

# Osnove informacijskih sistemov

## Univerzitetni študij

Smer: programska oprema, logika in sistemi



UNIVERZA V LJUBLJANI

Fakulteta za računalništvo in informatiko

Doc. dr. Marko Bajec

Študijsko gradivo, verzija 1.0

## Splošne informacije...



### ▪ Predavatelj

- Doc. dr. Marko Bajec, univ. dipl. inž.

 Elektronska pošta: [marko.bajec@fri.uni-lj.si](mailto:marko.bajec@fri.uni-lj.si)

 Informacije: <http://infolab.fri.uni-lj.si/marko/>

### ▪ Asistenta

- As. dr. Damjan Vavpotič, univ. dipl. inž.

- As. mag. Alenka Rožanec, univ. dipl. inž.

 Elektronska pošta: [damjan.vavpotic@fri.uni-lj.si](mailto:damjan.vavpotic@fri.uni-lj.si)

 Informacije: <http://aris.fri.uni-lj.si/~damjan/>



# Splošne informacije

- **Predavanja**
  - Vsak četrtek od 11:15 do 13:00 (2 šolski uri);
- **Izpit**
  - Seminarska naloga (predpogoj)
  - Pisni izpit
  - Ustni izpit



# Vsebina predmeta

- I. Osnovni pojmi
  - Opredelitev IS
  - Zgodovina in razvoj IS
  - Vloga IS v poslovnem sistemu <sup>I4</sup>
  - Vrste IS
  - Upravljanje z IT – priložnosti in izzivi
  - Poslovno okolje
  - Opredelitev sistema
  - IS kot podsistem
  - Konceptualni model IS
  - Podatek in informacija



I4

- \* Kaj je metodologija
  - \* Vrste metodologij (objektne, strukturne,....)
  - \* Teža metodologije (lahke, agilne)
- Infolab; 12.2.2006

## Vsebina predmeta

- II. Poslovne aplikacije
  - E-poslovanje
  - Več-funkcijski poslovni sistem
  - Poslovno informacijska arhitektura
  - Funkcionalni informacijski podsistemi
    - Finančni podsistem,
    - Računovodski podsistem,
    - Prodajni podsistem,
    - Proizvodni podsistem in
    - Kadrovski podsistem
  - CRM
  - ERP



- I2
- \* Kaj je metodologija
  - \* Vrste metodologij (objektne, strukturne,....)
  - \* Teža metodologije (lahke, agilne)
- Infolab; 12.2.2006

## Vsebina predmeta

- III. Osnove informacijskih sistemov
  - Osnovni pojmi
  - Življenjski modeli razvoja IS
  - Metodologije razvoja IS<sup>I5</sup>
- IV. Strukturni razvoj IS
  - Osnovne značilnosti strukturnega pristopa
  - Strateško načrtovanje
  - Analiza
  - Načrtovanje
  - Izvedba
  - Testiranje
  - Uvajanje



- \* Kaj je metodologija
  - \* Vrste metodologij (objektne, strukturne,....)
  - \* Teža metodologije (lahke, agilne)
- Infolab; 12.2.2006

## Vsebina predmeta...

- V. Objektni razvoj IS
  - Osnovni principi objektne usmerjenosti
  - Osnove modelirnega jezika UML
  - Objektna analiza in načrtovanje
- VI. Načrtovanje podatkovnih baz
  - Tri-nivojsko načrtovanje
  - Konceptualno načrtovanje
  - Osnove relacijskega modela in logično načrtovanje
  - Normalizacija



- \* Kaj je metodologija
  - \* Vrste metodologij (objektne, strukturne,....)
  - \* Teža metodologije (lahke, agilne)
- Infolab; 12.2.2006

## Priporočena literatura...



### ▪ Literatura:

- [1] Martin Fowler (2003). **UML Distilled: A Brief Guide to the Standard Object Modeling Language**, Third Edition. Addison-Wesley.
- [2] Thomas A. Pender (2002). **UML Weekend Crash Course**. Wiley Publishing.
- [3] Kent Beck (1999). **Extreme Programming Explained: Embrace Change**, Addison-Wesley.
- [4] Martin, R. C., **Agile Software Development, Principles, Patterns, and Practices** (2003), Prentice Hall

Citiranje: glej [1, 15-20] = glej v knjigi M. Fowler, strani od 15 do 20.



# Priporočena literatura...

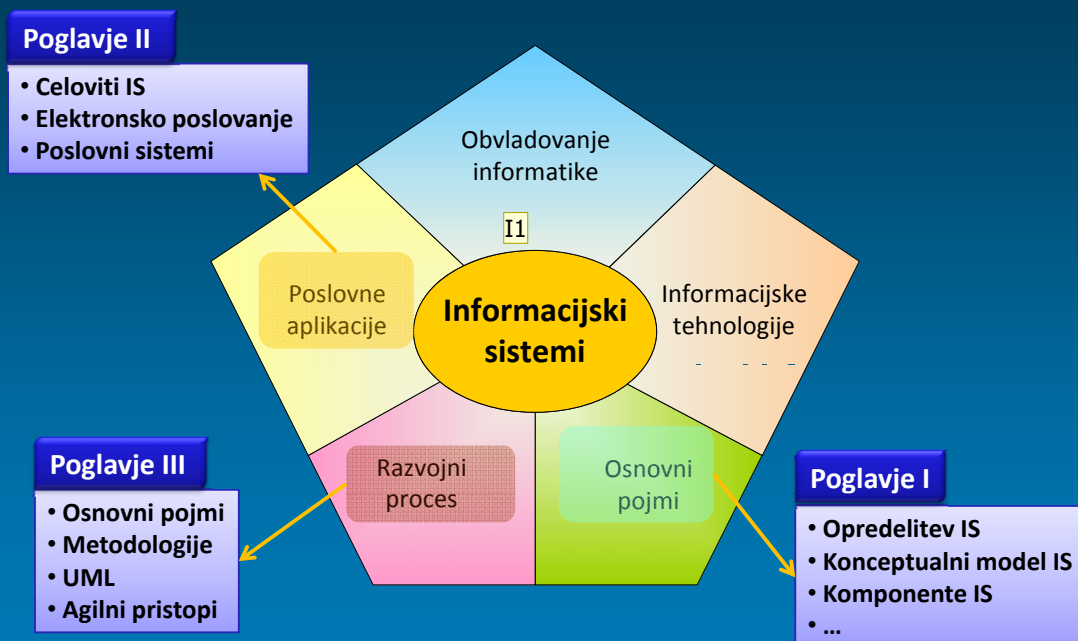


## Literatura:

- [5] Hoffer, J. A., George, J. F. in Valacich, J. S. (1999). **Modern Systems Analysis and Design**, Second edition, Addison-Wesley
- [6] O'Brien J. in Marakas, G. M. (2006). **Management Information Systems**, Seventh edition, McGraw-Hill

Citiranje: glej [1, 15-20] = glej v knjigi M. Fowler, strani od 15 do 20.

# Vsebina predmeta...



- I1
- \* Kaj je metodologija
  - \* Vrste metodologij (objektne, strukturne,....)
  - \* Teža metodologije (lahke, agilne)
- Infolab; 12.2.2006

## Poglavje I

# Osnovni pojmi

- Opredelitev IS
- Zgodovina in razvoj IS
- Vloga IS v poslovnem sistemu
- Vrste IS
- Upravljanje z IT - priložnosti in izzivi
- Poslovno okolje
- Opredelitev sistema
- IS kot podsistem
- Konceptualni model IS
- ...

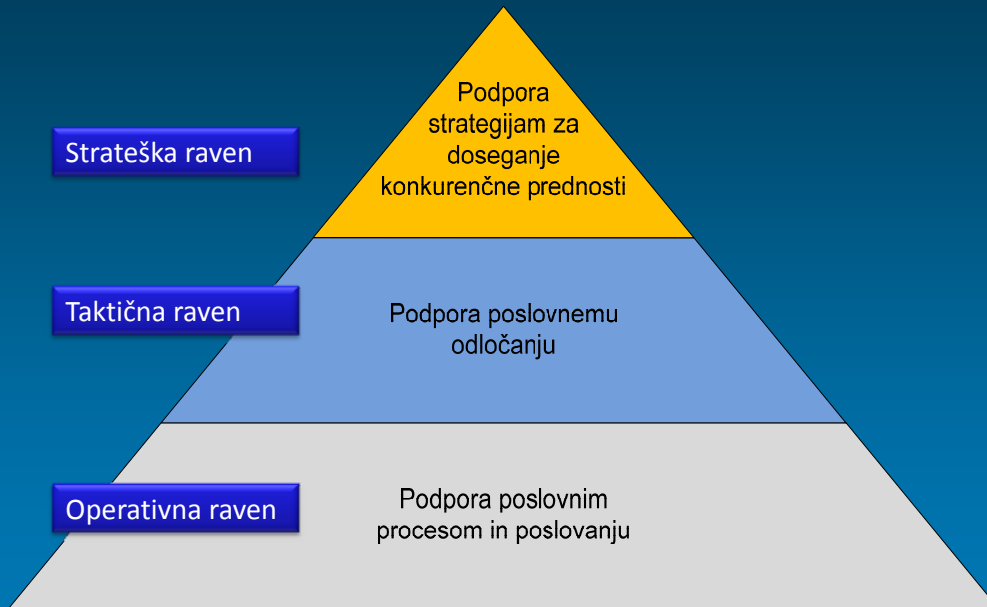




# Informacijski sistem...

- **Definicija:**
  - Informacijski sistem lahko opredelimo kot množico medsebojno odvisnih komponent (strojna oprema, programska oprema, ljudje), ki zbirajo, procesirajo, hranijo in porazdeljujejo podatke in s tem podpirajo delavne procese v organizaciji [5].
- Ločimo **formalne in neformalne IS**
- IS je lahko **računalniško podprt**

# Vloga IS v poslovnem sistemu



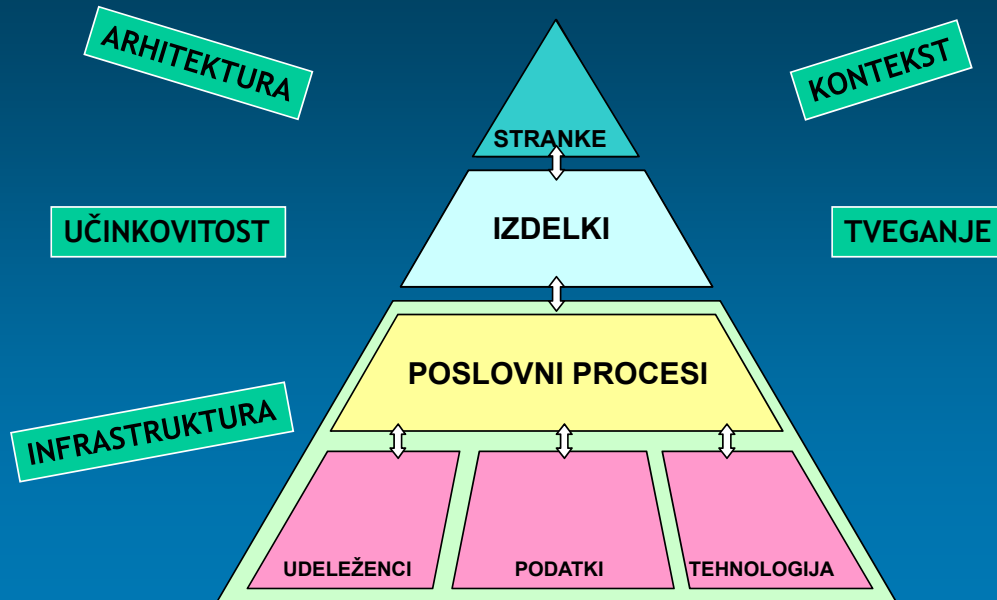
# Lastnosti dobrega IS

## ▪ Lastnosti "dobrega" informacijskega sistema:

- Zagotavlja podatke, iz katerih lahko zaposleni na različnih ravneh v združbi pridobivajo informacije, ki jih potrebujejo pri svojem delu.
- Daje podlago tako za reševanje vsakodnevnih vprašanj kot tudi za izvajanje upravljaljskih ukrepov ter sprejemanje strateških odločitev.

- Je usklajen s poslovnim sistemom!

# Analiza poslovnega sistema



# Strateško planiranje informatike

- **Strateško planiranje informatike** je proces izoblikovanja informacijskega sistema, ki organizaciji omogoča uresničitev njenih ciljev in ji s tem posredno zagotavlja konkurenčno prednost.

Fidler in Rogerson, 1996

## Primer IS v izobraževalnem podjetju...

- Podjetje se ukvarja z izvajanjem računalniških tečajev. IS v podjetju daje podlago za reševanje vprašanj, kot so:
- Vsakodnevna vprašanja:
  - Je Janez Novak prijavljen na tečaj Windows XP, ki se prične naslednji teden?
  - Je podjetje MIX d.o.o. plačalo račun za svojih sedem udeležencev tečaja iz prejšnjega tedna?
  - Kdo so udeleženci tečaja Visual Studio, ki se prične jutri?

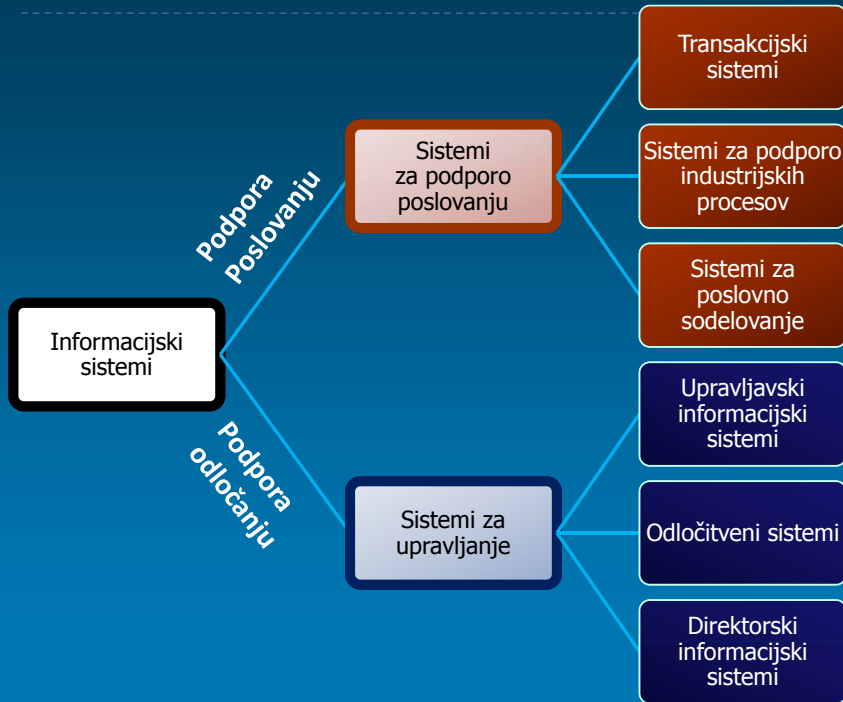


## Primer IS v izobraževalnem podjetju

- Upravljalvska vprašanja:
  - Je prijavljenih za tečaj JBuilder dovolj, da je izvedba tečaja upravičena?
  - Kakšen je bil dobiček s tečajem, ki je bil izveden v Mariboru?
  - Kateri tečaji so bili v zadnjem letu najbolj donosni?
- Strateška vprašanja:
  - Bi bilo smiselno dvigniti cene tečajev?
  - Je smiselno pripravljati nadaljevalne tečaje?
  - Informatika je v krizi. Je smiselno razmišljati o dodatni dejavnosti?



## Vrste IS



## Druge kategorije IS...

- **Ekspertni sistemi**
  - Sistemi, ki temeljijo na bazi znanja in so namenjen reševanju specifičnih problemov iz realnega sveta
- **Sistemi za upravljanje z znanjem**
  - Sistemi, ki temeljijo na bazi znanja in omogočajo izdelavo, urejanje in širjenje poslovnega znanja znotraj združbe

## Druge kategorije IS

- **Strateški informacijski sistemi**
  - Podpirajo strateško upravljanje poslovnih sistemov za doseganje konkurenčne prednosti (ključni izdelki, storitve in poslovni procesi)
- **Funkcionalni informacijski podsistemi**
  - Vključuje aplikativne sisteme, ki podpirajo osnovne poslovne funkcije (funkcionalna področja)



## Upravljanje z IT – izzivi in priložnosti...

### Poslovni sistem

Strategije / Procesi / Struktura / Kultura



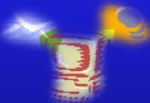
#### Poslovni / IT izzivi

- Hitrost in prilagodljivost razvojnih, proizvodnih in dostavnih ciklov
- Prenova in med-funkcijsko povezovanje poslovnih procesov z uporabo internetnih tehnologij
- Vpeljava e-poslovanja in e-trgovanja v strategijo, procese, strukturo in kulturo podjetja



## Upravljanje z IT – izzivi in priložnosti...

### Informacijska tehnologija



#### Poslovni / IT razvoj

- Uporaba interneta, intraneta, ektraneta in spleta, kot primarno IT infrastrukturo
- Širitev uporabe spletnih tehnologij na zaposlene, stranke in dobavitelje
- Omrežno računalništvo na globalnem nivoju, sodelovanje in sistemi za podporo odločanju



## Upravljanje z IT – izzivi in priložnosti

### Poslovna vrednost

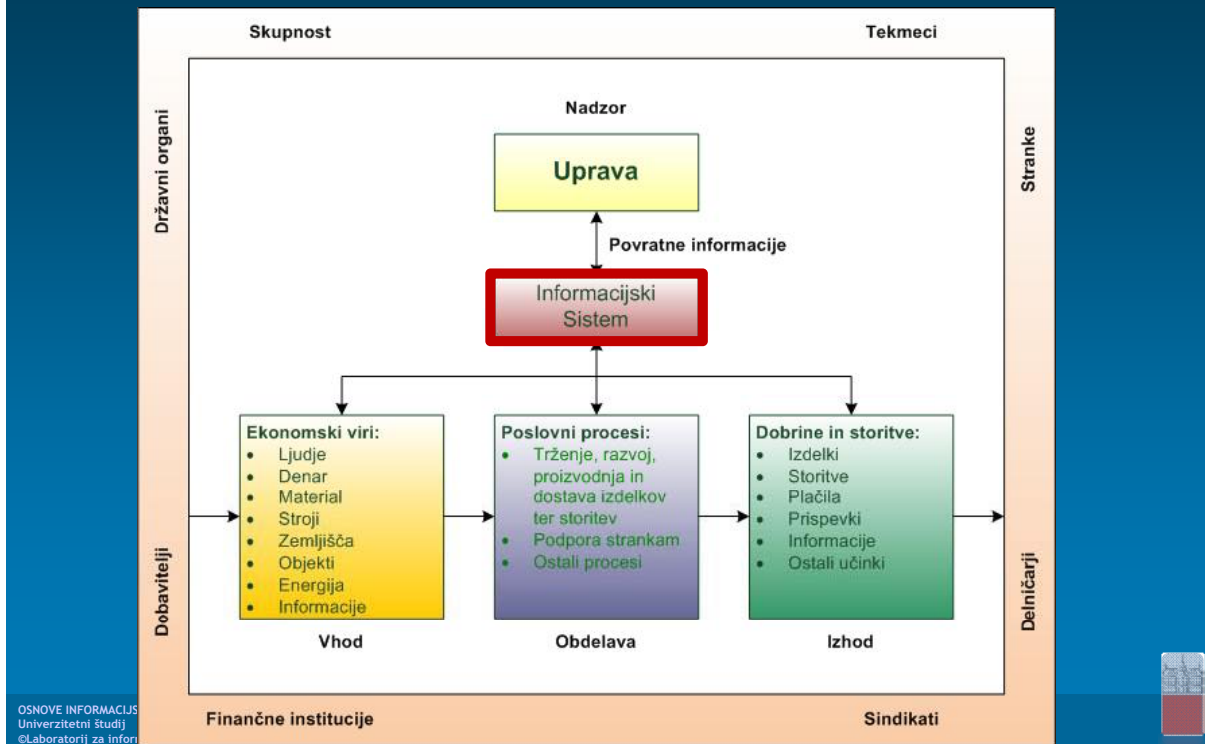


#### Poslovni / IT cilji

- Strankam ponuditi pravo stvar ob pravem času za najnižjo ceno
- Uskladitev proizvodnega in poslovnega procesa z dobavitelji in kupci
- Povezovanje z dobavitelji in distributerji na področju prodaje



# Poslovno okolje

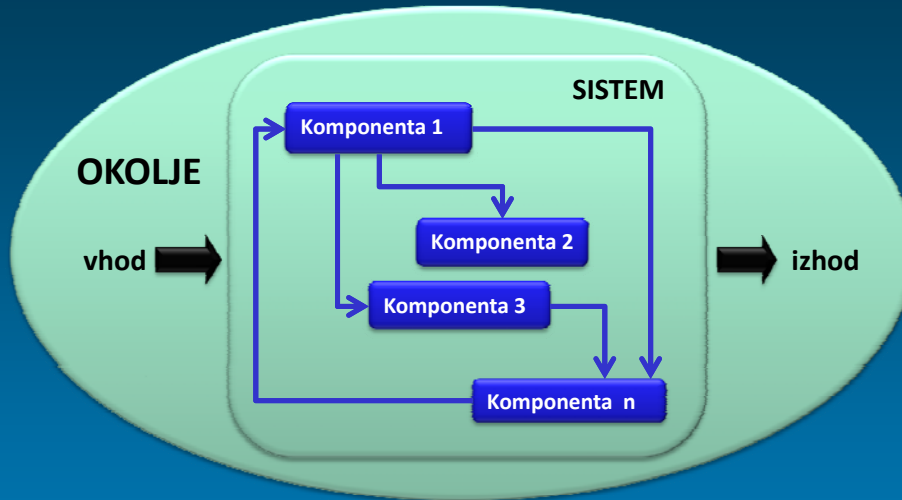


## Opredelitev sistema

- **Sistem** je celota, ki se sestoji iz več komponent ali podsistemov in množice povezav med njimi.
- Sisteme lahko razdelimo v tri temeljne skupine:
  - **Naravni sistemi:** iz naravnih sestavin, delujejo po naravnih zakonitostih, za naravne smotre; uravnavajo se sami;
  - **Tehnični sistemi:** iz naravnih snovi snuje človek, uporabljajo naravne zakonitosti, delujejo za cilje organizacije; krmiljenje je avtomatizirano, samodejno;
  - **Organizacijski sistemi:** snuje človek iz naravnih in tehničnih sistemov; delujejo po načelih in predpisih za smotre in cilje organizacije; krmiljenje je zavestno – iz človekove volje.
- S sistemi se ukvarja **teorija sistemov**



## Shema sistema



### Lastnosti sistema

- Vsaka komponenta je za sistem pomembna – obstoj in funkcija komponente vplivata na obstoj in funkcijo celotnega sistema.
- Nobena komponenta ni *izolirana*.
- Sistem s svojo funkcijo vpliva na funkcijo komponente.

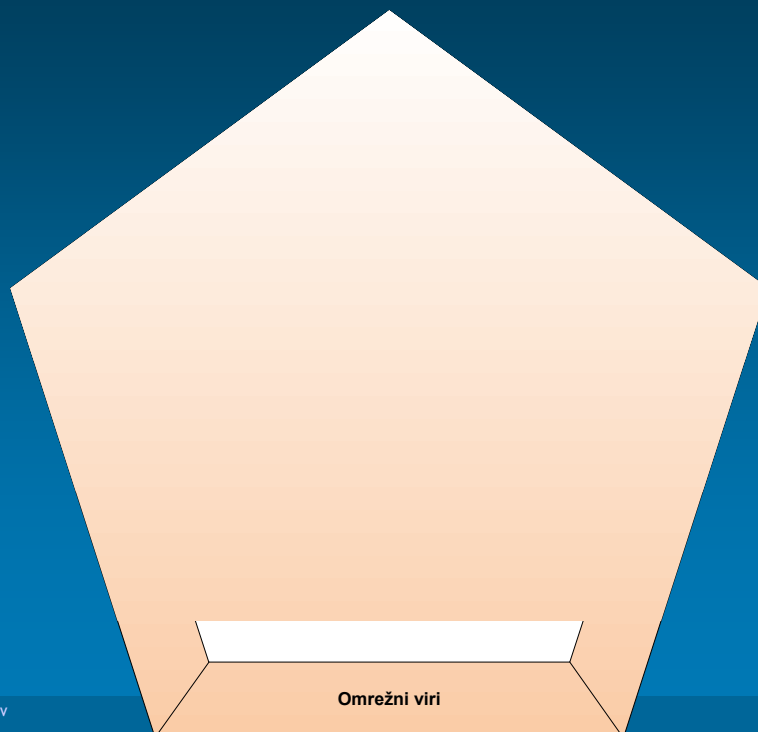
## IS kot podsistem poslovnega sistema



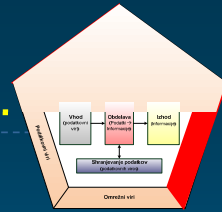
# Sinergija procesov v poslovnem okolju



# Konceptualni model IS...



## Konceptualni model IS - ključni viri...



### ▪ Zaposleni

- Informatiki (sistemski analitiki, razvijalci programske opreme, sistemski operaterji)
- Uporabniki (vsi, ki uporabljajo IS)

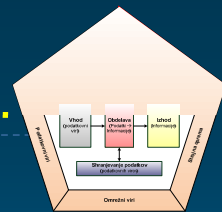


### ▪ Strojna oprema

- Strojna oprema
- Pomnilniki

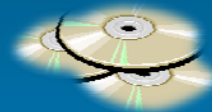


## Konceptualni model IS - ključni viri...

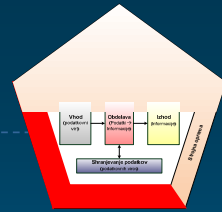


### ▪ Programska oprema in postopki

- Sistemska programska oprema (operacijski sistem...)
- Aplikacije (podpora poslovnim funkcijam oziroma procesom)
- Postopki oziroma navodila uporabnikom IS (postopki za zajem podatkov, postopki za odpravljanje napak, postopki za razdeljevanje plačilnih list...)



## Konceptualni model IS - ključni viri



- **Podatkovni viri**

- O izdelkih, o strankah, o zaposlenih, o sredstvih...

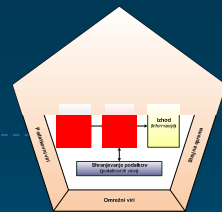


- **Omrežja**

- Komunikacijski kanali, dostopne točke, nadzorni sistemi...



## Konceptualni model IS - aktivnosti...



- **Vhod**

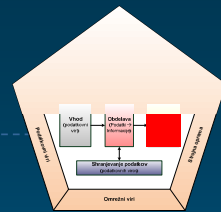
- Zajem in priprava podatkov
- Primer: odčitavanje črtnih kod, RFID

- **Obdelava**

- Obdelava podatkov (računanje, primerjanje, sortiranje, agregiranje...)



## Konceptualni model IS - aktivnosti...

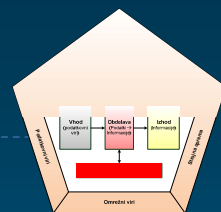


### ■ Izhod

- Posredovanje rezultatov obdelav (informacij) uporabnikom
- Osnovni cilj informacijskega sistema je pridobivanje "informacij" za uporabnike
- Primeri:
  - Poročila uprave in ostala poslovna dokumentacija v tekstovni in grafični obliki,
  - Zvočni zapisi.



## Konceptualni model IS - aktivnosti...



### ■ Shranjevanje podatkov

- Podatki se shranijo v podatkovno bazo za kasnejšo uporabo
- Predstavlja eno izmed ključnih komponent IS

### ■ Nadzor

- Nadzor zmogljivosti sistema
- Odkrivanje ozkih grl



## Podatek in informacija

- Podatek in informacija sta besedi, ki označujeta različna pojma.



## Podatek in informacija

### Podatek

Podatek je predstavitev informacije na formaliziran način, ki je primeren za komunikacijo, interpretacijo ali obdelavo (s strani človeka ali stroja). Predstavimo ga lahko s pomočjo simbolov ali analognih veličin, ki ji je pripisan, ali se ji lahko pripiše nek pomen.

