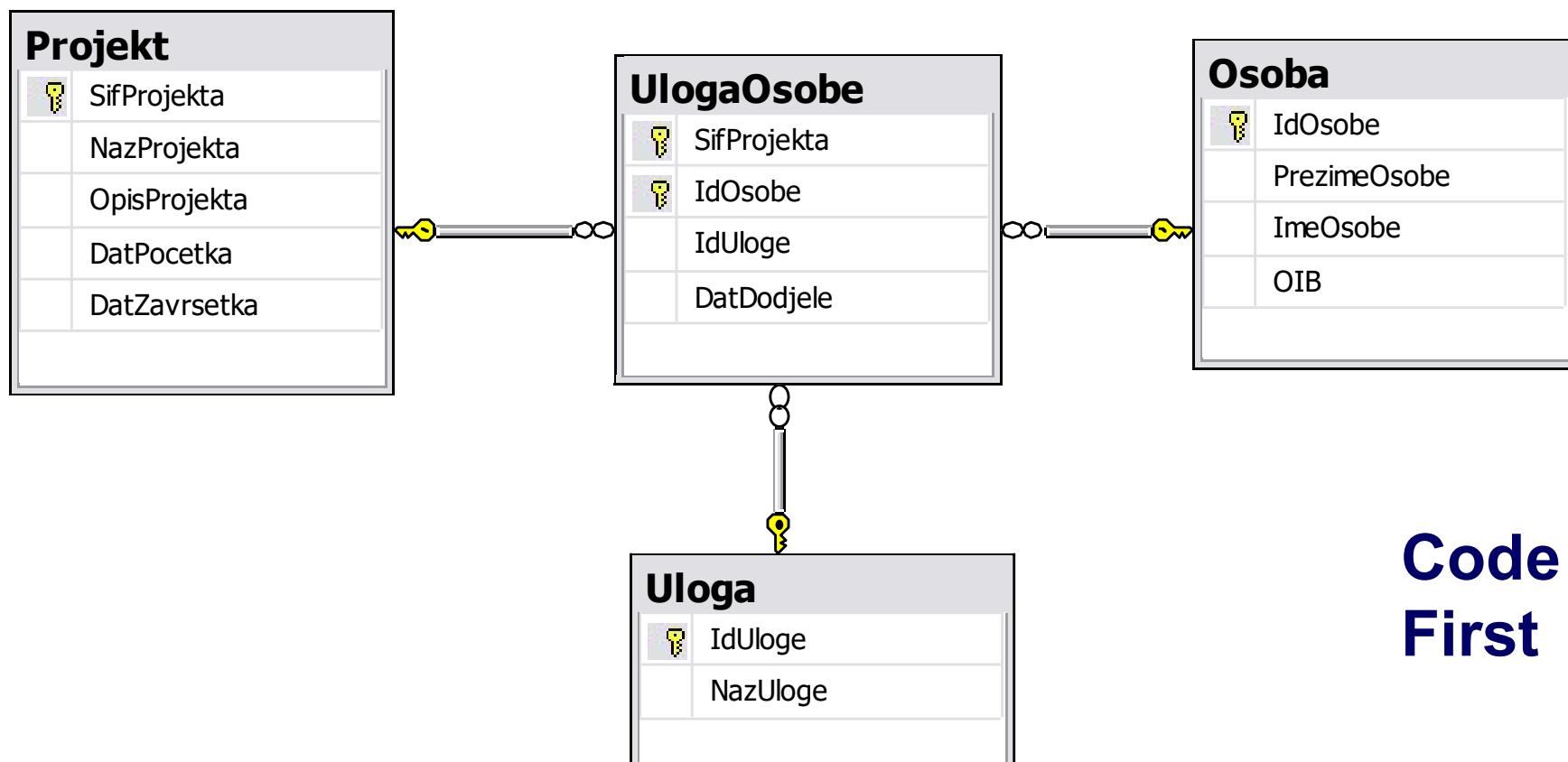
# Na kraju

## □ 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)

- tehnička konverzija podataka između relacijske BP i OO jezika



Code  
First



# Prilozi i reference

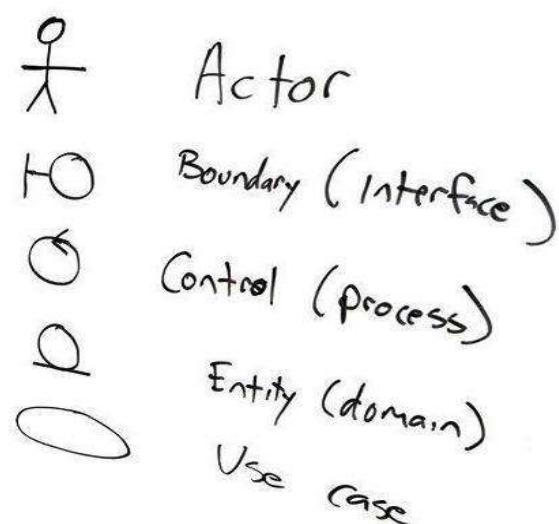
## Primjeri IS06-dodatak

### Reference

- Dennis et al. System Analysis and Design with UML, Wiley, 2015.
- Rockford Lhotka. Expert C# 2008 Business Objects, Apress, 2008.
- Rebecca Wirfs-Brock: [A Brief Tour of Responsibility-Driven Design](#)
- Vaughn Vernon. Domain-Driven Design Distilled, Addison-Wesley Professional, 2016.
- Avram i Marinescu. [Domain-Driven Design Quickly](#), InfoQ, 2007.

### Agilno modeliranje

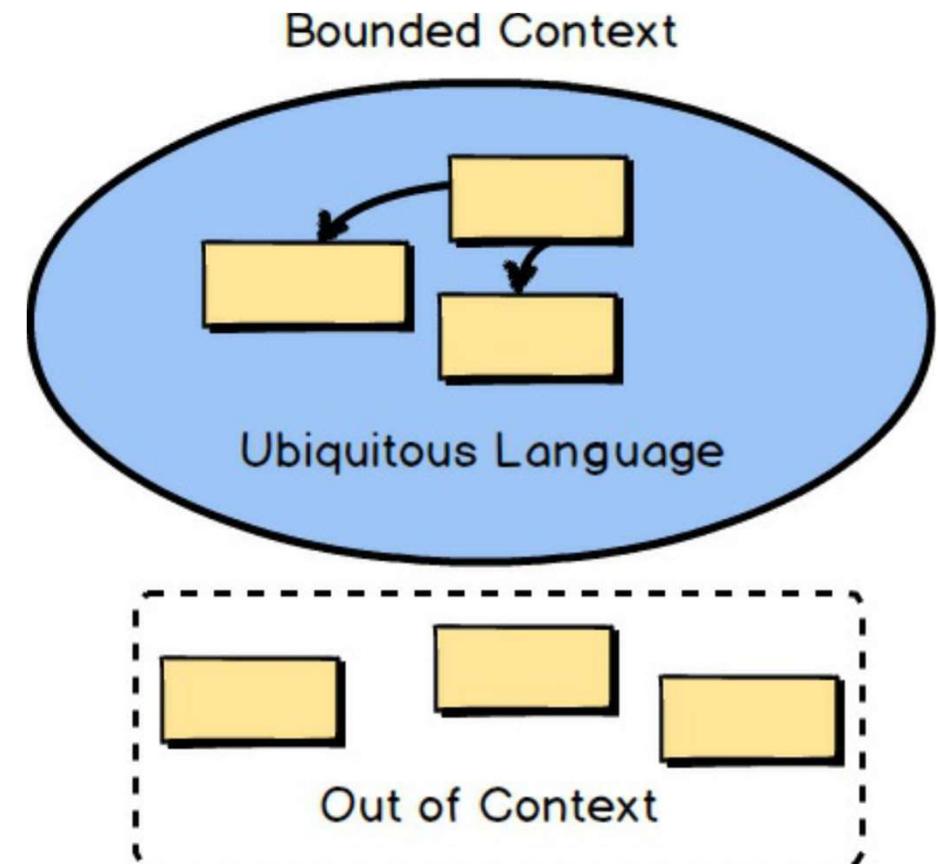
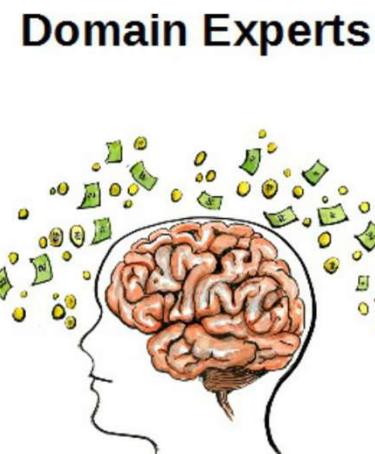
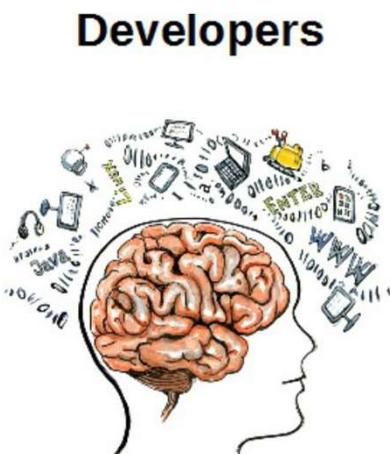
- <http://www.agilemodeling.com>
- .../artifacts/robustnessDiagram.htm



# Domenom vođen dizajn

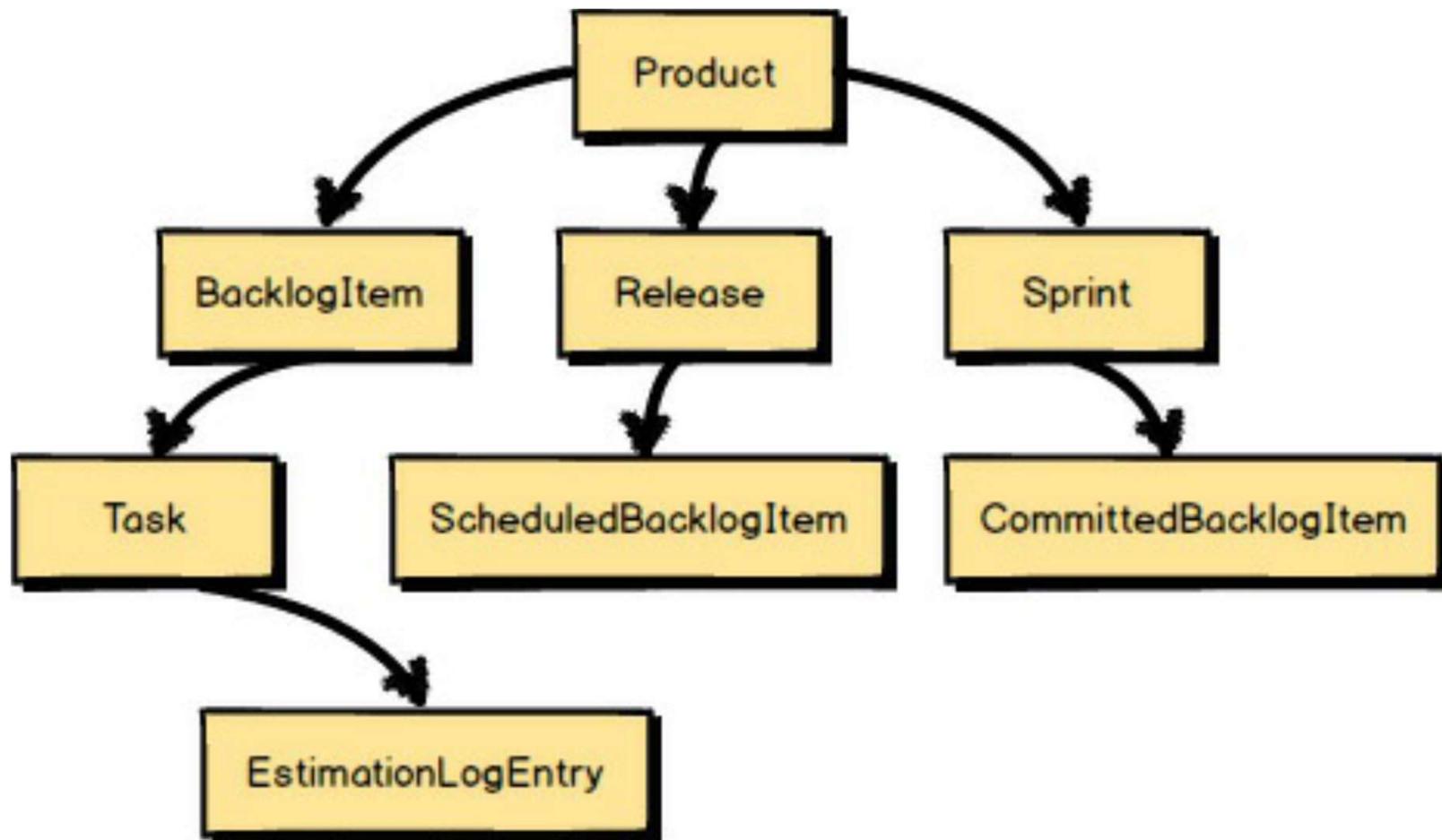
# Domain-Driven Design (DDD)

- Domena – problemsko područje
- DDD – projektiranje softvera na temelju ulaza stručnjaka domene
- *Sveprisutni jezik* – razumljiv razvojnicima i *domencima*
- *Ograničeni kontekst* – granica/opseg (pod)domene



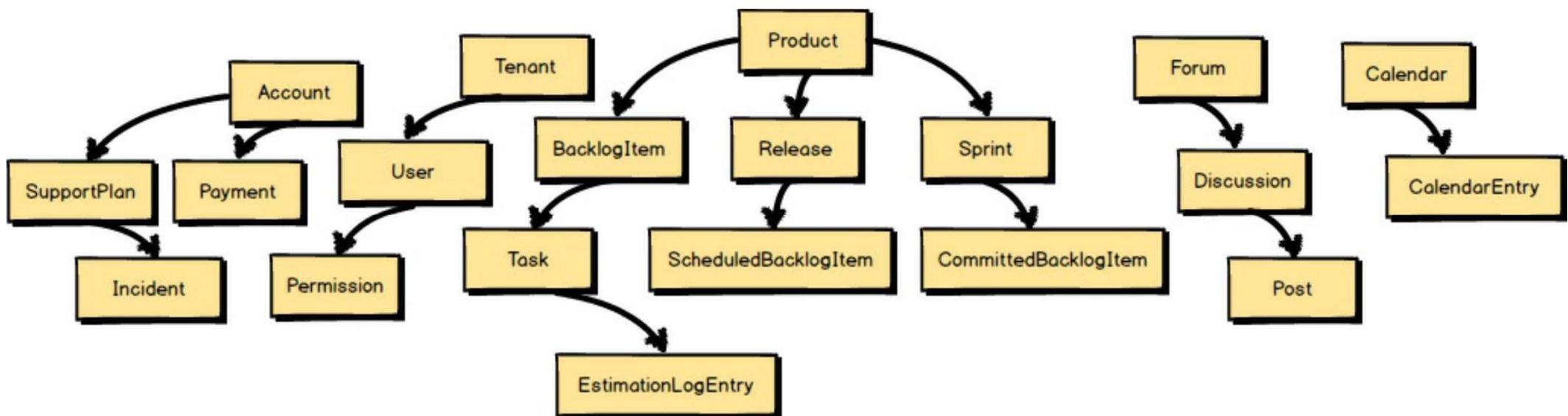
# Primjer, zahtjevi na programsko rješenje

## □ Platforma za agilno upravljanje projektom metodom Scrum

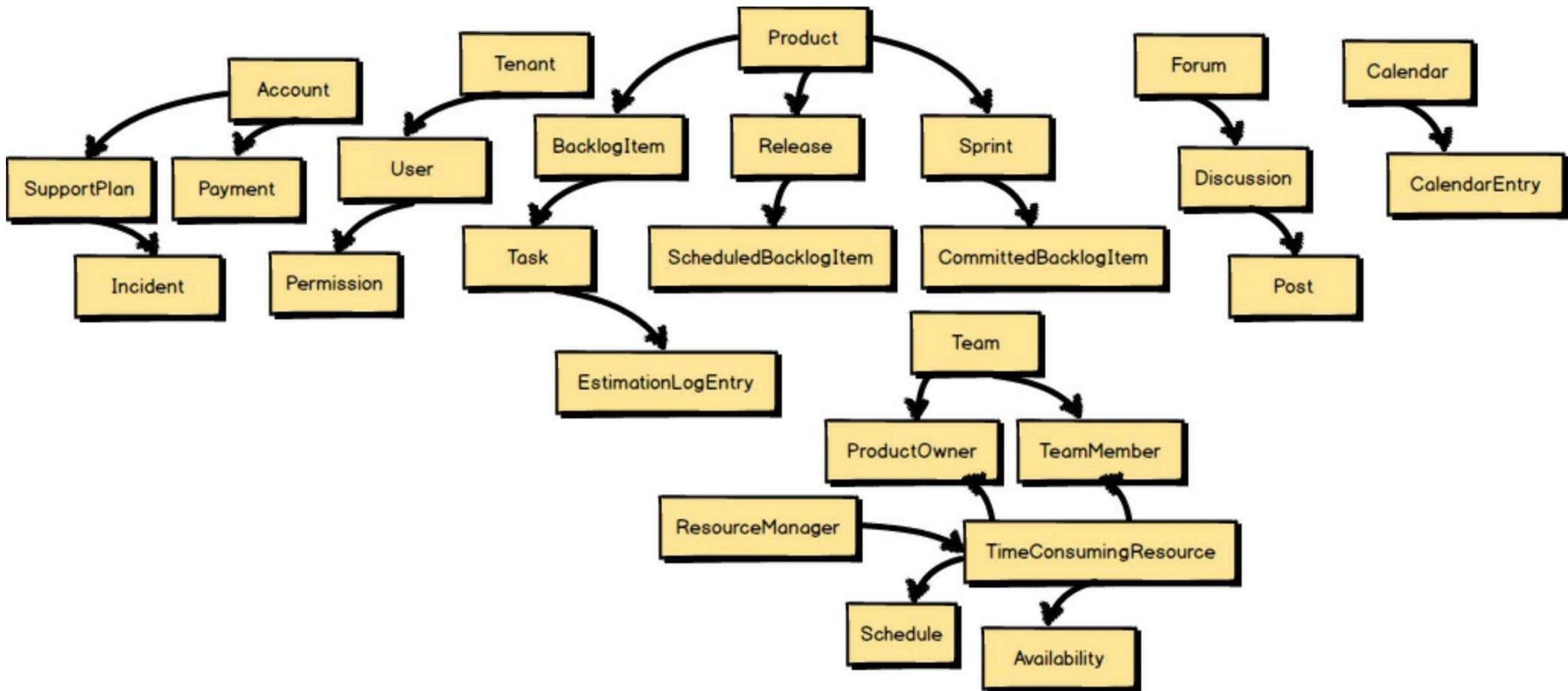


# Još zahtjeva ...

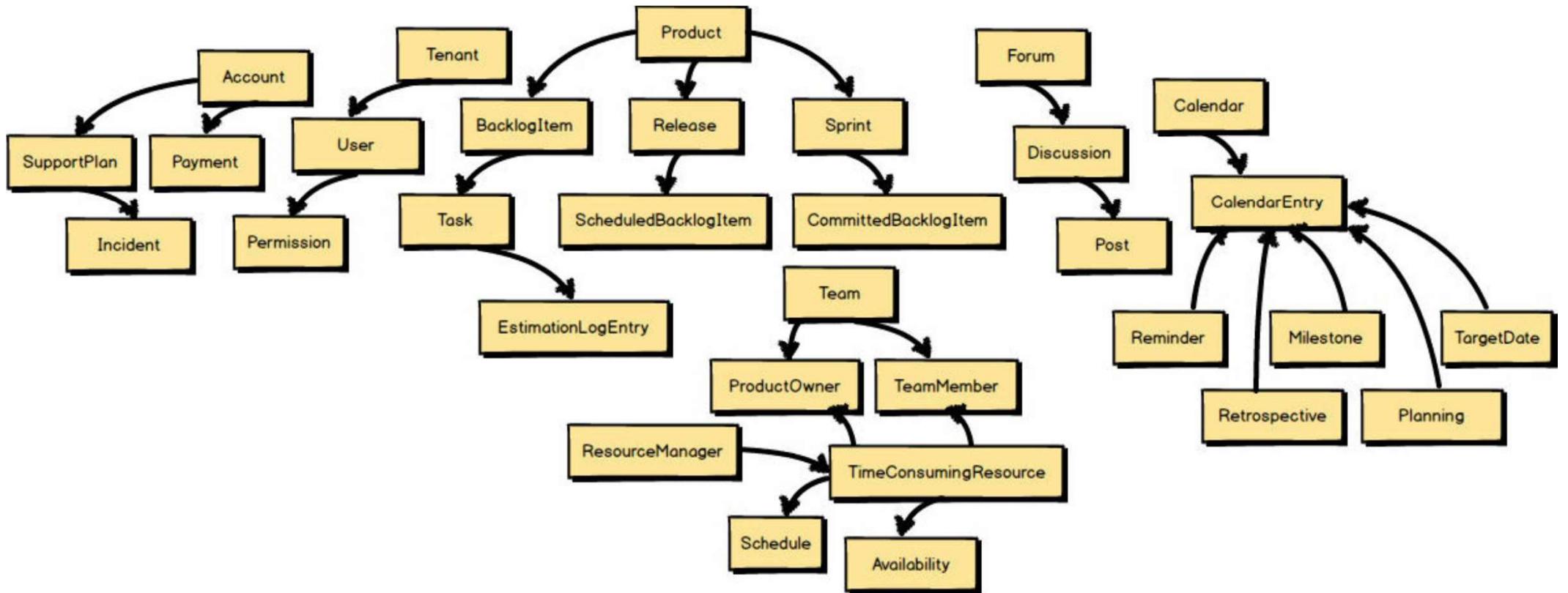
- ☐ Trebamo evidenciju korisnika iz različitih organizacija (Tenant) ... i kalendare ... i forume ... i naplatu korisnicima