### Informacija

Informacija je znanje, ki se nanaša na objekte, kot so dejstva, dogodki, stvari, procesi ali ideje, vključno s koncepti, ki imajo v okviru nekega konteksta določen pomen (ISO).



# Podatek in informacija

## ▪ Informacijska enačba

Borje Langefors

- Informacija je novo spoznanje, ki ga človek doda svojemu poznavanju sveta. Odnos med informacijo, podatki, časom in interpretatorjevim znanjem predstavlja informacijska enačba:

$$I = i (D, S, t)$$

- I – informacija, ki jo posredujejo podatki
- i – informacijska funkcija
- D – podatki
- S – prejemnikovo znanje
- t – čas, ki je na voljo prejemniku za interpretacijo podatkov



# Podatek in informacija

## ▪ Zaključki:

- Podatki niso informacija
- Podatki ne vsebujejo informacije
- Podatki posredujejo informacijo prejemniku, katerega znanje je konsistentno z izbrano predstavitvijo podatkov in modelom sveta, na katerega se nanašajo.
- Če je količina podatkov tako velika, da se jih v času, ki je na voljo za ukrepanje na njihovi osnovi, ne da interpretirati, se lahko zgodi, da s podatki ni posredovana nobena informacija.



## Poglavje II

# Poslovne aplikacije

- E-poslovanje
- Več-funkcijski poslovni sistem
- Poslovno informacijska arhitektura
- Funkcionalni informacijski podsistemi
- CRM
- ERP



## E-poslovanje

E-business

### E - poslovanje

**S terminom e-poslovanje označujemo uporabo informacijskih tehnologij in omrežij (internet, intranet, ekstranet...) za podporo elektronskemu trgovanju, podjetniškemu komuniciranju in sodelovanju ter spletnim poslovnim procesom.**

- E-poslovanje se lahko izvaja:
  - med podjetji (B2B),
  - med podjetjem in partnerji,
  - med podjetjem in strankami (B2C).





# Več-funkcijski poslovni sistem

Cross-functional enterprise systems

- Presegajo meje tradicionalnih poslovnih funkcij in se odpirajo navzven;
- Stranke, dobavitelji, partnerji in zaposleni postajajo pomemben člen poslovnih procesov;
- Prenova in izboljšanje ključnih poslovnih procesov.



# Poslovno informacijska arhitektura...

Enterprise Resource Planning

- ERP – Integriran poslovni informacijski sistem
  - zagotoviti učinkovite interne poslovne procese na področju proizvodnje, razpečave (distribucije) in financ

Customer Relationship Management

- CRM – Upravljanje odnosov s strankami
  - pridobiti in zadržati dobičkonosne stranke
  - trženje, prodaja, podpora



## Poslovno informacijska arhitektura...

### Partner Relationship Management

- **PRM – Upravljanje odnosov s partnerji**
  - pridobiti in zadržati partnerje, ki lahko povečajo prodajo in razpečavo (distribucijo) izdelkov ali storitev podjetja

### Supply Chain Management

- **SCM – Upravljanje oskrbovalne verige**
  - razviti učinkovite procese iskanja virov in nabave izdelkov ter storitev potrebnih za poslovanje



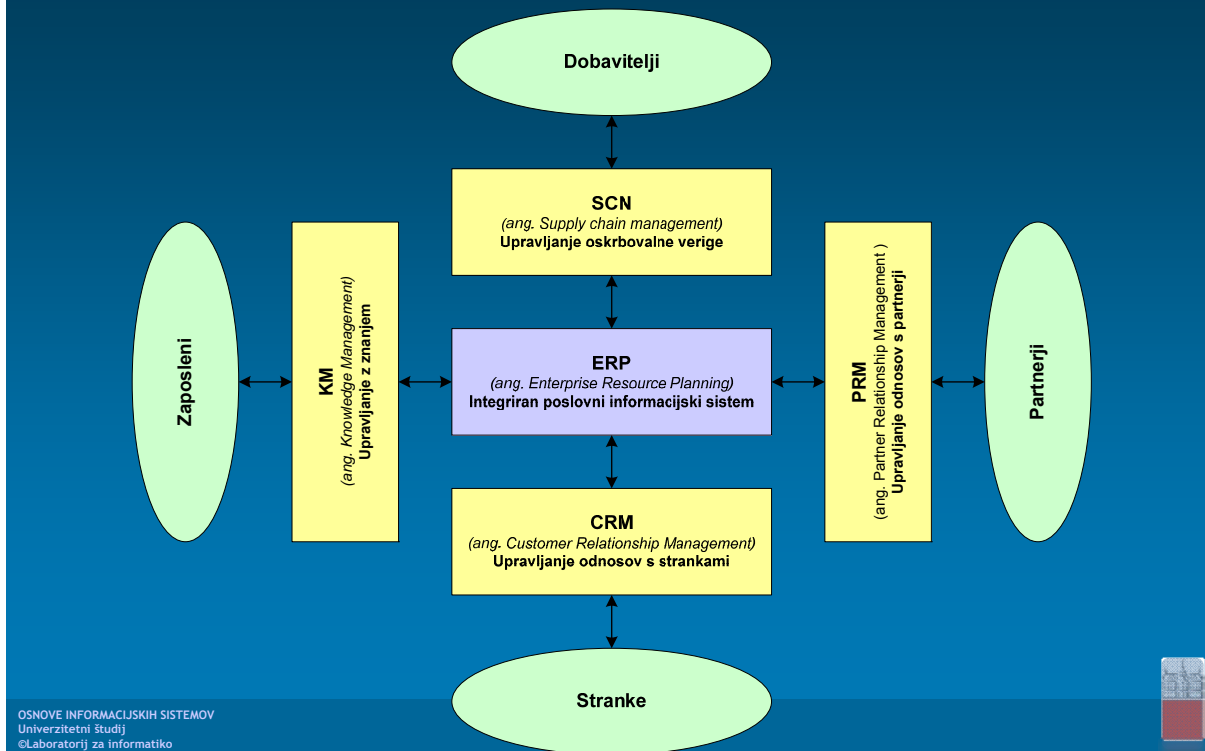
## Poslovno informacijska arhitektura...

### Upravljanje z znanjem

- **KM – Upravljanje z znanjem**
  - zaposlenim zagotoviti ustrezna odločitvena orodja in orodja, ki podpirajo skupinsko delo



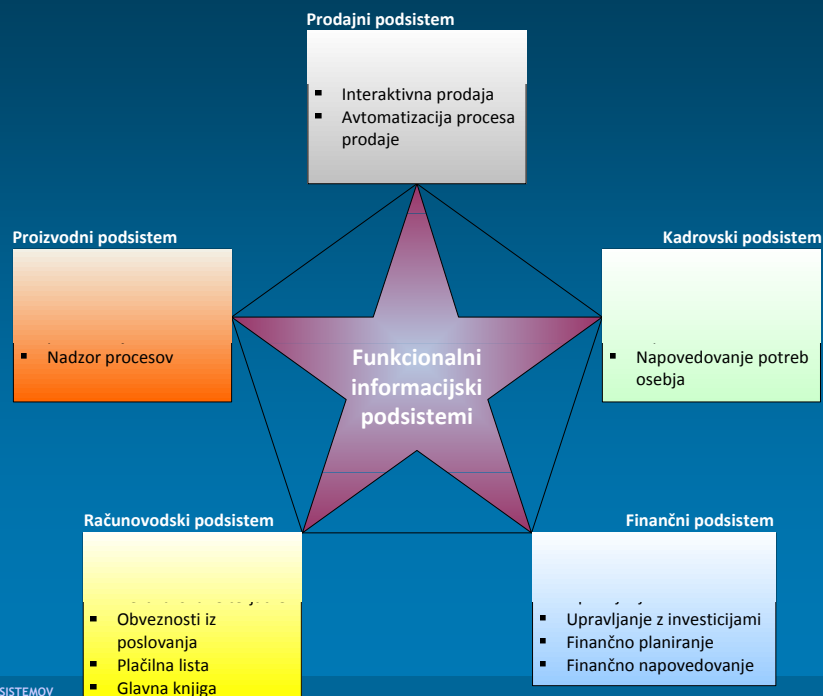
# Poslovno informacijska arhitektura



## Funkcionalni informacijski podsistemi...

- S pojmom sistem za upravljanje s poslovnimi funkcijami označimo nabor aplikativnih sistemov, ki v podjetju podpirajo
  - finančno,
  - računovodsko,
  - prodajno,
  - proizvodno in
  - kadrovsko poslovno funkcijo

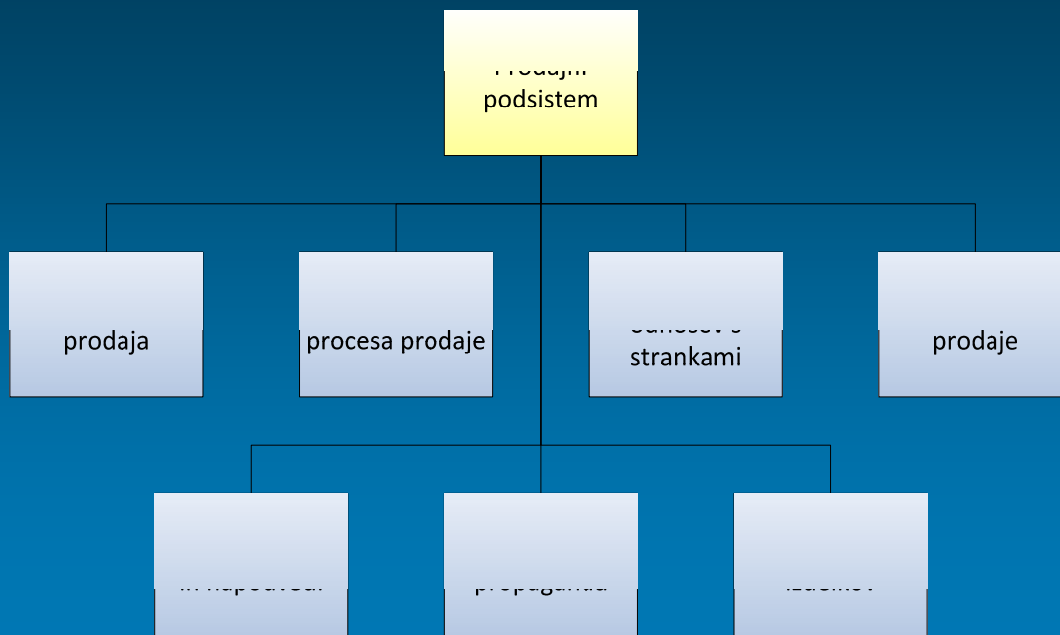
## Funkcionalni informacijski podsistemi...



## Prodajni podsistem...

- **Prodajni podsistem** je odgovoren za načrtovanje, nadzor in obdelavo transakcij povezanih s prodajno funkcijo (upravljanje prodaje, oglaševanje, promocije...)

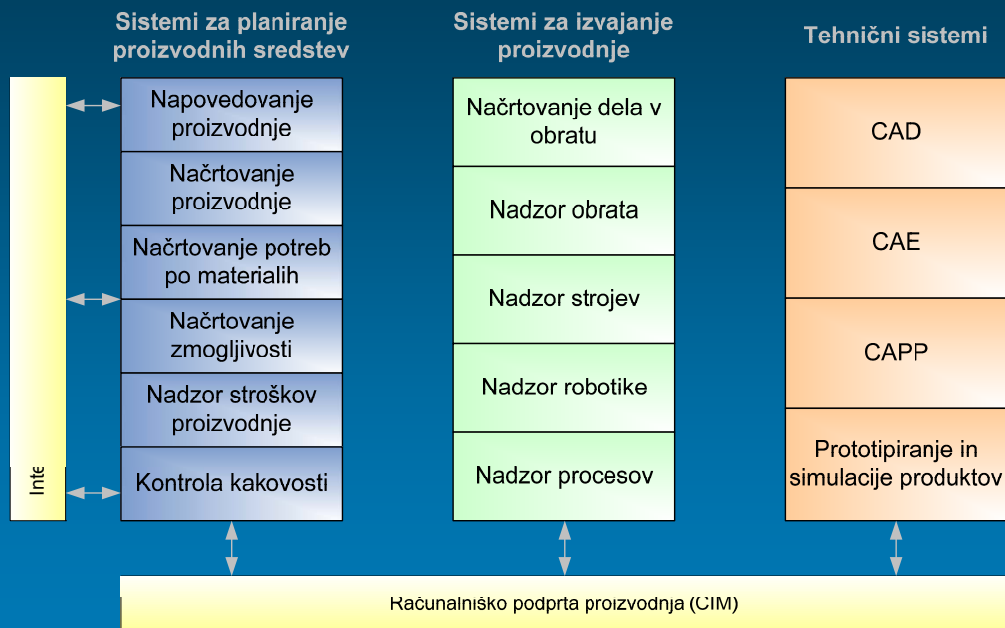
## Prodajni podsistem



## Proizvodni podsistem

- **Proizvodni podsistem** skrbi za načrtovanje, nadzor in izvajanje proizvodnega procesa.
- **Osnovni koncepti:**
  - CIM (Computer-Integrated Manufacturing)  
Računalniško podprta proizvodnja
  - CAM (Computer-Aided Manufacturing)  
Računalniško podprto krmiljenje strojev in naprav
  - CAPP (Computer-Aided Process Planning)  
Računalniško podprta izdelava tehnoloških postopkov
  - CAD (Computer-Aided Design)  
Računalniško podprto konstruiranje

# Proizvodni podsistem



# Kadrovski podsistem...

- **Kadrovski podsistem** podpira procese namenjene upravljanju s kadri oziroma zaposlenimi:
  - pridobivanje kadrov,
  - izbiranje in zaposlovanje novih kadrov,
  - razporeditev na delovna mesta in ocena uspešnosti,
  - usposabljanje
  - načrtovanje kariere

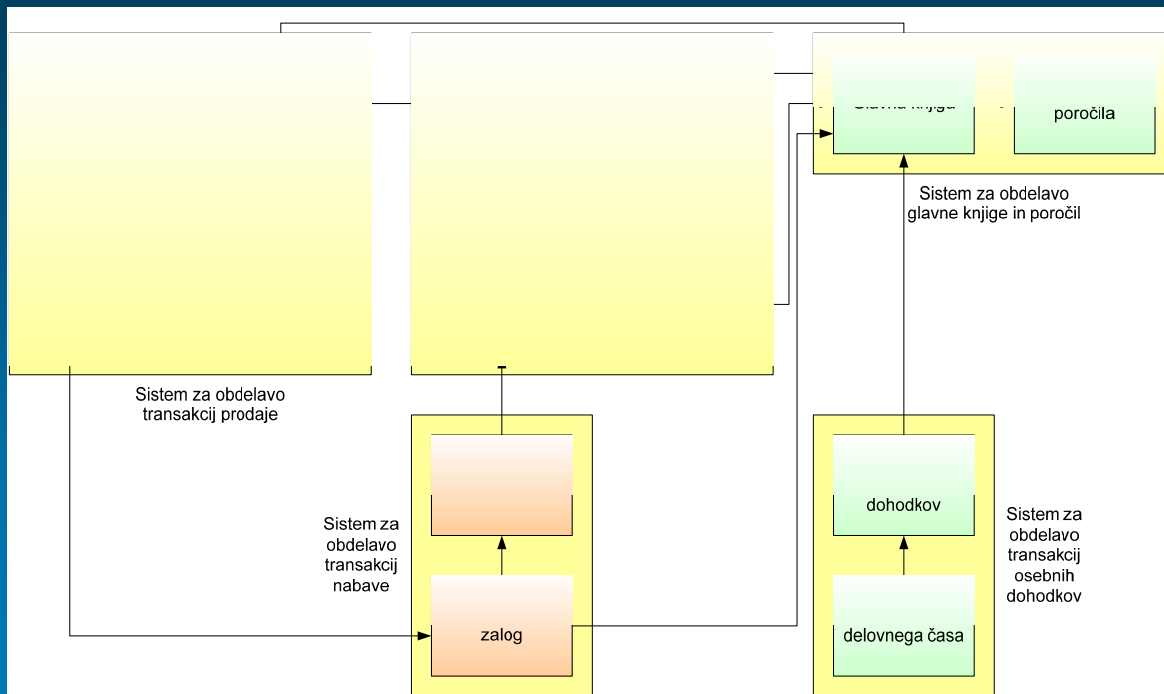
# Kadrovski podsistem



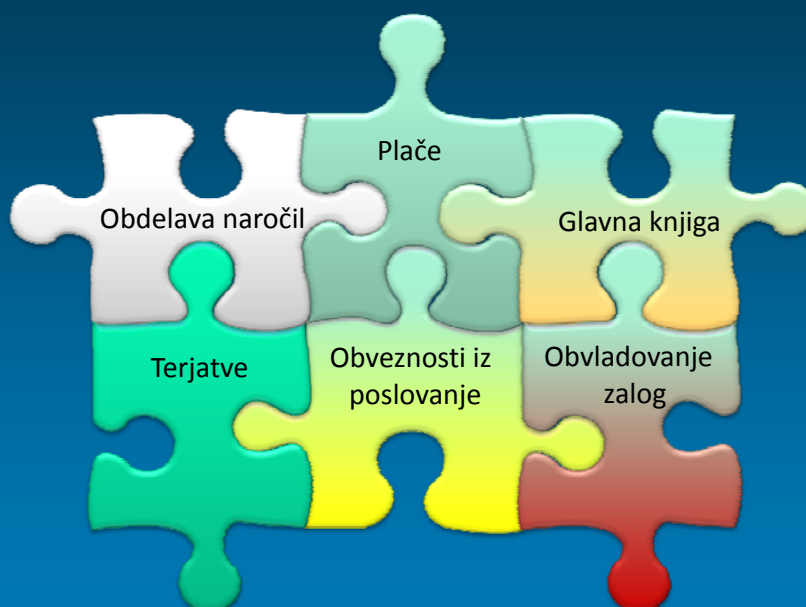
# Računovodski podsistem...

- **Računovodski podsistem podpira:**
  - evidentiranje in izdelavo poročil o poslovnih transakcijah
  - sledenje toku sredstev skozi podjetje
  - izdelavo finančnih poročil (izkazov stanja)
- Računovodski podsistem zagotavlja informacije potrebne za načrtovanje in vodenje poslovnih dejavnosti

# Računovodski podsistem



# Temeljni računovodski aplikativni sistemi...





## Temeljni računovodski aplikativni sistemi...

- **Obdelava naročil**
  - zajem in obdelava naročil strank
  - priprava podatkov za:
    - aplikativni sistem za obvladovanja zalog,
    - aplikativni sistem za terjatve
- **Obvladovanje zalog**
  - obdelava podatkov o stanju zalog
  - priprava podatkov za:
    - dostavo,
    - ponovna naročila



## Temeljni računovodski aplikativni sistemi...

- **Terjatve**
  - evidenca zneskov dolga strank
  - priprava:
    - faktur,
    - mesečnih izkazov,
    - poročil o vodenju kreditov
- **Obveznosti iz poslovanja**
  - evidenca nakupov, dolgov in izvedenih plačil dobaviteljem
  - priprava:
    - poročilo o upravljanju z denarnimi sredstvi



# Temeljni računovodski aplikativni sistemi

- **Plače**
  - evidenca dela zaposlenih in podatkov o nadomestilih
  - priprava:
    - izplačil plač,
    - dokumentov in poročil plačilne liste
- **Glavna knjiga**
  - združevanje podatkov iz drugih računovodskih sistemov
  - priprava:
    - periodičnih izkazov stanja,
    - poslovnih poročil



# Finančni podsistem...

- **Finančni podsistem podpira:**
  - upravljane s financami poslovnega sistema
  - razporejanje in nadzor finančnih virov
- **Upravljanje s finančnimi viri obsega:**
  - upravljanje z denarnimi sredstvi in vrednostnimi papirji
  - načrtovanje proračunskih sredstev
  - finančno napovedovanje
  - finančno planiranje



## Finančni podsistem...



## CRM – Upravljanje odnosov s strankami

### Customer Relationship Management

- CRM je poslovni aplikativni sistem, ki je v celoti osredotočen na stranko.
- CRM združuje avtomatizacijo procesov prodaje, neposredno trženje, upravljanje z računi, upravljanje z naročili in podporo strankam
- Primarna cilja CRM:
  - Podjetju oziroma zaposlenim zagotoviti enoten in celovit pogled nad vsemi podatki o strankah
  - Strankam omogočiti enoten in celovit pogled na podjetje

# CRM – Upravljanje odnosov s strankami

## Temeljni sklopi CRM

- **Upravljanje s stiki in računi**
  - Zajem in sledenje vseh stikov stranke s podjetjem
- **Prodaja**
  - Prodajnemu osebju zagotavlja potrebna programska orodja in podatke za učinkovito prodajo izdelkov
  - Zagotavlja hiter dostop do podatkov o strankah (pretekli nakupi, specifične zahteve, potencialna področja zanimanja...)



# CRM – Upravljanje odnosov s strankami

## Temeljni sklopi CRM

- **Trženje in izpolnitev**
  - Omogoča pripravo in izvedbo oglaševalskih akcij ter analizo odzivov nanje
  - Zagotavlja hiter odziv na zahteve strank
- **Podpora**
  - Podpornemu osebju zagotavlja programska orodja in podatke za učinkovito izvajanje podpornih aktivnosti
- **Zadržanje in zvestoba**
  - Omogoča identifikacijo in nagrajevanje najzvestejših in najdobičkonosnejših strank



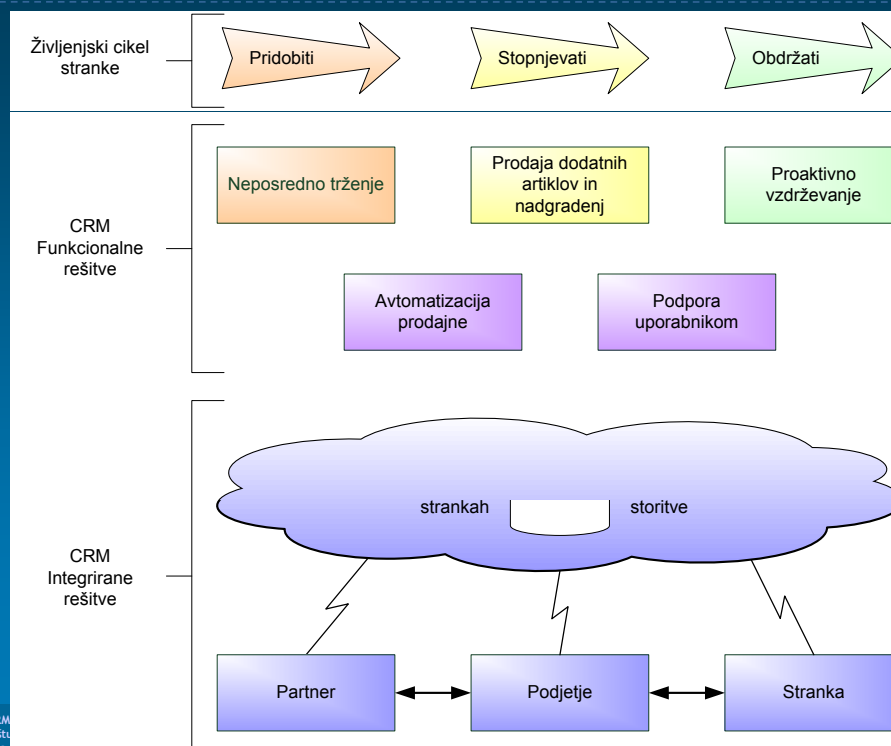
# CRM – Upravljanje odnosov s strankami

## Temeljni sklopi CRM

### ■ V razmislek:

- Prodaja izdelka novi stranki stane šestkrat več kot prodaja istega izdelka obstoječi stranki
- Povprečna nezadovoljna stranka bo svojo neprijetno izkušnjo delila s približno osmimi do desetimi ljudmi
- Verjetnost prodaje novi stranki je 15%, verjetnost prodaje obstoječi stranki je 50%
- 70% strank bo znova poslovalo s podjetjem ob hitri razrešitvi garancijskega zahtevka

## Poslovni proces CRM



# CRM – Upravljanje odnosov s strankami

## Prednosti

- Omogoča identifikacijo najboljših strank;
- Omogoča prilagajanje in personifikacijo produktov in storitev skladno z zahtevami, željami in navadami strank;
- Stranki omogoča enako izkušnjo neodvisno od mesta oziroma načina dostopa (neposredno v prodajalni, prek spleta, telefona...)



# CRM – Upravljanje odnosov s strankami

## Pasti

- CRM ne zagotavlja uspeha
  - Preko 50% CRM projektov ne izpolni začetnih pričakovanj
  - 20% implementacij CRM je škodovalo odnosu z obstoječimi dolgoletnimi strankami
- Glavna razloga za neuspeh sta:
  - Pomanjkanje razumevanja
  - Nezadostna priprava



# CRM – Upravljanje odnosov s strankami

## Trendi

- **Operativni CRM**
  - Stranki olajša komunikacijo s podjetjem (telefon, faks, e-pošta, klepet, mobilne naprave)
- **Analitični CRM**
  - Omogoča natančno analizo in napovedovanje navad, vrednot in dobičkonosnosti strank



# CRM – Upravljanje odnosov s strankami

## Trendi

- **Sodelovalni CRM**
  - Poenostavlja načine interakcije in sodelovanja s strankami, dobavitelji in partnerji
- **CRM Portal**
  - Vsem uporabnikom omogoča dostop do potrebnih orodij in informacij glede na njihove vloge in zahteve.



# ERP – Integriran IS

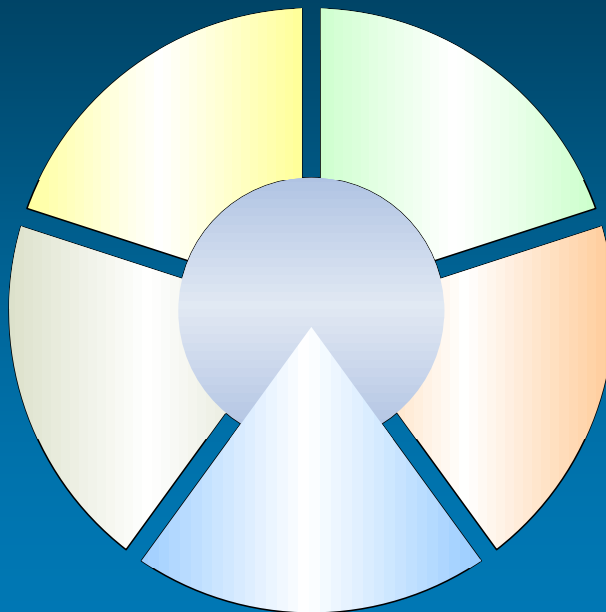
## Enterprise Resource Planning

- Integrirana več funkcijska programska oprema, ki s prenovo proizvodnih, razpečevalskih (distribucijskih), finančnih, kadrovskih in drugih osnovnih poslovnih procesov omogoča večjo učinkovitost, prilagodljivost in donosnost podjetja
- ERP je tehnološka hrbtnica e-poslovanja



# ERP – Integriran IS

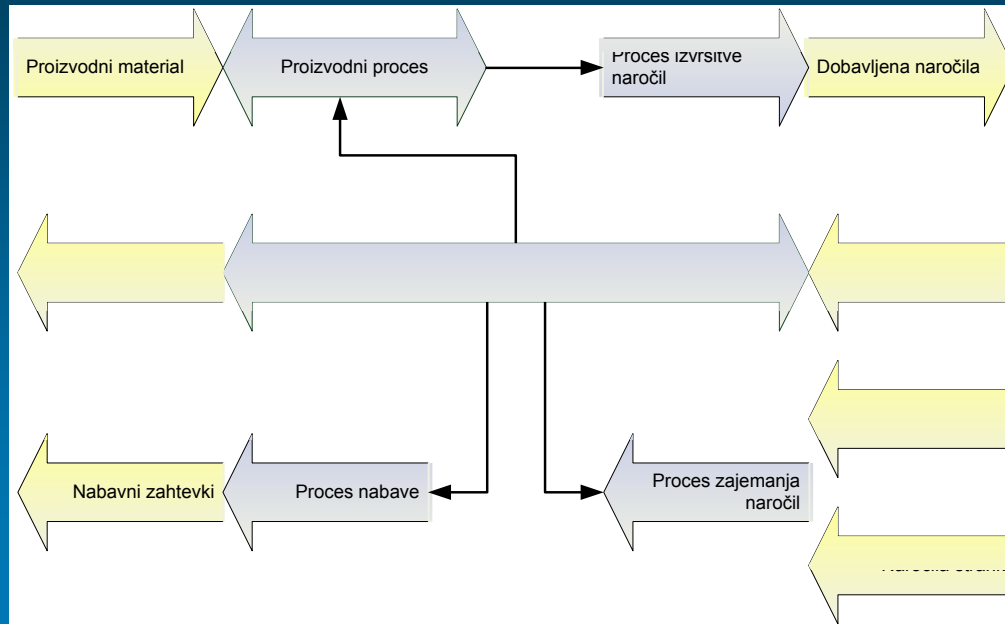
## Temeljni sklopi ERP





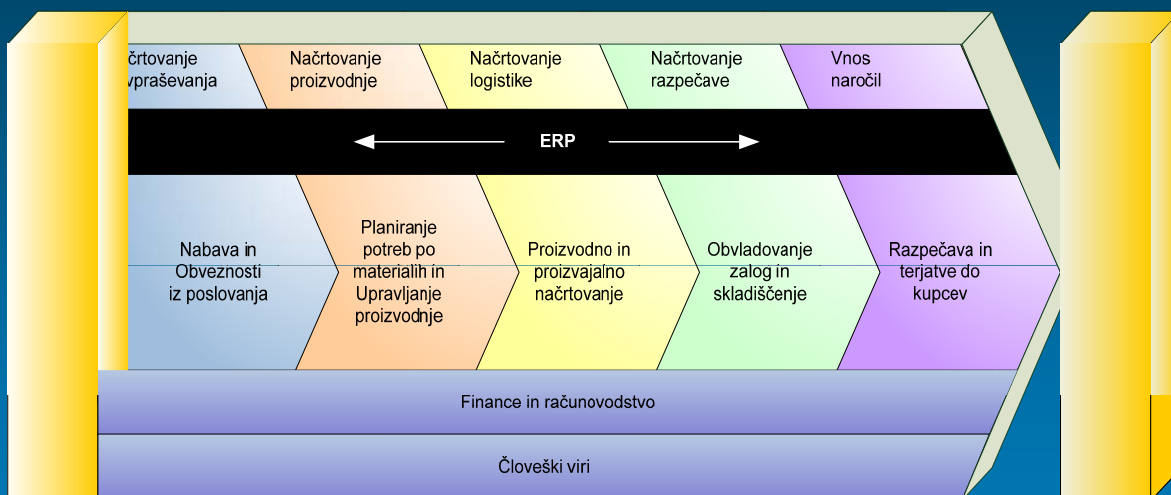
# ERP – Integriran IS

## Nekateri ključni procesi ERP



# ERP – Integriran IS

## Primer ERP – Colgate Palmolive



# ERP – Integriran IS

## Prednosti vpeljave ERP

- **Kakovost in učinkovitost**
  - ERP je ogrodje, ki služi kot osnova za integracijo in izboljšanje internih poslovnih procesov
  - Izboljšanje kvalitete in učinkovitosti proizvodnje, razpečave (distribucije) in strežbe strank
- **Zmanjšanje stroškov**
  - Opazno zmanjšanje stroškov na področju obdelave transakcij in IT podpore (programska, strojna in informacijska podpora)



# ERP – Integriran IS

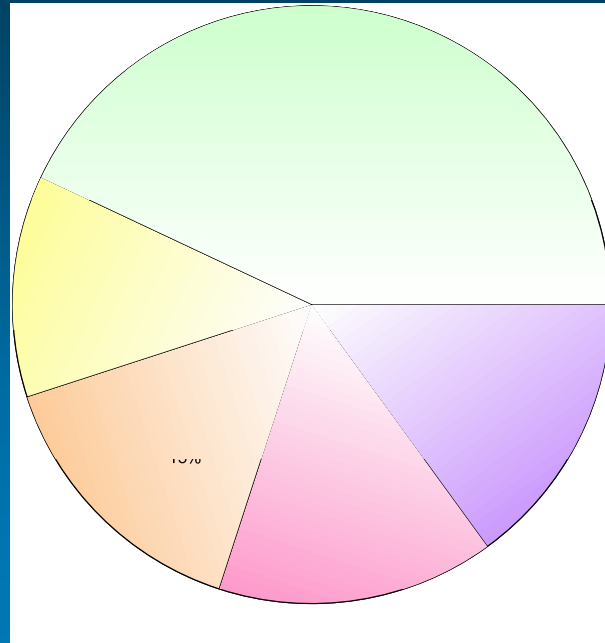
## Prednosti vpeljave ERP

- **Podpora odločanju**
  - ERP zagotavlja hiter in agregiran dostop do ključnih informacij o stanju in uspehu podjetja in tako omogoči vodstvu sprejemanje boljših predvsem pa pravočasnih odločitev
- **Poslovna agilnost**
  - Vpeljava ERP sistema podre ločnice med poslovnimi procesi, informacijskimi sistemi in viri informacij tako na oddelčnem, kot tudi na funkcijskem nivoju
  - Z ERP se vzpostavi prilagodljiva organizacijska struktura, ki se je sposobna učinkovito spoprijeti z novimi poslovnimi izzivi



# ERP – Integriran IS

## Stroški vpeljave ERP



# ERP – Integriran IS

## Trendi

- Podcenjevanje kompleksnosti načrtovanja in razvoja ERP sistema s strani vodstva in IT strokovnjakov;
- Zapostavljanje ključnih uporabnikov v procesu načrtovanja in razvoja;
- Neustrezen obseg usposabljanja;

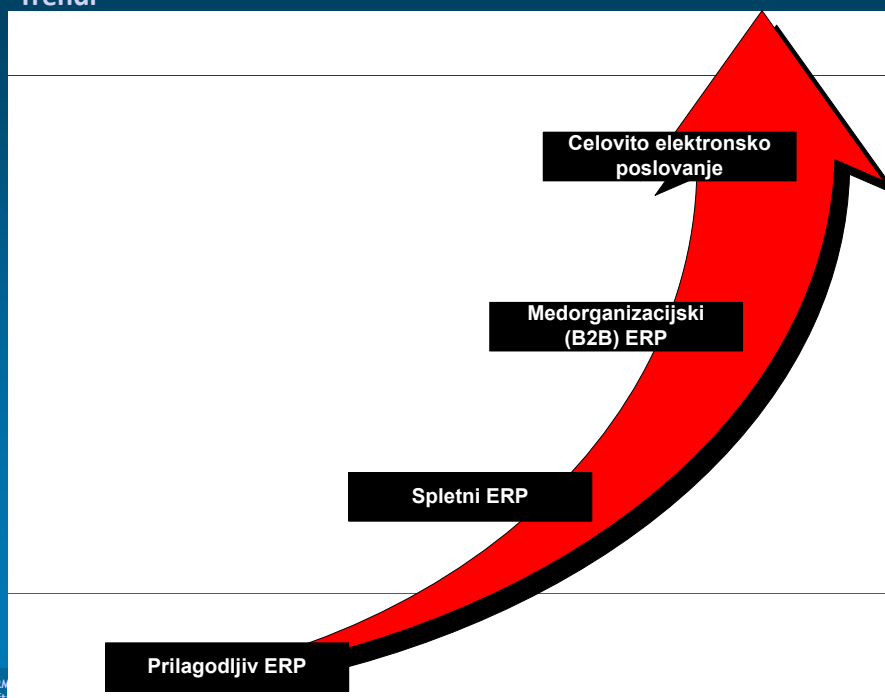
# ERP – Integriran IS

## Trendi

- Prehiter prehod na nov sistem;
- Napake pri pretvarjanju oziroma pri uvozu podatkov in pri testiranju;
- Zanašanje na trditve in obljube ponudnikov rešitev ERP in njihovih zastopnikov brez ustreznega predhodnega (neodvisnega) testiranja;

# ERP – Integriran IS

## Trendi



## Poglavje III

# Osnove informacijskih sistemov

- Osnovni pojmi
- Življenjski modeli razvoja IS
- Metodologije razvoja IS



## Kaj nas zanima pri razvoju IS?...

- V okviru razvoja IS nas zanima, kako razviti računalniške rešitve, ki bodo čim boljše podprle delovanje IS.
- V okviru razvoja IS se ukvarjamo z:
  - Razvojem računalniške rešitve
  - Nabavo ustrezne strojne opreme
  - Namestitvijo systemske programske opreme
  - Uvedbo rešitve
  - Vzdrževanjem rešitve



## Kaj nas zanima pri razvoju IS?

- Razvoj IS ni zgolj programiranje!
- Ni zgolj inženirsko delo...
- Pri razvoju IS imajo velik pomen tudi sociološki dejavniki:
  - Kako dojemamo problematiko?
  - Kako razumemo potrebe uporabnikov?
  - Kako uvedemo rešitve v prakso?
  - ...

**Information System is about an implementation of IT into a human enterprise!**



## Razvoj IS in vrste IS

- Pristop k razvoju IS je odvisen tudi od vrste IS.
- Nekaterе vrste IS zahtevajo specifične pristope:
  - Ekspertni sistemi
  - Odločitveni sistemi
  - Sistemi za podporo delovnim procesom
- Največ napora gre še vedno za razvoj IS za podporo operativnem delovanju PS.



# Sinonimi za računalniške programe

## Terminologija

- V okviru razvoja IS med drugim nastanejo računalniški programi.
- Izraz "Računalniški program" ima številne sinonime:
  - Program
  - Aplikacija
  - Aplikativni sistem
  - Informacijska rešitev
  - Računalniška rešitev
  - ...



## P1.2 – Življenjski modeli razvoja IS

# Življenjski modeli razvoja IS...

- Razvoj IS zajema številna opravila, navadno razdeljena v faze:
  - Analiza
  - Načrtovanje
  - Implementacija
  - Testiranje
  - Uvedba
  - Vzdrževanje
- Življenjski model razvoja IS (SDLC\*) pove, v kakšnem sosledju in na kakšen način si v okviru razvoja IS sledijo posamezne faze.

Kot večina razvojnih procesov sledi tudi razvoj IS določenemu življenjskemu ciklu oziroma razvojnemu modelu, ki določa zaporedje faz razvoja.

\* SDLC - System Development Life Cycle



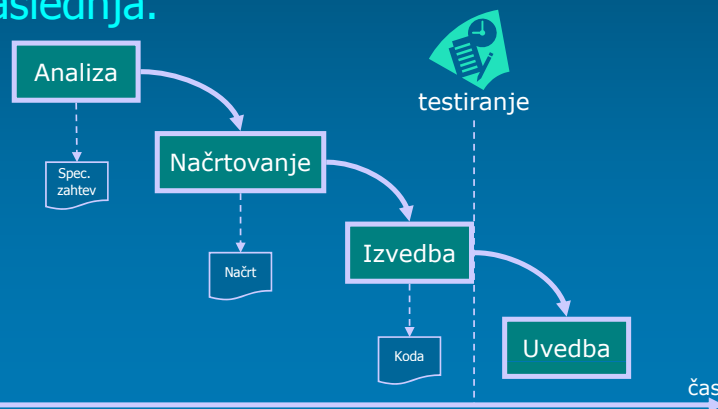
## Življenjski modeli razvoja IS...

- Poznamo različne življenjske modele:
  - Zaporedni ali slapovni model
  - Interativni model
  - Prototipni model
  - Inkrementalni model
- V praksi se večinoma uporablja kombinacija različnih modelov.



## Zaporedni ali slapovni model...

- Zaporedni ali slapovni model temelji na zaporednem izvajanju faz.
- Ko se ena faza v celoti konča, se začne naslednja.





## Zaporedni ali slapovni model...

### ▪ Značilnosti:

- Najstarejši razvojni model, značilen za prve oblike strukturnega pristopa.
- Faze si sledijo zaporedno.
- Vračanje nazaj ni mogoče.
- Primeren za relativno kompleksne projekte, če zahteve dobro razumemo in se med projektom ne bodo bistveno spreminjale.
- Omogoča dobro in natančno projektno vodenje.



## Zaporedni ali slapovni model...

### ▪ Prednosti:

- Pomaga zmanjševati količino režijskega dela, ki ni v neposredni povezavi z izdelavo programske opreme (npr. vodenje projekta), saj je mogoče načrtovanje v celoti izvesti vnaprej.

### ▪ Slabosti:

- Ni fleksibilen. Vsaka naknadna sprememba zahteva veliko dodatnega napora.
- Nenaraven: v praksi težko pričakovati, da se lahko nek postopek v celoti zaključiti, preden se začne z naslednjim.
- Ne omogoča paralelnega izvajanja delov postopkov.



## Zaporedni ali slapovni model

- Zaradi kritik pojav modificiranih različic slapovnega razvoja.
- Odprava nekaterih pomanjkljivost:
  - uvedba bolj enostavnega prehajanja med postopki,
  - paralelno izvajanje delov različnih postopkov,
  - ...
- Slapovni model kljub vsemu nudi zelo čvrsto oporo sistematičnemu razvoju.
- Možno uporabiti v kombinaciji z drugimi modeli.



## Iterativni model...



- Razvit kot odziv na pomanjkljivosti slapovnega pristopa.
- Faze razvoja izvajamo v več iteracijah.

### Iteracija

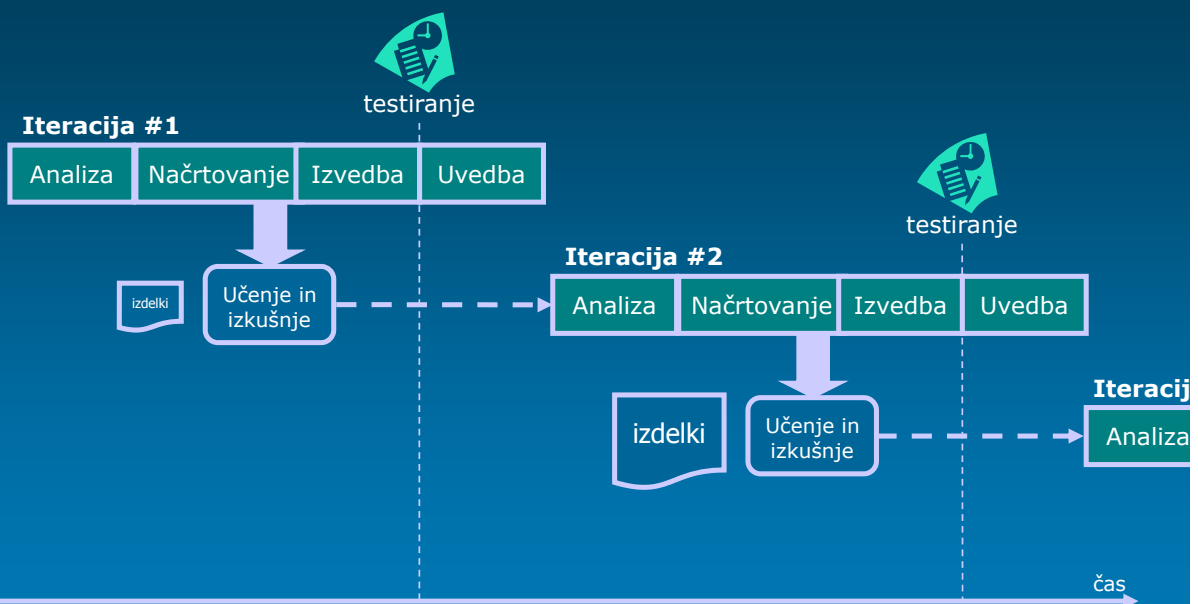
Iteracija je specifično zaporedje aktivnosti, izvedenih na osnovi načrta in z določenim kriterijem vrednotenja, ki se konča z izdajo izdelka.



## Iterativni model...

- V vsaki iteraciji razvijemo določen del funkcionalnosti celotnega sistema.
- Iteracija gre navadno čez vse faze razvoja: analizo, načrt, izvedbo...
- V začetnih iteracijah razvijemo najbolj tvegane dele sistema.
- Gre za **evolucijski razvoj**.

## Iterativni model...



Tipično trajanje ene iteracije: 7 do 14 dni

# Iterativni model...

## ▪ Lastnosti iteracij:

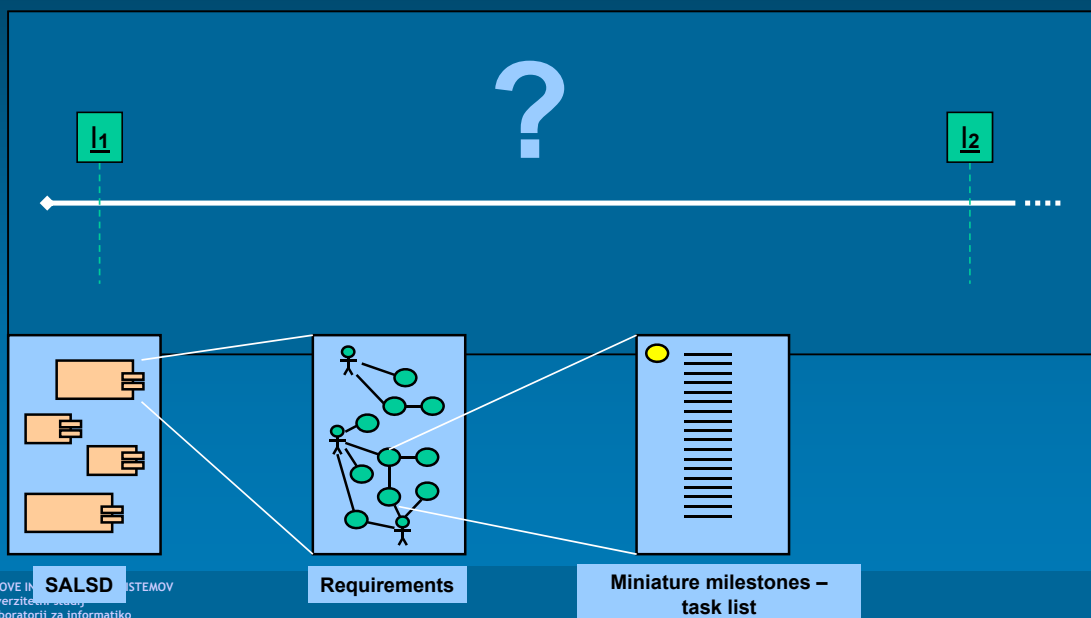
- Trajanje: od 7 do 14 dni (tipično)
- Vsaka iteracija gre čez vse faze (ne z enako intenzivnostjo)
- Naslednja iteracija se lahko začne šele takrat, ko je prejšnja končana.
- Vsebina naslednje iteracija je določena na osnovi rezultatov prejšnje.
- Med izvajanjem iteracije ne sprejemamo sprememb.

# Planiranje na makro in mikro ravni

1. Release Planning

2. Iteration Planning

3. Task Planning



## Iterativni model...

### ▪ Prednosti:

- Prednosti iterativnega razvoja (proti zaporednemu):
- Najbolj tvegani deli so razrešeni še preden postane investicija velika
- Začetne iteracije omogočijo zgodnje povratne informacije s strani uporabnikov
- Preizkušanje in povezovanje v sistem sta nepretrgana
- Ciljni mejniki omogočajo kratkoročno osredotočenje
- Napredek merimo z ocenjevanjem izvedenega dela
- Možna je predaja izvedenega dela projekta še preden je dokončan celoten projekt



## Iterativni model

### ▪ Slabosti:

- ne omogoča dobrega načrtovanja poteka projekta.
- ni mogoče točno predvideti, koliko iteracij bo potrebnih za razvoj dokončnega (dovolj dobrega) izdelka.
- vodenje projekta je zahtevno.



## Prototipni model...

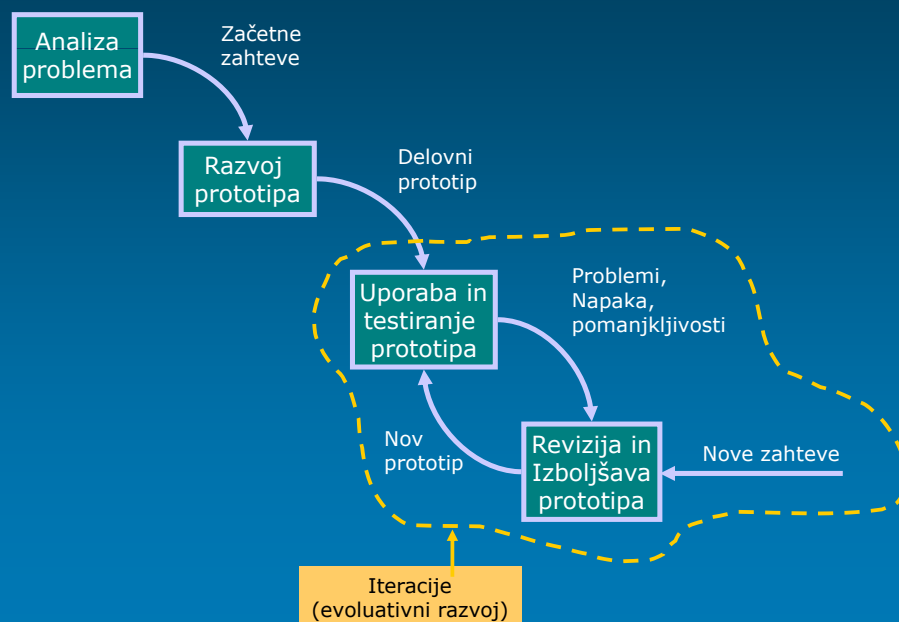


- Gre za različico iterativnega modela.
- Temelji na izdelavi **prototipov** in njihovi postopni izboljšavi, dokler ne dosežemo zadovoljive kakovosti.

### Prototip

Prototip označuje predhodno izdelane in navadno še nepopolne različice sistema ali dela sistema.

## Prototipni model



## Inkrementalni model...



- Temelji na postopni gradnji celotne IR in sproti predaji posameznih **inkrementov** uporabniku.

### Inkrement

**Inkrement predstavlja zaokroženo funkcionalnost sistema (sklop, podsistem, modul).**

- Ne razvijamo celotne IR hkrati. Omejimo se na posamezen sklop, ki ga razvijemo v celoti in predamo uporabniku.



## Inkrementalni model...

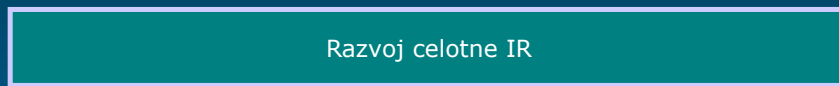
- Ne razvijamo celotne IR hkrati. Omejimo se na posamezen inkrement, ki ga razvijemo v celoti, predamo uporabniku ter nadaljujemo z naslednjim sklopom.
- Ob predaji novi sklop povežemo z ostalimi sklopi.
- Inkremente je moč razvijati tudi vzporedno.
- Rezultat razvoja po inkrementalnem modelu je IR, sestavljena iz integriranih sklopov.



# Inkrementalni model...

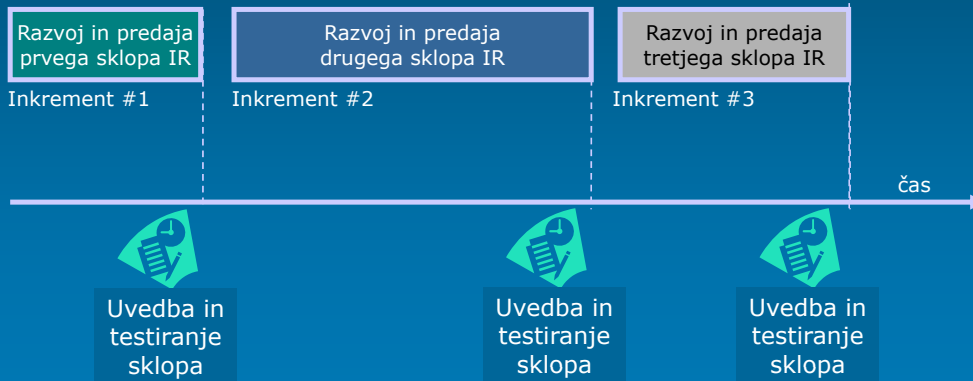


## Klasičen razvoj



Uvedba in testiranje celotnega sistema

## Inkrementalni razvoj



# Inkrementalni model...

- Priporočeno določiti ustrezen razpored razvoja sklopov.
- Zaporedje lahko določimo na različne načine:
  - na podlagi pomembnosti sklopov,
  - na podlagi upoštevanja tveganja,...
- Pri razporejanju sklopov na osnovi pomembnosti najprej razporedimo sklope z najpomembnejšo funkcionalnostjo. Na tak način funkcionalnost postopno nadgrajujemo do končnega sistema.





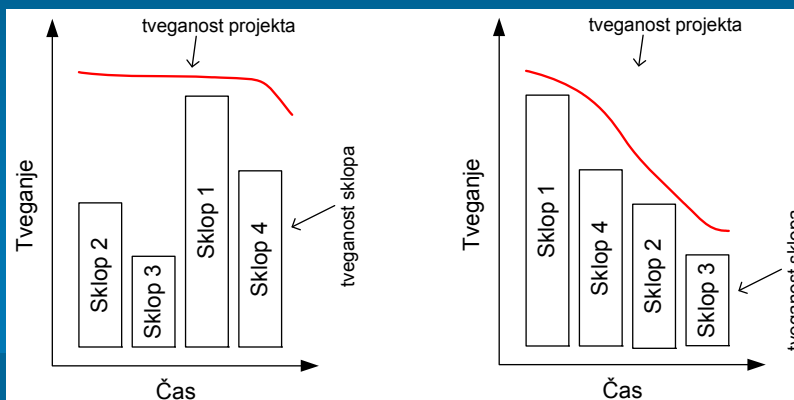
## Inkrementalni model...

- Pri razporejanju sklopov glede na pomembnost lahko uporabimo analizo MOSCOW, kjer funkcionalnosti IR razdelimo na:
  - funkcije, ki jih je nujno potrebno implementirati,
  - funkcije, ki bi jih bilo dobro implementirati,
  - funkcije, ki bi jih lahko implementirali – niso obvezne in
  - funkcije, ki jih ne bomo implementirali.



## Inkrementalni model...

- Na podlagi analize MOSCOW lahko funkcije sistema ustrezno časovno razporedimo.
  - Pri razporejanju sklopov upoštevajoč tveganost (v smislu tveganja, da razvit sklop ne bo ustrezal naročniku) najprej razvijemo najbolj tvegane in zahtevne sklope, kar omogoča hitrejše znižanje stopnje tveganja pri celotnem projektu



## Inkrementalni model...

- **Prednosti:**
  - Uporabnik prej dobi del zahtevane IR, saj se IR razvija po delih.
  - Rešitev, ki jo uporablja, se postopoma nadgrajuje, sam pa lahko sodeluje pri testiranju razvitih sklopov.
  - Naročnik lažje sledi napredovanju projekta.
- **Slabosti:**
  - Ni mogoče uporabiti pri vseh projektih. Nekaterih rešitev ni moč predati v uporabo po delih.
  - IR moramo razdeliti na sklope in predvideti odvisnosti med njimi. Pri neustreznem načrtovanju se lahko zgodi, da sklope neustrezno razporedimo.

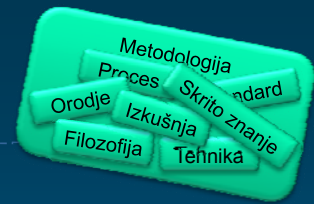


## Inkrementalni model

- Poseben primer inkrementalnega modela je razdelitev vsebine na samostojne projekte.
- Inkrementalni model možno učinkovito uporabiti v kombinaciji z iterativnim modelom:
  - razdelitev na inkremente,
  - vsak inkrement se izvaja iterativno.



## Kaj je metodologija razvoja IS?



### ■ Nekaj definicij:

- Metodologija je priporočena zbirka filozofij, faz, postopkov, pravil, tehnik, orodij, dokumentacije, upravljanja in izobraževanja za razvijalce IS (Maddison).
- Metodologija razvoja IS je priporočen način razvoja IS, ki temelji na filozofiji in množici principov (Avison, 2003).
- Metodologija je množica dogovorov (konvencij), s katerimi se (projektna) skupina/organizacija strinja (Cockburn, 2002).
- Metodologija je vse kar redno delamo, da bi dosegli končni rezultat – delujoča PO pri končnem uporabniku (Cockburn, 2002).