# I još uloge članova te njihove raspoloživosti ...



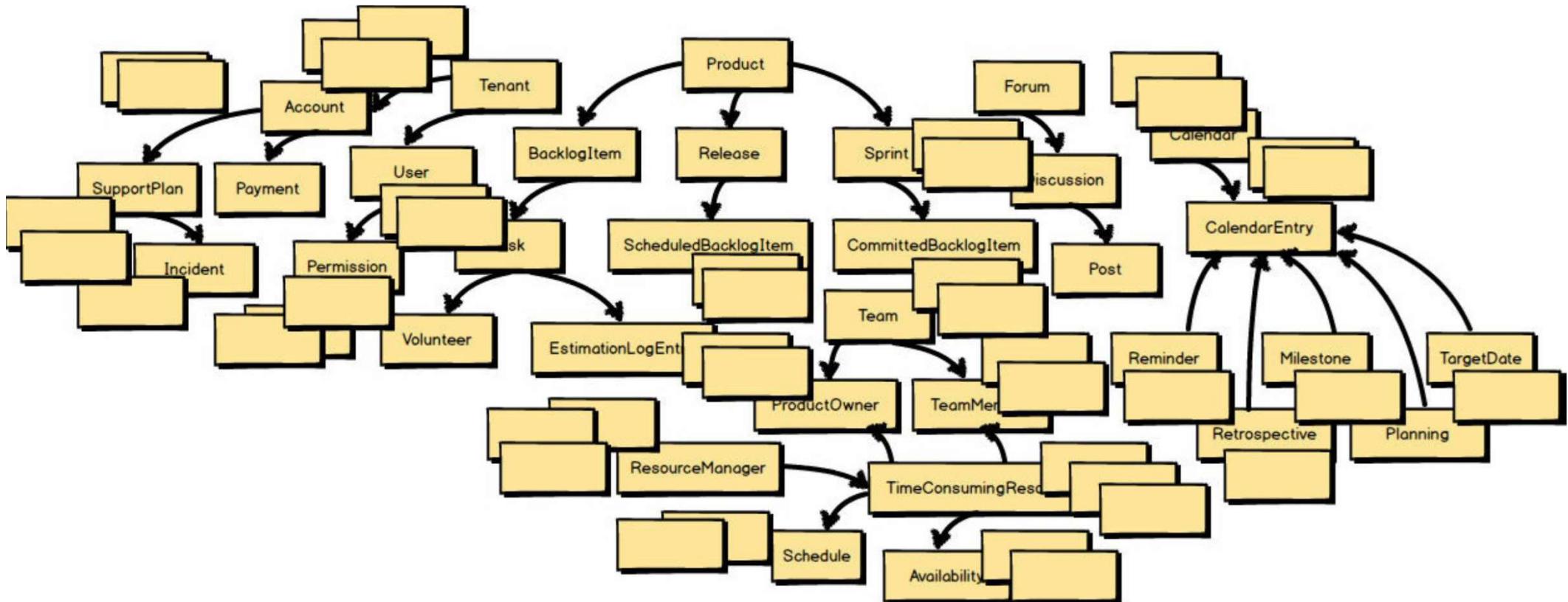
# I malo jači kalendar ...



# I još i još ...

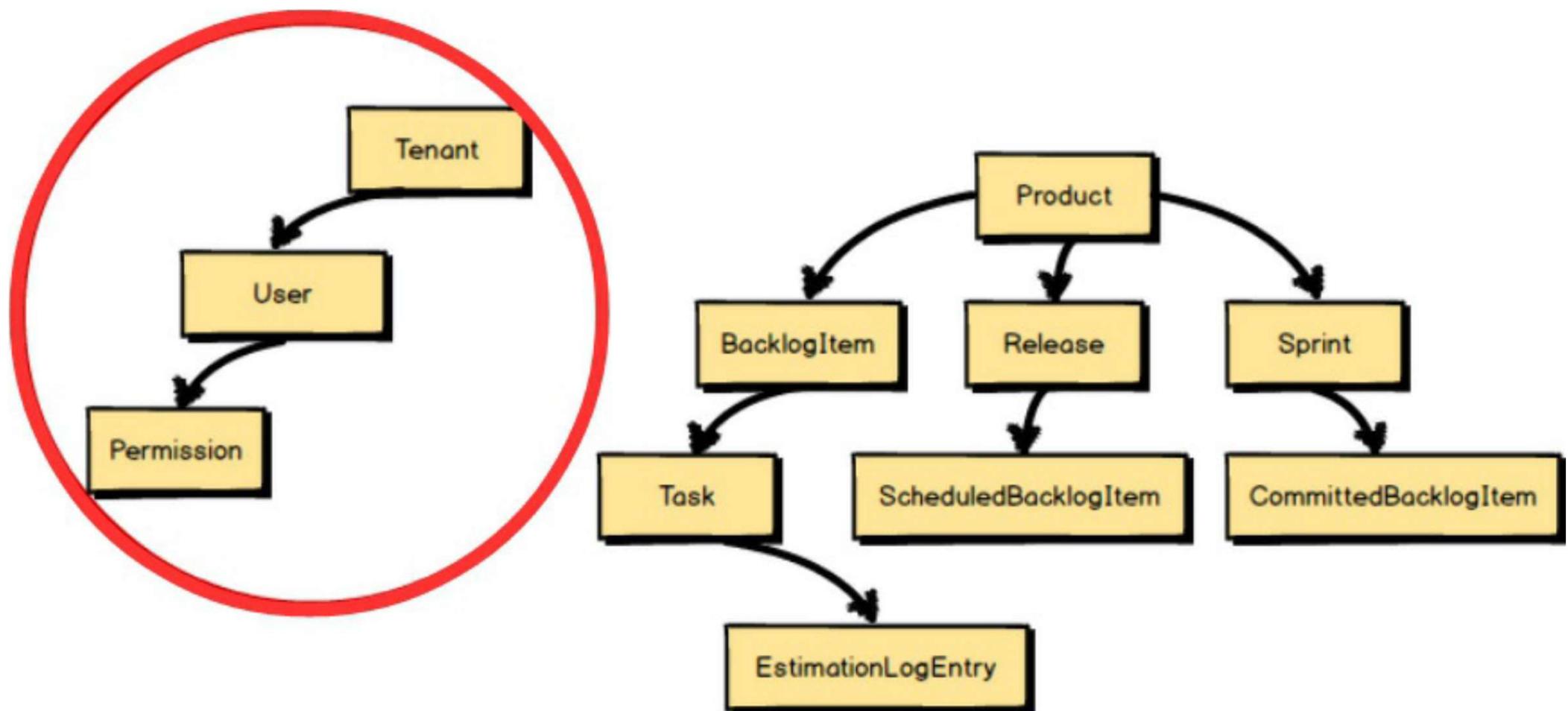


- Big Ball of Mud***
- ... No Big Design Up Front (jer će se promijeniti)***



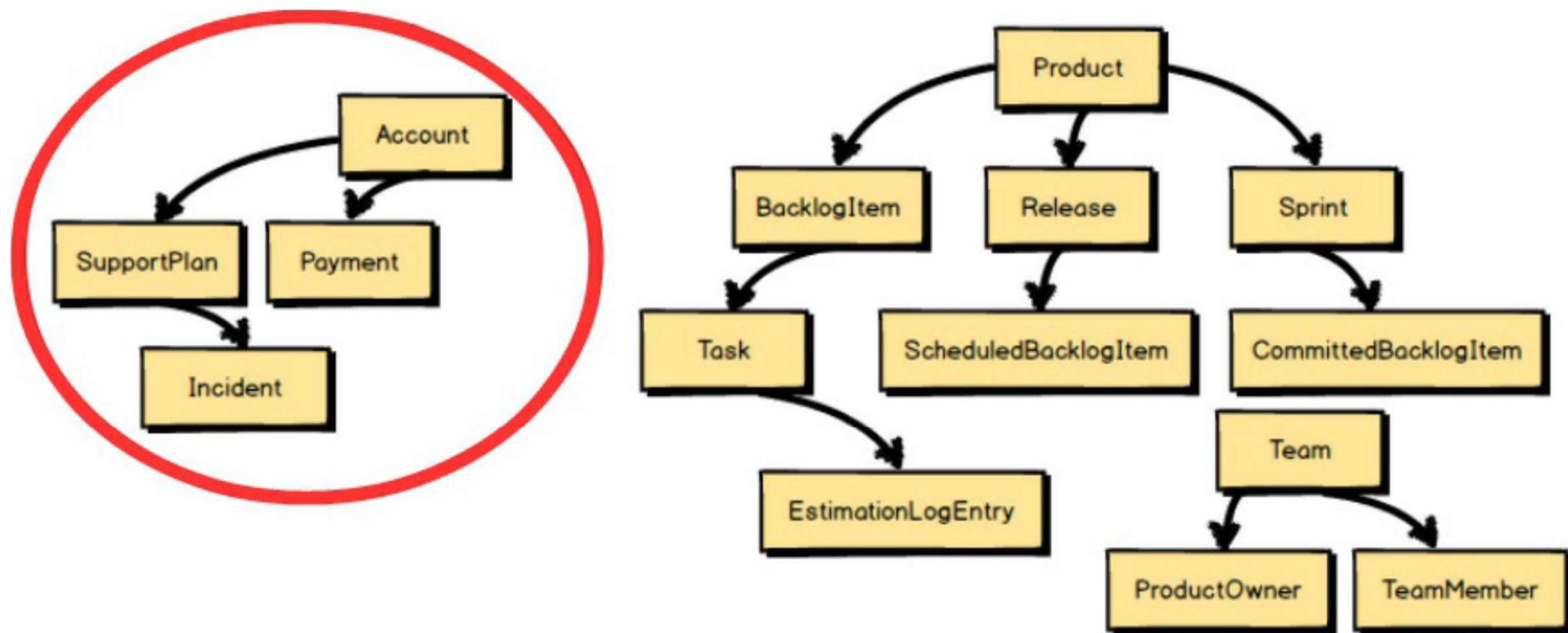
# Rješenje - korisnici

- Nema veze sa Scrumom, odvojiti



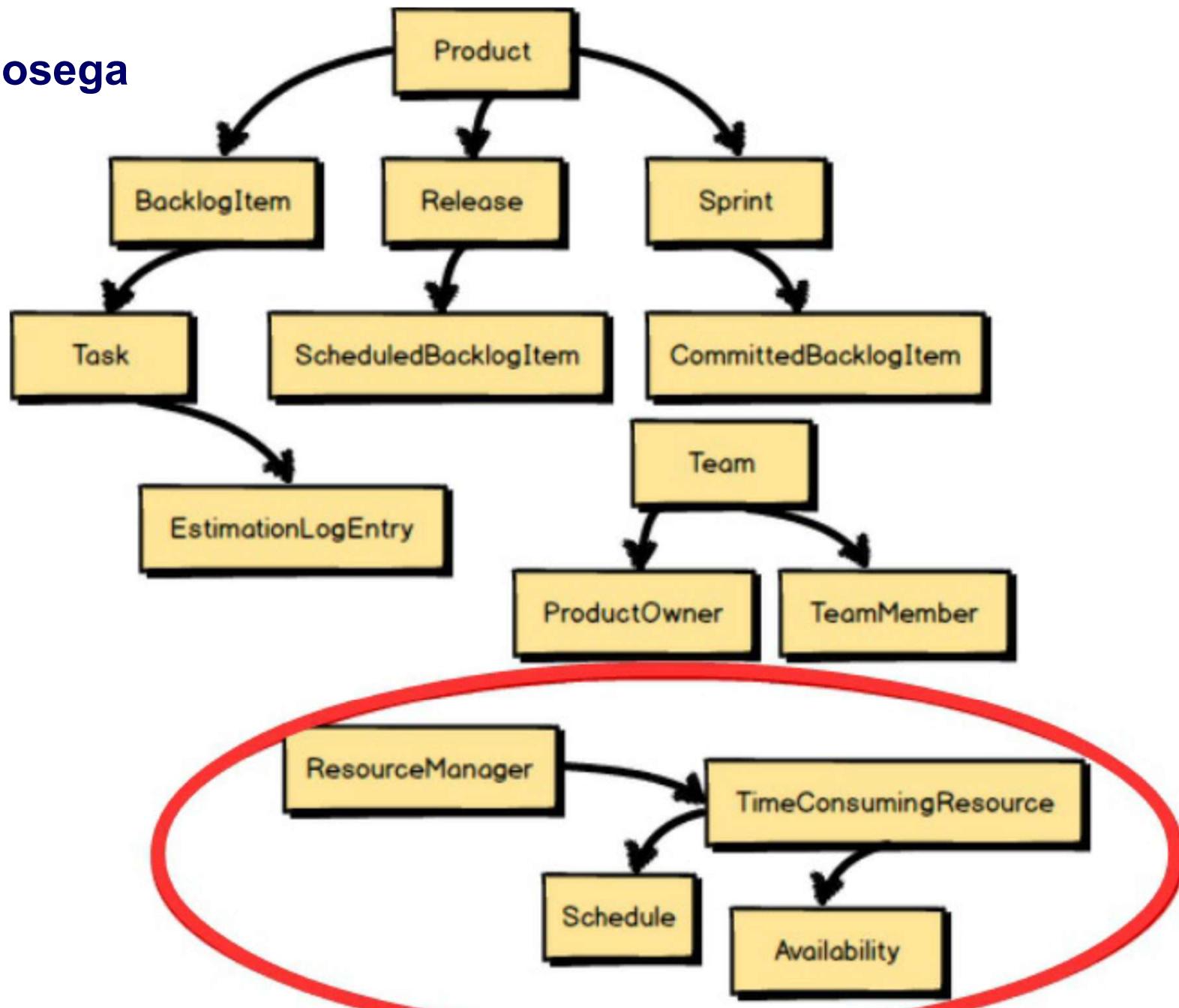
# Rješenje - podrška

- Team, ProductOwner i TeamMember ostaviti jer su dio Scrum
- Bavljenje podrškom i naplatom odvojiti



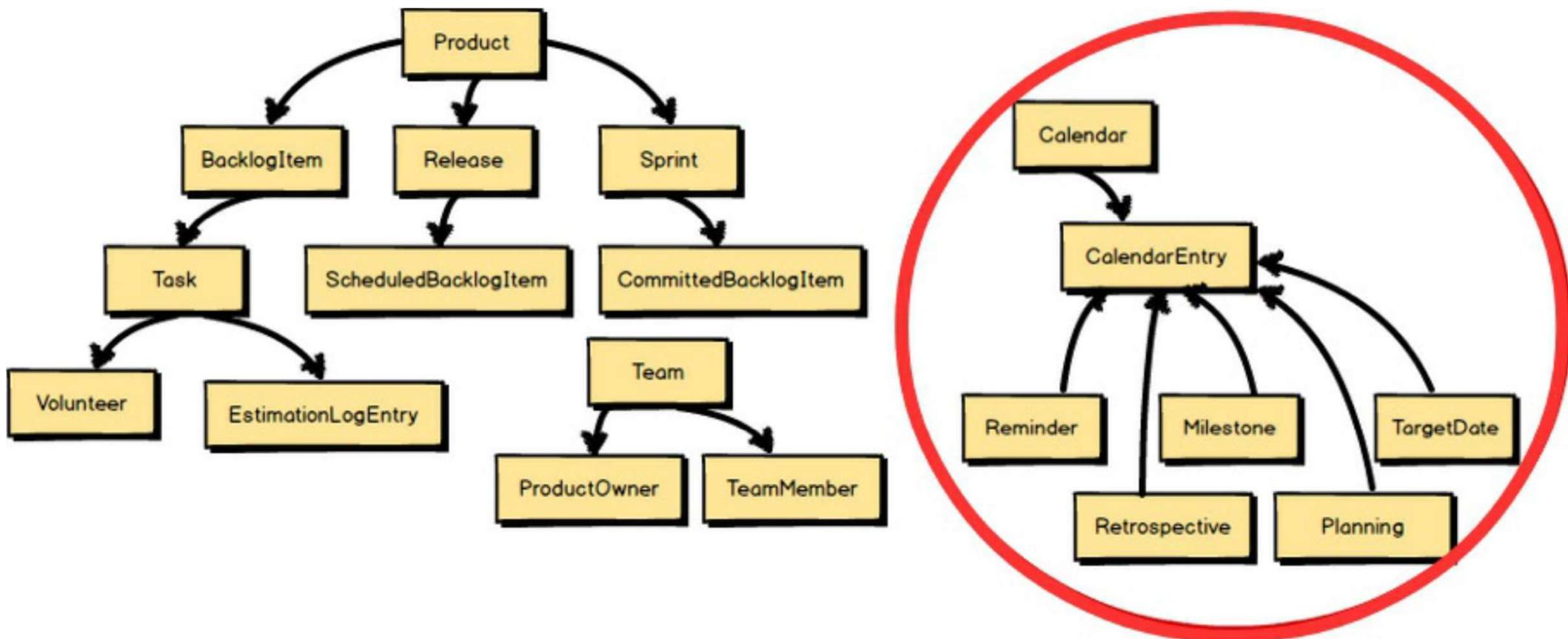
# Rješenje - kadroviranje

Izvan dosega



# Rješenje – napredni kalendar

- U kontekstu, ali privremeno izvan opsega



# Sveprisutni jezik

## □ Scenarij

- The product owner commits a backlog item to a sprint. The backlog item may be committed only if it is already scheduled for release, and if a quorum of team members have approved commitment. If it is already committed to a different sprint, it must be uncommitted first. When the commitment completes, notify the sprint from which it was uncommitted and the sprint to which it is now committed.