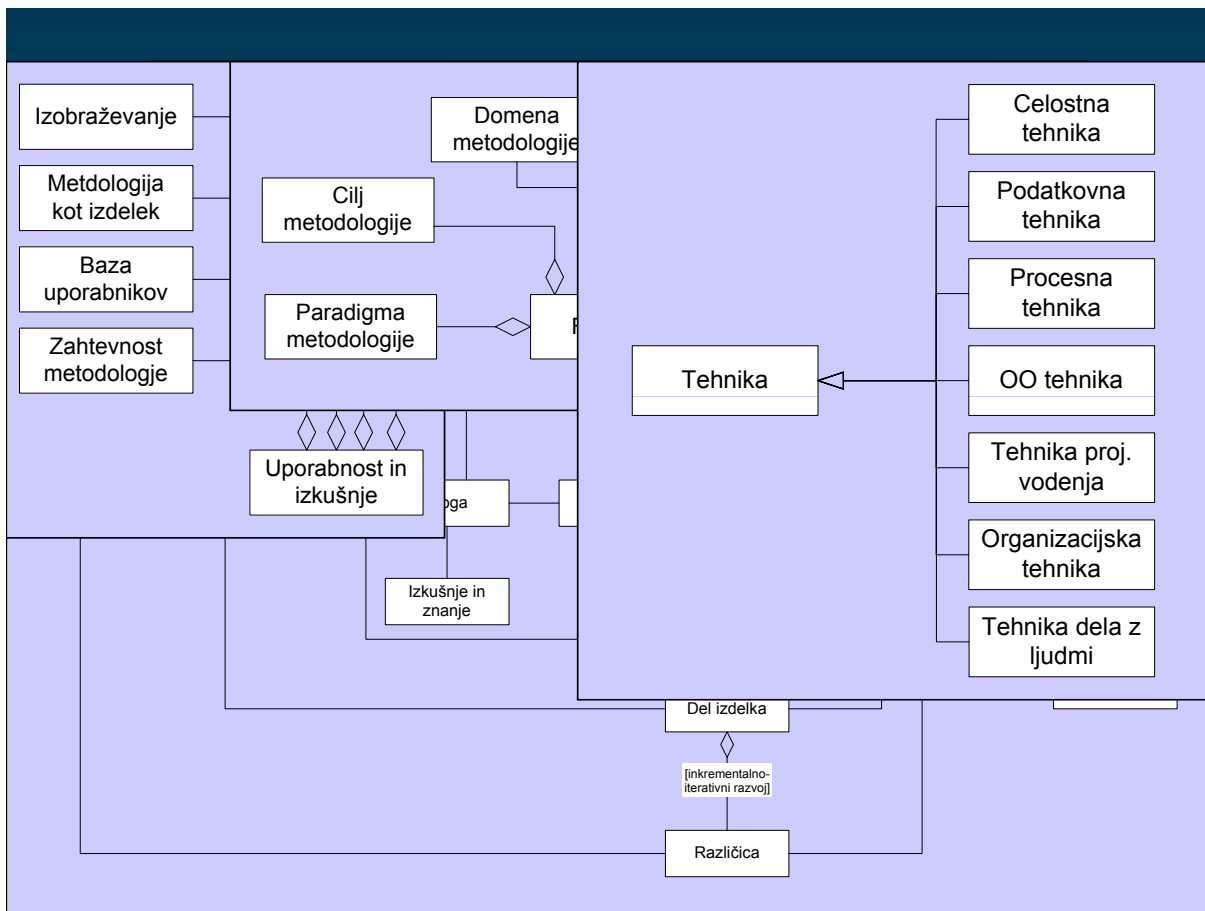
## Kaj je metodologija razvoja IS?

### ISlovar.

### ■ Iz terminološkega slovarja:

- **METODOLOGIJA**: skupek metod, postopkov in standardov, ki sestavljajo zaključeno celoto pri izvajanju inženirskih pristopov k razvoju produkta.
- **METODA**: seznam postopkov in pravil za izvedbo določene naloge.
- **METODOLOGIJA RAZVOJA IS**: postopen način razvoja informacijskega sistema, ki vključuje uporabo različnih tehnik in orodij, celovit v smislu korakov življenjskega cikla razvoja





## Zgodovina metodologij...

- **Obdobje pred pojavom metodologij (do zgodnjih 1970)**

- Ni formalnih metodologij, poudarek na reševanju tehničnih problemov, ključna vloga je programer

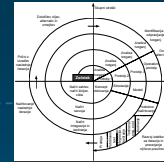


- **Zgodnje obdobje metodologij (do zgodnjih 1980)**

- SDLC, tehnike podatkovnega in procesnega modeliranja, večji poudarek na zajemu zahtev, analizi in načrtovanju
- Royce, Boehm, Codd



# Zgodovina metodologij



- **Obdobje metodologij (do sredine/konca 1990)**
  - iterativni in inkrementalni razvoj, RAD, objektna usmerjenost, večanje teže metodologij, strateško načrtovanje
  - Booch, Rumbaugh, Jacobson, Martin, Yourdon, Gamma
- **Obdobje ponovne ocenitve metodologij (danes)**
  - kritika težkih metodologij, lahke metodologije, spletne aplikacije, nove tehnologije, hitro spreminjanje zahtev, trend agilnosti
  - Beck, Ambler, Cockburn, Highsmith

Agilnost



# Metodologija kot sociološka komponenta

- **Metodologija razvoja IS ima pomembno sociološko komponento:**
  - Poleg tehnik in orodij zajema skupek dogovorov, ki v organizaciji oziroma skupini veljajo pri razvoju informacijskih rešitev;
  - Je prežeta s filozofijo in miselnostjo organizacije in njenih zaposlenih;
  - Je dinamična in odvisna od posameznikov, ki sestavljajo organizacijo.
  - Zajema vse kar počnemo, da izdelamo izdelek oziroma opravimo storitev, ki je cilj našega dela.

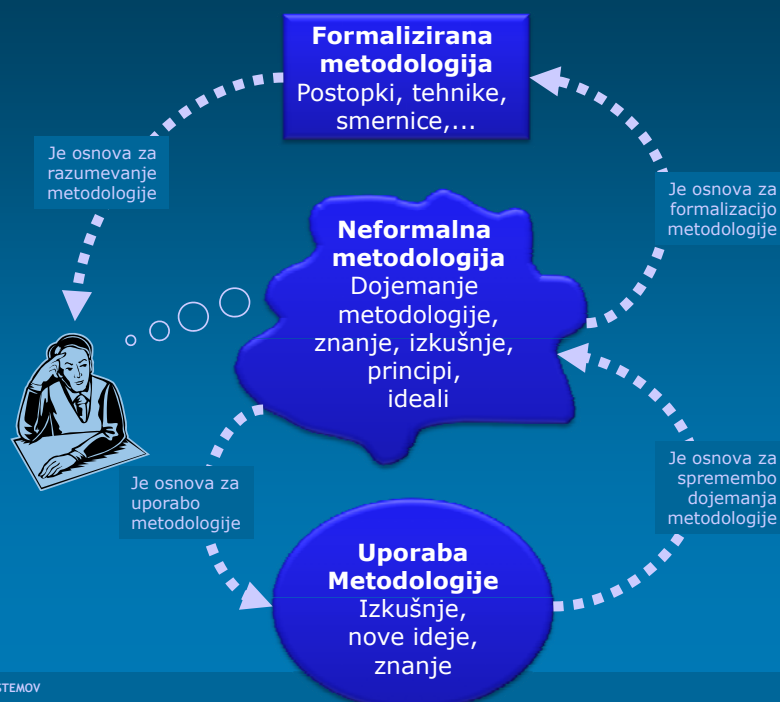


## Formalizacija metodologij

- Metodologije razvoja IS v mnogih podjetjih niso formalizirane!
  - Postopki, tehnike, orodja itd. niso dokumentirani;
  - Razvoj poteka stihijsko (na vsakem projektu je drugače; ni definirano, kako nek postopek izvedemo; kakšna orodja se uporabljajo je prepuščeno posameznikom,...
- Posledice
  - Slabša kakovost izdelka
  - Razvoj je nesledljiv in netransparenten
  - Tveganje, da izdelek ne bo pravi,
  - Težje vzdrževanje,...



## Dojemanje metodologij



## Ponudba komercialnih metodologij

- Obstaja tudi precej ponudnikov "komercialnih" metodologij.
- Nekaj primerov:
  - IE (Information Engineering) (Oracle CDM)
  - SSADM
  - Rational Unified Process
  - STRADIS
  - Agilne metodologije (XP, SCRUM, FDD,...)

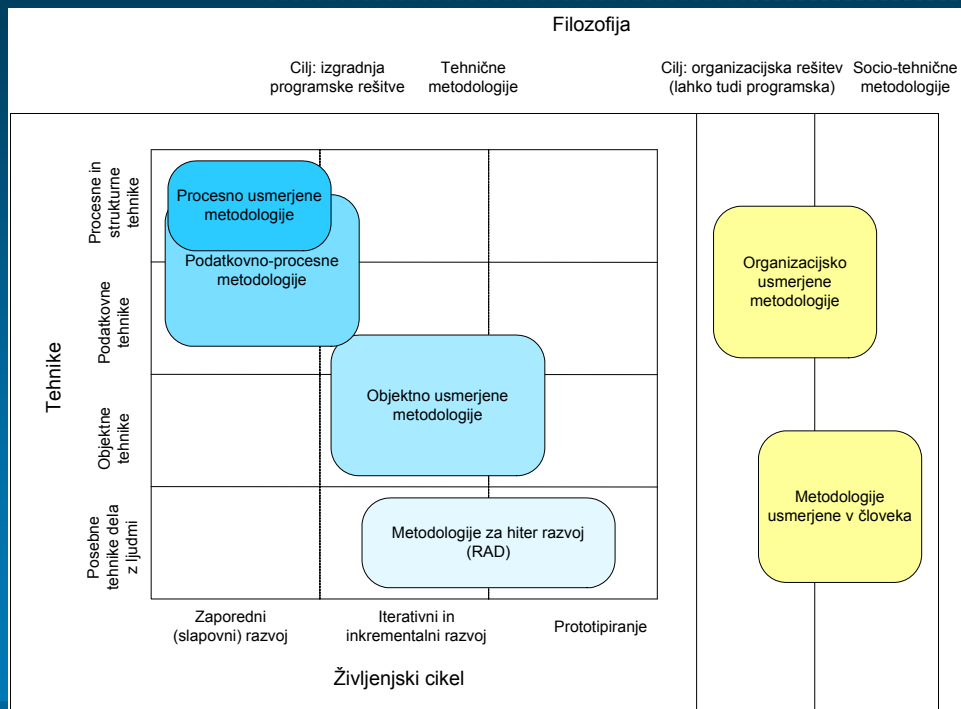


## Vrste metodologij

- Obstaja več načinov za delitev metodologij.
- Delitev metodologij nam pomaga pri izbiri ustrezne metodologije.
- Nekatere možne delitve:
  - glede na tip metodologije,
  - glede na težo metodologije,
  - glede na utežitev metodologije.

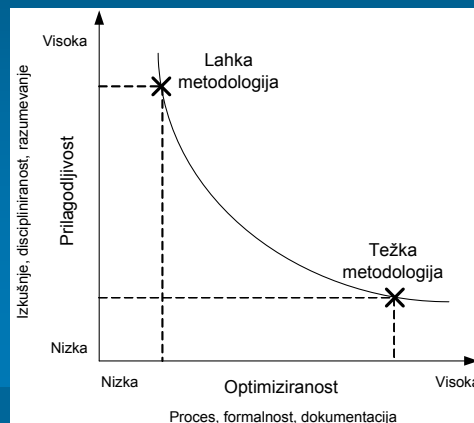


# Delitev glede na tip metodologije



# Delitev glede na težo metodologije

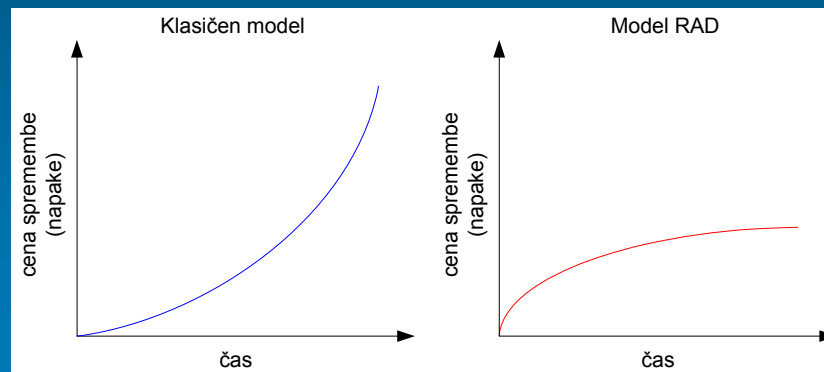
- Teža je določena z obsegom in gostoto
  - Obseg metodologije je določen s številom različnih elementov, ki jih metodologija opisuje.
  - Gostoto lahko definiramo kot zahtevan nivo podrobnosti oziroma formalnosti.





# Delitev glede na utežitev metodologije

- Tipi metodologij glede na utežitev
  - Spredaj utežene: dajejo poudarek analizi in načrtovanju
  - Zadaj utežene: dajejo poudarek kodiranju in testiranju
  - Uravnotežene: kombinacija obeh pristopov



# Pojem agilnosti

Agilnost

- Začetki: leto 2001, srečanje strokovnjakov s področja lahkih metodologij.
- Skupna izjava  
[Manifesto for Agile SW Development](#) ◆
- Na osnovi izjave je bilo izpeljanih več konkretnih priporočil za razvoj PO:
  - Načela in priporočila za agilno modeliranje (Ambler).
  - Principi agilnih metodologij (Cockburn, Highsmith).

## Osnovna načela agilnosti

Posamezniki in njihova komunikacija so pomembnejši kot sam proces in orodja.

Delujoča programska oprema je pomembnejša kot popolna dokumentacija.

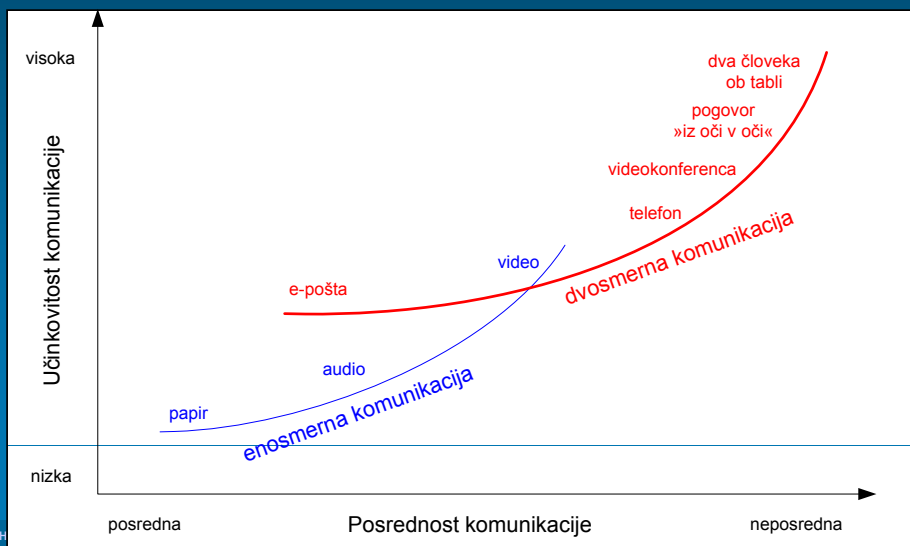
Vključevanje (sodelovanje) uporabnika je pomembnejše kot pogajanje na osnovi pogodb.

Upoštevanje sprememb je pomembnejše od sledenja planu.



## Principi agilnih metodologij...

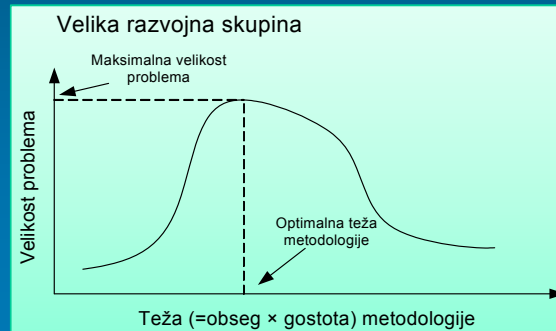
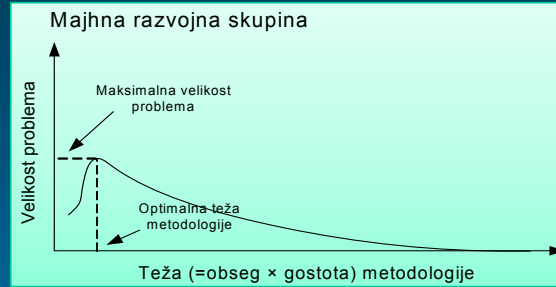
- **Neposredna komunikacija – najboljši način za izmenjavo informacij.**



## Principi agilnih metodologij...

Nepotrebna dodatna teža metodologije je draga.

Večje razvojne skupine potrebujejo težje metodologije.

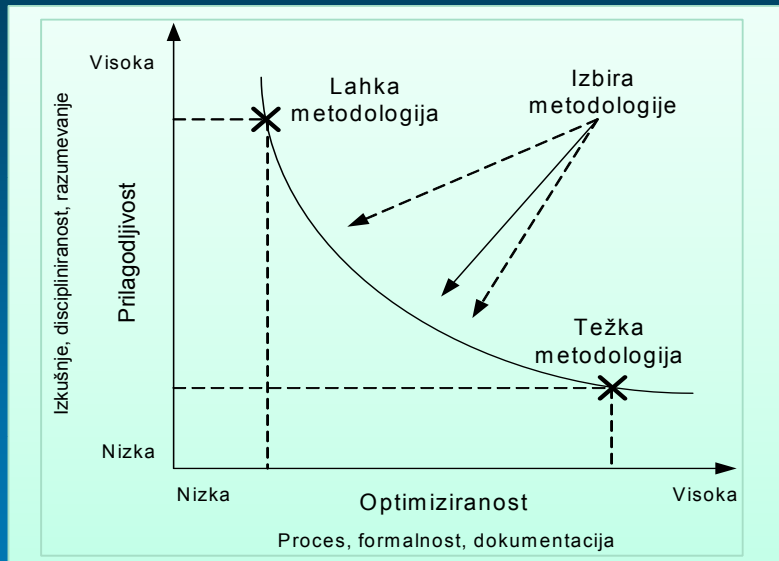


## Principi agilnih metodologij...

- Višja stopnja formalnosti metodologije je primerna za projekte z višjo stopnjo kritičnosti.
- Boljša komunikacija in več povratnih informacij zmanjšuje potrebo po vmesnih izdelkih.
- Čim večji so discipliniranost, izkušnje in znanje razvojne skupine, tem manjša je potreba po podrobno definiranem procesu, formalnosti in dokumentaciji.

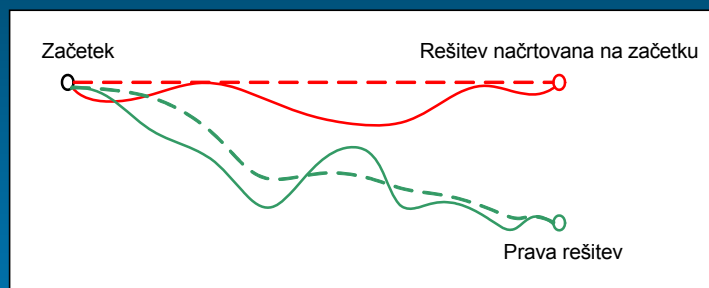


## Principi agilnih metodologij...



## Principi agilnih metodologij

- Odkloni od načrtovane rešitve nas vodijo k pravi rešitvi.



# Izbira metodologije

- Pri izbiri si pomagamo z:
  - Modeli za klasifikacijo metodologij
  - Modeli za klasifikacijo projektov
  - Izbira metodologije glede na podprte postopke in življenjski cikel
  - Izbira metodologije na podlagi agilnih principov



# Modeli za klasifikacijo metodologij...

Izbira glede na težo metodologije

- Lahke metodologije uporabimo v primeru, ko:
  - je glavni (in dokončno opredeljen) cilj razvoj programske rešitve,
  - imamo odgovorne, disciplinirane, izkušene in motivirane razvijalce, ki so seznanjeni s posebnimi tehnikami dela,
  - stranko, ki razume bistvo lahkih metodologij in je pripravljena sodelovati,
  - imamo nepredvidljive in spreminjajoče se zahteve za programsko rešitev,
  - je cilj razvoja relativno majhen sistem z nižjo stopnjo kritičnosti, ki ga je mogoče razviti z majhno razvojno ekipo,



## Modeli za klasifikacijo metodologij...

- Težke metodologije uporabimo kadar:
  - cilj ni le razvoj programskega sistema ampak tudi prenova organizacijskega sistema,
  - imamo manj izkušene razvijalce, pri katerih točno opredeljena formalna pravila nadomeščajo izkušnje in znanje,
  - naročnik zahteva visoko stopnjo formalizma (izdelovanje dokumentacije),
  - imamo relativno dobro definirane in stabilne zahteve,
  - je cilj razvoja obsežnejši sistem z višjo stopnjo kritičnosti, ki zahteva izdelavo ustreznih načrtov in dokumentacije.



## Izbira metodologije...

Izbira glede na lastnosti projekta

- Primer modela za klasifikacijo projektov

Kritičnost (odpoved sistema lahko povzroči izgubo...)		ponovljivost						
		sledljivost						
		produktivnost						
		L6	L20	L40	L100	L200	L500	L1000
življenje	L6	L20	L40	L100	L200	L500	L1000	
nenadomestljiv denar	E6	E20	E40	E100	E200	E500	E1000	
nadomestljiv denar	D6	D20	D40	D100	D200	D500	D1000	
udobje	C6	C20	C40	C100	C200	C500	C1000	
		1-6	7-20	21-40	41-100	101-200	201-500	501-1000
		Število ljudi, ki sodelujejo na projektu ± 20%						



# Izbira metodologije...

## Obseg metodologije

### ■ Obseg nekaterih komercialnih metodologij

Metodologija	STRADIS	YSM	IE	SSADM	RUP	DSDM	ETHICS	SSM	EMRIS	XP	MNova
Strategija		□	■	□	□	□		■	■		
Izvedljivost	■	■	□	■	□	■		□	□		□
Analiza	■	■	■	■	■	■	■	■	■	□	□
Logično načrtovanje	■	■	■	■	■	■	■		■	□	□
Fizično načrtovanje	■	■	■	■	■	■	□		■	□	□
Programiranje	□		□		□	□			□	■	□
Testiranje	□		■		■	□			■	■	□
Uvedba	□		□	□	■	□	□		■	□	
Evalvacija	□		□	□	□		□				
Vzdrževanje			□								

Legenda:

- Samo omenjeno
- Osnovna navodila, več je prepuščeno lastni interpretaciji
- Manj podroben opis
- Podroben opis skupaj s tehnikami

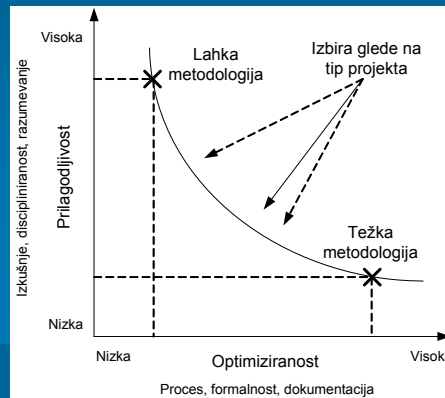
# Izbira metodologije...

## Izbira glede na podprte življenjske cikle

Kriteriji	Življenjski cikel	Slapovni razvoj	Modificiran slapovni razvoj	Iterativni razvoj	Inkrementalni razvoj	Evolucijsko prototipiranje	Kodiraj in popravi
Deluje s slabo razumljenimi zahtevami		Slabo	Srednje do odlično	Odlično	Slabo do srednje	Srednje do odlično	Slabo
Deluje s slabo razumljeno arhitekturo		Slabo	Srednje do odlično	Odlično	Slabo	Slabo	Slabo
Rezultat je zanesljiv sistem		Odlično	Odlično	Odlično	Srednje do odlično	Srednje do odlično	Slabo
Rezultat je nadgradljiv sistem		Odlično	Odlično	Odlično	Odlično	Odlično	Slabo do srednje
Upravlja s tveganji		Slabo	Srednje	Odlično	Srednje do odlično	Srednje	Slabo
Omogoča predvidljiv razvoj		Srednje	Srednje	Srednje	Odlično	Srednje	Slabo
Potrebuje malo režije		Srednje	Odlično	Srednje	Srednje	Srednje	Odlično
Dovoljuje spremembe med projektom		Slabo	Srednje	Srednje	Slabo do srednje	Srednje do odlično	Slabo do odlično
Stranki omogoča nadzor napredovanja		Slabo	Srednje	Odlično	Srednje	Odlično	Srednje

## Izbira metodologije

- Discipliniranost, izkušnje in znanje proti procesu, formalnosti in dokumentaciji
- Višja stopnja formalnosti metodologije za projekte z višjo stopnjo kritičnosti
- Večje razvojne skupine potrebujejo težje metodologije



## Uporaba metodologij v praksi

- Empirične raziskave kažejo na **nizko uporabo metodologij**.
- Ključni razlogi:
  - Nizka strukturiranost procesa razvoja (vsak projekt ima svoje lastnosti in specifičnosti; različne zahteve in pogledi uporabnikov,...)
  - Neprilagodljivost metodologij,
  - Zastarelost metodologij,
  - Sociološka neprimernost,
  - Neosveščenost uporabnikov.



## Poglavje IV

# Strukturni razvoj

- Osnovne značilnosti strukturnega pristopa
- Primer strukturne metodologije
- Strateško načrtovanje
- Analiza
- Načrtovanje
- Izvedba
- Testiranje
- Uvajanje



## Strukturni razvoj

### Osnovne značilnosti strukturnega pristopa

- Eden prvih sistematičnih pristopov k razvoju IS
- Zgleduje se po standardnih postopkih razvoja tehničnih izdelkov: aktivnosti si sledijo zaporedno.
- Izoblikoval se je konec 60 in v začetku 70 let.
  - **Razlog:** uvedba discipliniranega izvajanja analize in načrtovanja.
  - **Cilj:** zmanjšanje stroškov izgradnje in uvajanja IS.
- Pristop "iz vrha navzdol".



# Primer strukturne metodologije

## Informacijski inženiring - IE

- **Informacijski inženiring** je primer metodologije, ki opisuje razvoj IS po strukturnem pristopu.
- Nastane leta 1981, glavni avtor je **James Martin**.
- Uveljavitev v sredini 80-tih let, uporablja se še danes.
- IE je zasnovan na teoretičnih in praktičnih dosežkih 80-tih let iz metodološkega in tehnološkega vidika.



# Informacijski inženiring – IE

## Osnovne značilnosti IE

- **Osnovne značilnosti IE so:**
  - sloni na povezani množici tehnik za planiranje, analizo, načrtovanje, razvoj in vzdrževanje IS celotne združbe ali vsaj njenih glavnih delov;
  - uporablja pristop od vrha navzdol;
  - je podatkovno usmerjen;
  - podpira avtomatizacijo razvoja;
  - uveljavlja strateško planiranje;
  - povečuje produktivnost.



# Informacijski inženiring – IE

## Osnovne značilnosti IE

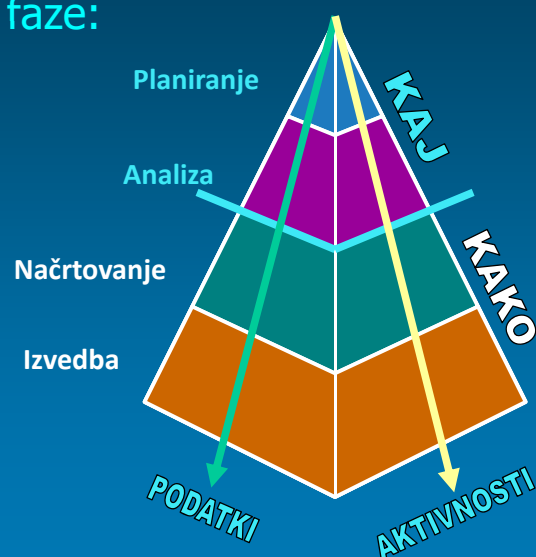
- IE predpostavlja, da so poslovni sistemi večinoma **podatkovno usmerjeni**, tehnični sistemi pa **procesno ali dogodkovno**.
- Podatki so stabilnejši od procesov in dogodkov.