## □ Izvršna specifikacija – tehnika *Specification by Example*

- Drugim riječima: **Behavior-Driven Development (BDD)**

---

Scenario: The product owner commits a backlog item to a sprint  
Given a backlog item that is scheduled for release  
And the product owner of the backlog item  
And a sprint for commitment  
And a quorum of team approval for commitment  
When the product owner commits the backlog item to the sprint  
Then the backlog item is committed to the sprint  
And the backlog item committed event is created

# Pripadni jedinični test

```
[Test]
public void ShouldCommitBacklogItemToSprint()
{
    // Given
    var backlogItem = BacklogItemScheduledForRelease();

    var productOwner = ProductOwnerOf(backlogItem);

    var sprint = SprintForCommitment();

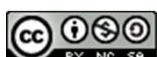
    var quorum = QuorumOfTeamApproval(backlogItem, sprint);

    // When
    backlogItem.CommitTo(sprint, productOwner, quorum);

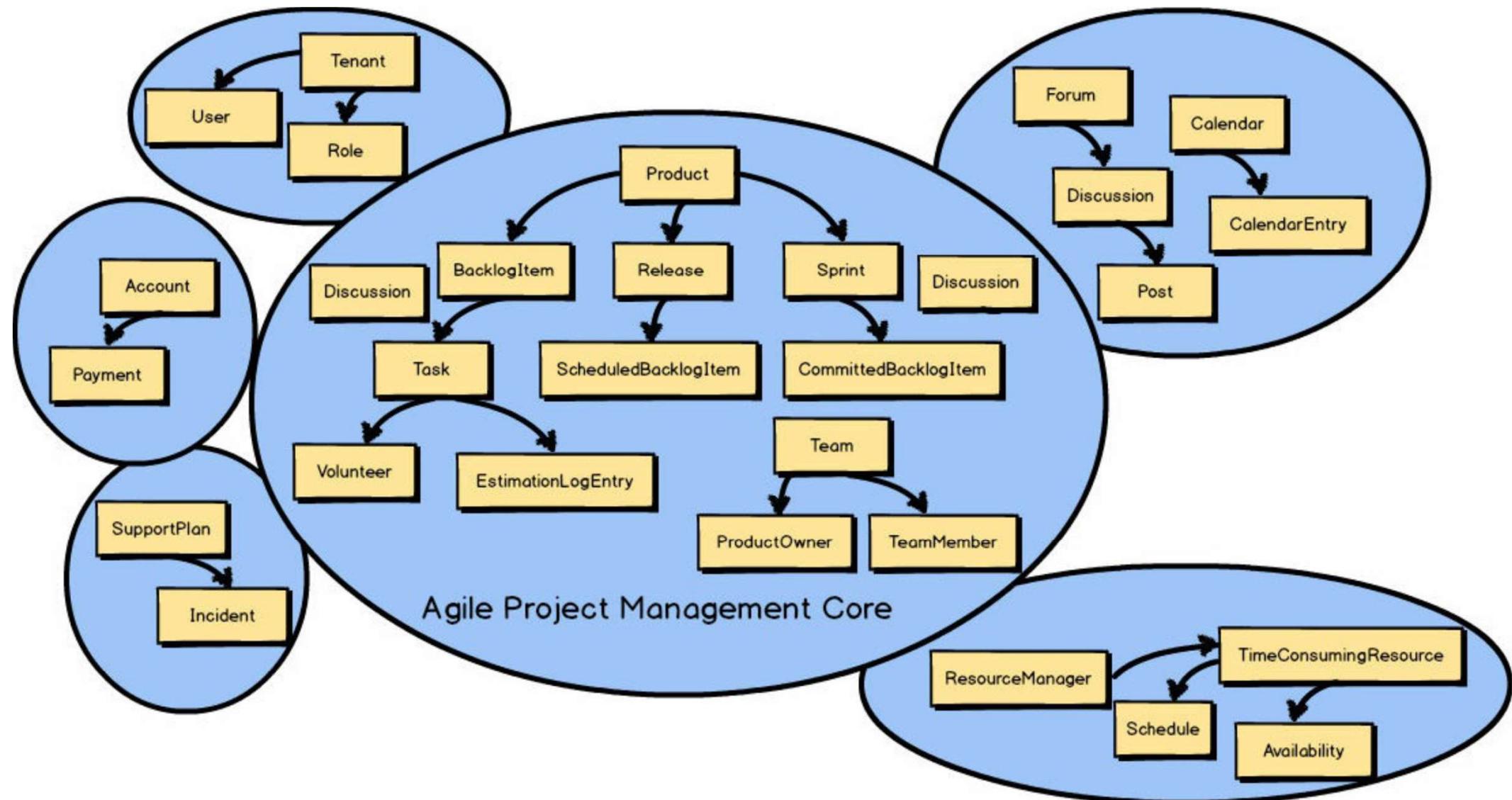
    // Then
    Assert.IsTrue(backlogItem.IsCommitted());

    var backlogItemCommitted =
        backlogItem.Events.OfType<BacklogItemCommitted>().SingleOrDefault();

    Assert.IsNotNull(backlogItemCommitted);
}
```

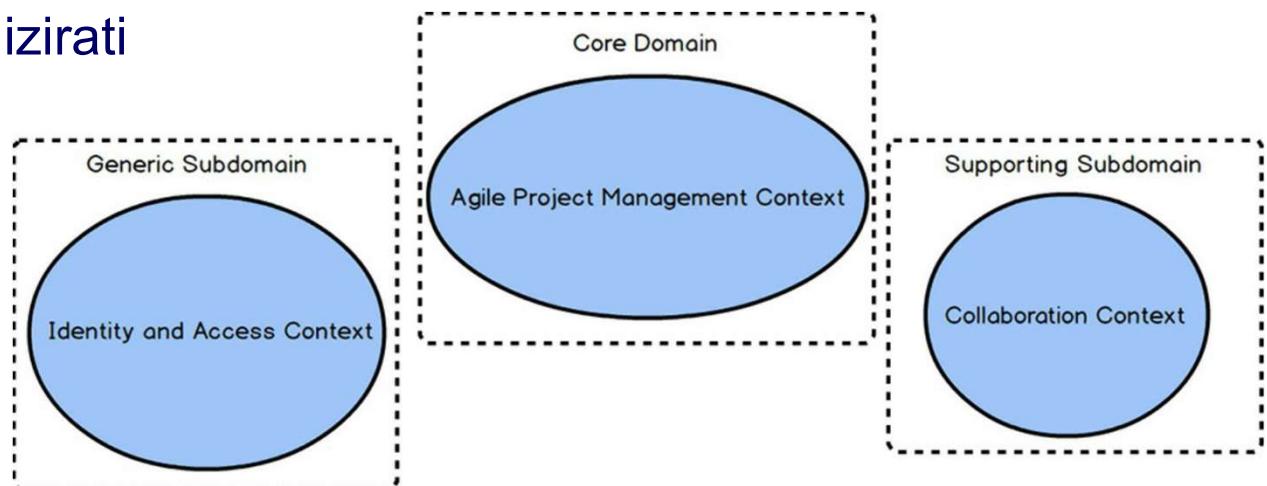


# Domena i poddomene



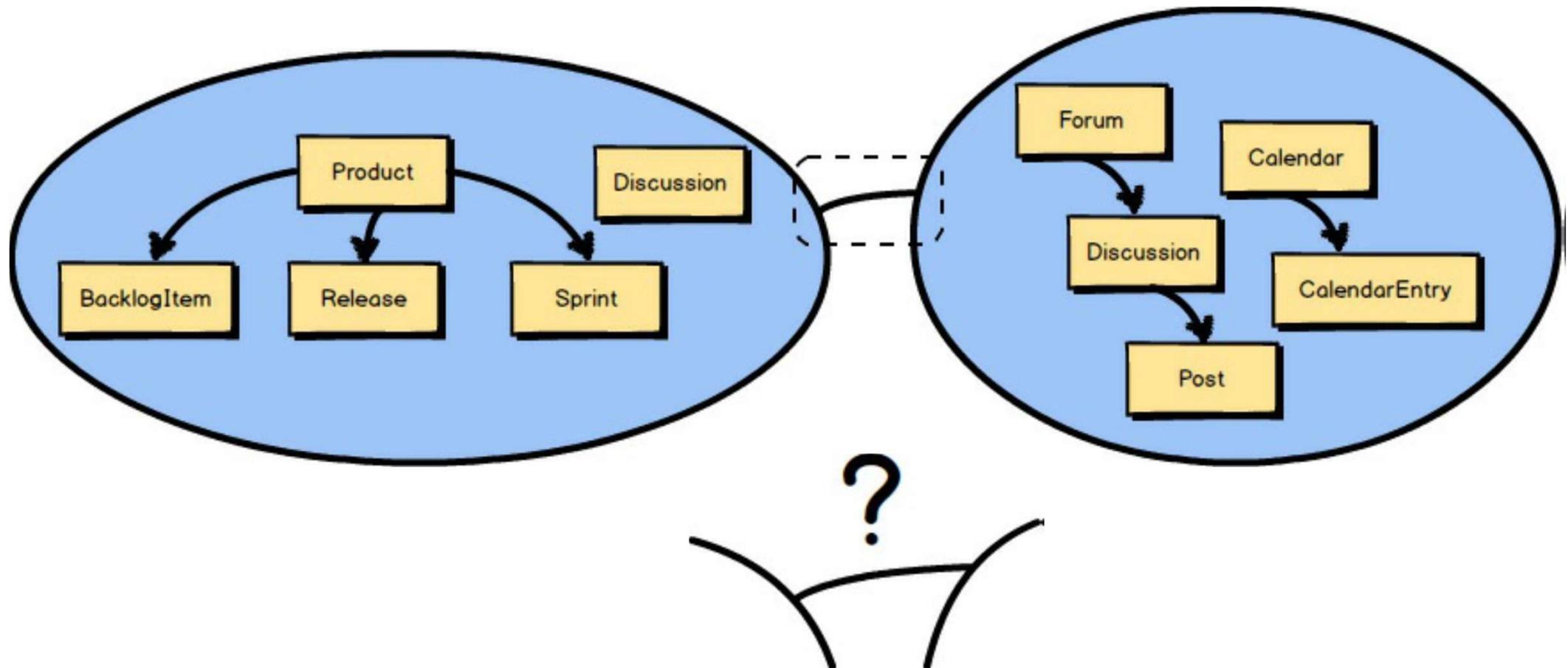
# Poddomene (*build vs buy vs rent još jedared!*)

- „pojedinačni, logički model područja”
- **Jezgrena (Core Domain)**
  - Dobro definiran model
  - Posebnost organizacije - kompetitivna prednost – vlastiti razvoj
- **Potporna (Supporting Subdomain)**
  - Zahtijeva razvoj po mjeri
  - Ne postoji gotovo rješenje na tržištu – vanjski razvoj (outsourcing)
- **Opća poddomena (Generic Subdomain)**
  - Kupiti gotovo ili eksternalizirati



# Strateški dizajn mapiranjem konteksta

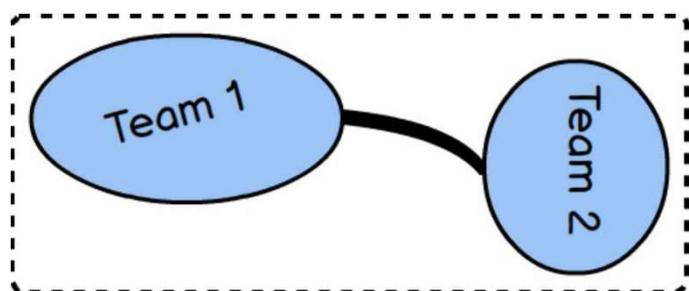
- Veliki projekti, više ekipa - očuvanje integriteta modela



# Mapiranje konteksta (2)

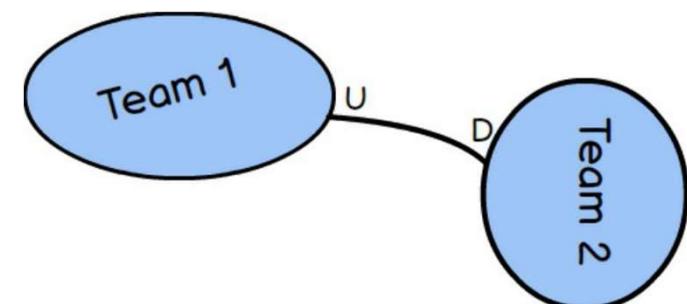
## □ Partnership

- Sinkronizacija
- Kontinuirana integracija



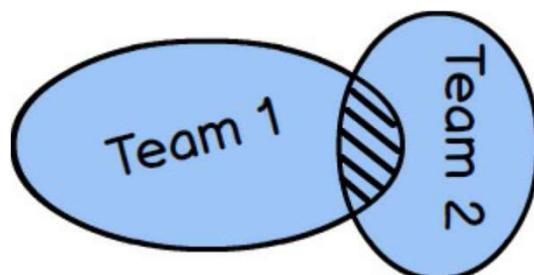
## □ Customer-Supplier

- Supplier = upstream (U)
  - Određuje kad će i što Customer dobiti
- Customer = downstream (D)



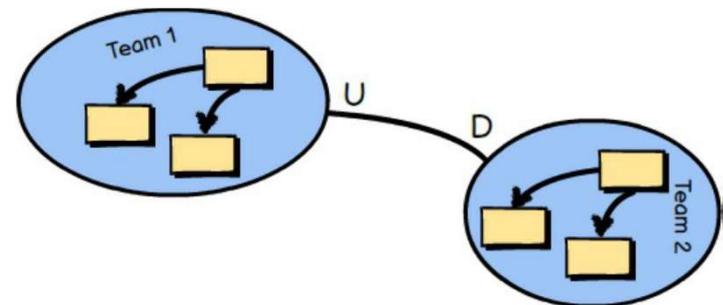
## □ Shared kernel

- Dijeljeni dio modela



## □ Conformist

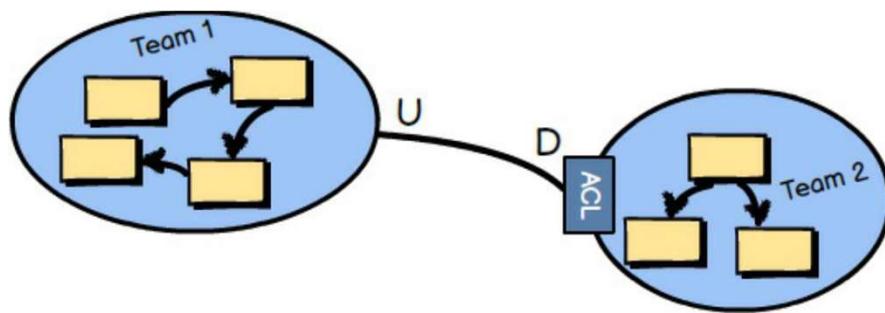
- Preuzimanje U modela umjesto preslikavanja
- Pr. kad je dobar (amazon.com)



# Mapiranje konteksta (3)

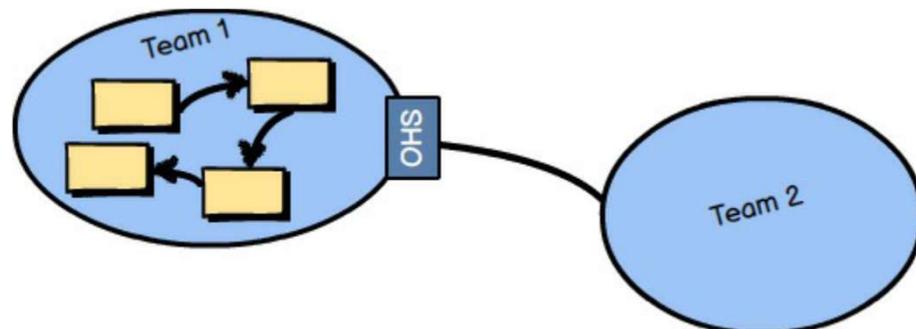
## **Anticorruption Layer**

- Translacija modela u prikladni



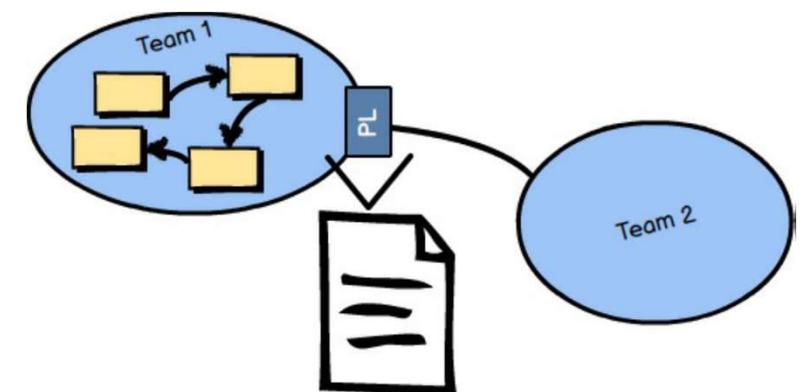
## **Open Host Service**

- OHS definira protokol ili sučelje
- Otvoren i lagan za korištenje – provokira konformista



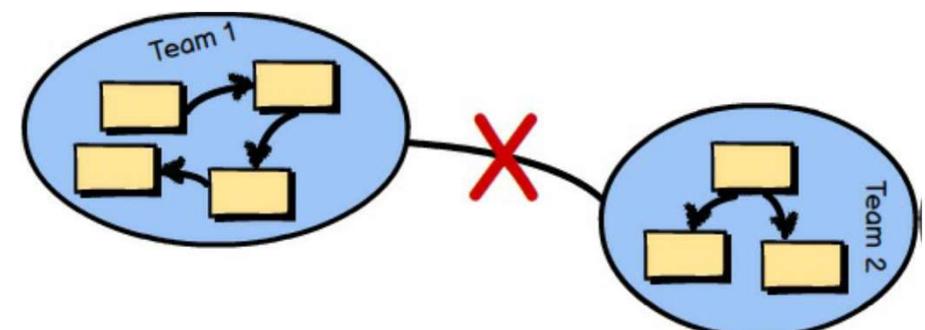
## **Published Language**

- Dobro dokumentirani jezik razmjene
- Npr. XML schema, JSON schema

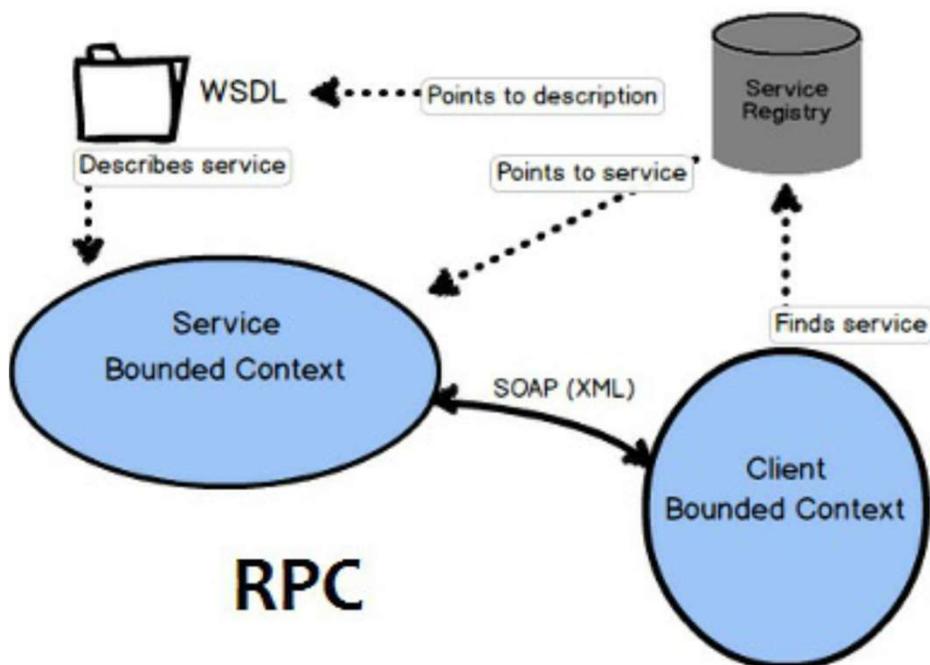


## **Separate Ways**

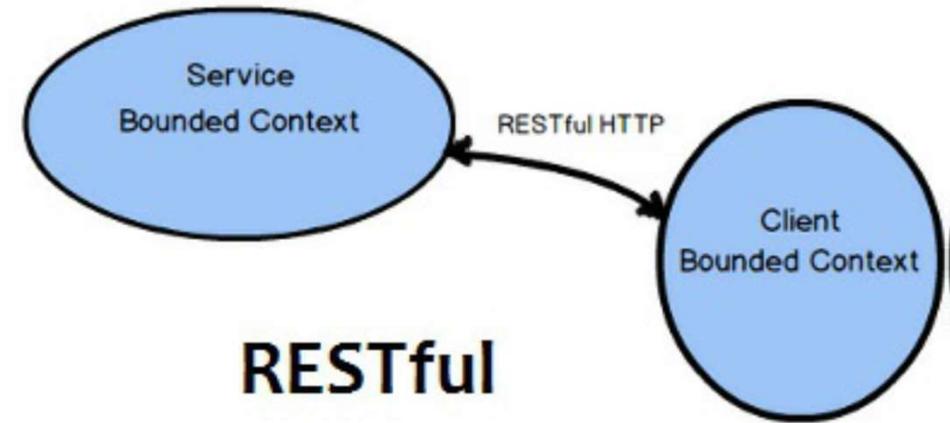
- Vlastito specijalizirano rješenje
- Za tu domenu, ne nužno „uvijek“