# Informacijski inženiring – IE

## Glavne faze IE

- IE v grobem zajema štiri faze:
  - Strateško planiranje
  - Analiza
  - Načrtovanje
  - Izvedba
- IE posebej obravnava podatke, posebej aktivnosti



# Strateško planiranje informatike – SP



## Opredelitev SP

- Razvoj IS združbe se po IE začne s fazo strateškega planiranja informatike.

## Strateško planiranje informatike

Strateško planiranje informatike je proces izoblikovanja informacijskega sistema, ki organizaciji omogoča uresničitev njenih ciljev in ji s tem posredno zagotavlja konkurenčno prednost.

Fidler in Rogerson, 1996



# Strateško planiranje informatike – SP

## Cilji SP

- Cilji strateškega planiranja so:
  - Povezati razvoj IS s poslovno strategijo organizacije.
  - Izboljšati komunikacijo med vodstvom in informatiki.
  - Načrtovati pretok informacij in procesov.
  - Zmanjšati stroške in čas, potreben za razvoj aplikacij.
  - Predlagati optimalno zaporedje nadaljnjih korakov pri planiranju in razvoju IS.
  - Pripraviti izhodišča za nadaljnje korake informatizacije.
  - Zagotoviti uporabo standardov za enotne tehnološke rešitve.
  - Pokazati na organizacijske probleme pri uvajanju informacijske podpore in predlagati organizacijske rešitve za doseg racionalnejše uporabe informacijske podpore.



# Strateško planiranje informatike – SP

## Problemi brez SP

- Problemi, če gre za investicije v informatiko na osnovi sprotnih potreb:
  - Investiranje v sisteme, ki ne podpirajo poslovnih usmeritev.
  - Sistemi niso integrirani (podvojitev naporov in podatkov).
  - Ni sredstva za določitev prioritete projekta, plani se pogosto spreminjajo.
  - Problemi z obvladovanjem podatkov: niso dostopni, so neskladni, netočni ali nepravočasni.
  - Nezdostne infrastrukturne investicije.



# Strateško planiranje informatike – SP

## Problemi brez SP

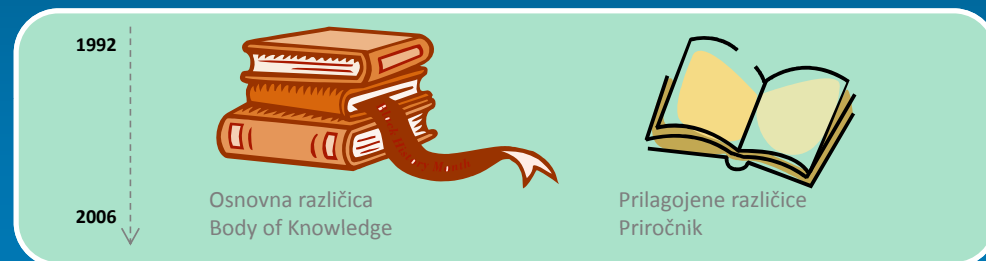
- Problemi (nadaljevanje):
  - Vsi projekti so ovrednoteni le na finančni bazi.
  - Problemi glede IT investicij povzročajo konflikte med različnimi oddelki znotraj organizacije.
  - Nerazumevanje med uporabniki in informatiki vodi v konflikte in nezadovoljstvo.
  - Sistemi imajo večinoma krajšo življenjsko dobo, pogosteje je potreben ponoven razvoj, kar povzroča večje stroške.



# Strateško planiranje informatike – SP

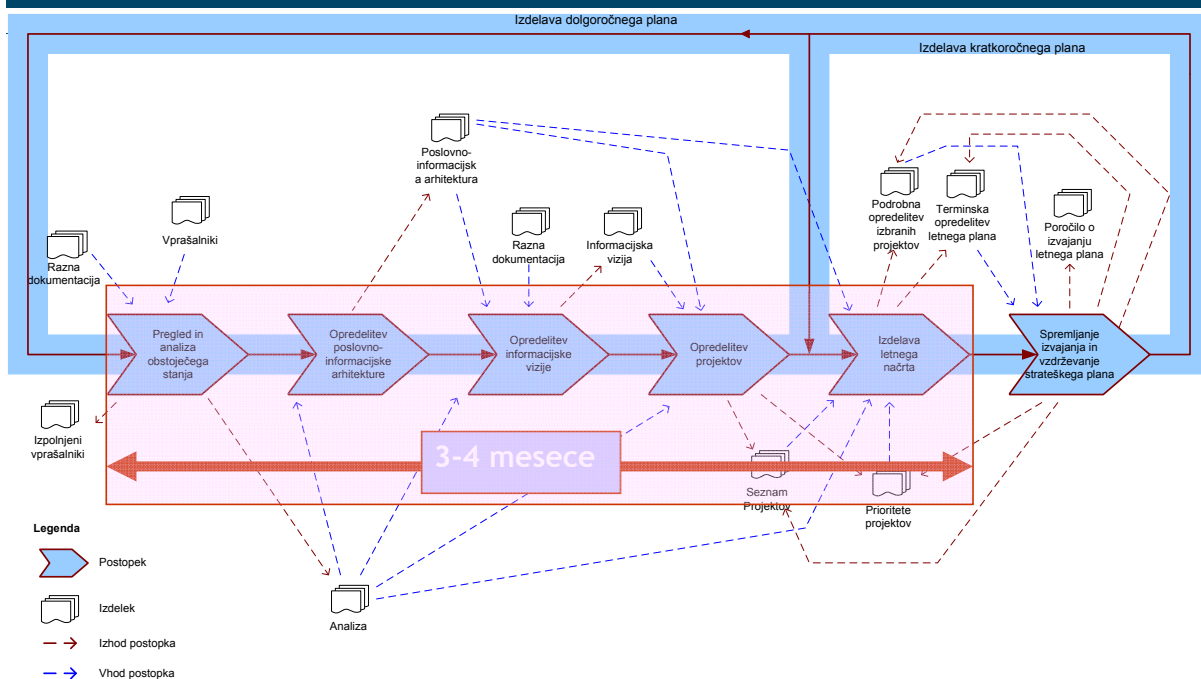
## Metodologija IS

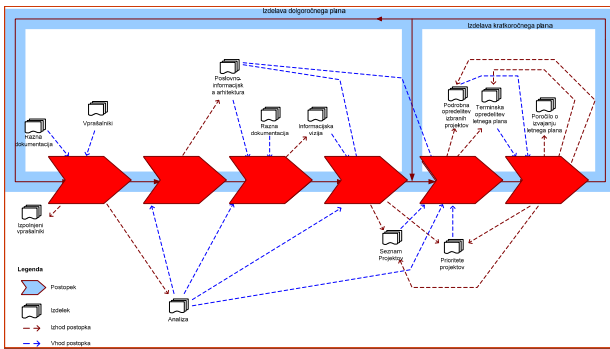
- IE priporoča postopek za izdelavo.
- V LINF razvitih več različic.
- Osnovna različica temelji na metodologiji IE.
- Prilagojene različice za potrebe posameznih podjetij.



# Strateško planiranje informatike – SP

## Shema procesa izdelave SP





## Informatike – SP

Pregled in  
analiza  
obstoječega  
stanja

## Strateški plan razvoja informatike

Trajanje izdelave SP in sodelujoči

- Izdelava strateškega plana traja približno od 3 do 6 mesecev.
- Pri izdelavi sodelujejo:
  - Zunanji svetovalci
  - Metodologi (informatike izven organizacije)
  - Ključni uporabniki
  - Člani vodstvene skupine organizacije
- Strateški plan je potrebno osveževati!

# Razvoj po strukturnem pristopu

## Osnovne značilnosti strukturnega pristopa k razvoju

- **Strukturni pristop k razvoju**
  - temelji na strukturni izvedbi analize in načrtovanja.
  - podatki se obravnavajo ločeno od aktivnosti postopkov.
  - ključen element pri strukturnem modelu je podatkovna baza, ki predstavlja strukturo, okrog katere se razvije programske module.
- **Strukturni razvoj se z uveljavitvijo objektnih programskih jezikov ukinja.**
- **Danes se uporablja hibriden pristop:** temelji na objektni filozofiji, vendar podatkovna baza še vedno ohranja ključen pomen.



# Razvoj po strukturnem pristopu

## Ključni postopki strukturnega razvoja

- **Strukturni pristop k razvoju zajema več postopkov:**
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Vzdrževanje.







## Razvoj po strukturnem pristopu

Kje smo?

- **Postopki strukturnega pristopa:**

- ➔ Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Vzdrževanje.



## Zajem in specifikacija zahtev

Opredelitev in namen

- **Zajem in specifikacija zahtev je ena pomembnejših aktivnosti razvoja oziroma nakupa IR.**
- **Osnovni namen** zajema in specifikacije zahtev je **opredeliti želeno IR na način, ki bo omogočal:**
  - Pri nakupu IR izbirati med obstoječimi rešitvami,
  - Pri razvoju IR opredeliti osnovno funkcionalnost ter tehnološke in druge nefunkcionalne zahteve in omejitve za izgradnjo želene IR.



## Zajem in specifikacija zahtev

Končni izdelek 📁

- Rezultat zajema in specifikacije zahtev je dokument, kjer so zabeležene vse funkcionalne in nefunkcionalne zahteve v zvezi z želeno IR.
- V nadaljnjih korakih se dokument lahko uporablja:
  - kot vhod v postopek analize,
  - pri pripravi razpisne dokumentacije za nakup IR oziroma izbiro zunanjega razvijalca,
  - kot priloga k pogodbi med naročnikom in izvajalcem sistema.



## Zajem in specifikacija zahtev

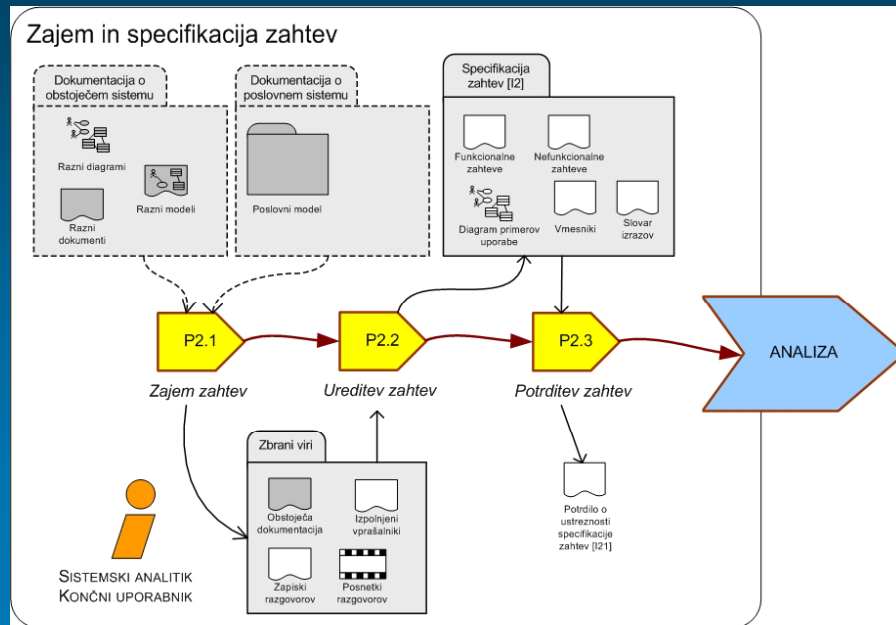
Vloge in koraki 👤 ⚡

- Zajem zahtev izvede sistemski analitik ob tesnem sodelovanju s poznavalci problemske domene oziroma ključnimi uporabniki.
- Osnovni koraki zajema:
  - Zajem zahtev,
  - Ureditev zahtev in
  - Potrditev zahtev.



# Zajem in specifikacija zahtev

Postopek zajema in specifikacije zahtev ▶



# Razvoj po strukturnem pristopu

Kje smo?

## Postopki strukturnega pristopa:

➔ Zajem in specifikacija zahtev;

➔ Zajem zahtev,

- Ureditev zahtev in
- Potrditev zahtev

- Analiza;
- Načrtovanje;
- Izvedba;
- Testiranje;
- Uvajanje
- Izobraževanje.

# Zajem in specifikacija zahtev

## Zajem zahtev

- **Obstajajo različne tehnike zajema zahtev:**
  - Razgovori
  - Vprašalniki
  - Opazovanje pri delu
  - Analiza obstoječega sistema
  - Skupinsko načrtovanje aplikacij



# Zajem in specifikacija zahtev

## Zajem zahtev

- **Splošni napotki za uspešno izvedbo zajema zahtev:**
  - Analitik mora biti objektiven,
  - Analitik mora upoštevati vse možnosti v okviru nekega problema,
  - Analitik posveča pozornost podrobnostim,
  - Analitik mora strmeti k novim in boljšim rešitvam,
  - Analitik ne daje obljub uporabnikom,
  - Analitik nima zadržkov pri zajemanju zahtev.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:

- ➔ Zajem in specifikacija zahtev;

- Zajem zahtev,

- ➔ Ureditev zahtev in

- Potrditev zahtev

- Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



## Zajem in specifikacija zahtev

Ureditev zahtev

- V aktivnosti **ureditev zahtev** skušamo naše razumevanje želene IR opredeliti še v okviru dokumenta, s katerim bodo zahteve IR podane bolj formalno in jedrnato.
- Izdelek, ki nastane, imenujemo **specifikacija zahtev**.
- Specifikacija zahtev lahko služi kot temeljna podlaga pri dogovarjanju med naročnikom in izvajalcem.



# Zajem in specifikacija zahtev



## Specifikacija zahtev

- Specifikacija zahtev ima navadno naslednjo strukturo:

1. Kratek opis namena IR ali njenega podsistema
2. Opis funkcionalnih zahtev
3. Opis nefunkcionalnih zahtev
4. Opis vmesnikov
5. Slovar izrazov



# Zajem in specifikacija zahtev

## Specifikacija zahtev

### Funkcionalne zahteve

**Funkcionalne zahteve so zahteve, ki se nanašajo na želeno funkcionalnost sistema.**

#### **Odjava iz izpitnega roka**

Sistem naj študentom omogoča odjavo iz izpitnega roka.

Poslovna pravila:

- Študent se ne more odjaviti iz izpitnega roka, če je do roka še manj kot tri dni.
- ...



# Zajem in specifikacija zahtev

## Specifikacija zahtev

### Nefunkcionalne zahteve

Nefunkcionalne zahteve so zahteve, ki se nanašajo na tehnične in druge nevsebinske zahteve sistema.

- Sistem naj bo narejen v tri-nivojski arhitekturi z lahkim odjemalcem.
- Podatki naj se hranijo v podatkovni bazi Oracle.
- Za avtentikacijo nej se uporabi digitalno potrdilo.
- Izdajatelj digitalnega potrdila je lahko CVI RS, NLB, PS.
- ...



# Razvoj po strukturnem pristopu



Kje smo?

## Postopki strukturnega pristopa:

➔ Zajem in specifikacija zahtev;

- Zajem zahtev,
- Ureditev zahtev in

➔ Potrditev zahtev

- Analiza;
- Načrtovanje;
- Izvedba;
- Testiranje;
- Uvajanje
- Izobraževanje.





## Zajem in specifikacija zahtev

### Potrditev zahtev

- Namen aktivnosti **potrditev zahtev** je predstavitev specifikacije zahtev naročniku in pridobitev soglasja o tem, da so zajete zahteve res to, kar si naročnik želi.



## Razvoj po strukturnem pristopu

### Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Vzdrževanje.





# Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Vzdrževanje.



## Analiza

Oprelitev in namen

- Glavni namen analize je izdelati razumljiv opis realnega sveta oziroma poslovnega okolja, na katerega se nanaša razvoj IS.
- Izdelamo model sistema, ki na formalen način opredeli potrebne podatkovne strukture in funkcije, ki te podatke uporabljajo.



# Analiza

## Opredelitev in namen

- Analiza daje odgovor na vprašanje, **KAJ** naj IS podpira. Kaj se izvaja v poslovnih funkcijah in kakšne podatke te rabijo?
- Analiza služi kot:
  - sredstvo za definicijo zahtev,
  - osnova za dogovor med naročnikom in izvajalcem
  - osnova za kasnejše faze razvoja.



# Analiza

## Končni izdelek 📁

- Rezultat analize je model sistema.
- Na osnovi modelov se v nadaljnjih korakih izdelava podatkovna baza ter potrebni programski moduli.
- Poleg modela sistema v fazi analize izdelamo tudi predlog tehnične arhitekture sistema ter opsijsko prototipe komponent uporabniškega vmesnika.
- Med izdelke analize sodi tudi strategija testiranja.




# Analiza

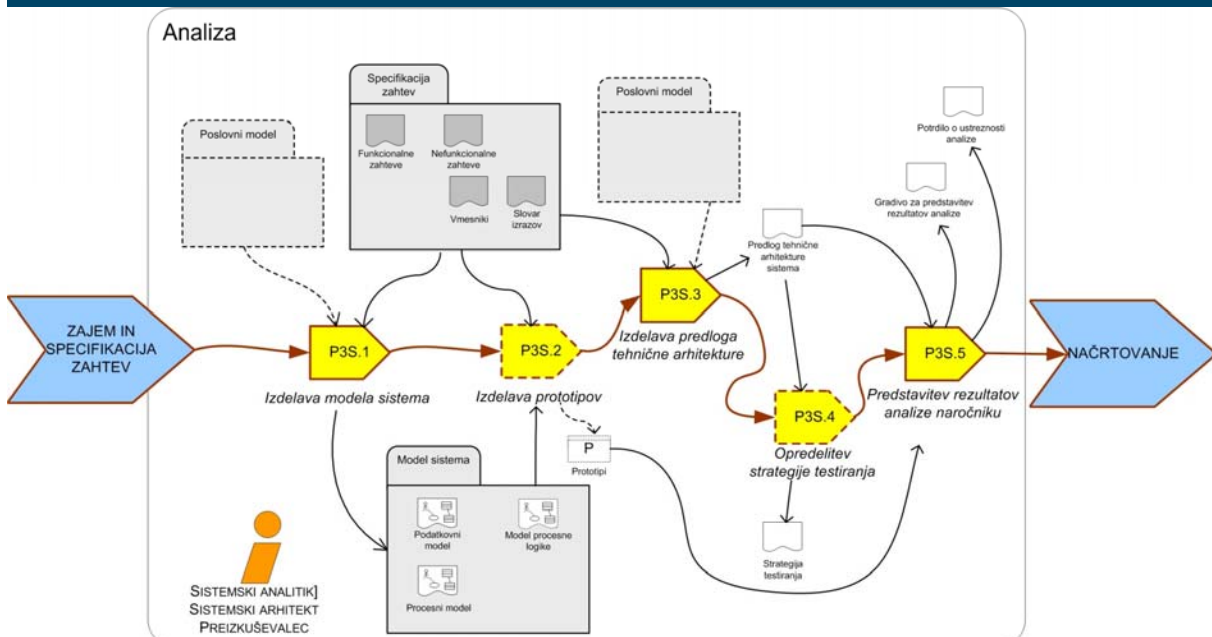
Vloge in koraki  

- Postopek analize izvajajo **systemski analitik, systemski arhitekt in ključni uporabniki**.
- Postopek analize tipično zajema naslednje aktivnosti:
  - Izdelava modela sistema;
  - Izdelava prototipov;
  - Izdelava predloga tehnične arhitekture sistema;
  - Opredelitev strategije testiranja;
  - Predstavitve rezultatov analize naročniku.



# Analiza

Shema postopka 





# Razvoj po strukturnem pristopu

Kje smo?

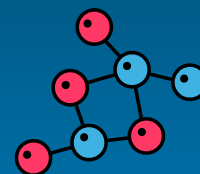
- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza
    - ➔ Izdelava modela sistema;
      - Izdelava prototipov;
      - Izdelava predloga arhitekture sistema;
      - Opredelitev strategije testiranja;
      - Predstavitve rezultatov analize naročniku.;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



# Analiza

Splošno o modeliranju

- Modeliranje je uveljavljena inženirska tehnika na mnogih področjih:
  - Gradbeništvo,
  - Strojništvo,
  - Kemija,
  - Ekonomija,
  - Sociologija,
  - Računalništvo...



# Analiza

## Splošno o modeliranju

- Modele razvijamo zato, da bi sisteme boljje razumeli.



# Analiza

## Splošno o modeliranju

- **Model** je poenostavitev realnosti, pri čemer je abstrakcija realnosti poljubno natančna.
- Pomembno je, da model prikazuje pomembne elemente in izpušča tiste, ki nas ne zanimajo.



# Analiza

## Splošno o modeliranju

- **Modeliranje prinaša naslednje bistvene prednosti:**
  - Omogoča vizualizacijo sistema,
  - Prikazuje tako statične kot dinamične lastnosti sistema,
  - Predstavlja šablono za nadaljnjo gradnjo sistema,
  - Dokumentira sprejete odločitve.



# Analiza

## Splošno o modeliranju

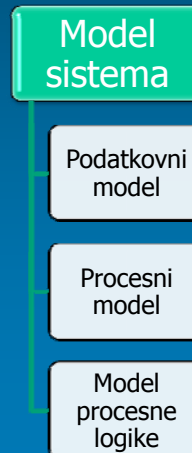
- **Izbira modelov**
  - Za modeliranje sistema lahko izberemo različne modele.
  - Izbira modelov določa, kako bomo pristopili k reševanju problema ter kako oblikovali rešitev.
  - Modeli morajo podpirati izražanje na različnih ravneh natančnosti.
  - Najboljši modeli so tesno povezani z realnostjo.
  - En sam model nikoli ni dovolj. Sistem je potrebno modelirati iz različnih vidikov. Najboljši pristop je izbira nekaj modelov, ki kar najboljše pokrijejo najpomembnejše vidike sistema.
  - Metodologije razvoja IS predlagajo različne modele.



# Analiza

## Analiza po strukturnem pristopu

- **Model, ki ga izdelamo v fazi analize po strukturnem pristopu, se v grobem deli na:**
  - Podatkovni model,
  - Procesni model in
  - Model procesne logike.



# Analiza – Izdelava modela sistema

## Izdelava modela sistema po strukturnem pristopu

- **Podatkovni model:**
  - prikazuje sistem s podatkovnega vidika tako, da opisuje podatkovne strukture, ki so potrebne za delovanje sistema. Poleg podatkovnih struktur zajema tudi vse povezave med njimi.
- **Procesni model:**
  - prikazuje sistem z vidika aktivnosti ali procesov, ki se v sistemu izvajajo. Definirani so tokovi podatkov med procesi.
- **Model procesne logike:**
  - natančneje definira procese, definirane v procesnem modelu.

# Analiza – Izdelava modela sistema

## Strukturne diagramске tehnike

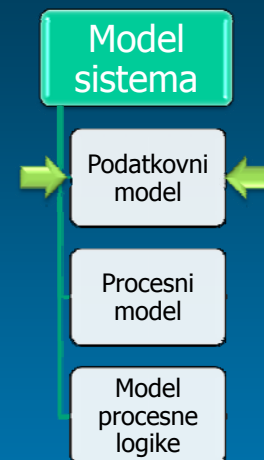
- Za predstavitev posameznih modelov sistema uporabljamo formalne, semi-formalne in tudi neformalne tehnike.
  - Podatkovni model: diagram entiteta-razmerje
  - Procesni model: procesni diagram, diagram podatkovnih tokov, funkcionalna razgradnja
  - Model procesne logike: naravni jezik, strukturiran jezik, odločitvene tabele, odločitveni grafi, diagrami prehajanja stanj



# Analiza – Izdelava modela sistema

## Podatkovni model

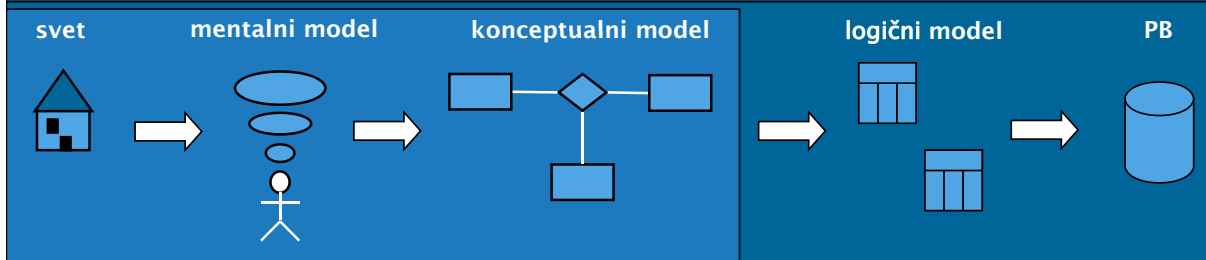
- **Podatkovni model:**
  - Je eden izmed najpomembnejših izdelkov faze analize in predstavlja vse podatkovne kategorije, za katere na nekem delovnem področju obstaja potreba, da se o njih podatki spremljajo, obdelujejo in hranijo.
  - V analizi izdelamo **konceptualni podatkovni model**.
  - Za izdelavo uporabljamo diagramsko tehniko **entiteta-razmerje**.





# Analiza – Izdelava modela sistema

## Podatkovni model



# Analiza – Izdelava modela sist.

## Podatkovni model – primer

### ▪ Konceptualni podatkovni model

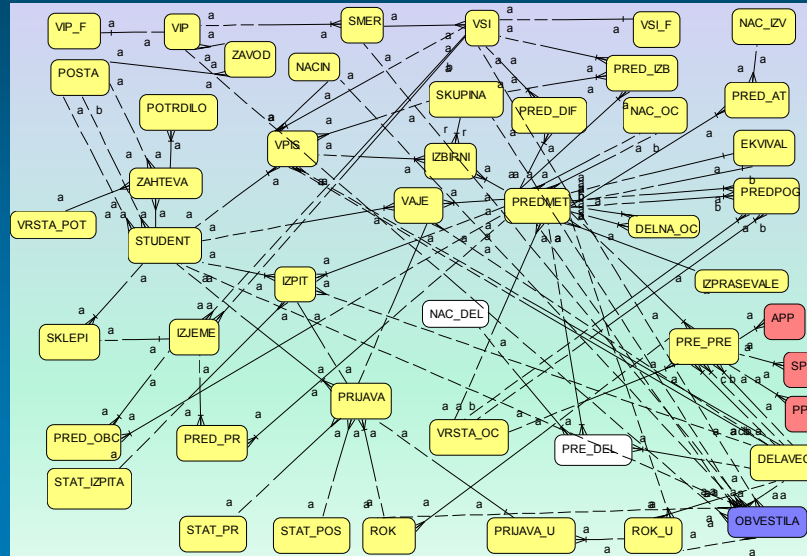


V konceptualnem modelu lahko nastopajo tudi **sestavljeni** in **večrednostni** atributi!

# Analiza – Izdelava modela sistema

## Podatkovni model – primer

- Primer nerazumljivega podatkovnega modela



- 190 -

# Analiza – Izdelava modela sistema

## Podatkovni model – primer

- Entitetne tipe je potrebno dokumentirati
- Primer dokumentacije:

Naziv entitetnega tipa	Opis	Sinonim	Število entitet
Delavec	Predstavlja pedagoškega delavca, ki je nosilec enega ali več predmetov	Pedagoški delavec	Vsaka katedra ima enega ali več pedagoških delavcev. Niso vsi delavci nosilci predmetov.
Rok	Predstavlja datum, na katerega je za nek predmet in določeno ciljno skupino (letnik, smer,...) razpisan izpitni rok.	Rok, pisni izpit, kolokvij	Na leto se razpiše okrog 300 pisnih izpitov. Vsak predmet mora imeti vsaj tri roke letno
...			