# Mehanizmi mapiranja (o arhitekturi drugi put)

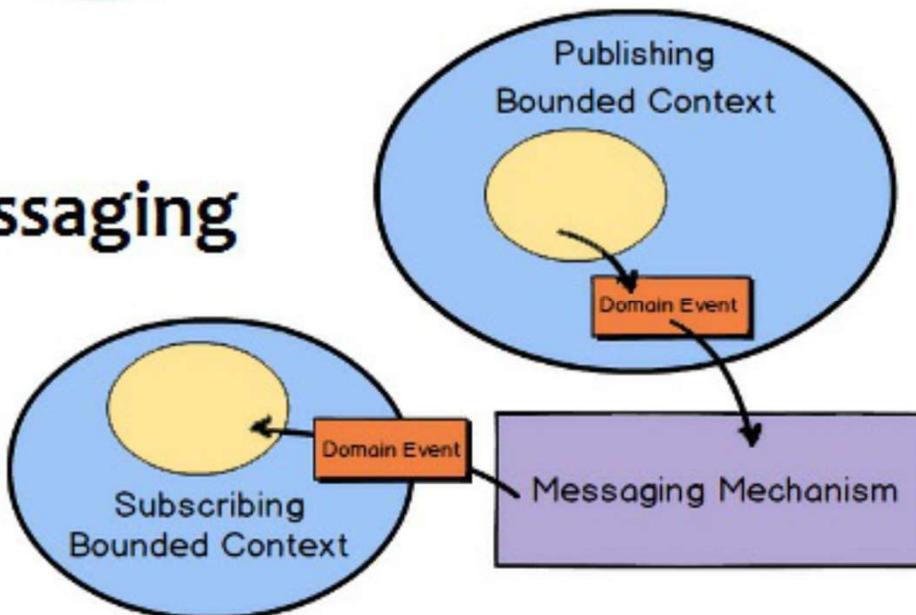


**RPC**

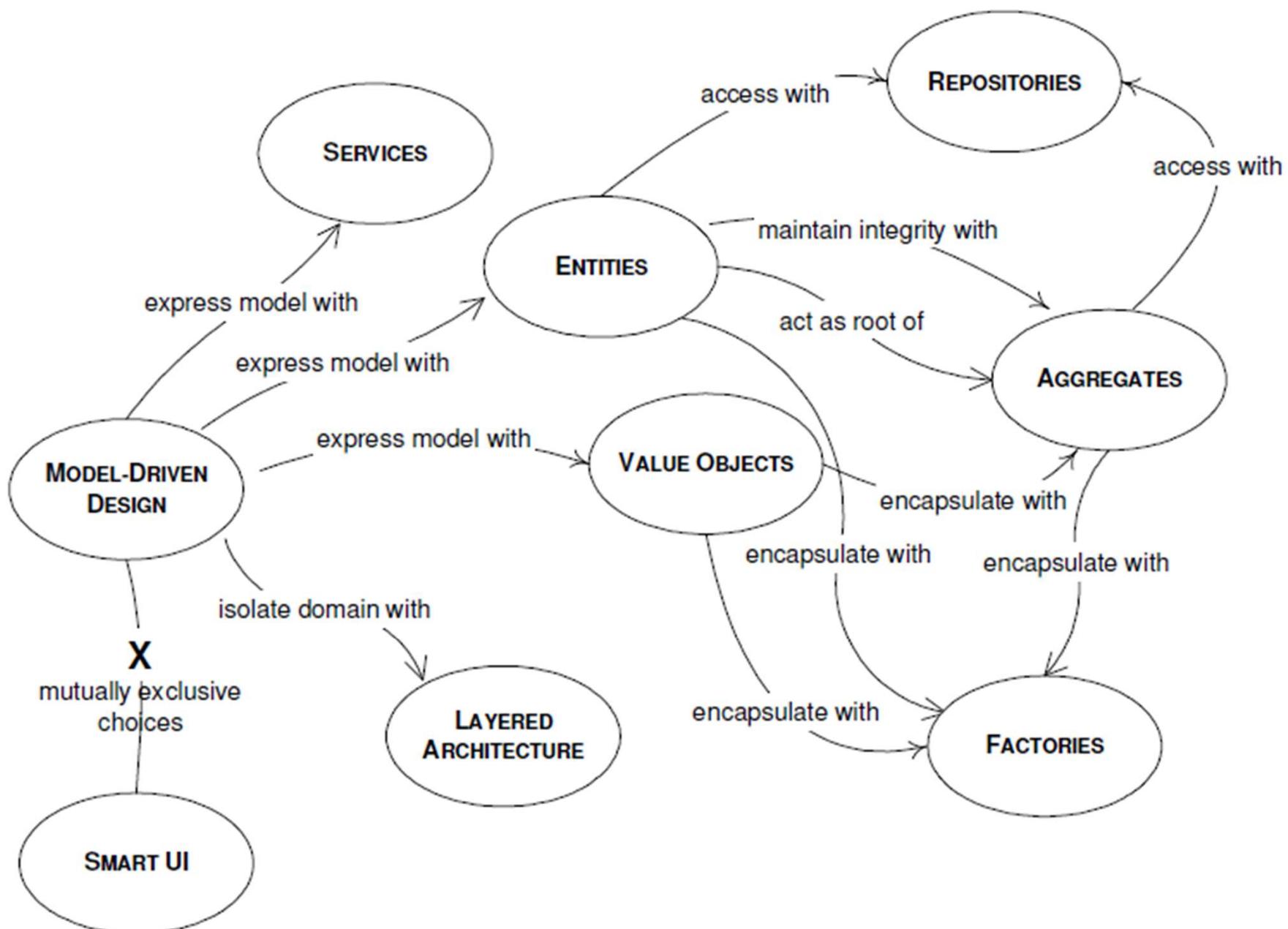


**RESTful**

**Messaging**



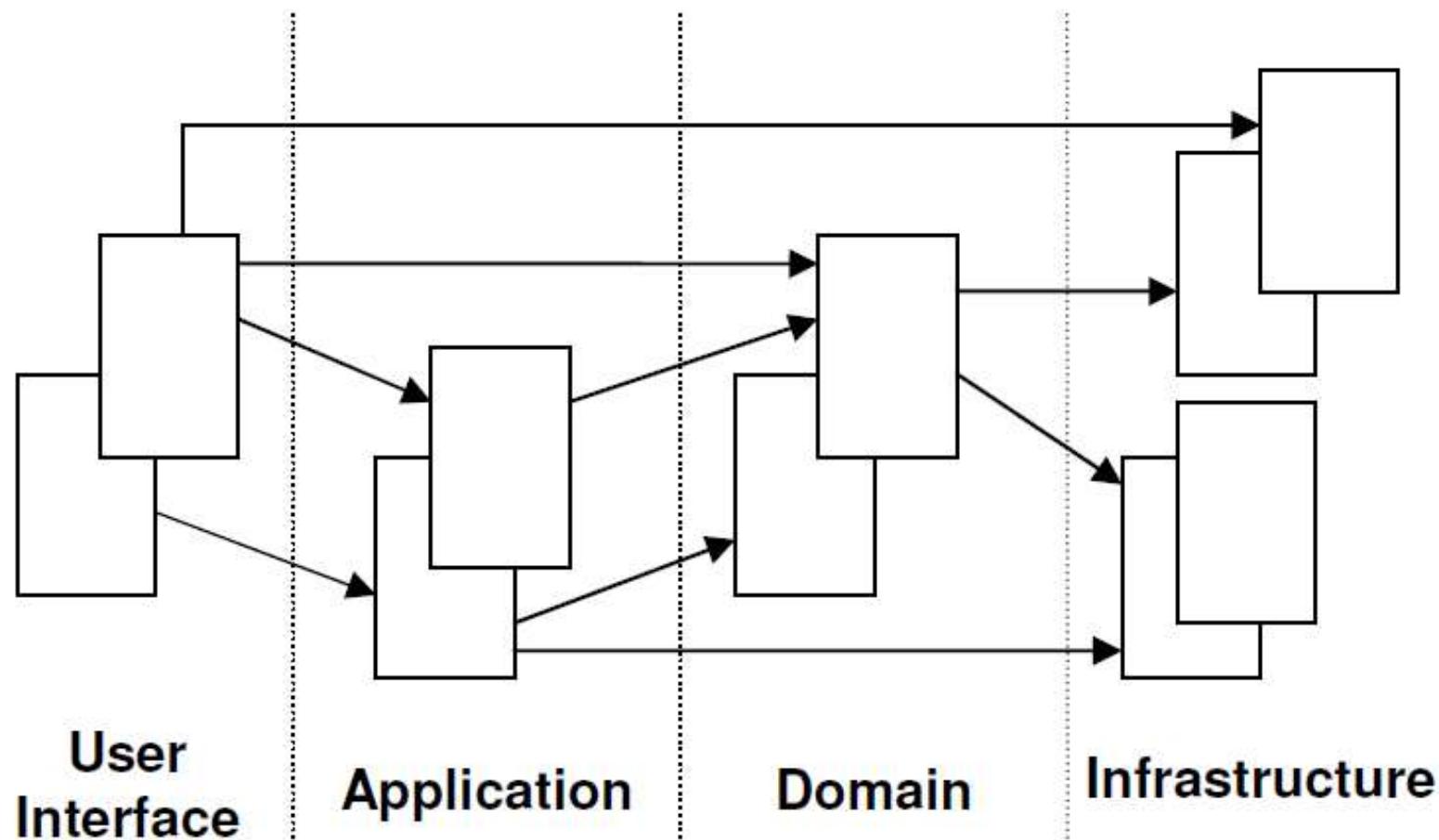
# Gradjevni blokovi modelom vodenog dizajna (MDD)



# Slojevita arhitektura

## □ Slojevi

- Korisničko sučelje – prezentacija
- Aplikacijski – koordinacija aplikacijskih aktivnosti
- Domena – stanje poslovnih objekata, poslovna logika
- Potporna knjižnica - komunikacija slojeva, implementacija pohrane



# Entiteti i vrijednosni objekti

## □ Entiteti (entities) – business entity (BE)

- Imaju identitet koji se ne mijenja promjenom stanja softvera
- Atributi entiteta se mogu mijenjati (*mutable*)
- Imaju ponašanje - poslovna logika

## □ Vrijednosni objekti (value objects - VO)

- „stvari“ bez identiteta, jedinstvenosti
- Jednaki ako imaju jednake vrijednosti svih atributa
- Atributi često nepromjenjivi (*immutable*) – uništiš pa stvorиш drugi

### ■ Varijanta – Data Transfer Object (DTO)

- Klasa podataka bez logike

### ■ Izvorno: Plain Old Simple Class Object (POCO)

- .NET: Plain Old Simple CLR Object
- Java: Plain Old Simple Java Object (POJO)

```
// Entity
public class Person
{
    public int Id { get; set; }
    public string Name { get; set; }
    public Address Address { get; set; }
}

// Value Object
public class Address
{
    public string City { get; set; }
    public string ZipCode { get; set; }
}
```

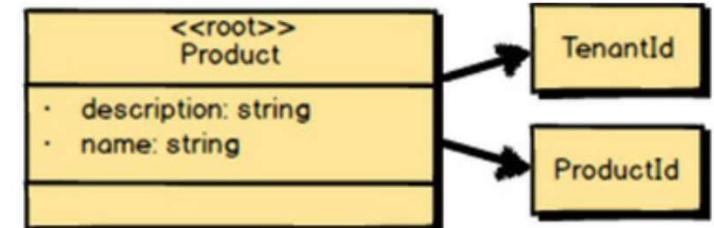
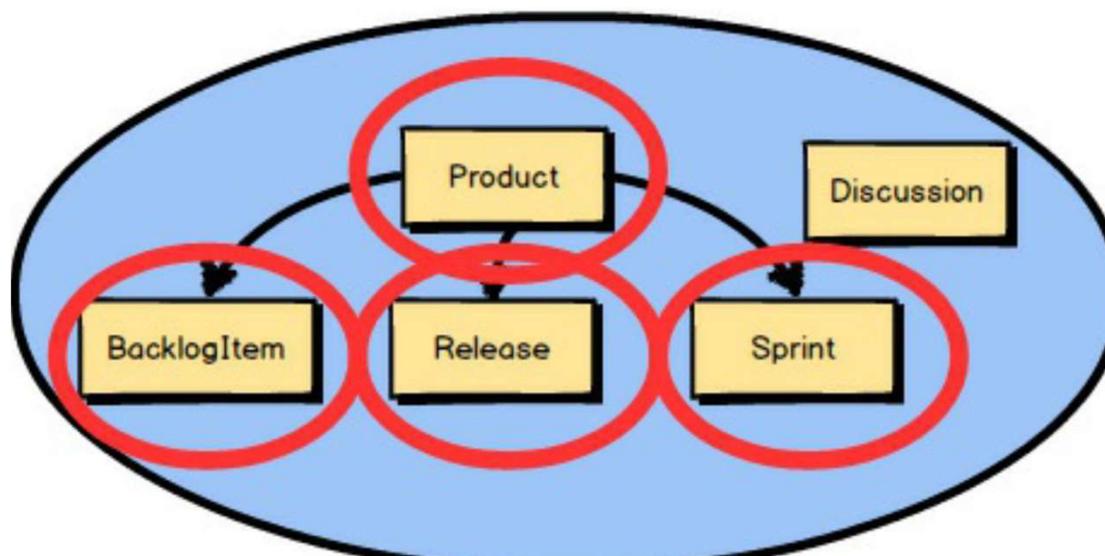


# Agregati

- Grupe entiteta i vrijednosti, tretirani kao cjelina
- Svaki agregat ima jedan korijenski entitet (root)
- Identitet korijena je globalan, identiteti unutarnjih entiteta lokalni
- Vanjski objekti mogu referencirati samo korijen
- Interni objekti ne mogu biti promijenjeni izvan agregata
- Svaki agregat treba biti izmijenjen i pohranjen u zasebnoj transakciji

## □ Primjer:

- Lijevo - svaki zaokruženi je agregat, osim diskusije koja je VO
- Desno – agregat *Product*, kompozitnog identiteta 2 VO, bez gettera/settera



```
public class Product : Entity
{
    private string description;
    private string name;
    private ProductId productId;
    private TenantId tenantId;
}
```

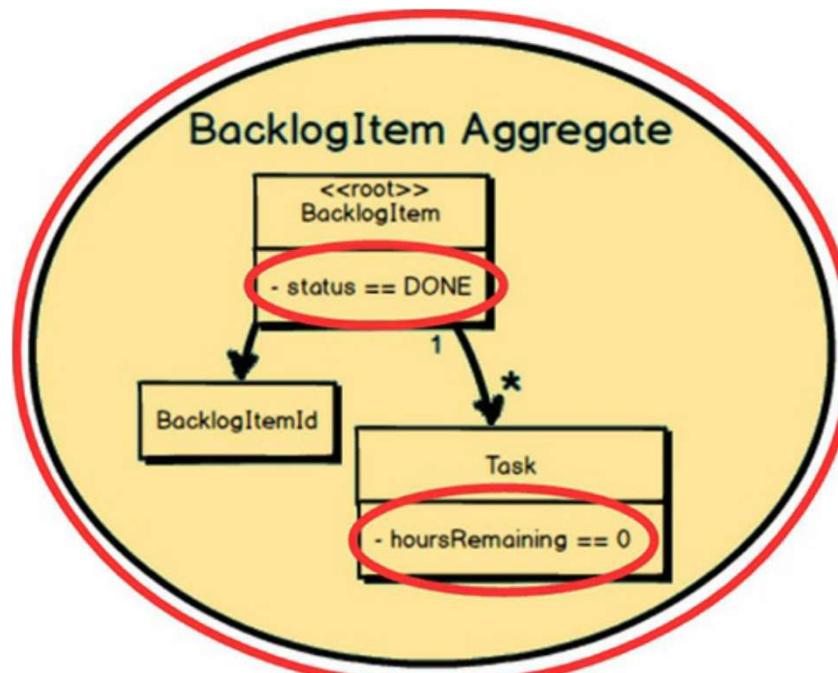
# Taktički dizajn agregatima – osnovna pravila (1)

## □ **Protect business invariants inside Aggregate boundaries**

- Invarijanta = Karakteristika nepromjenjivosti  
<https://www.enciklopedija.hr/natuknica.aspx?ID=27707>

## □ U našem slučaju konzistentnost

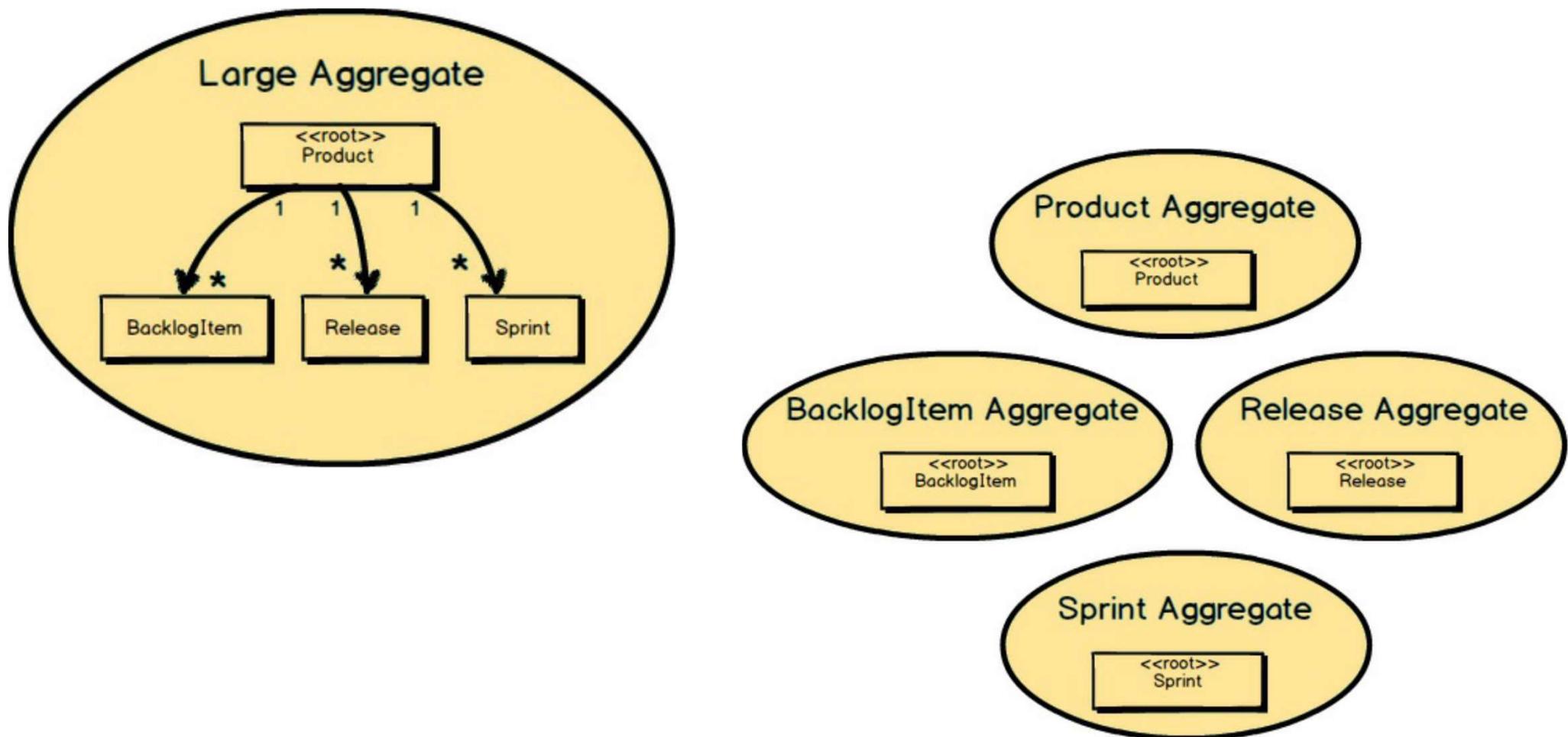
- „po završetku transakcije svi zavisni dijelovi moraju biti konzistentni u odnosu na korijen“
- Pr: kad ponestane posla na zadacima, zaostatak posla bude gotov (DONE)



# Taktički dizajn agregatima – osnovna pravila (2)

## □ *Design small Aggregates*

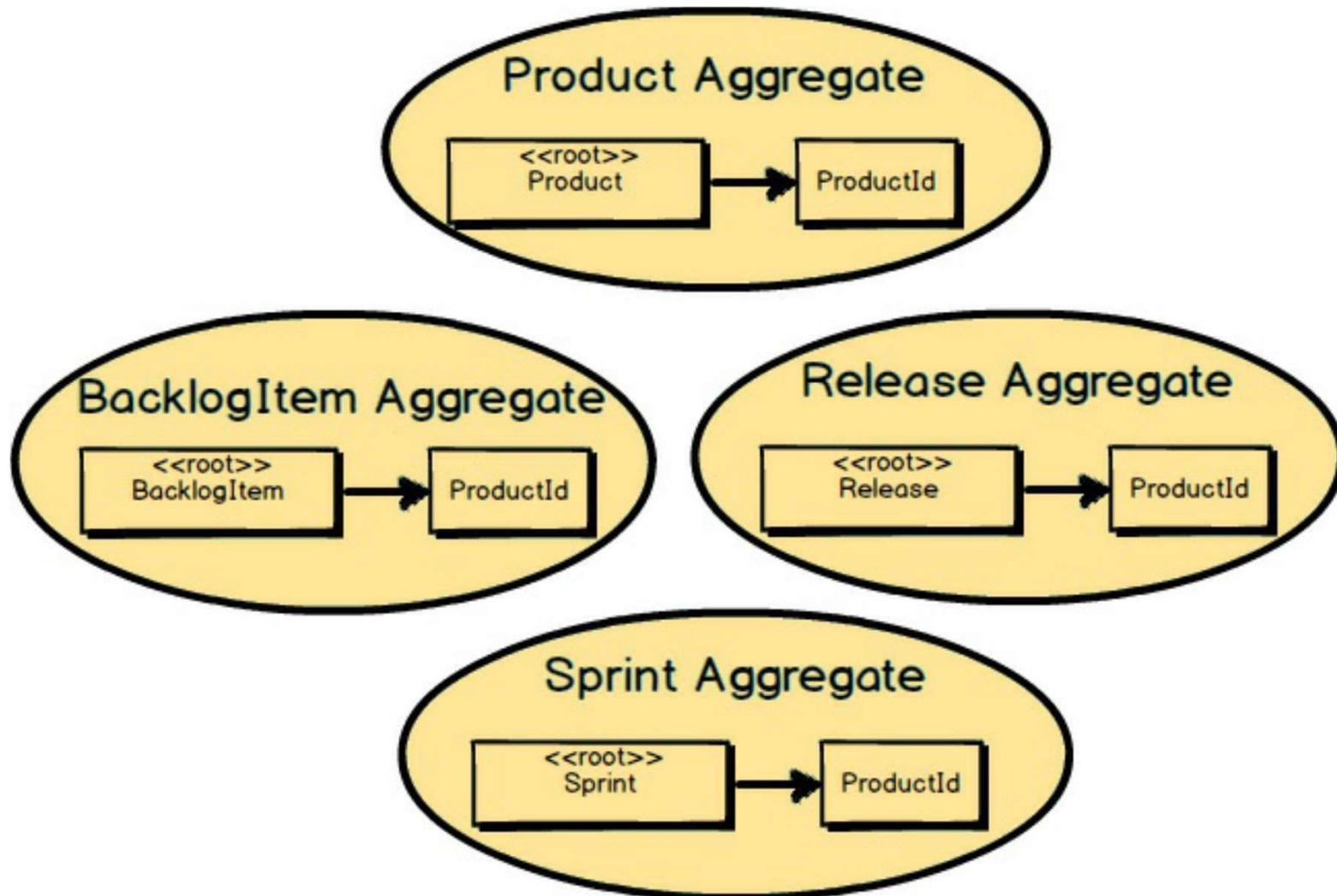
- Manje je lakše savladivo
- Načelo jednostrukih odgovornosti (pogledati Responsibility Driven Design)



# Taktički dizajn agregatima – osnovna pravila (3)

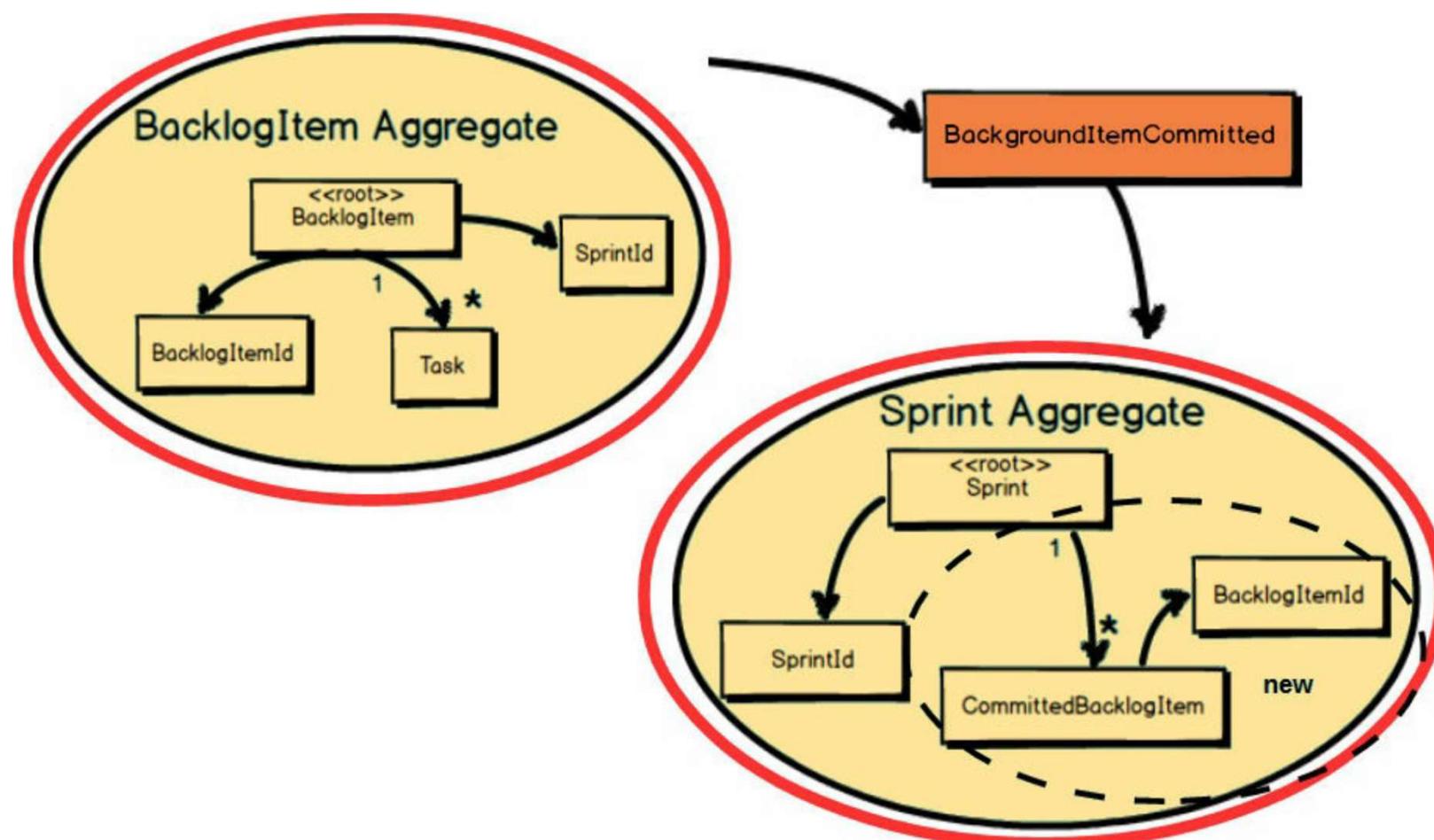
## *Reference Other Aggregates by Identity Only*

- Referencirati druge aggregate samo identitetom



# Taktički dizajn agregatima – osnovna pravila (4)

- ***Update Other Aggregates Using Eventual Consistency***
  - Konačna konzistentnost
- **Primjer: završetak preostalog posla posljedično ažurira sprint**



# Usluge (Services)

## □ Servisi mogu postojati u različitim slojevima

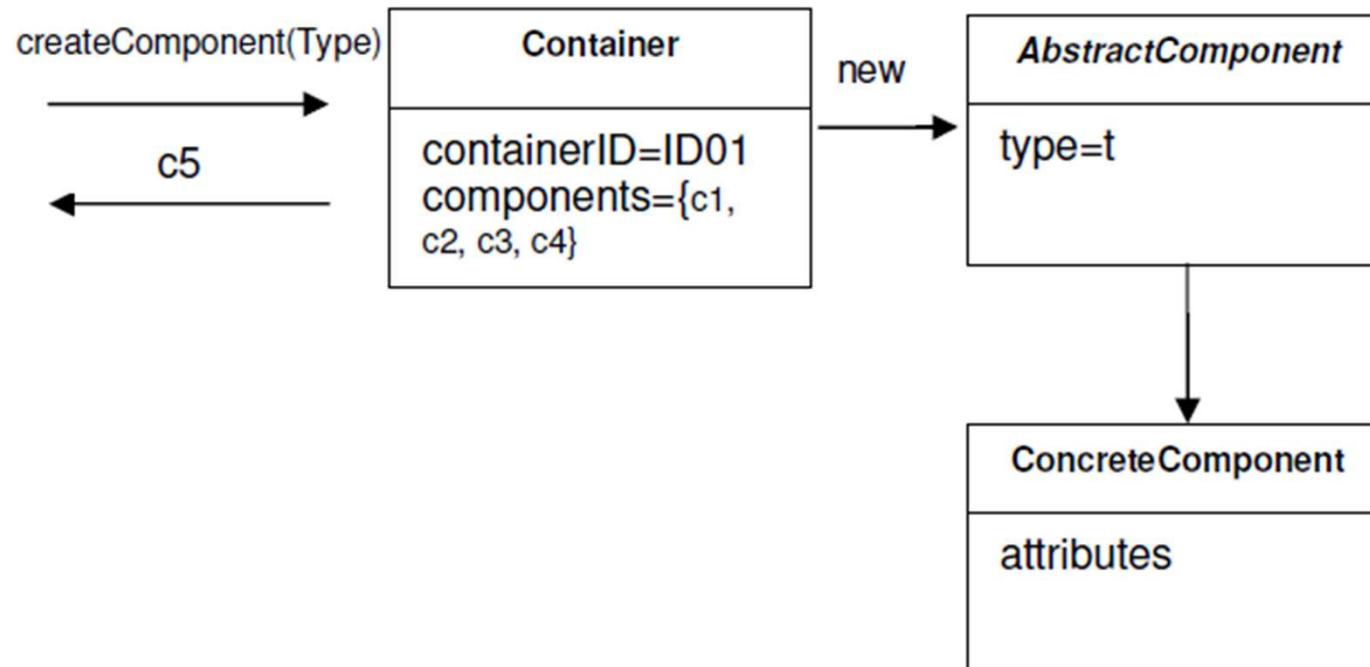
- Aplikacijska usluga
  - „bankarska aplikacija može izvesti transakcije u proračunsku tablicu“
- Domenska usluga
  - „knjiženje uplata i isplata po računu“
- Infrastrukturna usluga
  - „sučelje koje učahujuje email sustav za slanje poruka o promjeni stanja“

## □ Domenska usluga

- Poslovna logika domene koja nije dio entiteta ili vrijednosnog objekta
- Obično rukuje s više entiteta i vrijednosnih objekata
- Nema stanje! (Services are stateless!)
- Izložena kao sučelje definirano modelom
- Parametri i rezultati su domenski objekti

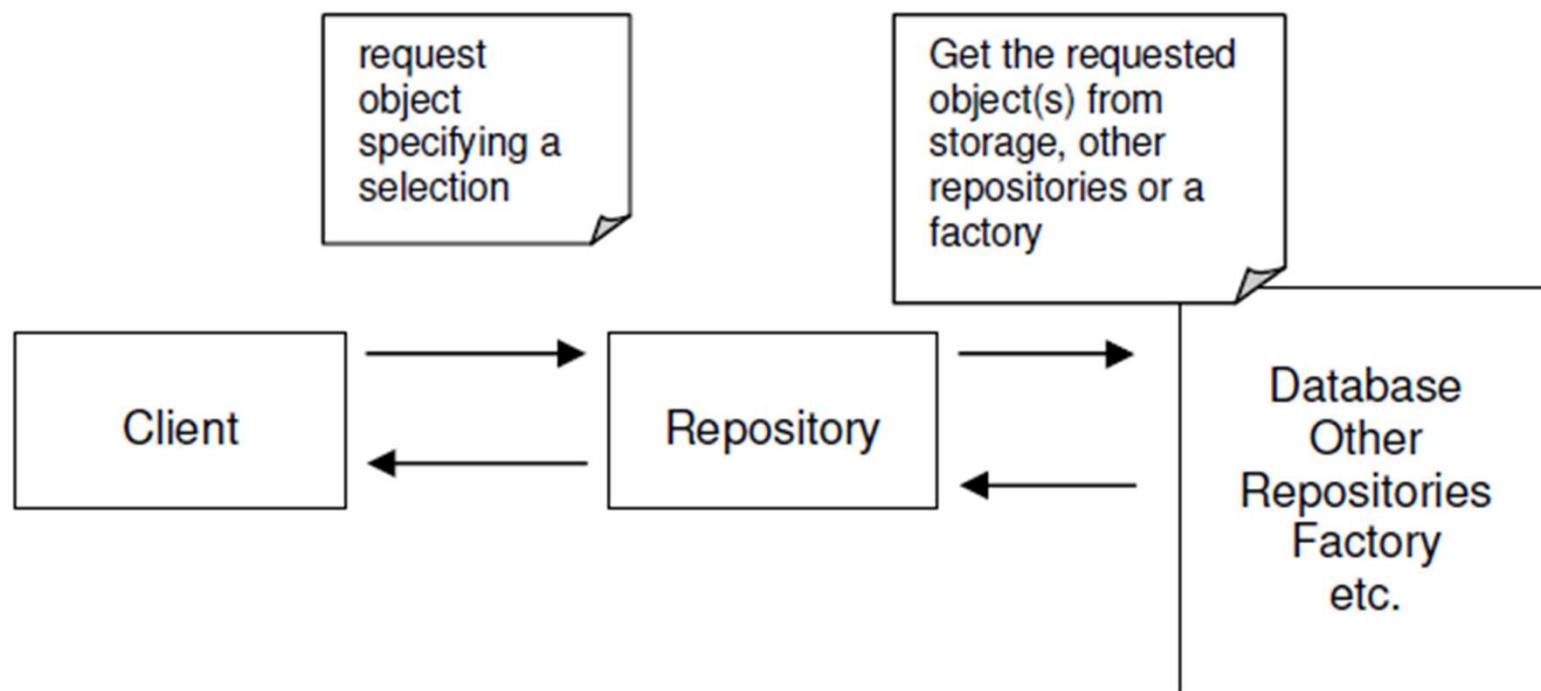
# Tvornice (Factories)

- Objekt koji kreira druge objekte
- Kreira i upravlja složenim domenskim objektima
- Posebno koristan za stvaranje agregata

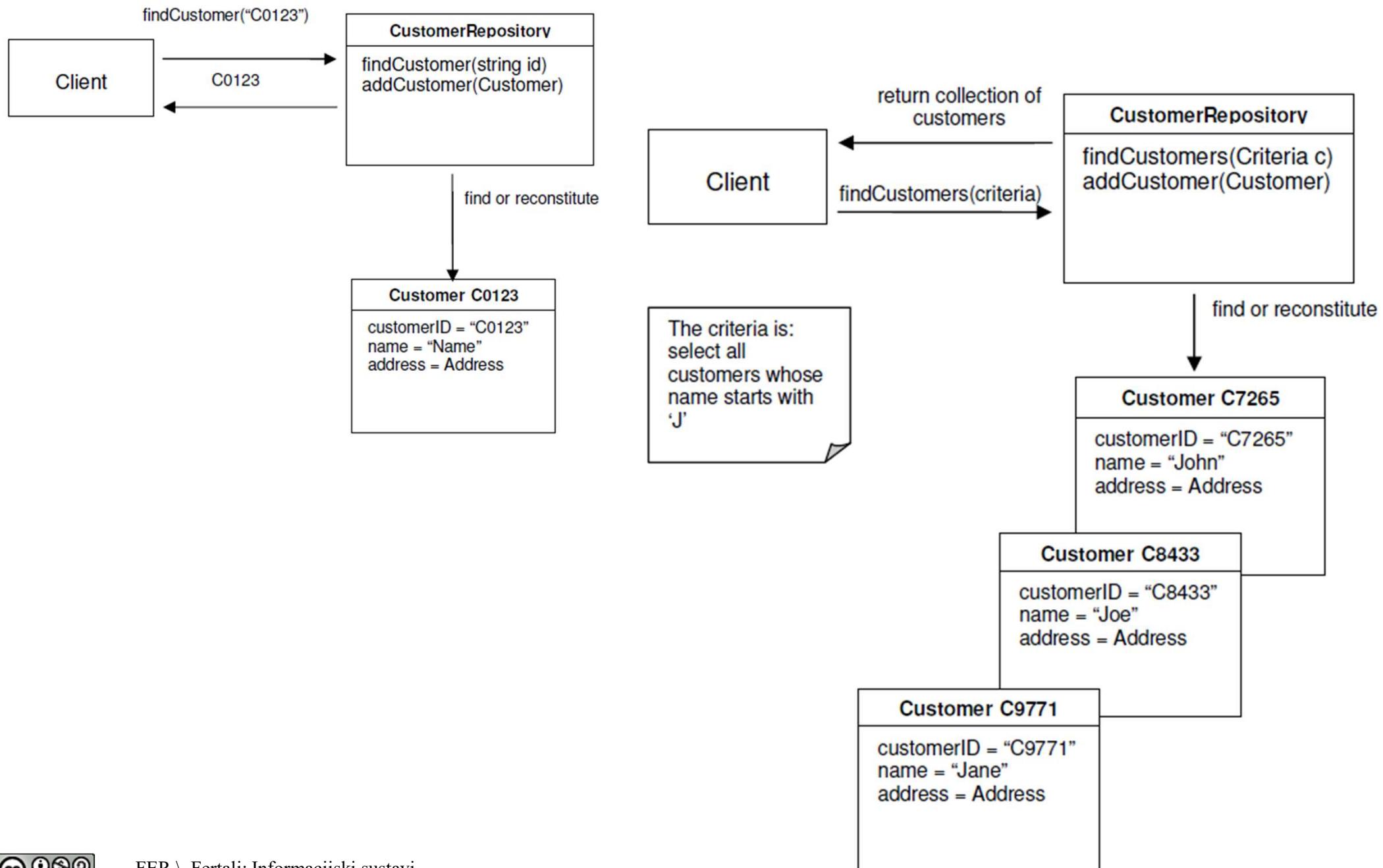


# Repozitoriji (Repositories)

- Repozitorij učahuje pohranu i dohvat domenskih objekata (persistence and retrieval)
- Čisto razdvajanje i jednosmjerna zavisnost između domene i sloja za preslikavanje podataka
- Jedan po agregatu!