# Analiza – Izdelava modela sistema

## Podatkovni model – metoda načrtovanja

- Možni koraki konceptualnega načrtovanja:
  - K 1.1: Identificiraj entitetne tipe
  - K 1.2: Identificiraj povezave
  - K 1.3: Identificiraj in z entitetnimi tipi poveži attribute
  - K 1.4: Atributom določi domene
  - K 1.5: Določi kandidate za ključe; izmed kandidatov izberi primarni ključ
  - K 1.6: Po potrebi uporabi elemente razširjenega diagrama entiteta – razmerje
  - K 1.7: Preveri, če v modelu obstajajo odvečni elementi
  - K 1.8: Preveri, če model "zdrži" transakcije
  - K 1.9: Preveri model z uporabnikom

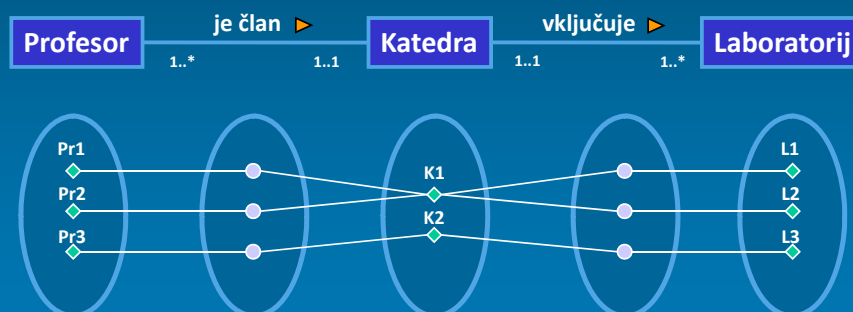
# Analiza – Izdelava modela sistema

## Podatkovni model – metoda načrtovanja – identifikacija povezav

- V postopku identifikacije povezav smo pazljivi na dvoumne in nepopolne povezave.

**K 1.2**

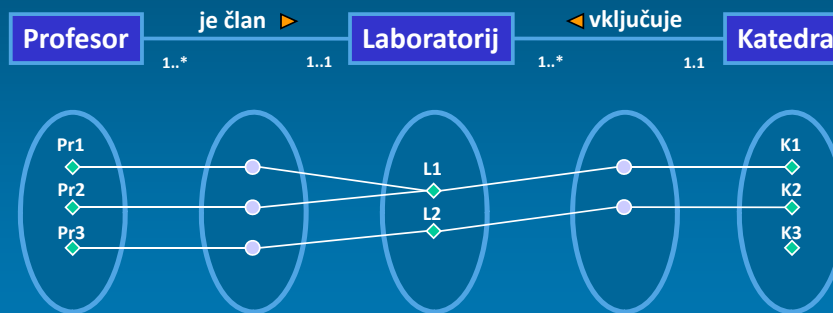
### Primer dvoumne povezave



# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – identifikacija povezav

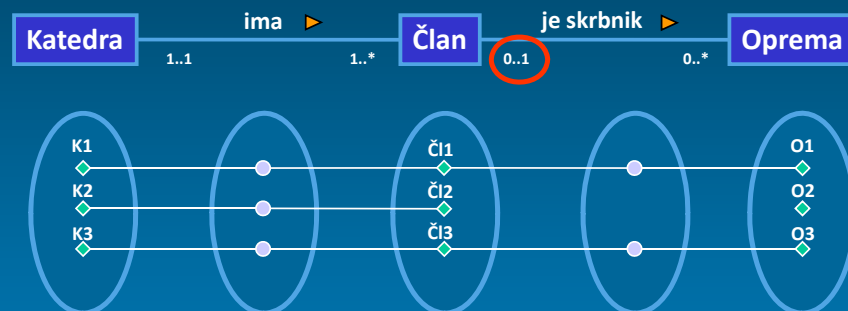
- Dvournno povezavo odpravimo z restrukturiranjem modela



# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – Identifikacija povezav

## Primer nepopolne povezave

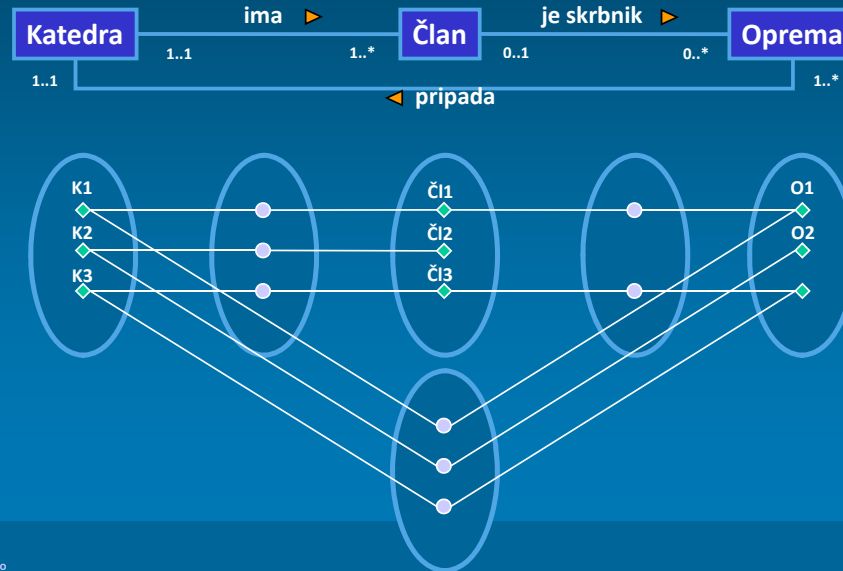


Kateri katedri pripada oprema O2?

# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – identifikacija povezav

- Odpravimo z prestrukturiranjem modela.



PODATKOVNE BAZE - II del  
3. Letnik UNI, Informatika  
©Laboratorij za informatiko

# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – odvečne povezave

- Identifikacija odvečnih povezav **K 1.7**
  - Povezava je odvečna, če je možno priti do iste informacije prek drugih povezav!
  - Izdelati želimo minimalen podatkovni model → odvečne povezave zato odstranimo.
  - Zgolj pregledovanje poti med entitetnimi tipi ne zadošča (povezave imajo lahko različen pomen)

PODATKOVNE BAZE - II del  
3. Letnik UNI, Informatika  
©Laboratorij za informatiko

# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – odvečne povezave

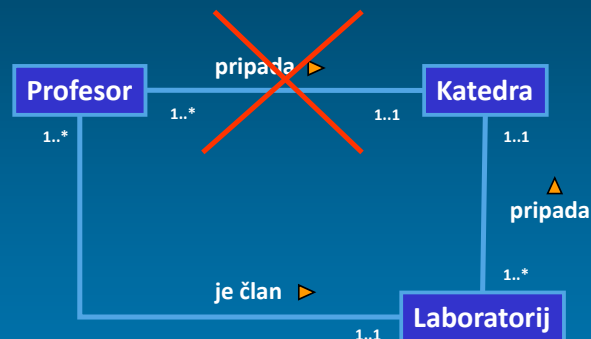
- Ali je kakšna povezava odveč?



# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – odvečne povezave

- Ali je kakšna povezava odveč?



## Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – preverjanje transakcij

- Preveriti moramo, če model podpira vse zahtevane transakcije. **K 1.8**
  - Transakcije izvajamo ročno
  - Če neke transakcije ne uspemo izvesti, je model pomanjkljiv (manjka bodisi entitetni tip, povezava ali atribut)
- Možna dva pristopa:
  - Preverjanje prek opisa transakcij
  - Preverjanje prek transakcijskih poti



## Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – preverjanje prek opisa transakcij

- Preverjanje prek opisa transakcij
  - Vsako transakcijo opišemo;
  - Preverimo, če model zajema vse entitetne tipe, povezave in attribute, ki jih transakcija potrebuje.



# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – preverjanje prek opisa transakcij

- Primer opisa transakcijskih zahtev
  - Vnos podatkov:
    - Vnesi podatke o študentih (npr. 24010637, Monika Jemec,...)
    - Vnesi podatke o predmetih (npr. 70029, Razvoj IS, Letni,...)
    - ...
  - Urejanje in brisanje podatkov:
    - Uredi/briši podatke o študentu
    - Uredi/briši podatke o predmetih
    - ...
  - Poizvedbe
    - Izpiši vse študente, ki so se vpisali v določen letnik, določene smeri, določenega programa
    - Izpiši vse predmete, ki jih je opravil določen študent
    - ...



# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – preverjanje prek transakcijskih poti

- Preverjanje prek transakcijskih poti
  - Transakcije preverimo na modelu – pot transakcije narišemo
  - Pristop načrtovalcu omogoča:
    - Da identificira **pomanjkljivosti modela** (če pot za neko transakcijo ni možna)
    - Da identificira dele modela, ki so **transakcijsko kritični**
    - Da odkrije **odvečne dele modela** (deli, ki jih ne potrebuje nobena transakcija)





# Analiza – Izdelava modela sistema

Podatkovni model – metoda načrtovanja – preverjanje prek transakcijskih poti

- Izpiši vse predmete, ki jih je opravil določen študent
- Izpiši vse študente, ki so se vpisali v določen letnik, določene smeri, določenega programa

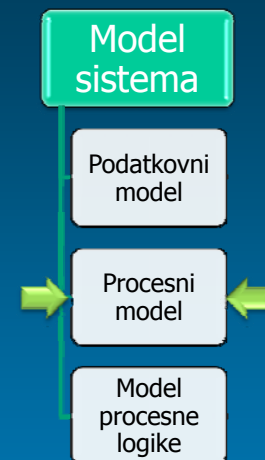


# Analiza – Izdelava modela sistema

Izdelava modela sistema – procesni model

## Procesni model:

- prikazuje sistem z vidika aktivnosti ali procesov, ki se v sistemu izvajajo. V procesnem modelu lahko nastopajo tudi podatkovne strukture, ki jih procesi potrebujejo pri svojem delovanju.
- Za izdelavo procesnega modela uporabimo dve diagramski tehniki: **diagram razgradnje** ter **diagram podatkovnih tokov**. Uporabimo lahko tudi **procesni diagram**.



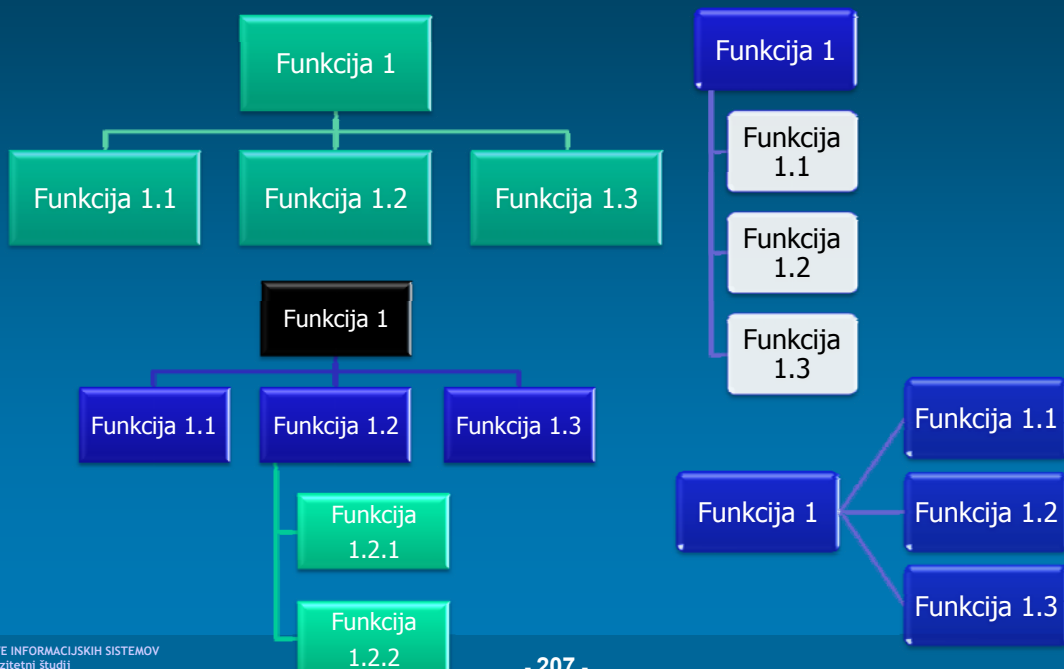
# Analiza – Izdelava modela sistema

Procesni model – diagram funkcionalne razgradnje – opredelitev

- Z diagramom funkcionalne razgradnje prikažemo hierarhijo funkcij, ki jih želimo s sistemom podpreti.
- Hierarhijo funkcij lahko prikažemo na različne načine, najbolj običajno kot navpično hierarhijo.

# Analiza – Izdelava modela sistema

Procesni model – diagram funkcionalne razgradnje



# Analiza – Izdelava modela sistema

Procesni model – diagram funkcionalne razgradnje – smernice

- Smernice za izdelavo diagramov funkcionalne razgradnje:
  - Število nivojev in število enot na enem nivoju običajno ni omejeno, čeprav velja priporočilo, naj ima vsak element največ devet podrejenih elementov.
  - Za vsako enoto velja, da ima nič, eno ali več podrejenih enot (vej) in da vedno pripada natanko eni nadrejeni enoti na prvem višjem nivoju.
  - Enote na istem nivoju razporedimo od leve proti desni po neki sekvenčni karakteristiki, ki jo natančno definiramo in k diagramu dokumentiramo.



# Analiza – Izdelava modela sistema

Procesni model – diagram funkcionalne razgradnje - primer



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – opredelitev

- **Diagrame podatkovnih tokov (DFD)** uporabimo za prikaz okolja, v katerem bo sistem deloval, ter za prikaz odvisnosti med procesi, ki jih bo sistem podprl.
- DFD združuje podatkovni in procesni pogled na obravnavano področje.
- DFD je uvedel **Tom DeMarco** leta 1978. Od takrat nastalo več različic DFD tehnike. Razlikujejo se predvsem v notaciji.

DFD – Data Flow Diagram



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – osnovni gradniki

- DFD sodi med enostavnejše diagramске tehnike.
- **Osnovni gradniki DFD so:**
  - Proces
  - Tok podatkov
  - Podatkovno skladišče
  - Zunanja entiteta



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – proces

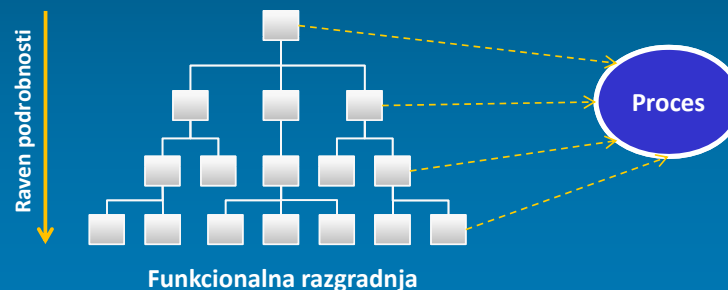
- Proces predstavlja množico aktivnosti, ki vhodne podatke pretvorijo v izhodne.



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – proces

- Proces je generičen pojem in lahko predstavlja dogajanje na različnih ravneh (funkcija, proces, pod-proces, naloga, aktivnost ipd.)



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – proces

- Vsakemu procesu je dodeljen **naziv** in **številčna oznaka**, ki se v diagramu vpišeta v grafični simbol procesa.
- Za naziv procesa običajno uporabimo glagol, glagolski samostalnik ali zaporedje besed, ki opisujejo vrsto dejavnosti.
- Številčna oznaka enolično določa proces.



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – tok podatkov

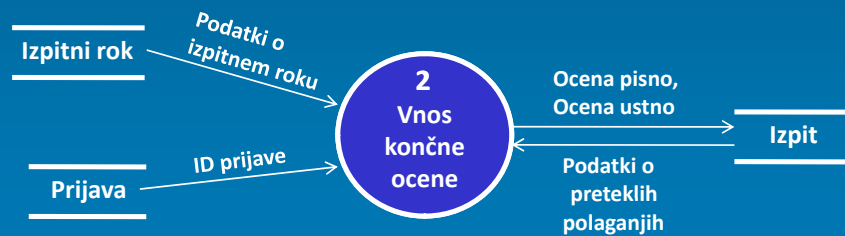
- **Tok podatkov** predstavlja množico vhodnih ali izhodnih podatkov, ki imajo enolično definirano vsebino in strukturo.
- Podatki lahko predstavljajo:
  - elementarne podatke (npr. ime, priimek, vpisna številka,...)
  - dokumente (vpisni list, kartotečni list,...)
  - elektronske dokumente (elektronski indeks,...)



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – tok podatkov

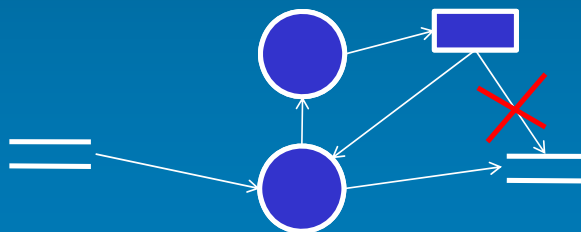
- Vsakemu toku podatkov določimo **naziv**, ki pove kaj tok prenaša.
- Nazivi so samostalniki, običajno v ednini, ali pa kombinacija samostalnika in pridevnika



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – tok podatkov

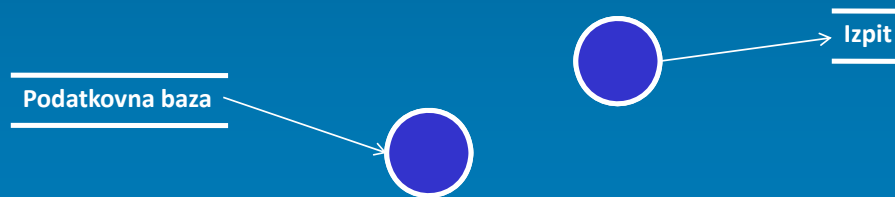
- Podatkovni tok se lahko giblje:
  - iz zunanje entitete v proces ali iz procesa k zunanji entiteti,
  - iz procesa v drug proces in
  - iz procesa v skladišče podatkov ali obratno.
- Podatkovni tok ne more povezovati zunanje entitete s skladiščem podatkov!



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – podatkovna shramba

- **Podatkovna shramba** predstavlja prostor, kamor procesi shranjujejo podatke za druge procese ali kasnejšo uporabo.
- Podatkovna shramba je lahko **enostavna** (npr. zajema le elementarne podatke) ali **kompleksna** (npr. predstavlja celo zbirko podatkov).



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – podatkovna shramba

- V fazi analize s podatkovno shrambo opišemo **logične sklope** podatkov neodvisno od bodoče fizične organizacija podatkov.
- **Naziv** podatkovne shrambe je največkrat enak nazivu vhodnih podatkovnih tokov (skladišče je pravzaprav podatkovni tok v mirovanju)

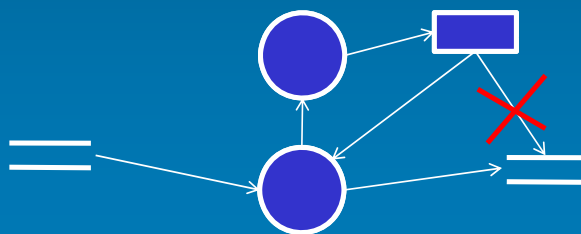




## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – podatkovna shramba

- V vsako shrambo mora pisati vsaj en proces, sicer je shramba odveč!
- Velja logično pravilo, da iz shrambe ni mogoče brati podatkov, ki niso bili vanj zapisani.
- Shramba je interna podatkovna struktura, zato dostop do nje ni omogočen zunanjim entitetam!



## Analiza – Izdelava modela sistema

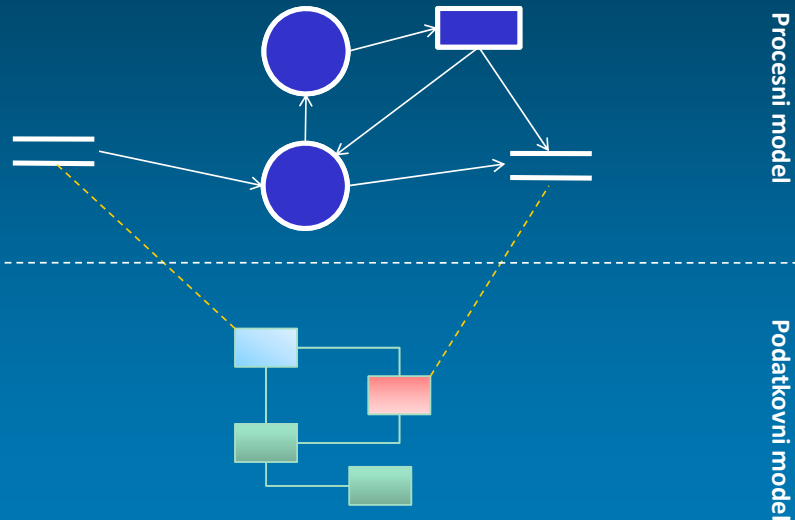
Procesni model – diagram podatkovnih tokov – podatkovna shramba

- Povezava med podatkovnim in procesnim modelom:
  - Eden od načinov uporabe DFD je, da najprej izdelamo podatkovni model, potem pa z DFD pokažemo, kako se podatki med procesi prenašajo.
- Podatkovna shramba tedaj ustreza enemu ali več entitetnim tipom iz podatkovnega modela.



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – podatkovna shramba



# Analiza – Izdelava modela sistema

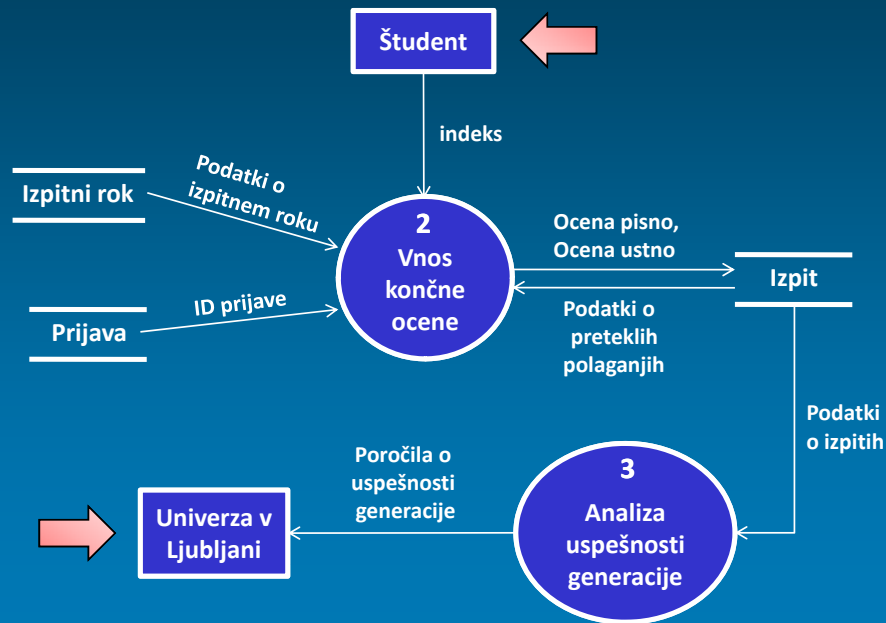
Procesni model – diagram podatkovnih tokov – zunanja entiteta

- **Zunanje entitete** predstavljajo zunanje procese ali zunanje sisteme.
- Nahajajo se izven interesnega področja analize.
- Ne zanima nas njihova struktura ali obnašanje, pač pa le podatkovni tokovi, ki se prenašajo med obravnavanim področjem in njimi.



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – zunanja entiteta



## Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – način risanja DFD

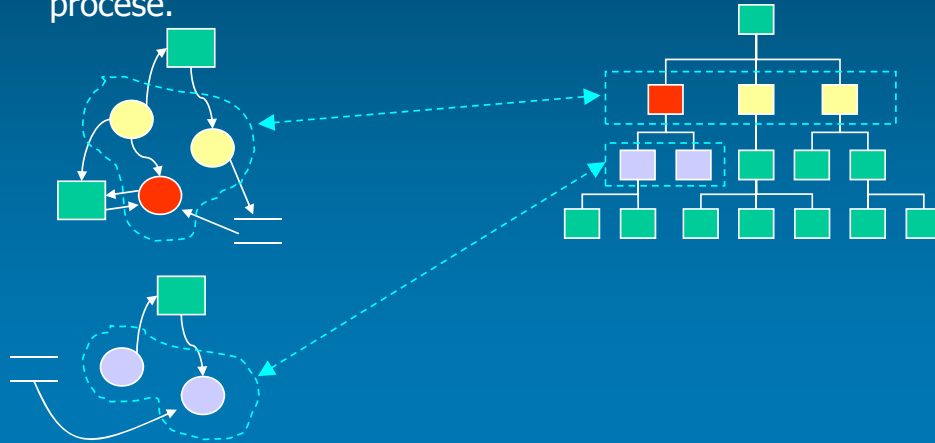
- V analizi pogosto identificiramo večje število procesov.
- Predstavitev vseh procesov enem diagramu je nepregledna, sama vsebina pa nerazumljiva.
- Zato uporabljamo **razčlenjevanje**. Diagrame rišemo od "vrha navzdol":
  - Začnemo z najvišjo ravno, kjer nastopajo obsežnejši procesi,
  - nadaljujemo do najnižje ravni, kjer nastopajo zelo podrobni procesi.

# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – način risanja DFD

## ▪ Razčlenjevanje:

- Za vsak proces, ki je predstavljen v DFD na višji ravni, izdelamo poseben DFD, kjer proces razbijemo na pod-procese.

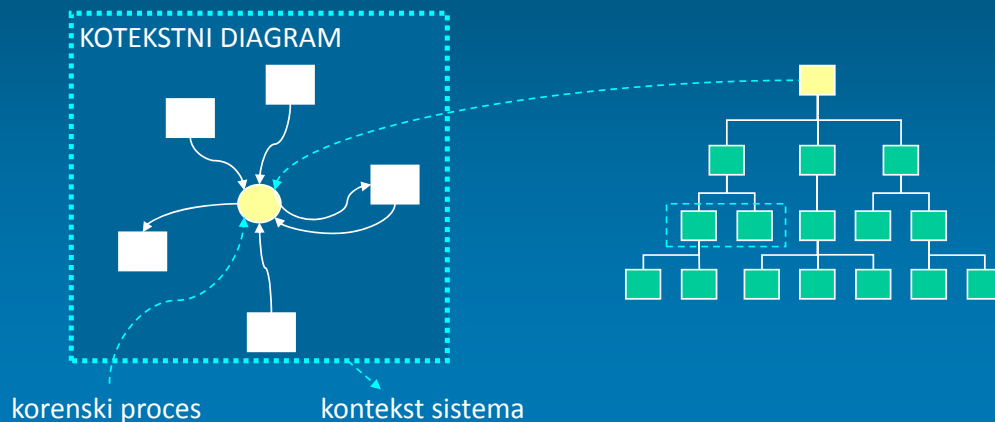


# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – način risanja DFD

## ▪ Kontekstni diagram:

- razčlenjevanje DFD začnemo na najvišji ravni, kjer nastopa en sam proces – **korenski proces**.



# Analiza – Izdelava modela sistema

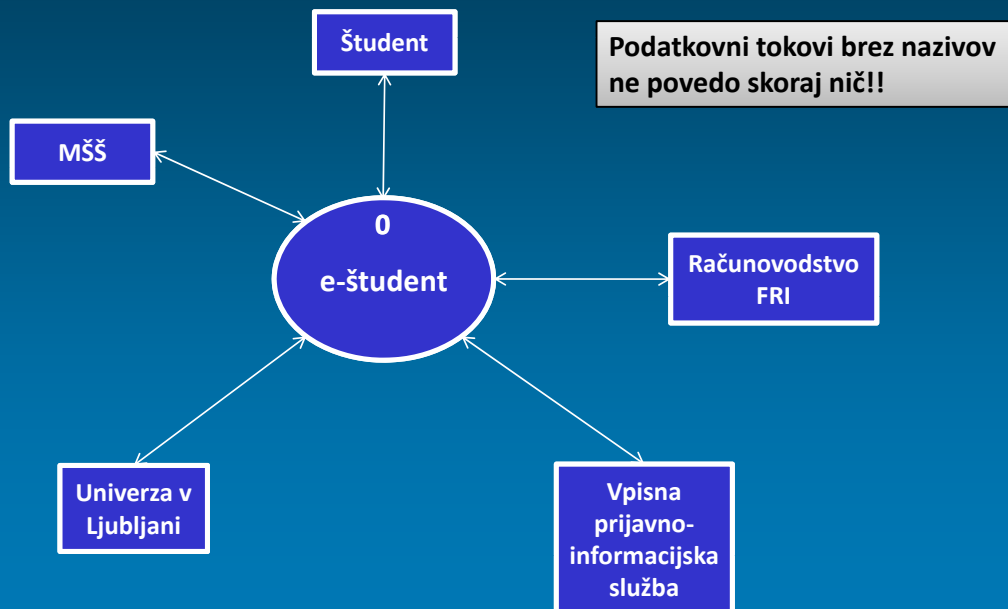
Procesni model – diagram podatkovnih tokov – način risanja DFD

- **Značilnosti kontekstnega diagrama:**
  - Kontekstni diagram prikazuje **kontekst sistema** – sistem v sodelovanju z okoljem.
  - Kontekstni diagram ima en sam proces – **korenski proces**.
  - Kontekstni diagram **nima podatkovnih shramb**. Shrambe so namenjene odlagališču podatkov pri prenosu le-teh med procesi. Podatkovna shramba je del sistema!
  - Podatkovni tokovi med korenskim procesom in zunanji entitetami opredeljujejo **vmesnike** med sistemom in okoljem.