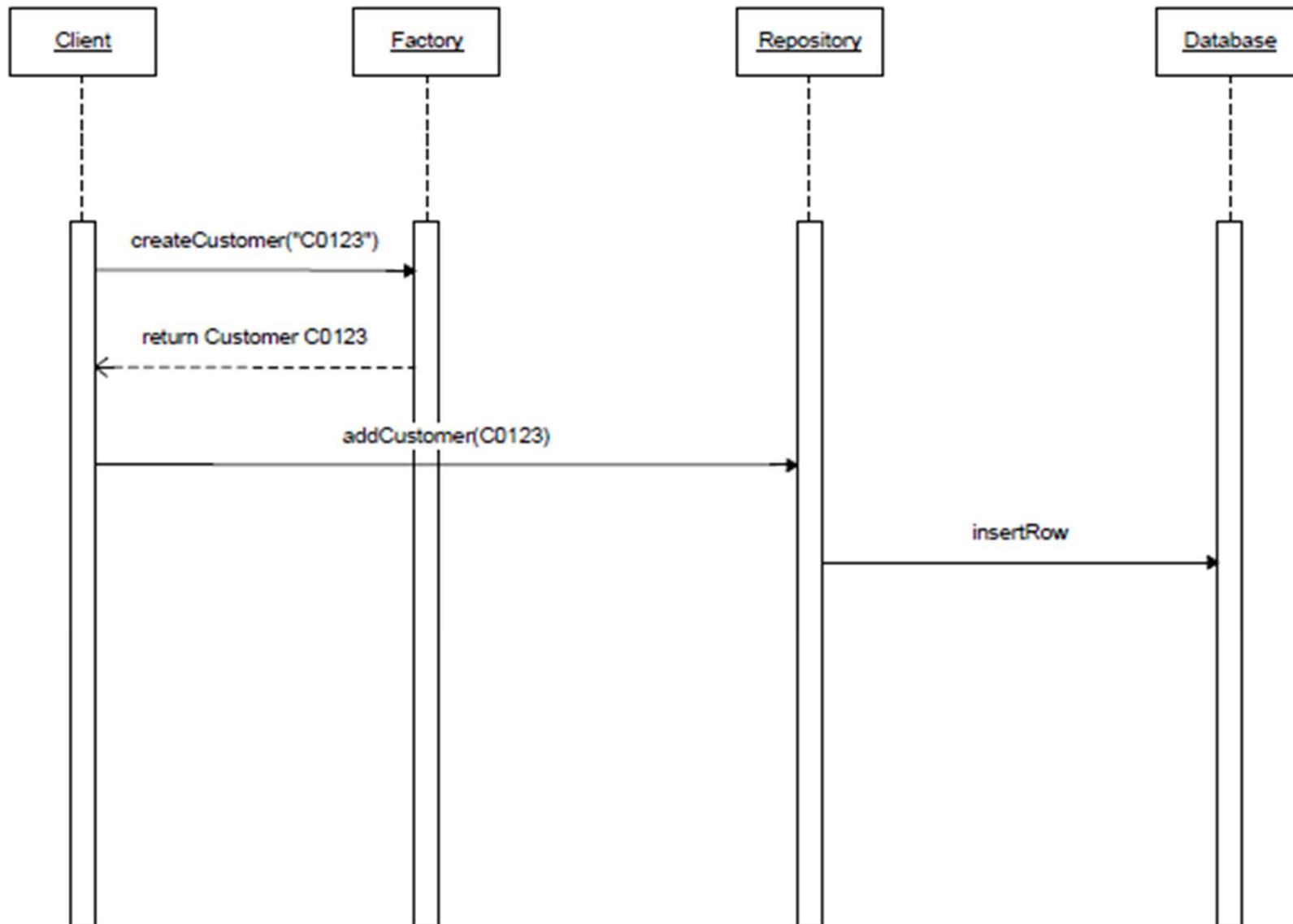


# Primjer: dohvati jednog, dohvati više



# Tvornice i rezervorije

- Zasebne uloge, jedan stvara, drugi sprema i pronalazi



## 1. blic 2015./2016.

**Kako se zove kad A treba ponašanje B? Coupling**

**Entity-relation dijagram, što je netočno?** u svakom mjestu živi točno jedna osoba, svaka organizacija ako proizvodi avion može proizvoditi točno jedan

**Usko poslovno područje, nizak trošak, neki rizik, pristojna poslovna korist? BPA**

**Za koju metodu prikupljanja informacija vrijedi da se neki razgovori moraju ponoviti zbog proturječnosti? intervju**

**Koja je uloga MIS-a?** Obrada transakcijskih podataka.

**Kolektivni naziv za ulaze, izlaze, funkcije, podatke...?** Funkcijske točke (sve ponuđeno je “[pridjev] točke”).

**Što je zahtjev na sustav (system request)?** Ne sjećam se odgovora, ali ima u 2. predavanju, 16. slajd.

**Iz čega se može raditi procjenjivanje?** Iz povijesnih podataka.

**Koja je opća tehnika temelj automatizacije poslova (BPA)?** Analiza problema (sve ponuđeno je “Analiza [nešto]”).

**Tehnika koja spada u refaktoriranje tijekom oblikovanja (tijekom kodiranja)?** Izračun/pohrana izvedenih svojstava. Također bilo ponuđeno neka pretvorba asocijacijskih razreda, izdvajanje koda u nove module, dodavanje razreda koji nedostaju.

**Kada razred A ovisi o B, pa se onda ponašanje iz A, o kojem B ovisi, izdvoji u C, onda je to...?** Normalizacija ponašanja.

**Kojim dijagramima se modeliraju poslovni zahtjevi?** Dijagramima toka.

**Koja je prednost radnih sjednica?** Rješavanje problema važnih za čitavu organizaciju ili veći dio poslovanja.

**Što znači meta?** Viša razina apstrakcije.

**Što ne spada u slojeve poslovnih sustava?** MIS (bilo je ponuđeno Management Information System i nešto s drugačjom prvom riječi, to drugo je bilo krivo)

**Što spada u varijabilne troške?** -> 2. predavanje, 38. slajd. Troškovi koji su proporcionalni poslovnim aktivnostima. Režije, putni troškovi, materijalni troškovi, troškovi održavanja.

**Što je planiranje sustava?** Odgovaranje na pitanje zašto se gradi sustav

**Bila su zadana 2 entiteta (lijevo mjesto, desno osoba) s vezom (relationship) između njih. Koje kardinalnosti moraju biti napisane na mjesta X i Y da vrijedi sljedeće: svako se mjesto sastoji od bar jednog stanovnika te svaka osoba mora živjeti u bar jednom mjestu, a može i u više njih?**

Lijevo 1, N desno 1, M

**Bila su zadana 2 entiteta (organizacijska jedinica, proizvod) s vezom između njih (proizvodi) i bile su napisane kardinalnosti 0, X lijevo i 0, Y (ili takvo nešto) desno. Bilo je pitanje što znači kada se kardinalnost na lijevoj strani promijeni u 1, X? Proizvod će se proizvoditi u bar jednoj organizacijskoj jedinici.**

**Kao prijašnje pitanje, ali što znači ako se na desnoj strani promjeni u 1, X? Svaka organizacijska jedinica proizvodi barem jedan proizvod.**

**Što je denormalizacija?** Redukcija modela podataka u nižu normalnu formu.

**Što je dizajn vođen odgovornostima?** Postupak oblikovanja softvera koji naglašava modeliranje uloga i odgovornosti objekata te suradnje objekata

**Što je projektiranje informacijskog sustava?** Slijed izrade fizičkog i logičkog modela postojećeg i budućeg IS (ali je drugačiji odgovor bio ponuđen)

**Čime se uklanja rizik nezamijećenih aktivnosti?** Zajedničkom arhitekturom

**Što je vlastiti razvoj (outsourcing)?** Najam usluge razvoja informacijskog sustava ili njegovih dijelova

**Što ne spada u elemente dijagrama dekompozicije sustava?** Ponuđeno funkcije, procesi, događaji, spojnice, vanjski spojevi → odgovor je događaji

**Što ne spada u elemente dijagrama toka podataka?** Ponuđeno tok podataka, spremište podataka, vanjski entitet, proces i događaj → odgovor je događaj

**Bio je zadan dijagram aktivnosti i bilo je pitanje što predstavlja puna strelica?** Tok podataka

**Koja je prednost promatranja poslovnog sustava?** Pouzdanost prikupljenih podataka, ostali ponuđeni su bili nedostaci promijenjeni da zvuče kao prednosti

**Što je poboljšanje poslovnih procesa?** Manje promjene radi boljeg iskorištenja tehnologije i učinkovitosti/djelotvornosti

**Tko koristi Decision Support System (DSS)?** Više poslovodstvo

**Što prikazuje dijagram konteksta?** Prikazuje sustav na najvišoj razini hijerarhije prikaza (top level diagram)

**Suradnja kod CRC kartice ( nešto takvo )?** klijent-poslužitelj-ugovor

**Što je složeni atribut?** Atribut koji je nastao grupiranjem drugih jednostavnih atributa, n-torka od jednostavnih atributa

**Što ima najveći rizik kod analize procesa/poslovanja?** Ponuđena automatizacija (BPA), poboljšanje (BPI), preoblikovanje (BPR) i još nešto što nema smisla, točno preoblikovanje

**Neka pitanja za blic iz prijašnjih godina (s materijala, bez pitanja iz primjera blica)**

**Model razvoja najsličniji objektnom** -> fazni

**Koraci vremenske izvedivosti** -> postavlja se čvrsti rok, određuju se prioriteti, gradi se jezgra sustava, odgadja se ugradnja funkcionalnosti koja ne može biti dovršena na vrijeme, u roku se isporučuje sustav, ponavljaju se koraci

**Kada se koristi prototipiranje** -> kada korisnik ne zna kako sustav treba izgledati

**Što je outsourcing** -> vanjski razvoj - najam usluge razvoja IS ili njegovih dijelova

**Izbaci uljeza od strateškog planiranja** -> strateško planiranje je izrada poslovne strategije, dugoročno planiranje resursa i akcija

**Izbaci uljeza, vrste IS** -> transaction processing system, management information system, decision support system, executive information system, expert system, office automation system, office support system, group support system, supply chain management, customer relationship management

**Čemu služi supply chain management** -> integrira razvoj proizvoda

**Koji model omogućava ponovnu procjenu rizika** -> spiralni

**Što nije element funkcijске dekompozicije** -> događaji

**Čemu služe radne sjednice** -> za prikupljanje ideja, zajedničko pronalaženje rješenja

**Što je snimak poslovnog plana?, snimka stanja** -> početno istraživanje, „je li projekt vrijedan pažnje?“

**U kojoj se fazi izvodi studija izvedivosti** -> planiranje

**Tehnika analize, mali rizik, mali trošak, poboljšava učinkovitost** -> BPA

**Tehnika analize, mali rizik, mali do umjeren trošak, bolje iskorištenje tehnologije b** -> BPI

**Tehnika analize, veliki rizik, veliki trošak, redizajn** -> BPR

**Mape dijaloga** -> prikazuju: alternativne putove, opcionalne putove, posebna stanja

**Što je verifikacija zahtjeva** -> osigurava da su izjave o zahtjevima precizne

**Što je modeliranje funkcija** -> izrada općeg modela funkcija promatranog sustava u fazi planiranja, modeliranje od globalnih funkcija...

**Analitičar iz jedne firme ide u drugu i analizira. Iz druge zatim u prvu i analiziraju. Koja je to tehnika** -> Baždarenje i usporedba (Informal benchmarking)

**Što je svojstvo robusnosti** -> stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pogrešnih podataka, defekata u komponentama ili nepredviđenim operativnim uvjetima

**Što je svojstvo prenosivosti** -> napor prijenosa iz jedne operativne okoline u drugu

**Svojstva ankete** -> svojstva su: može obuhvatiti više ispitanika, pitanja zatvorenog tipa, prikladna za popis resursa, sugestivnost odgovora

**Što modelira dijagram toka podataka** -> prikazuje protok, strukturu i obradu podataka

**Što modelira dijagram stanja** -> vremenski zavisno ponašanje čitavog sustava

**Najvažniji korak procjenjivanja** -> određivanje linija koda i funkcijskih točaka (procjena veličine projekta: funkcijeske i objektne točke)

**Model razvoja koji se jako oslanja na validaciju i verifikaciju prethodnih faza** -> V model

**Tko koristi managment information system (MIS)** -> srednje poslovodstvo

**Što je BPA** -> unaprjeđenje rada korištenjem računalne tehnologije

**Korisnik traži više nego što mu je potrebno** -> taktika dimne zavjese

**Na koje pitanje odgovor traži tehnoško-tehnička izvedivost** -> može li se napraviti? Može li se kupiti?

**Određivanje koji sustavi postoje, s kojim podacima rade, itd.** -> dio je planiranja IS-a (snimka stanja) **kužim da je snimka stanja u planiranju...ali ovo nije snimka stanja**) zar ne bi ovo trebala bit analiza... u analizi se analiziraju i vlastiti i tuđi sustavi

**Što je zadaća oblikovanja sustava** -> izgled sustava s obzirom na hardver, softver, skladišta podataka

**Koja je mjeru napora projekta** -> čovjek-mjesec

**Postupak od .. do gotovog programskog koda je** -> objektno orijentirano oblikovanje (oo razvoj?)

**Što nije dio snimke stanja** -> ponuđeno: procjena problema, procjena prilika, nešto sa direktivama, **određivanje troška**

**Što fali na sljedećoj slici** -> u ovom primjeru falilo je inženjerstvo zahtjeva



**Čime se bavi Management Information System (MIS)** -> obrada transakcijskih podataka, izrada izvješća potrebnih za upravljanje i nadzor poslovanja

**Što je meta-modeliranje** -> modeliranje podataka o podacima

**Bile su navedene karakteristike načina prikupljanja zahtjeva** -> u ovom primjeru odgovor je bio anketa

**Objektno orientiran pristup mora biti vođen** -> use-case

**Procjena analogijom** -> skaliranje poznavanjem ciklusa i provedene aktivnosti, skaliranje analizom povijesnih podataka

**Outsourcing varijanta** -> varijante su: ugovoreni razvoj isporuke gotovog projekta (**contract out**), dugoročna suradnja s isporučiteljom ili izdvajanje odjela informatike u preferiranog izvođača (**preferred contractor**)

**Aktivnosti analize sustava** -> detaljna analiza postojećeg sustava te utvrđivanje potreba i zahtjeva, detaljna specifikacija zahtjeva na IS, daljnja razrada granica projekta

**Analiza problema** -> otkrivanje problema sustava i predlaganje rješenja uz pomoć korisnika

**Korisnički zahtjevi** -> opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijama

## Dodatak – Zahtjevi na kvalitetu programske podrške:

- **DOSTUPNOST** – postotak predviđenog vremena tijekom kojeg sustav treba biti funkcionalan
- **UČINKOVITOST** – stupanj iskorištenosti resursa
- **PRILAGODLJIVOST** – lakoća dodavanja novih mogućnosti
- **INTEGRITET** - sigurnost
- **INTEROPERABILNOST** – razmjena podataka ili usluga sa drugim sustavima
- **POUZDANOST** – vjerojatnost da će softver u nekom razdoblju raditi bez greške
- **ROBUSNOST** – stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pogrešnih podataka, defekata u komponentama ili nepredviđenim operativnim uvjetima

- **UPOTREBLJIVOST** – mjera napora za pripremu, obradu ili tumačenje podataka, tj. lakoću korištenja
- **LAKOĆA ODRŽAVANJA** – mjera lakoće popravka pogreške ili izmjene
- **PRENOSIVOST** – napor prijenosa iz jedne operativne okoline u drugu
- **PONOVNA UPOTREBLJIVOST** – mjera upotrebe u drugim aplikacijama
- **PODLOŽNOST TESTIRANJU** – lakoća provjere softverske komponente
- **PLANIRANJE UNAPRIJED** – mogućnost neodabira napornih kolegija <- epic
- **LINIJA MANJEG OTPORA** – zašto nisi uzeo FMOS? <- ko da bi to pomoglo

## Primjer blica s RIS portala

1. Što je informacijski sustav?

- a. Skup međusobno povezanih komponenti koji prikuplja, obrađuje, pohranjuje i pruža informacije potrebne za obavljanje poslovne zadaće
- b. Skup međusobno disjunktnih komponenti koji prikuplja, obrađuje, pohranjuje i pruža informacije potrebne za obavljanje poslovne zadaće
- c. Uređeni poredak međusobno odvojenih komponenti namijenjenih postizanju međusobno odvojenih ciljeva
- d. Sustav namijenjen donošenju strateških odluka
- e. Sustav namijenjen izradi izvješća potrebnih za upravljanje i nadzor poslovanja

2. Na što se odnosi sustav za potporu odlučivanju (Decision Support System (DSS))?

- a. **Donošenje odluka na temelju podataka iz različitih izvora**
- b. Obradu transakcijskih podataka
- c. Evidenciju i obradu podataka o poslovnim transakcijama
- d. Donošenje strateških odluka
- e. Izradu izvješća potrebnih za upravljanje i nadzor poslovanja

3. Tko su korisnici transakcijskog informacijskog sustava (Transaction Processing System (TPS))?

- a. **Niže poslovodstvo**
- b. Srednje poslovodstvo
- c. Više poslovodstvo
- d. Sve razine poslovodstva
- e. Nitko iz poslovodstva

4. Što je karakteristično za informacijske sustave?

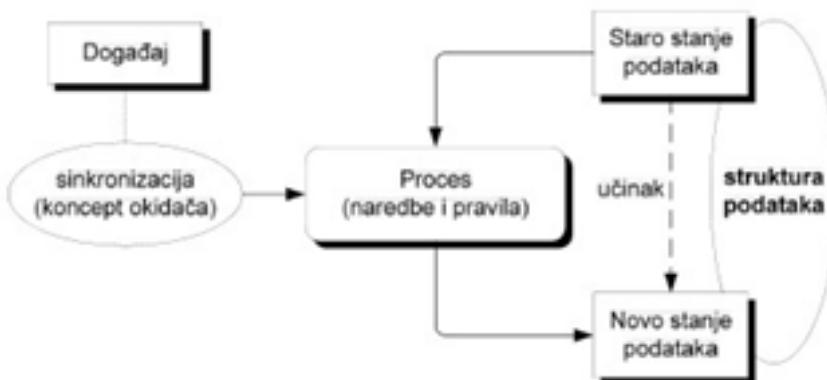
- a. **Složena okolina koju je teško u potpunosti definirati**
- b. Jednostavno sučelje prema okolini, koje uključuje različite ulaze i izlaze
- c. Jednostavne veze između ulaza i izlaza
- d. Jednostavno projektiranje
- e. Jednostavna izrada i održavanje

5. Što ne bi bilo načelo razvoja informacijskih sustava?

- a. **Korisnici i vlasnici sustava ne smiju biti uključeni u razvoj sve do njegovog kraja**
- b. Treba koristiti pristup koji vodi k rješavanju problema
- c. Uspostaviti faze i aktivnosti
- d. Uspostaviti standarde za konzistentan razvoj i dokumentiranje
- e. Ne okljevati ako treba revidirati doseg ili otkazati projekt

6. Što opisuje sljedeća slika?

- a. **Modeliranje informacijskog sustava**
- b. Projektiranje informacijskog sustava
- c. Strategiju projekta
- d. Životni ciklus razvoja informacijskog sustava
- e. Vodopadni model razvoja informacijskog sustava



7. Što su modeli objekata?

- a. **Dijagrami koji dokumentiraju strukturu objekata i njihove interakcije**
- b. Dijagrami koji dokumentiraju strukturu programske modula
- c. Dijagrami koji dokumentiraju modele funkcija i procesa
- d. Dijagrami koji dokumentiraju modele resursa
- e. Dijagrami koji dokumentiraju ključne procese

8. Što je uloga sistem analitičara?

a. Proučavanje problema i potreba poslovanja radi određivanja kako poslovni sustav i informacijska tehnologija mogu najbolje riješiti problem i postići unaprjeđenje poslovanja

- b. Analiza poslovanja, identifikacija koristi, oblikovanje (novih) procesa i procedura
- c. Upravljanje ekipom analitičara, razvojnika i drugih stručnjaka
- d. Izrada plana, raspodjela resursa i nadzor provedbe
- e. Izrada modela podataka

9. Što se radi u fazi analize životnog ciklusa?

a. Proučavaju se postojeći sustavi, ustanovljavaju moguća poboljšanja i razvija koncept novog sustava

- b. Specifikacija ili konstrukcija računalom podržanog rješenja identificiranih poslovnih zahtjeva
- c. Razmatranje učinka koji događaji imaju na procese i podatke te opis stanja
- d. Traženje odgovora na pitanje zašto graditi sustav
- e. Izrada plana rada, kadroviranje projekta, upravljanje i nadzor projekta+

10. Što je studija izvodljivosti (feasibility study)?

a. Analiza problemskog područja i određivanje (granica) projekata

- b. Određivanje poslovnih ciljeva, problema i ideja njihovog rješavanja
- c. Izrada plana rada, kadroviranje projekta, upravljanje i nadzor projekta
- d. Modeliranje cjelokupnog sustava
- e. Snimka stanja, sažetak poslovnih potreba

11. Koji je model razvoja prikazan na sljedećoj slici?

a. Vodopadni model

- b. Pseudostruktturni vodopadni model
- c. Radikalni vodopadni model
- d. Prototipski model
- e. Evolucijski model



12. Što nije vrsta prototipa?

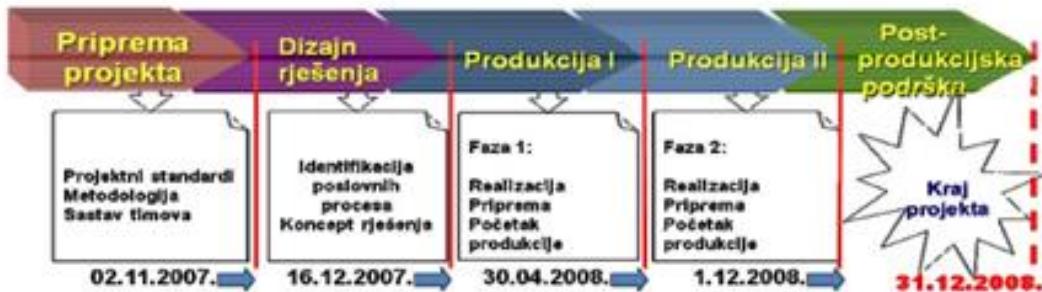
- a. **Pseudostruktturni model**
- b. Model oponašanja (mock-up)
- c. Istraživački model (research model)
- d. Ugradbeni model (implementation model)
- e. Ne postoje vrste prototipa

13. Što treba napraviti ukoliko je rizik prevelik, a razvoj se odvija primjenom spiralnog modela?

- a. **Obustaviti ili prekinuti projekt**
- b. Ponovno isplanirati sve iteracije
- c. Ponovno razmotriti rizike
- d. Preskočiti iteracije gdje je rizik najveći
- e. Ukoliko je projekt veliki prekinuti ga, ukoliko je mali, nastaviti s izradom

14. Što prikazuje sljedeća slika?

- a. **Životni ciklus ERP proizvoda**
- b. Vodopadni model razvoja
- c. "I" model razvoja
- d. Pseudostruktturni model razvoja
- e. Evolucijski model razvoja



15. Što je od navedenog najvažnije za uspješnu izgradnju informacijskog sustava?

- Aktivno sudjelovanje korisnika i vlasnika
- Sudjelovanje korisnika i vlasnika u što je moguće manjim količinama
- Detaljno dokumentiranje svih koraka procesa
- Skraćivanje koraka u rješavanju problema
- Korištenje vodopadnog modela razvoja

16. Tko procjenjuje troškove i rokove prilikom proširenja područja projekta, a kao rezultat nepotrebno prihvata odgovornost za premašaj troškova i rokova?

- Sistem analitičar
- Upravitelj projekta
- Projektant
- Vlasnik sustava
- Programer

17. Kada se može reći da je sustav zastario?

- Kada tijekom faze podrške sustavu, troškovi održavanja premaše troškove izgradnje novog sustava
- Kada se tijekom faze podrške sustavu, troškovi održavanja počnu povećavati
- Kada korisnici prestanu koristiti sustav
- Sustav nikad ne može zastarjeti
- Kada tijekom faze podrške sustavu nema troškova održavanja

18. Što je prednost vodopadnog modela razvoja?

a. Kvaliteta proizvoda jer su zahtjevi za kvalitetom proizvoda naglašeniji od zahtjeva za smanjivanjem troškova i vremena razvoja proizvoda

- b. Sustav nije upotrebljiv dok nije gotov u potpunosti
- c. Uvođenje prema gore (bottom up): moduli, podsustavi, sustav
- d. Problem predodžbe o proizvodu na temelju pisane specifikacije
- e. Korisnici prekasno uoče nedostatke

19. Što je nedostatak brzog prototipiranja?

a. Dokumentacija proizlazi iz izrade

- b. Uklanaju se moguća iznenađenja na kraju razvoja
- c. Moguće su promjene zahtjeva korisnika
- d. Povećanje kreativnosti i brzine razvoja
- e. Brzo prototipiranje nema nedostataka

20. Što je uloga poslovnog analitičara?

a. Analiza poslovanja, identifikacija koristi, oblikovanje (novih) procesa i procedura

- b. Proučavanje problema i potreba poslovanja radi određivanja kako poslovni sustav i informacijska tehnologija mogu najbolje riješiti problem i postići unaprjeđenje poslovanja
- c. Upravljanje ekipom analitičara, razvojnika i drugih stručnjaka
- d. Izrada plana, raspodjela resursa i nadzor provedbe
- e. Izrada modela podataka

**Model razvoja najsličniji objektnom** -> fazni

**Koraci vremenske izvedivosti** -> postavlja se čvrsti rok, određuju se prioriteti, gradi se jezgra sustava, odgađa se ugradnja funkcionalnosti, u roku se isporučuje sustav, ponavljaju se koraci

**Kada se koristi prototipiranje** -> kada korisnik ne zna kako sustav treba izgledati

**Što je outsourcing** -> nabava gotovog rješenja

**Izbaci uljeza od strateškog planiranja** -> strateško planiranje je izrada poslovne strategije, dugoročno planiranje resursa i akcija

**Izbaci uljeza, vrste IS** -> transaction processing system, management information system, decision support system, executive information system, expert system, office automation system, office support system, group support system, supply chain management, customer relationship management

**Čemu služi supply chain management** -> integrira razvoj proizvoda

**Koji model omogućava ponovnu procjenu rizika** -> fazni?!

**Što nije element funkcijске dekompozicije** -> događaji

**Čemu služe radne sjednice** -> za prikupljanje ideja i definiranje ..

**Što je snima poslovnog plana?, snimka stanja** -> početno istraživanje, „je li projekt vrijedan pažnje?“

**U kojoj se fazi izvodi studija izvedivosti** -> planiranje

**Tehnika analize, mali rizik, mali trošak, poboljšava učinkovitost** -> BPA

**Tehnika analize, mali rizik, mali do umjeren trošak, bolje iskorištenje tehnologije b** -> BPI

**Tehnika analize, veliki rizik, veliki trošak, redizajn** -> BPR

**Mape dijaloga** -> prikazuju: alternativne putove, opcionalne putove, posebna stanja

**Što je verifikacija zahtjeva** -> osigurava da su izjave o zahtjevima precizne

**Što je informacijski sustav** -> skup međusobno povezanih komponenti koji prikuplja, obrađuje, pohranjuje i pruža informacije potrebne za obavljanje poslovne zadaće

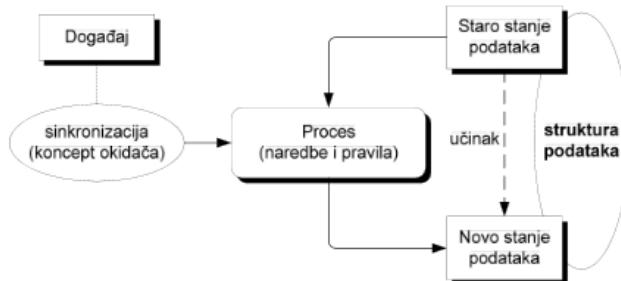
**Na što se odnosi sustav za potporu odlučivanju (DSS)** -> donošenje odluka na temelju podataka iz različitih izvora

**Tko su korisnici transakcijskog IS (TPS)** -> niže poslovodstvo

**Što je karakteristično za IS** -> složena okolina koju je teško u potpunosti definirati

**Što ne bi bilo načelo razvoja IS** -> korisnici i vlasnici ne smiju biti uključeni u razvoj sve do njegovog kraja

**Što opisuje sljedeća slika** -> modeliranje informacijskog sustava



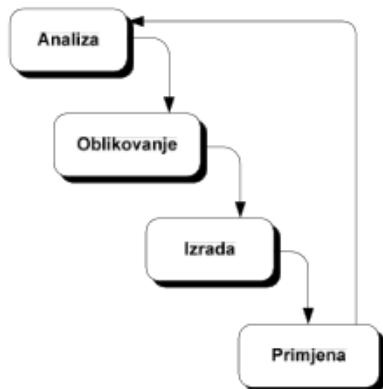
**Što su modeli objekata** -> dijagrami koji dokumentiraju strukturu objekata i njihove interakcije

**Što je uloga sistem analitičara** -> proučavanje problema i potreba radi određivanja kako poslovni sustav i informacijska tehnologija mogu najbolje riješiti problem i postići unaprjeđenje poslovanja

**Što se radi u fazi analize životnog ciklusa** -> proučavaju se postojeći sustavi, ustanovljavaju moguća poboljšanja i razvija koncept novog sustava

**Što je studija izvodljivosti** -> analiza problemskog područja i određivanje 8granica) projekta

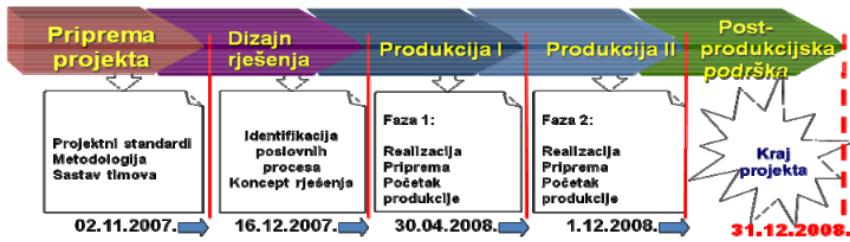
**Koji je model razvoja prikazan na sljedećoj slici** -> vodopadni model



**Što nije vrsta prototipa** -> vrste prototipa: model oponašanja, istraživački model, ugradbeni model

**Što treba napraviti ukoliko je rizik prevelik, a razvoj se odvija primjenom spiralnog modela** -> obustaviti ili prekinuti projekt

Što prikazuje sljedeća slika -> životni ciklus ERP proizvoda



Što je od navedenog najvažnije za uspješnu izgradnju IS -> Aktivno sudjelovanje korisnika i vlasnika

Tko procjenjuje troškove i rokove prilikom proširenja područja projekta, a kao rezultat nepotrebno prihvata odgovornost za premašaj troškova i rokova -> sistem analitičar

Kada se može reći da je sustav zastario -> kada tijekom faze podrške sustavu, troškovi održavanja premašće troškove izgradnje novog sustava

Što je prednost vodopadnog modela razvoja -> kvaliteta proizvoda

Što je nedostatak brzog prototipiranja -> dokumentacija proizlazi iz izrade

Što je uloga poslovnog analitičara -> analiza poslovanja, identifikacija korisnosti oblikovanje (novih) procesa i procedura

Što je modeliranje funkcija -> modeliranje od globalnih funkcija ..

Analitičar iz jedne firme ide u drugu i analizira. Iz druge zatim u prvu i analiziraju. Koja je to tehnika -> formal benchmarking

Što je svojstvo robustnosti -> stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pogrešnih podataka, defekata u komponentama ili nepredviđenim operativnim uvjetima

Što je svojstvo prenosivosti -> napor prijenosa iz jedne operativne okoline u drugu

Svojstva ankete -> svojstva su: može obuhvatiti više ispitanika, pitanja zatvorenog tipa, prikladna za popis resursa, sugestivnost odgovora

Što modelira dijagram toka podataka -> prikazuje protok, strukturu i obradu podataka

Što modelira dijagram stanja -> vremenski zavisno ponašanje čitavog sustava

Najvažniji korak procjenjivanja -> određivanje linija koda i funkcijskih točaka

Model razvoja koji se jako oslanja na validaciju i verifikaciju prethodnih faza -> V model

Tko koristi management information system (MIS) -> srednje poslovodstvo

Što je BPA -> unaprjeđenje rada korištenjem računalne tehnologije

**Korisnik traži više nego što mu je potrebno** -> problem kuhinjskog sudopera

**Na koje pitanje odgovor traži tehnološko-tehnička izvedivost** -> može li se napraviti? Može li se kupiti?

**Određivanje koji sustavi postoje, s kojim podacima rade, itd.** -> dio je planiranja IS-a

**Što je zadaća oblikovanja sustava** -> izgled sustava s obzirom na hardver, softver, skladišta podataka

**Koja je mjera napora projekta** -> čovjek-mjesec

**Postupak od .. do gotovog programskog koda je** -> objektno orijentirano oblikovanje

**Što nije dio snimke stanja** -> ponuđeno: procjena problema, procjena prilika, nešto sa direktivama, određivanje troška?

**Što fali na sljedećoj slici** -> u ovom primjeru falilo je inženjerstvo zahtjeva



**Čime se bavi Management Information System (MIS)** -> obrada transakcijskih podataka, izrada izvješća potrebnih za upravljanje i nadzor poslovanja

**Što je meta-modeliranje** -> modeliranje podataka o podacima

**Bile su navedene karakteristike načina prikupljanja zahtjeva** -> u ovom primjeru odgovor je bio anketa

**Objektno orijentiran pristup mora biti vođen** -> use-case

**Procjena analogijom** -> skaliranjem poznavanjem ciklusa i provedene aktivnosti, skaliranje analizom povijesnih podataka

**Outsourcing varijanta** -> varijante su: ugovoren razvoj isporuke gotovog projekta (**contract out**), dugoročna suradnja s isporučiteljom ili izdvajanje odjela informatike u preferiranog izvođača (**preferred contractor**)

**Aktivnosti analize sustava** -> detaljna analiza postojećeg sustava te utvrđivanje potreba i zahtjeva, detaljna specifikacija zahtjeva na IS, daljnja razrada granica projekta

**Analiza problema** -> otkrivanje problema sustava i predlaganje rješenja uz pomoć korisnika

**Korisnički zahtjevi** -> opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijama

## **DODATAK:**

### **Zahtjevi na kvalitetu programske podrške:**

- **DOSTUPNOST** – postotak predviđenog vremena tijekom kojeg sustav treba biti funkcionalan
- **UČINKOVITOST** – stupanj iskorištenosti resursa
- **PRILAGODLJIVOST** – lakoća dodavanja novih mogućnosti
- **INTEGRITET** - sigurnost
- **INTEROPERABILNOST** – razmjena podataka ili usluga sa drugim sustavima
- **POUZDANOST** – vjerojatnost da će softver u nekom razdoblju raditi bez greške
- **ROBUSNOST** – stupanj do kojeg sustav nastavlja ispravno funkcionirati u slučaju pogrešnih podataka, defekata u komponentama ili nepredviđenim operativnim uvjetima
- **UPOTREBLJIVOST** – mjera napora za pripremu, obradu ili tumačenje podataka, tj. lakoću korištenja
- **LAKOĆA ODRŽAVANJA** – mjera lakoće popravka pogreške ili izmjene
- **PRENOSIVOST** – napor prijenosa iz jedne operativne okoline u drugu
- **PONOVNA UPOTREBLJIVOST** – mjera upotrebe u drugim aplikacijama
- **PODLOŽNOST TESTIRANJU** – lakoća provjere softverske komponente

**Napomena:** ovo su samo neka od pitanja koja su se pojavila na 1. blicu 2019/20. Moguće je da postoji još puno više pitanja koja su se pojavila drugim studentima.

**Što nije vrsta informacijskog sustava? (Ponuđeno: OAS, CRM, ES, CCS, SCM) CCS**

**Na što se odnosi sustav za upravljanje odnosom s kupcima (Customer relationship management (CRM))?** Podržava marketing, prodaju i servis, uključujući interakciju s korisnicima.

**Što je to analiza sustava (systems analysis)?** Proučavanje poslovanja s ciljem preporuke poboljšanja i specificiranja zahtjeva za rješenje.

**Što je to dizajn sustava (systems design)?** Specifikacija ili konstrukcija računalom podržanog rješenja identificiranih poslovnih zahtjeva.

**Što je postavljanje zahtjeva za sustav (system request)?** Određivanje poslovnih ciljeva, problema i ideja njihovog rješavanja.

**Što je integracija i provjera sustava (system integration and evaluation)?** Udruživanje dijelova provjera cjeline, da bi se dokazalo da sustav radi (da je ispravno napravljen) te da radi ono što je zahtjevano (da je napravljen pravi sustav, koji ispunjava zahtjeve).

**Što je strateško planiranje poslovanja?** Definiranje poslovne strategije, dugoročno planiranje resursa i akcija.

**Što od navedenog spada u fiksni trošak razvoja sustava?** Trošak opreme.

**Što je varijanta vanjskog razvoja (outsourcing)?** Dugoročna suradnja isporučiteljom ili izdvajanje odjela informatike u preferiranog izvođača.

**Što ne podrazumijeva snimka stanja?** Procjena ciljeva.

**Koja je jedinica procjene veličine napora projekta?** Čovjek-mjesec.

**Što je nedostatak promatranja poslovnog sustava?** Utrošak vremena.

**Što je procjena analogijom?** Skaliranje poznavanjem ciklusa i provedenih aktivnosti.

**Što je baždarenje i usporedba procesa (informal benchmarking)?** Proučavanje poslovanja drugih organizacija i predstavljanje novih ideja koje mogu imati dodatnu vrijednost.

**Koja opća tehnika se koristi za utvrđivanje poboljšanja u automatizaciji poslovnih procesa?** Analiza problema.

**Kada je dobro imati vlastiti razvoj?** Kada postoje dodatni ili posebni razlozi kao što su povećana tajnost podataka i poslovnih procesa ili povećana zaštita informacijskog sustava.

**Što znači tehnika modeliranja u širinu kod razrade poslovnih procesa?** Svaki dijagram se detaljizira prije dekomponiranja

**Ako je dubina informacije mala, raspon veliki, a integracija informacije, sudjelovanje korisnika i trošak mali, o kojoj je tehnicu prikupljanja zahtjeva riječ?** Analiza dokumenata.

**Što nije element dijagrama toka podataka?** (*Ponuđeno: Događaj, vanjski entitet, tok podataka, proces, spremište podataka*) Događaj

**Dijagram prijelaza stanja (State Transition Diagram) najčešće opisuje:** Vremenski zavisno ponašanje čitavog sustava

**Preustroj poslovnih procesa je obično skup zbog količine utrošenog vremena i zbog količine preoblikovanja procesa. Kakav je doseg analize (breadth of analysis)?** Doseg je vrlo širok i radi s glavnim poslovnim procesima.

**“Radi se storno ulaznih dokumenata, unosom odgovarajućeg dokumenta. Dokument storna “nasljeđuje” konto dokumenta koji bude storniran. Stavka storna može se obrisati.**

**Storno... (ne znam kako je dalje tekst išao)** Funkcionalnog zahtjeva.

**Prednost meta-modeliranih aplikacija je:** Pojednostavljenje modela podataka.

**Što od navedenog nije preporuka pri izradi šifarskog sustava.** Koristiti samogovoreće šifre.

**Što je strukturirana analiza?** Tehnika modeliranja poslovnih zahtjeva na sustav.

**Što je poslovna politika?** Skup poslovnih pravila.

**Strukturirana analiza je tehnika modeliranja poslovnih zahtjeva na sustav:** usmjereni procesima, ali obuhvaća i podatke.

**Što nije element dijagrama dekompozicije funkcija?** Događaji

Što je vizija organizacije?

- a** Svrha (postojanja) organizacije, određuje postojeće stanje i procese
- b** Definiranje poslovne strategije, dugoročno planiranje resursa i akcija
- c** Određivanje prioritetnih zadataka i vremenskih okvira prioriteta
- d** Prepoznavanje problema i potreba
- e** Namjeravano, očekivano stanje u daljoj budućnosti

Meta-modeliranje je:

- a** Modeliranje surogatnog ključa
- b** Modeliranje nepoznatog
- c** Modeliranje podataka o podacima
- d** Modeliranje u metričkom sustavu
- e** Modeliranje memorijskih varijabli

Što je dijagram organizacije (organization chart)?