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – primer kontekstnega diagrama



# Analiza – Izdelava modela sistema

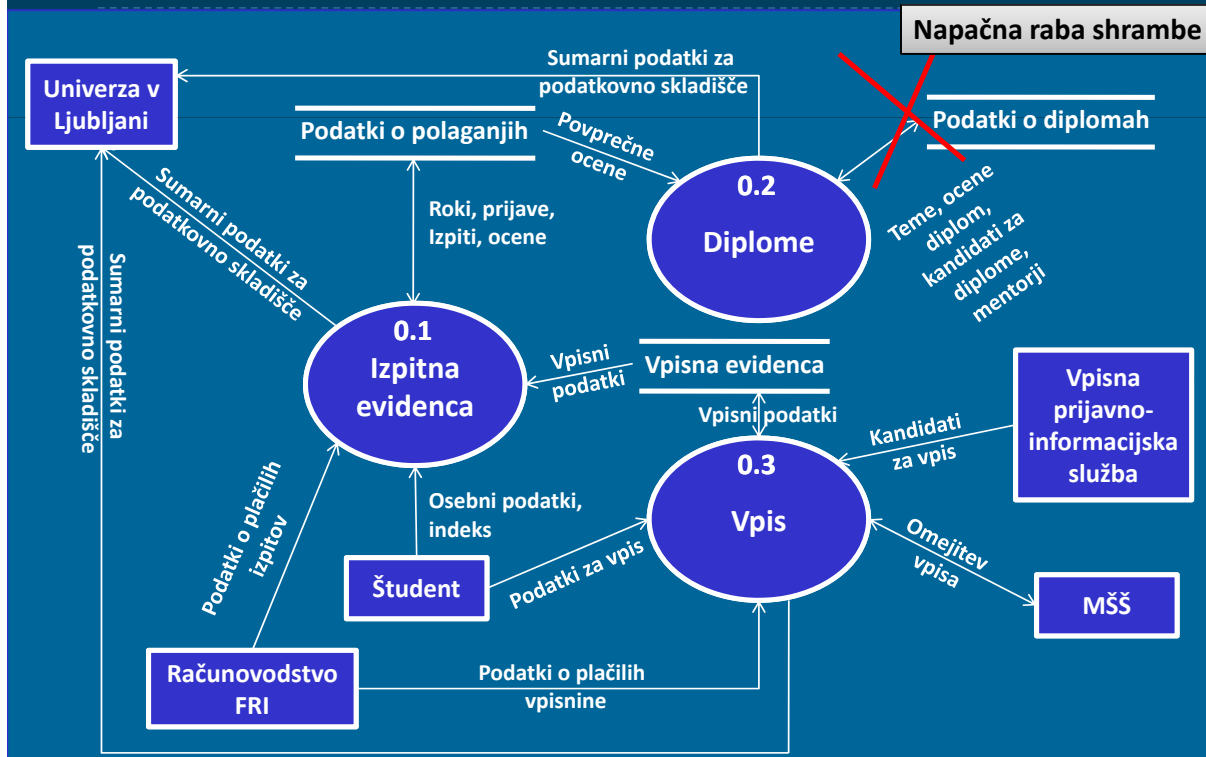
Procesni model – diagram podatkovnih tokov – način risanja DFD

## Prvi nivo diagrama podatkovnih tokov

- Prvi nivo razčlenitve kontekstnega diagrama predstavlja **DFD na hierarhičnem nivoju 1**.
- DFD na prvem hierarhičnem nivoju prikažemo z eno sliko, kjer korenski proces razčlenimo na potrebno število podprocesov (priporočljivo do 9).
- Pri razčlenjevanju procesa je potrebno **ohraniti vso funkcionalnost**: vsota funkcionalnosti vseh podrejenih procesov je enaka funkcionalnosti nadrejenega procesa.
- Potrebno je zagotoviti, da so evidentirani procesi približno **enakovredni oziroma uravnoteženi**.



# Analiza – Izdelava modela sistema



# Analiza – Izdelava modela sistema

Procesni model – diagram podatkovnih tokov – način risanja DFD

- **Kako podrobno razčlenjujemo DFD?**
  - Razčlenjevanje je smiselno do nivoja procesov, pri katerih ugotovimo, da je težko definirati shrambe podatkov, ki povezujejo njihove pod-procese. Na tem nivoju se procesi povezujejo neposredno s podatkovnimi tokovi.
  - Ta pogoj je vedno izpolnjen na nivoju elementarnih procesov, ki jih lahko opišemo z **zaporedjem korakov**.
  - Za opis procesov na najnižji ravni DFD tehnika ni primerna, ker **ne prikazuje zaporedja**. Uporablja se druge tehnike, ki so del **modeliranja procesne logike**.



# Analiza – Izdelava modela sistema

Procesni model – procesni diagram

- **Procesni diagram** uporabimo, ko želimo prikazati tok dogodkov ali potek določenega procesa.
- Za modeliranje procesov obstajajo številne tehnike, ki se razlikujejo predvsem po številu gradnikov ter notaciji.
- Diagrami **eEPC** spadajo med eno popularnejših tehnik modeliranja poslovnih procesov.

**eEPC – Extended Event-Driven Process Chain**



# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

- Tipični gradniki procesnih diagramov:
  - Dogodek
  - Aktivnost
  - Krmilni tok
  - Operator
  - Vloga
  - Aplikacija
  - Informacijski objekt



# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

- Dogodek:
  - vsaka aktivnost procesa ima praviloma vhodni in izhodni dogodek.
  - Vhodni dogodek se zgodi ob določenem trenutku, ko je izpolnjen nek pogoj in ima za posledico začetek izvajanja neke aktivnosti.
  - Ko se aktivnost izvede, lahko rezultat vpliva na izhodni dogodek.
  - Primeri dogodkov so: prijava zaključena, izpitni rok vnesen, ocena vpisana, prijava zavrnjena ipd.

Prijava  
zaključena

Ocena  
vpisana



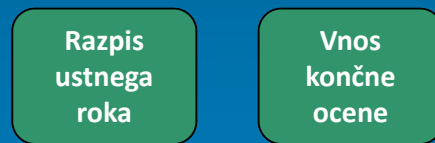


# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

### ▪ Aktivnost:

- aktivnost je najmanjša enota poslovnega procesa.
- Pomeni zaokroženo celoto procesiranja.
- Primeri aktivnosti: razpis ustnega roka, prijava na izpit, vnos končne ocene, odjava iz izpita ipd.
- Aktivnost lahko poteka v sodelovanju z uporabnikom ali popolnoma avtomatsko.

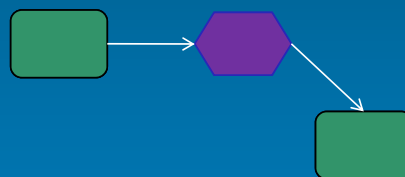


# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

### ▪ Krmilni tok

- krmilni oziroma kontrolni tok v obliki puščice nakazuje zaporedje dogodkov in aktivnosti v modeliranemu procesu.
- Kontrolni tok lahko razumemo kot nosilec kontrolnih podatkov in drugih pomembnih podatkov za izvajanje procesa.

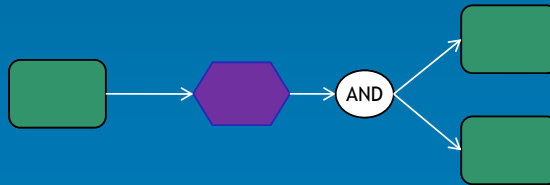


# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

### ▪ Operator

- Operator predstavlja mesto razdruževanja kontrolnega toka ali združitve iz več kontrolnih tokov v enega.
- Na nekem mestu v modeliranem procesu se lahko kontrolni tok, ki izhaja iz aktivnosti ali dogodka, razdruži v več tokov, ki vodijo naprej do dogodkov ali aktivnosti.
- Obratno se lahko kontrolni tokovi, ki izhajajo iz več aktivnosti ali dogodkov, združijo v en kontrolni tok, ki vodi do dogodka ali aktivnosti. Operatorji so AND, OR, XOR.



# Analiza – Izdelava modela sistema

## Procesni model – procesni diagram

### ▪ Vloga

- Vloga predstavlja subjekt, ki aktivnost izvaja oz. je zanj odgovoren (posameznik, skupina ljudi, organizacijska enota, ipd.)

### ▪ Aplikacija

- Predstavlja informacijsko rešitev, ki podpira izvajanje neke aktivnosti.

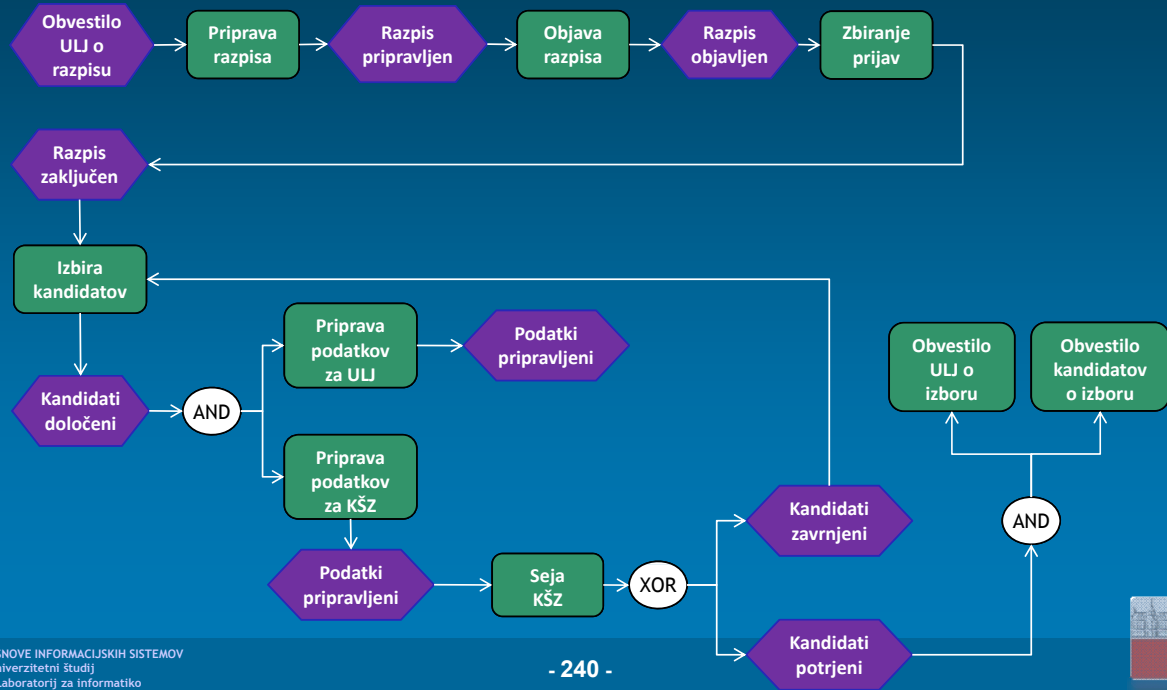
### ▪ Informacijski objekt

- Predstavlja nosilec podatkov, ki bodisi vstopa kot vhod v aktivnost ali v obliki izhoda iz nje izstopa.



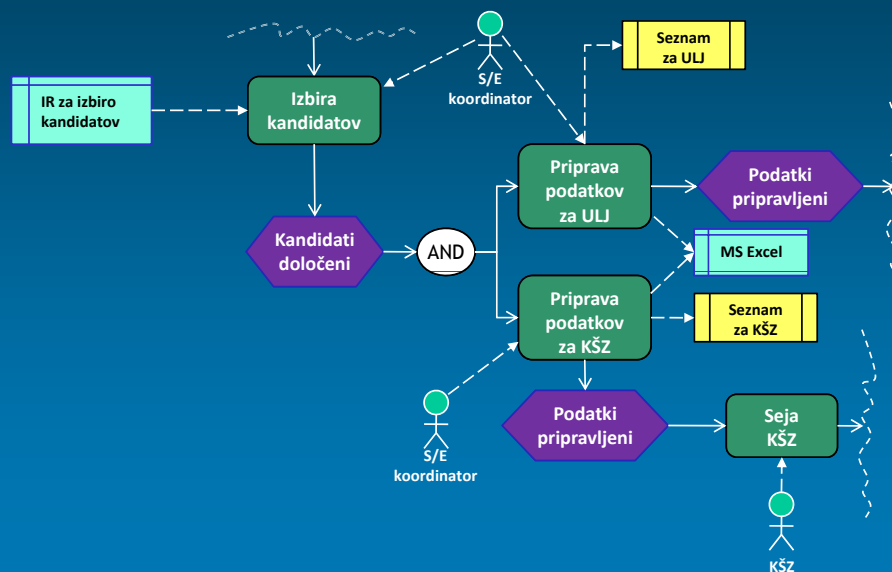
# Analiza – Izdelava modela sistema

Procesni model – procesni diagram – primer



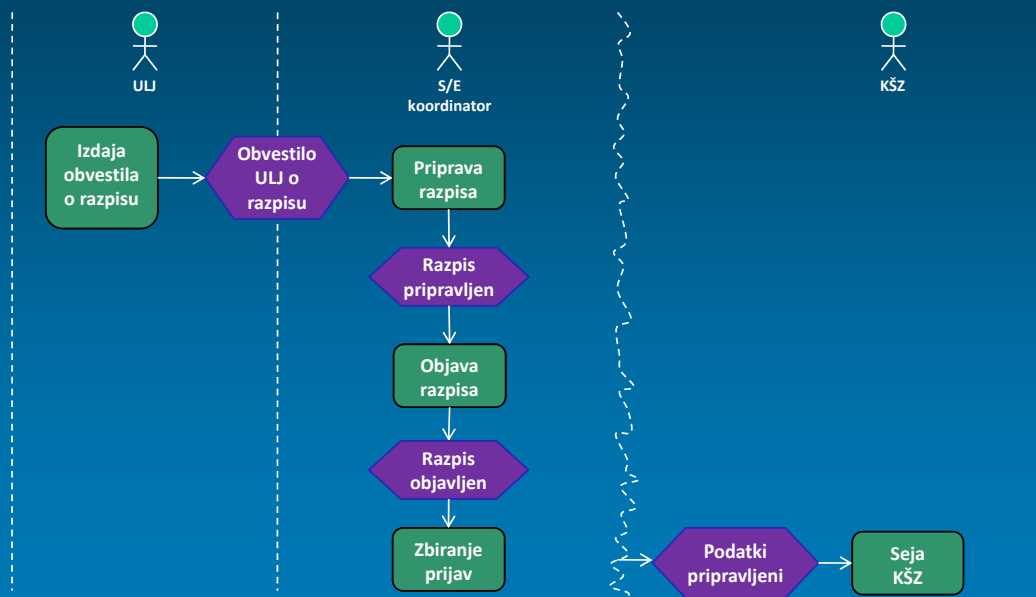
# Analiza – Izdelava modela sistema

Procesni model – procesni diagram – primer – dodatni gradniki



# Analiza – Izdelava modela sistema

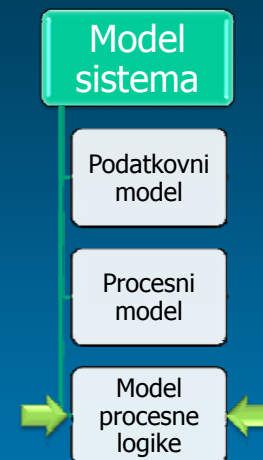
Procesni model – procesni diagram – primer – steze



# Analiza – Izdelava modela sistema

Izdelava modela sistema – model procesne logike

- **Model procesne logike:**
  - Dopolnjuje procesni model.
  - Osredotoči se na tiste procese, ki v procesnem modelu niso dovolj jasno opisani: zaporedje korakov, kompleksne odločitvene situacije.
  - Model procesne logike izdelamo s pomočjo diagramskih tehnik, kot so: **strukturiran jezik**, **odločitvene tabele**, **odločitvena drevesa**, **diagrami prehajanja stanj** idr.



# Analiza – Izdelava modela sistema

## Model procesne logike – strukturiran jezik

- Najpreprostejši način opisa procesne logike je z naravnim jezikom.
- Strukturiran jezik je oblika naravnega jezika:
  - kjer so opisi kratki in jedrnati stavki, sestavljeni iz glagolskih in samostalniških oblik naravnega jezika.
  - kjer ne uporabljamo drugih besednih oblik, npr. pridevnikov, prislovov itd.
  - kjer pišemo z zamiki, da poudarimo strukturo posameznih delov opisa.



# Analiza – Izdelava modela sistema

## Model procesne logike – strukturiran jezik – primer

- Primer opisa:
  - Vnos diplomske teme

```
Izberi študijski program
//izpišejo se vsi študijski programi
Izberi študenta
//izpišejo se vsi kandidati, ki so pri mentorju dvignili temo
Vnesi naslov teme
Vnesi opis teme
Potrdi ali prekliči temo
//če uporabnik vnosa ne potrdi, se podatki ne zabeležijo
//prikaži opcije za vnos/spreminjanje/izpis tem diplomskih nalog
```



# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvene tabele in drevesa

- Odločitvene tabele in odločitvena drevesa uporabimo pri modeliranju zapletenejše procesne logike.
- Tehniki sta primerni predvsem, ko v procesni logiki nastopa **veliko pogojev**, ki v različnih kombinacijah sprožajo različne akcije.



# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvene tabele

- **Zgradba odločitvene tabele:**
  - v zgornjem delu prikazuje pogoje, ki nastopajo v procesu ter vrednosti, ki jih ti pogoji lahko zavzamejo. Posameznim kombinacijam vrednosti pogojev pravimo **pravilo**.
  - v spodnjem delu tabele so navedene **akcije**, ki se morajo izvesti ob določenem pravilu.

# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvene tabele

Pravilo – kombinacija vrednosti pogojev.

	Pravilo 1	Pravilo 2	...	Pravilo p
Pogoj 1	$V(p_1)$	$V(p_1)$	...	$V(p_1)$
Pogoj 2	$V(p_2)$	$V(p_2)$	...	$V(p_2)$
...	...	...	...	...
Pogoj n	$V(p_n)$	$V(p_n)$		$V(p_n)$
Akcija 1	D/N	D/N	...	D/N
Akcija 2	D/N	D/N	...	D/N
...	...	...	...	...
Akcija m	D/N	D/N	...	D/N

Vrednost pogoja 2 v pravilu p

Akcije, ki jih izvedemo, če velja pravilo p.

# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvene tabele

- Pravila v odločitveni tabeli predstavljajo kombinacije vrednosti, ki jih posamezni pogoji lahko zavzamejo.

- Število pravil:

$$\prod_{i=1}^n zv(pog_i)$$

- $zv$ : zaloga vrednosti
- $n$ : število pogojev

### Primer:

$p_1$  = Pogoj 1: status vpisa

$p_2$  = Pogoj 2: letnik

$p_1$  = {redno, izredno, ponavlja, pavzira}

$zv(p_1) = 4$

$p_2$  = {1, 2, 3, 4, 5, ABS}

$zv(p_2) = 6$

Število pravil =  $4 * 6 = 24$

# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvene tabele – primer

### Prijava na izpitni rok

		Pravila							
Pogoji	Izpitni rok razpisan	D	D	D	D	D	D	D	N
	Dovoljen pristop k izpitu	D	D	D	D	D	D	D	N
	Pravočasna prijava	D	D	D	D	D	D	N	N
	Izpolnjeni vsi predpogoji	D	D	D	D	D	N	D	D
	Število zaporednih polaganj presega 3	N	N	N	D	D	N	N	N
	Prvo polaganje v izpitnem obdobju	D	N	N	D	N	D	D	D
	Drugo polaganje v jesenskem obdobju	N	D	N	N	D	N	N	N
	Četrto polaganje v šolskem letu	N	N	N	N	D	N	N	N
Akcije	Sprejmi prijavo	x	x		x				
	Zavrni prijavo			x		x	x	x	x
	Sestavi komisijo				x				
	Izdaj plačilni nalog				x				

# Analiza – Izdelava modela sistema

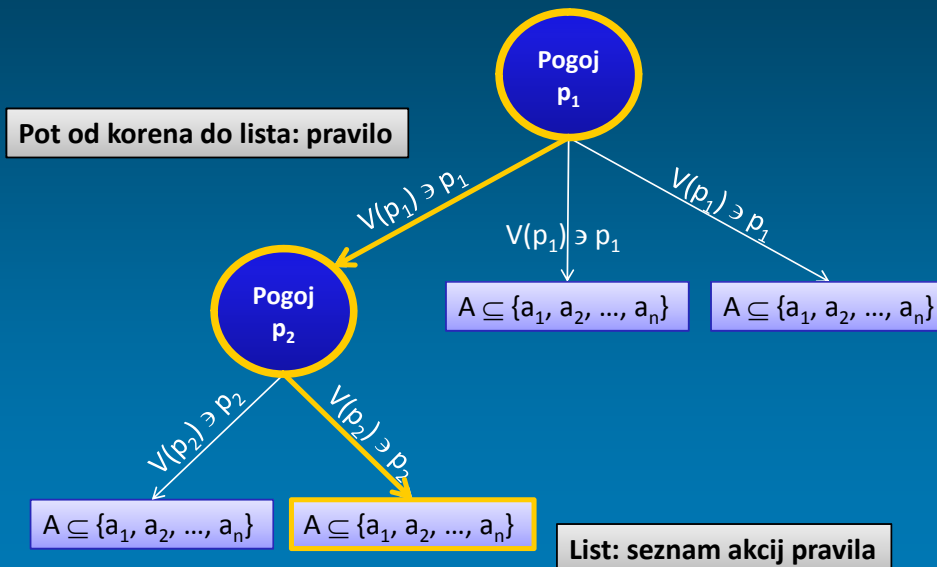
## Model procesne logike – odločitvena drevesa

- **Zgradba odločitvenega drevesa:**
  - Odločitveno drevo je sestavljeno iz **vozlišč** ter **povezav** med njimi.
  - Vozlišča predstavljajo **pogoje**, povezave med njimi pa možne **vrednosti posameznih pogojev**.
  - Posamezna pot v drevesu, od korena do predzadnjega vozlišča, predstavlja kombinacijo pogojev ali **pravilo**.
  - List drevesa, ki je na koncu poti, prikazuje **seznam akcij** pravila.



# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvena drevesa



# Analiza – Izdelava modela sistema

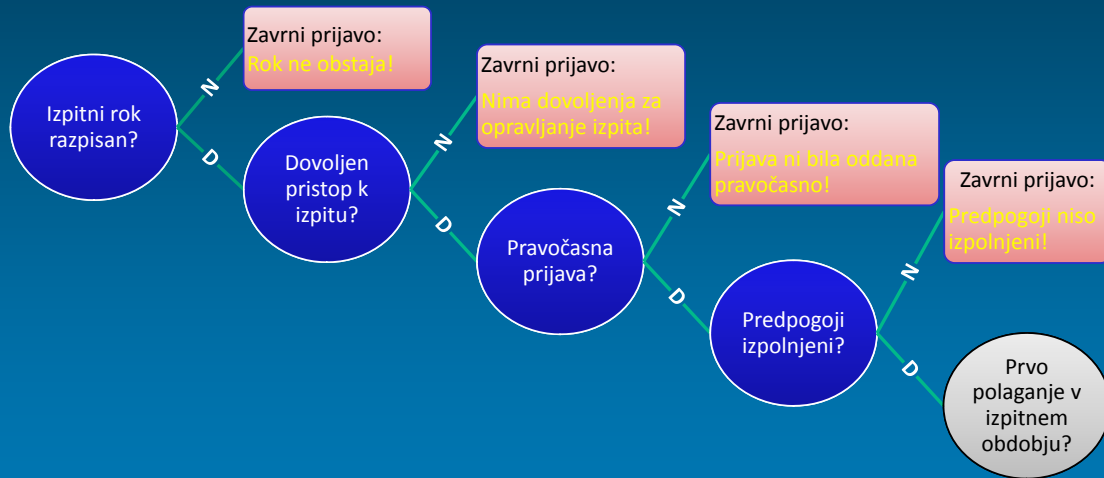
## Model procesne logike – odločitvena drevesa

### ▪ Sestavljanje odločitvenega drevesa:

- Odločitveno drevo rišemo iz leve v desno.
- Začnemo s prvim vozliščem, v katerega vpišemo prvi pogoj.
- Glede na zalogo vrednosti, ki jo pogoj ima, narišemo iz vozlišča ustrezno število puščic.
- Na konec vsake puščice narišemo novo vozlišče, ki predstavlja naslednji pogoj. Če gre za list, vpišemo seznam akcij, ki se izvedejo ob pravilu.
- Če ugotovimo, da na koncu določenih poti ni nobenih akcij, lahko te poti brišemo iz drevesa.

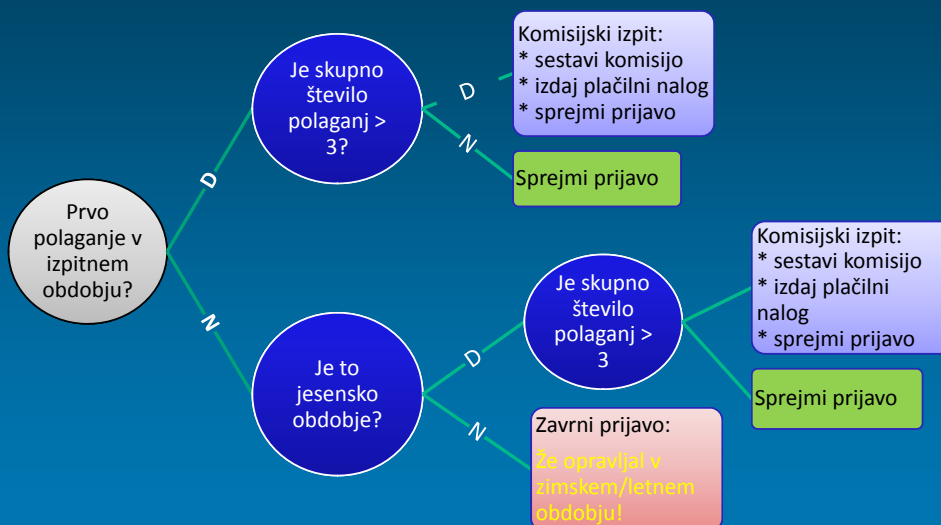
# Analiza – Izdelava modela sistema

## Model procesne logike – odločitvena drevesa – primer



# Analiza – Izdelava modela sistema

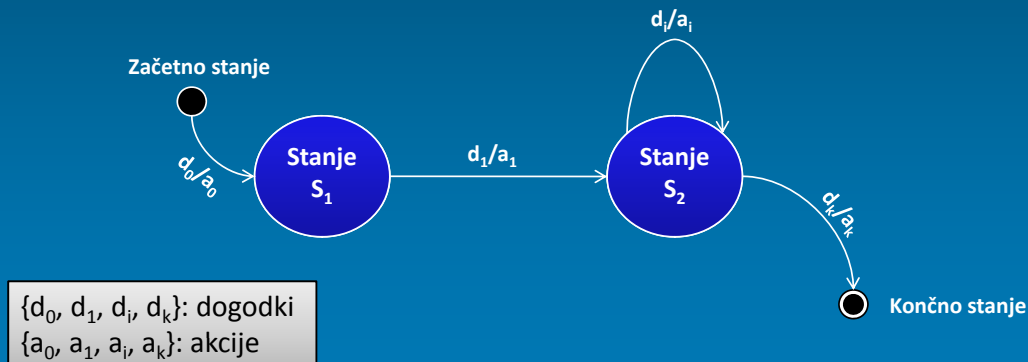
## Model procesne logike – odločitvena drevesa – primer



# Diagramске tehnike

## Model procesne logike – diagram prehajanja stanj

- Z diagramom prehajanja stanj prikažemo stanja, v katerih se lahko nahaja opazovan proces ter odzive oziroma prehajanja med stanji kot odziv na različne dogodke.