- a** Prikaz strukture organizacije hijerarhijom pravokutnika
- b** Definira okruženje sustava i područje analize (environmental model)
- c** Prikazuje sustav na najvišoj razini hijerarhije prikaza (top level diagram)
- d** Polazni dijagram ili dijagram konteksta (context diagram) hijerarhijski se razlaže na poddijagrame do razine osnovnih procesa
- e** Skup dijagonala za dokumentiranje fizičkog i logičkog modela sustava te zahtjeva

Što je oblikovanje sustava?

- a** Analiza problemskog područja i određivanje (granica) projekata
- b** Određivanje poslovnih ciljeva, problema i ideja njihovog rješavanja
- c** Određivanje kako će sustav funkcionirati, sa stanovišta hardvera, softvera i mrežne infrastrukture, korisničkog sučelja, programa te spremišta podataka
- d** Izrada plana rada, kadroviranje projekta, upravljanje i nadzor projekta
- e** Traženje odgovora na pitanje zašto graditi sustav

Što treba napraviti, pri izradi dijagrama toka podataka, nad procesima koji ne obavljaju pretvorbu podataka?

- a Ako je izlazni tok jednak ulaznom izbaciti jedan od tokova
- b Ako je izlazni tok nije jednak ulaznom pretvoriti dva toka u jedan
- c Ništa od navedenog
- d Ako je izlazni tok nije jednak ulaznom dodati jedan tok
- e Ako je izlazni tok jednak ulaznom preimenovati jedan od tokova ili prespojiti tokove

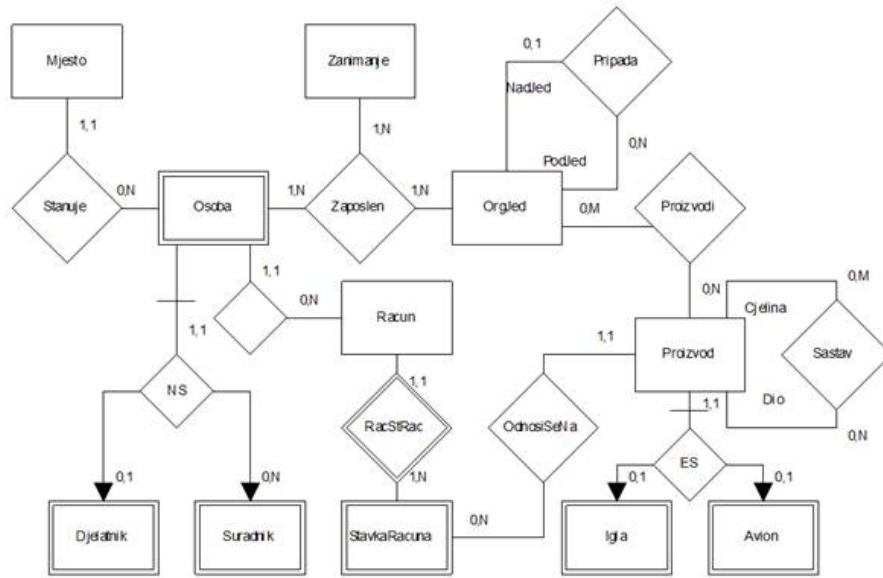
Što je snimka stanja?

- a Procjena mogućih izvođača projekta
- b Uočavanje problema, uzroka i mogućih posljedica
- c Traženje razloga za odobrenje projekta
- d Brzo istraživanje i procjena problema, mogućih prilika i direktiva
- e Analiza podataka s ciljem preoblikovanja poslovnih procesa

Preustroj poslovnih procesa je obično skup zbog količine utrošenog vremena i zbog količine preoblikovanja procesa. Kakav je doseg analize (breadth of analysis) ?

- a Doseg je vrlo širok i radi s glavnim poslovnim procesima
- b Doseg je ograničen i uvijek radi s jednim procesom
- c Nema uvjeta na doseg
- d Doseg je ograničen i obično radi s jednim procesom
- e Doseg je ograničen i radi s nekoliko procesa

Sto od navedenog **nije istina** za model na slici



- a** Neke organizacije ne proizvode niti jedan proizvod
- b** Jedan proizvod se može proizvoditi u više organizacijskih jedinica
- c** Neki proizvodi su sastavljeni od više drugih proizvoda
- d** Svaka organizacija (ako proizvodi avion) može proizvoditi maksimalno jedan avion
- e** Neki proizvodi se uopće ne proizvode

IZVEDIVOST

U kojoj se izvedivosti procjenjuju fiksni i varijabilni troškovi?

a

Tehnička izvedivost

b

Varijabilna izvedivost

c

Organizacijska izvedivost

d

Vremenska izvedivost

e

Ekonomска изведивост

Za koji je model razvoja karakteristična podjela sustava po verzijama koje se izrađuju slijedno, pri čemu svaki od sljedova vodi proizvodu koji se isporučuje i generira zahtjeve?

a

Model postupne isporuke

b

"V" model

c

Vodopadni model

d

MOOSAD pristup

e

Prototipski model

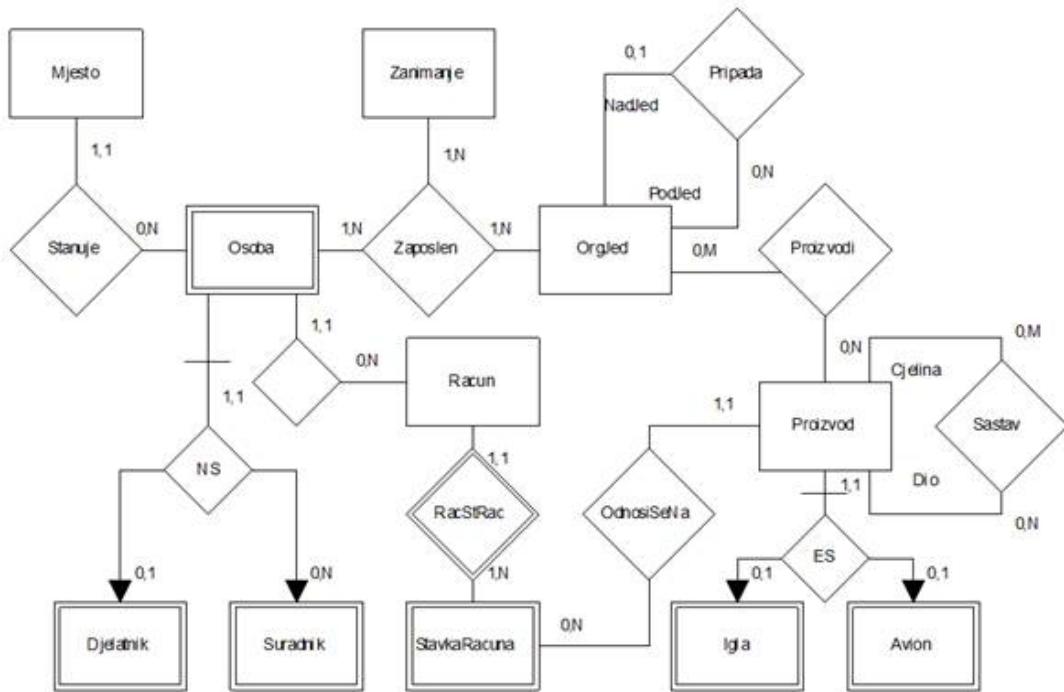
Kakvi su to vanjski događaji?

- a** Ulazni upravljački tokovi
- b** Vremenski uvjetovani
- c** Potaknuti od strane unutarnjih entiteta, koji zahtijevaju informaciju ili ažuriranje podataka
- d** Potaknuti od strane vanjskih entiteta, koji zahtijevaju informaciju ili ažuriranje podataka
- e** Posljedica prijelaza sustava iz jednog stanja u drugo

Što od navedenog spada u trošak primjene?

- a** Trošak opreme
- b** Trošak izobrazbe
- c** Sve od navedenog
- d** Trošak obnove licenci
- e** Honorari konzultanata

Što od navedenog **nije istina** za model na slici



a

Osoba ne može biti zaposlena u više organizacijskih jedinica

b

Svaki račun se sastoji od jedne ili više stavki

c

Svakoj osobi pridruženo je točno jedno mjesto stanovanja

„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.“  
Ovo su primjeri:

- a** Nefunkcionalnog zahtjeva
- b** Korisničkog zahtjeva
- c** Poslovog zahtjeva
- d** Funkcionalnog zahtjeva
- e** Nije primjer zahtjeva

Što je to analiza sustava (systems analysis)?

- a** Proučavanje poslovanja s ciljem preporuke poboljšanja i specificiranja zahtjeva na rješenje
- b** Izrada plana rada, kadroviranje projekta, upravljanje i nadzor projekta
- c** Specifikacija ili konstrukcija računalom podržanog rješenja identificiranih poslovnih zahtjeva
- d** Traženje odgovora na pitanje zašto graditi sustav
- e** Razmatranje učinka koji događaji imaju na procese i podatke te opis stanja

Što od navedenog nije osobina elementarnog procesa u logičkom procesu?

- a** Obavlja se uvijek na jednak način (za određeni ulaz daje isti izlaz)
- b** Proces je postupak, način rada ili dosljedna izmjena stanja
- c** Za obavljanje se koriste sredstva (ljudska, materijalna, finansijska)
- d** Procesom se opisuje kako sustav treba implementirati određeni zadatak.
- e** Trajanje je konačno i određivo

Koja strateška analiza u pravilu košta najmanje?

**a** automatizacija poslovnih procesa

**b** pojednostavljenje poslovnih procesa

**c** preoblikovanje poslovnih procesa

**d** poboljšanje poslovnih procesa

**e** obnova poslovnih procesa

Koliko okvirno traje prvi razgovor kod tehnike intervjuiranja?

**a** 4 sata

**b** 30 minuta

**c** 5 sati

**d** 2 sata

**e** 15 minuta

Na što se odnosi upravljački informacijski sustav (Management Information System (MIS))?

- a Donošenje strateških odluka
- b Obradu transakcijskih podataka**
- c Analizu hijerarhijski strukturiranih podataka
- d Evidenciju i obradu podataka o poslovnim transakcijama
- e Donošenje odluka na temelju podataka iz različitih izvora

Što ne obuhvaća analiza zahtjeva?

a

Modeliranje zahtjeva

b

Postavljanje prioriteta

c

Analizu ostvarivosti zahtjeva

d

Izradu izvješća

e

Izradu dijagrama konteksta sustava

Pri modeliranju CRC kartice objekti u suradnji su:

a ugovor – funkcija - objekt

b klijent – ugovor – razvojnik

c subjekt – predikat - objekt

d klijent – analitičar – razvojnik

e klijent – poslužitelj - ugovor

Što od navedenog **nije** istina za aggregate u modelom vođenom razvoju?

- a** Svaki agregat ima jedan korijenski entitet (root).
- b** Identitet korijena je lokalni, identiteti unutarnjih entiteta globalni.
- c** Agregati su grupe entiteta i vrijednosti, tretirani kao cjelina.
- d** Svaki agregat treba biti izmijenjen i pohranjen u zasebnoj transakciji.
- e** Vanjski objekti mogu referencirati samo korijen.

U primjeni su različite vrste matričnog prikaza modela događaja, ovisno o metodologiji. Što je primjer?

- a** Matrica entiteti/događaji (Entity/Event Matrix), npr. SSADM
- b** Matrica entiteti / entiteti (Entity / Entity Matrix), npr. SSDEE
- c** Matrica događaji /događaji (Event /Event Matrix), npr. SSAAA
- d** Matrica relacije / relacije (Relation / Relation Matrix), npr. SSDRR
- e** Matrica relacije / entiteti (Relation / Entity Matrix), npr. SSDER

Za koju je tehniku prikupljanja informacija karakteristično da se informacije prikupljaju nizom pojedinačnih razgovora koji bi se trebali voditi prema unaprijed dogovorenom planu i rasporedu?

- a Anketiranje
- b Promatranje
- c Proučavanje dokumenata
- d Intervjuiranje
- e Združeni razvoj

Cilj koje strateške analize je napraviti male ili umjerene promjene poslovnih procesa kako bi ih se učinilo učinkovitijim i djelotvornijim?

- a Poboljšanje poslovnih procesa
- b Preoblikovanje poslovnih procesa
- c Vrednovanje poslovnih procesa
- d Automatizacija poslovnih procesa
- e Sistematizacija poslovnih procesa

Što je snimka stanja?

- a Traženje razloga za odobrenje projekta
- b Početno istraživanje i procjena problema, mogućih prilika i direktiva
- c Analiza podataka s ciljem preoblikovanja poslovnih procesa
- d Uočavanje problema, uzroka i mogućih posljedica
- e Procjena mogućih izvođača projekta

Na što se odnosi sustav za upravljanje odnosom s kupcima (Customer relationship management (CRM))?

**a** Podržava marketing, prodaju i servis, uključujući interakciju s korisnicima

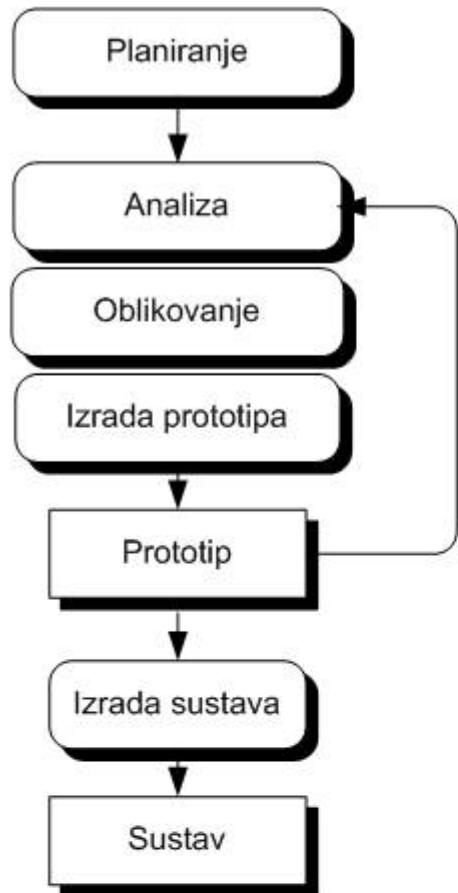
**b** Sustav pravila i mehanizme zaključivanja

**c** Potporu uredskom poslovanju

**d** Potporu grupnom radu

**e** Integrira razvoj proizvoda, nabavu proizvoda, proizvodnju i upravljanje zalihamama

Koji je model razvoja prikazan na sljedećoj slici?



**a** Prototipski razvoj – brzo prototipiranje

**b** Evolucijski model

**c** MOOSAD pristup

**d** Paralelni razvoj po potprojektima

**e** Prototipski model – ograničeno prototipiranje

Logičkim procesima se opisuju:

a Tehnologije

b Ništa od navedenog

c Funkcije, događaji i elementarni procesi

d Razgovori sudionika na projektu

e Utrošak vremena

Što je dijagram konteksta (context diagram)?

a Dijagram koji prikazuje podatke ovisno o kontekstu

b Definira okruženje sustava i područje analize (environmental model)

c Prikaz strukture organizacije hijerarhijom pravokutnika

d Skup dijagrama za dokumentiranje fizičkog i logičkog modela sustava te zahtjeva

e Dijagram koji prikazuje više procesa i jedan vanjski entitet

U kojoj se izvedivosti procjenjuju fiksni i varijabilni troškovi?

**a** Organizacijska izvedivost

**b** Vremenska izvedivost

**c** Ekonomска izvedivost

**d** Tehnička izvedivost

**e** Varijabilna izvedivost

Traženje odgovora na pitanje "Postoje li aplikacije čiji je razvoj u tijeku? U kojem su stadiju razvojni projekti?" je dio:

**a** Identificiranja projekta

**b** Određivanja dosega projekta

**c** Planiranja informacijskog sustava

**d** Rješavanja postavljenih problema

**e** Izrade početnog plana projekta

Što je planiranje sustava?

a Traženje odgovora na pitanje kada će sustav biti korišten

b Traženje odgovora na pitanje gdje će sustav biti korišten

c Traženje odgovora na pitanje što će sustav raditi

d Traženje odgovora na pitanje zašto graditi sustav

e Traženje odgovora na pitanje tko će koristiti sustav

Što je svrha, cilj i dubina analize sustava?

a Potvrđivanje da nije potrebno poboljšanje poslovnih procesa

b Utvrđivanje da nije potrebno preoblikovanje poslovnih procesa

c Utvrđivanje troška projekta

d Analiziranje učestalosti mijenjanja poslovnih potreba korisnika

e Automatizacija, poboljšanje i preoblikovanje poslovnih procesa

Tri vrste modela u modeliranju podataka su

**a** konceptualni, logički, fizički

**b** model konteksta podataka, logički, potpuno opisani model

**c** konceptualni, model konteksta podataka, fizički model

**d** globalni, logički, potpuno opisani model

**e** logički, fizički, potpuno opisani model

Što ne obuhvaća analiza zahtjeva?

a Analizu ostvarivosti zahtjeva

b Modeliranje zahtjeva

c Postavljanje prioriteta

d Izradu izvješća

e Izradu dijagrama konteksta sustava

Kako su definirani zahtjevi po IEEE standardu?

a IEEE standard ne definira zahtjeve

b Zahtjevi su definirani od strane voditelja projekata

c Detalji koji se odnose na planiranje projekta

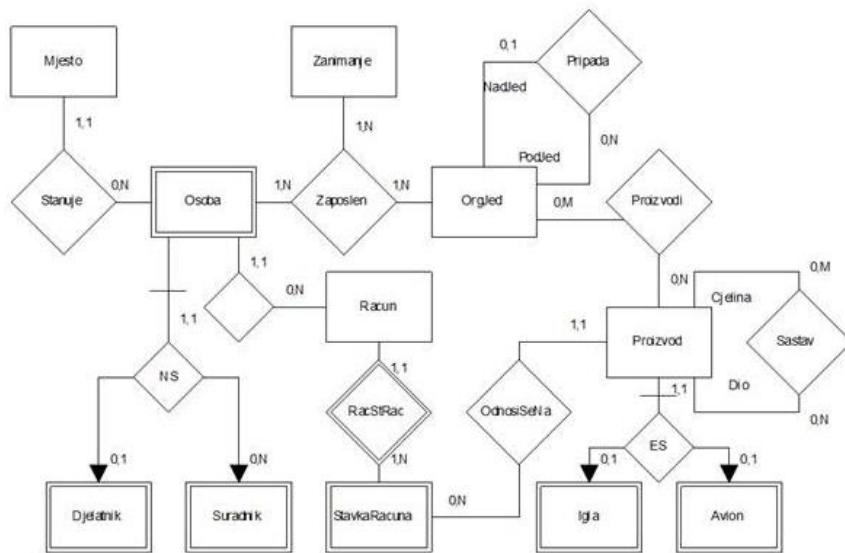
d Uvjet ili sposobnost koje korisnik treba da bi riješio problem ili ostvario cilj

e Detalji koji se odnose na dizajn, implementaciju

Što je varijanta vanjskog razvoja (outsourcing)?

- a** Razvoj vlastitim informatičkim snagama
- b** Osposobljavanje netehničkog osoblja
- c** Povremeni razvoj unajmljenim osobljem
- d** Dugoročni razvoj unajmljenim osobljem
- e** Ugovoren razvoj isporuke gotovog proizvoda (contract out)

Što od navedenog **nije istina** za model na slici



a Osoba može biti djelatnik i suradnik

b Svaki račun se sastoji od jedne ili više stavki

c Proizvod može biti igla ili avion

d Osoba ne može biti zaposlena u više organizacijskih jedinica

e Svakoj osobi pridruženo je točno jedno mjesto stanovanja

Jedan od problema pri određivanju zahtjeva je kada korisnik traži više mogućnosti, a zapravo mu je potrebna samo jedna ili dvije. Ovaj se problem naziva:

a Taktika "Treba mi isto, ali u boljem obliku"

b Taktika zelenih košulja

c Taktika dimne zavjese

d Taktika "Treba mi sve"

e Taktika kuhinjskog sudopera