# Diagramске tehnike

## Model procesne logike – diagram prehajanja stanj

- Glavni gradniki diagrama prehajanja stanj:
  - Stanje
  - Dogodek
  - Akcija
- Stanje:
  - Z vidika dogajanja v sistemu se lahko proces nahaja v različnih stanjih.
  - Stanja so določena z vrednostjo lastnosti, ki nas v okviru dogajanja procesa zanimajo.
  - Proces se nahaja v določenem stanju vse dokler se katera izmed opazovanih vrednosti ne spremeni. Tedaj nastopi novo stanje.

# Diagramске tehnike

## Model procesne logike – diagram prehajanja stanj

### ▪ Dogodek:

- Prehajanja med stanji povzročajo dogodki.
- Dogodek vpliva na spremembo ene ali več opazovanih lastnosti procesa.
- Z vidika prehajanja med stanji je dogodek nekaj, kar se pripeti v določenem trenutku in nima časovne razsežnosti.

### ▪ Akcija:

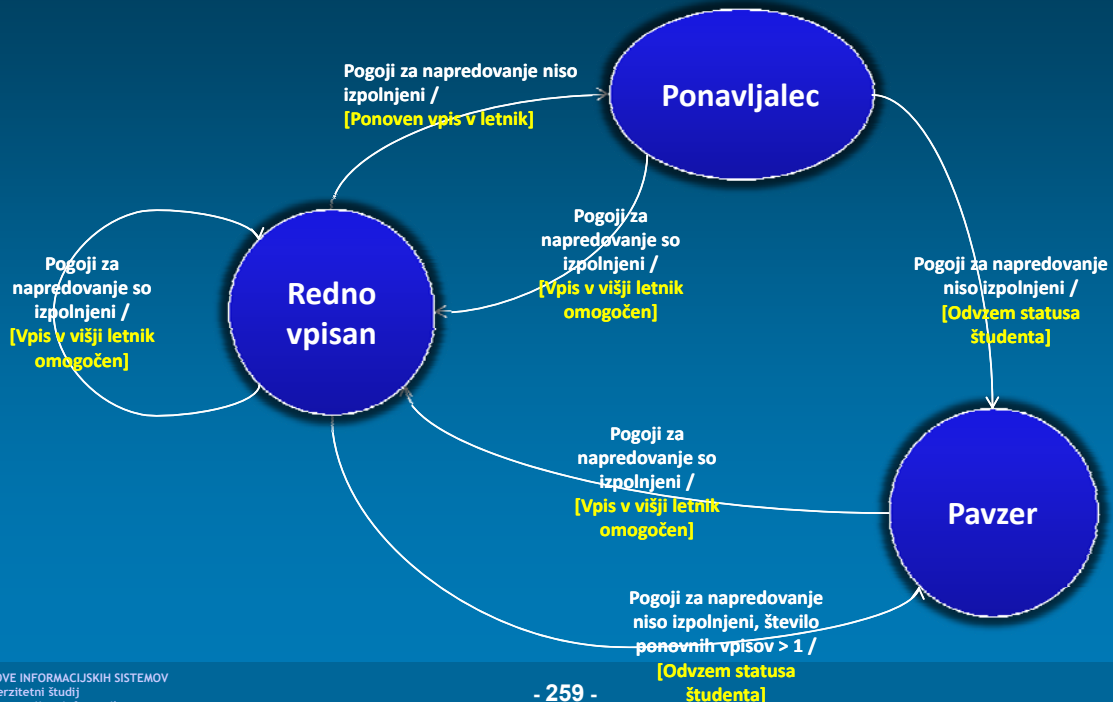
- Ob dogodku se lahko zgodijo različne akcije.
- Kot dogodki tudi akcije niso časovno trajajoče; zgodijo se v trenutku.



# Diagramске tehnike

Status študenta

## Model procesne logike – diagram prehajanja stanj – primer – prikaz z grafom



# Diagramске tehnike

Status študenta

Model procesne logike – diagram prehajanja stanj – primer – prikaz s tabelo

Dogodek	Stanje		
	Redno vpisan S1	Ponovno vpisan S2	Pavzer S3
Pogoji za napredovanje so izpolnjeni	S1 / [Vpis v višji letnik omogočen]	S1 / [Vpis v višji letnik omogočen]	S1 / [Vpis v višji letnik omogočen]
Pogoji za napredovanje niso izpolnjeni	S2 / [Ponoven vpis s letnik]	S3 / [Odvzem statusa študenta]	
Pogoji za napredovanje niso izpolnjeni, število ponovnih vpisov > 1	S3 / [Odvzem statusa študenta]		

## Vaja

- Za opisan problem želimo izdelati informacijsko rešitev. Izdelaj model sistema, ki bo problemsko domeno kar najbolje opisal.
- Izbiraš lahko med poljubnimi diagramskimi tehnikami.



# Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza
    - Izdelava modela sistema;
    - ➔ Izdelava prototipov;
    - Izdelava predloga arhitekture sistema;
    - Opredelitev strategije testiranja;
    - Predstavitve rezultatov analize naročniku.;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



# Analiza – Izdelava prototipov

Izdelava prototipov v okviru analize

- Izdelava prototipov je neobvezna
  - v okviru analize učinkovito sredstvo za prikaz izgleda ter osnovne funkcionalnosti uporabniškega vmesnika.
- Razvita posebna razvojna okolja, ki omogočajo vizualno sestavljanje zaslonskih mask, izpisov in poizvedb ter vsebujejo mehanizme za avtomatsko generiranje kode.



# Analiza – Izdelava prototipov

## Izdelava prototipov v okviru analize

- Prototipi se navadno uporabljajo le kot del specifikacije sistema, za pridobitev jasnejše podobe bodočega sistema in se v nadaljevanju zavržejo.
- Obstajajo tudi metode, ki tako izdelane prototipe izkoristijo kot osnovo za izdelavo produkcijskega sistema (Rapid Application Development – RAD).



# Analiza – Izdelava prototipov

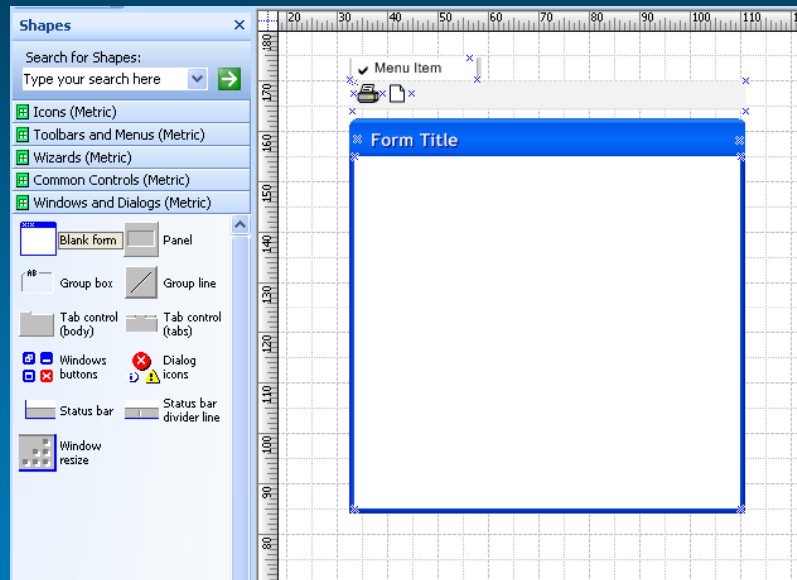
## Izdelava prototipov v okviru analize

- Nekaj primerov orodij za izdelavo prototipov:
  - Vizualna razvojna okolja: npr. Delphi;
  - Risarska orodja: npr. MS Visio;
  - CASE orodja: npr. Oracle Designer.



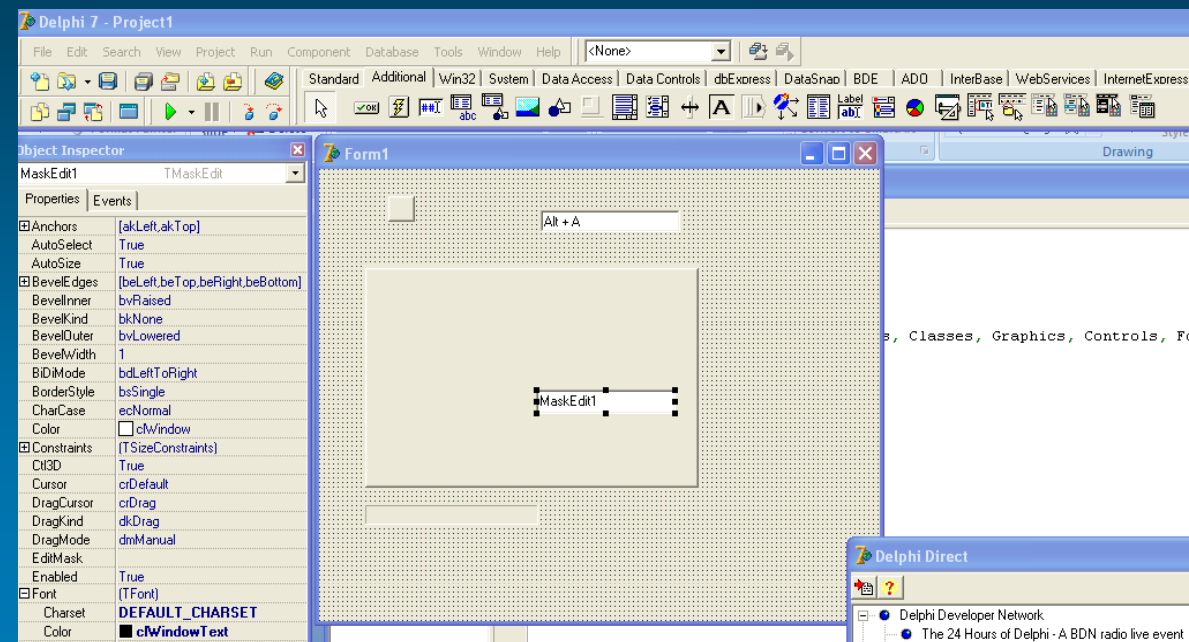
# Analiza – Izdelava prototipov

## Izdelava prototipov v okviru analize – Primer MS Visio



# Analiza – Izdelava prototipov

## Izdelava prototipov v okviru analize – Primer Delphi







# Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza
    - Izdelava modela sistema;
    - Izdelava prototipov;
    - ➔ Izdelava predloga arhitekture sistema;
    - Opredelitev strategije testiranja;
    - Predstavitve rezultatov analize naročniku.;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



## Analiza – Predlog tehnične arhitekture

Izdelava predloga tehnične arhitekture

- V okviru analize ne analiziramo zgolj funkcionalnih zahtev, temveč tudi **tehnične** in druge **nefunkcionalne** zahteve.
- Na osnovi tega podamo **predlog tehnične arhitekture sistema**.
- Upoštevamo tudi strategijo podjetja oziroma standarde, ki so v podjetju določeni za celoten informacijski sistem.



# Analiza – Predlog tehnične arhitekture

## Izdelava predloga tehnične arhitekture

- V okviru predloga tehnične arhitekture sistema opredelimo strojno, komunikacijsko in programsko opremo, ki je potrebna za vzpostavitev ustreznega razvojnega, testnega in produkcijskega okolja.
- V fazi analize navadno izdelamo le predlog, ki služi kot eden pomembnih virov za oceno stroškov razvoja oziroma nakupa IR.
- Upoštevati moramo standarde in predpise, ki so določeni za posamezna področja.



# Analiza – Predlog tehnične arhitekture

## Izdelava predloga tehnične arhitekture – vsebina

### 1. Arhitektura sistema

- 1.1 Strojna oprema
- 1.2 Sistemska programska oprema
- 1.3 Komunikacijska oprema
- 1.4 Druga oprema
- 1.5 Postavitveni diagram (opsijsko)

### 2. Postopki, predpisi in standardi

- 2.1 Zagotavljanje varnosti
- 2.2 Varnostne kopije (Backup)
- 2.3 Vzpostavitev sistema (Recovery)
- 2.4 Obravnava napak





# Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza
    - Izdelava modela sistema;
    - Izdelava prototipov;
    - Izdelava predloga arhitekture sistema;
    - ➔ Opredelitev strategije testiranja;
      - Predstavitve rezultatov analize naročniku.;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



# Analiza – Opredelitev strategije testiranja

Strategija testiranja

- Način testiranja je s standardi kakovosti podrobno določen. Njegova izvedba vseeno odvisna od vrste projekta.
- V okviru opredelitve strategije testiranja se odločimo:
  - Kaj je predmet testiranja? Npr. programske enote, programski sklopi, integracija, tehnične zahteve...
  - Kdo bo izvajal testiranje, kje in kako bo testiranje potekalo? Npr. teste programskih enot izvajajo programerji sami v okviru razvojnega okolja.
  - Kje bo nameščeno testno okolje (pri izvajalcu ali pri uporabniku)?



# Analiza – Opredelitev strategije testiranja

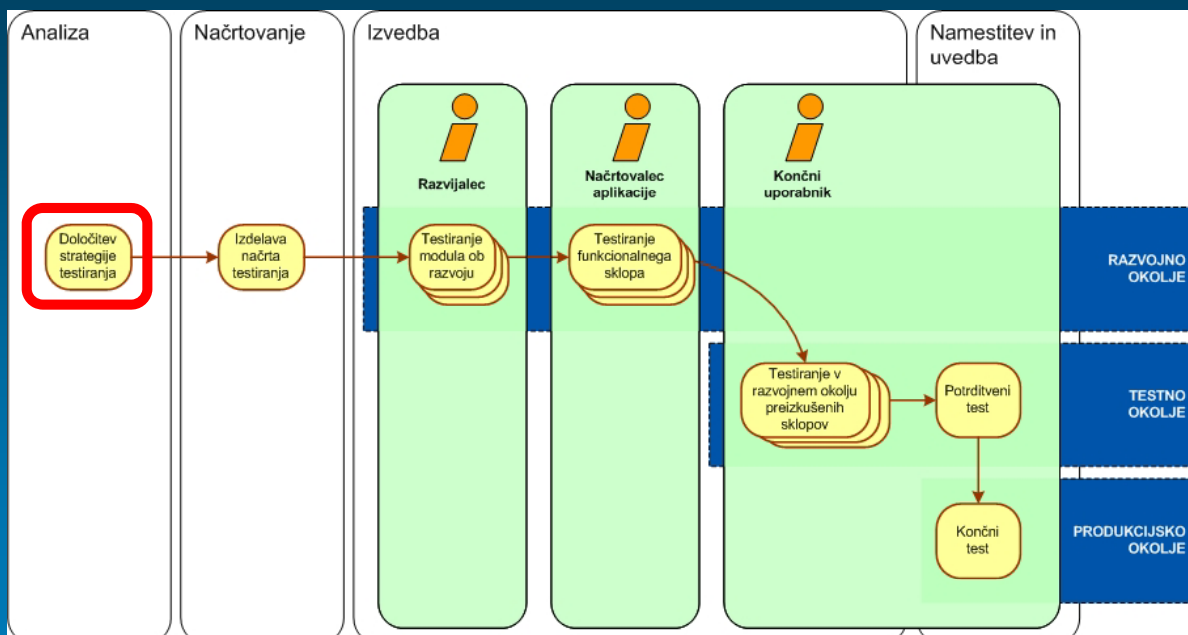
## Strategija testiranja

- **Opredelitev strategije testiranja (nadaljevanje):**
  - Na kakšni strojni opremi bo nameščeno testno okolje?
  - V kateri točki projekta se bo namestilo testno okolje?
  - Bo testiranje v testnem okolju potekalo v več iteracijah ali v celoti?
  - Kakšna orodja se bo uporabljalo za pripravo testov?
  - Kakšna orodja se bo uporabljalo za izvajanje testov?
  - ...



# Analiza – Opredelitev strategije testiranja

## Strategija testiranja – prva aktivnost v okviru postopka testiranja





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - ➔ Analiza
    - Izdelava modela sistema;
    - Izdelava prototipov;
    - Izdelava predloga arhitekture sistema;
    - Opredelitev strategije testiranja;
    - ➔ Predstavitev rezultatov analize naročniku;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



## Analiza – Predstavitev rezultatov

Predstavitev rezultatov analize uporabniku

- Ko izdelamo vse izdelke, ki jih predvideva faza analize, pripravimo predstavitev za naročnika.
- Osnovni namen predstavitve je pridobitev potrdila s strani naročnika o pravilnosti razumevanja problema, ki ga rešujemo z IR.
- Potrebno upoštevati, da naročnik morda nima strokovnih znanj za poglobljeno razumevanje rezultatov analize; predstavitev mora biti temu primerna.



## Analiza – Predstavitev rezultatov

### Predstavitev rezultatov analize uporabniku

- Predstavitev se zaključi s potrditvijo naročnika, da rezultati analize ustrezajo zahtevam.
- Potrditev se izvede s podpisom dokumenta, ki pogosto predstavlja eno izmed kontrolnih točk v okviru projektnega vodenja.



## Razvoj po strukturnem pristopu

### Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza
  - ➔ Načrtovanje;
  - Izvedba;
  - Testiranje;
  - Uvajanje
  - Izobraževanje.



# Načrtovanje

## Opredelitev in namen

- **Glavni namen načrtovanja** je izdelati načrt zgradbe sistema glede na specifikacije, ki so bile zbrane v fazi analize.
- **Načrt** daje odgovor na vprašanje, **KAKO** izdelati sistem, da bo ustrezal zahtevam, ki smo jih evidentirali v fazi analize.



# Načrtovanje

## Cilji načrtovanja

- **Cilji načrtovanja so:**
  - Izdelati načrt IR, ki ustreza ugotovitvam iz analize in upošteva tehnološke omejitve sistema;
  - Dokumentirati specifikacije načrta na način, ki bo omogočal vzdrževanje sistema;
  - Zasnovati strategijo prehoda iz obstoječe na novo aplikacijo.



# Načrtovanje

Končni izdelek 📁

- Osnovna rezultata načrtovanja sta **načrt podatkovne baze in načrt programskih modulov**, s katerima pripravimo vse potrebno za izdelavo podatkovnih in programskih komponent IR.
- V fazi načrtovanja izdelamo tudi:
  - načrt dokumentacije,
  - načrt testiranja in
  - načrt namestitve in uvedbe.



# Načrtovanje

Vloge in koraki 👤 ⚡

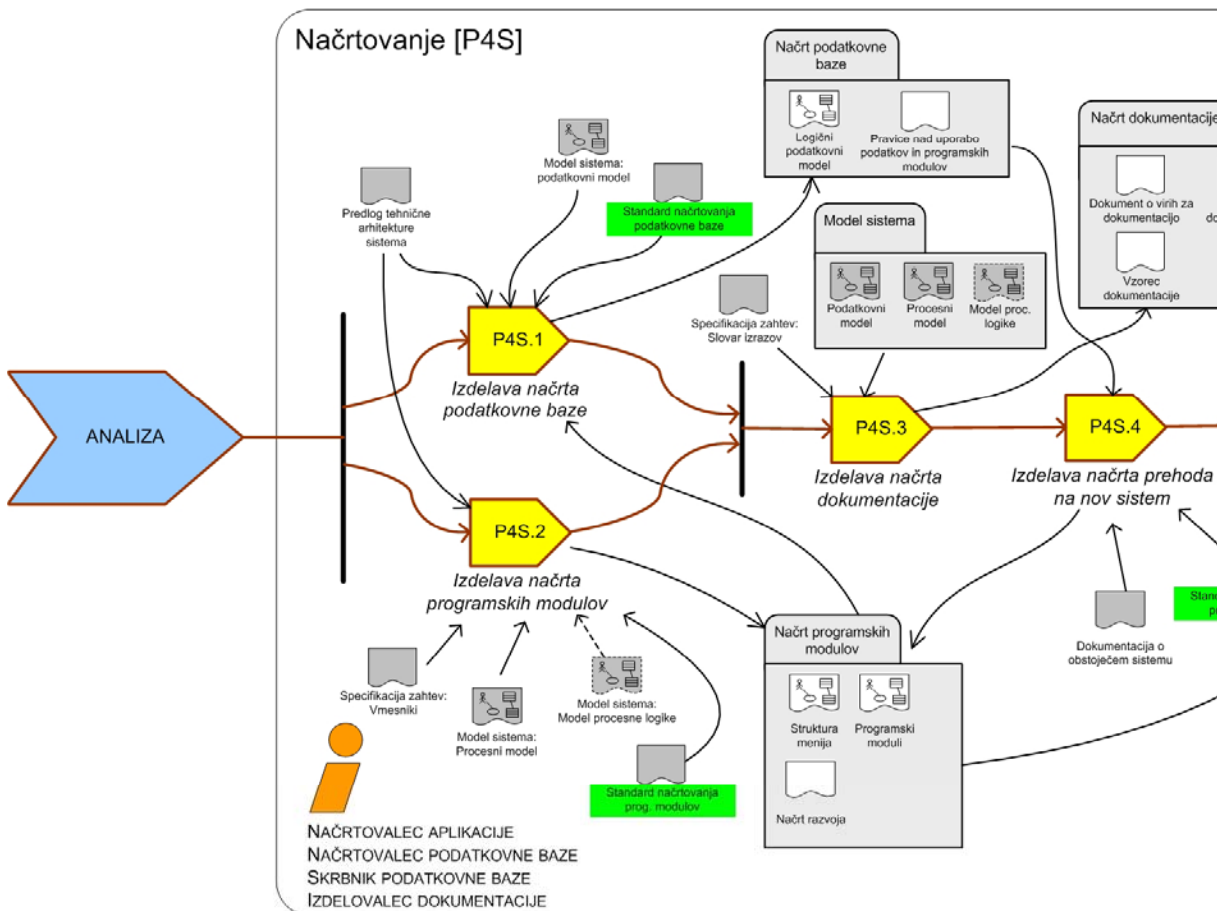
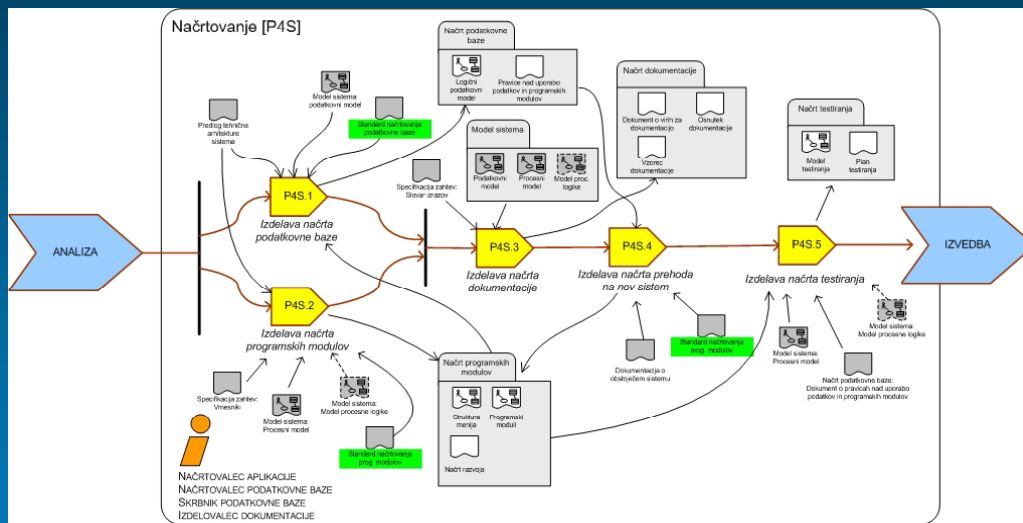
- Pri načrtovanju sodelujejo **načrtovalec podatkovne baze, načrtovalec aplikacije, skrbnik podatkovne baze, izdelovalec dokumentacije, uvajalec, poslovni lastnik in končni uporabnik**.
- Tipične aktivnosti načrtovanja so:
  - Izdelava načrta podatkovne baze,
  - Izdelava načrta programskih modulov,
  - Izdelava načrta dokumentacije,
  - Izdelava načrta testiranja,
  - Izdelava načrta namestitve in uvedbe in
  - Predstavitve načrta naročniku.





# Načrtovanje

## Schema postopka ▶



## Načrtovanje – Izdelava načrta PB

### Namen aktivnosti

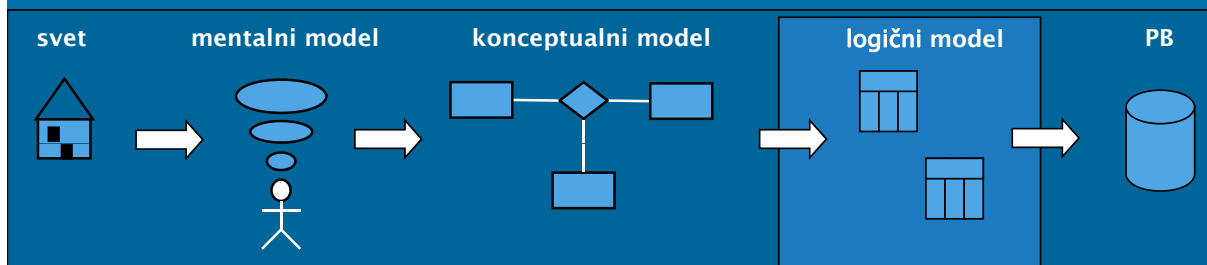
- **Namen aktivnosti** je na podlagi konceptualnega modela iz analize izdelati logični podatkovni model in izvesti ostale korake, potrebne za vzpostavitev učinkovite fizične podatkovne baze.
- V sklopu faze načrtovanja izdelamo **logični podatkovni model** in določimo **sistem pravic** za uporabo podatkov in programskih modulov.



## Načrtovanje – Izdelava načrta PB

### Izdelava logičnega podatkovnega modela

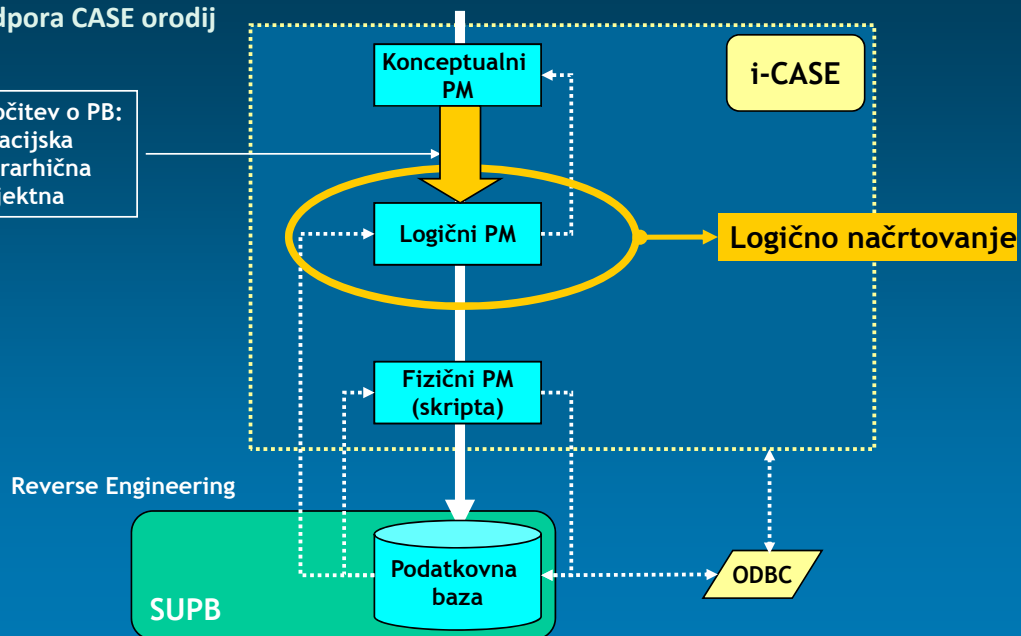
- **Logično modeliranje** podatkovne baze nastopi za konceptualnim modeliranjem.
- Osnova logičnega modela je jezik, ki je razumljiv ciljnemu SUPB.
- Če izberemo **relacijski SUPB**, potem govorimo o **relacijskem modelu**.



# Načrtovanje – Izdelava načrta PB

Podpora CASE orodij

Odločitev o PB:  
-Relacijska  
-Hierarhična  
-Objektna



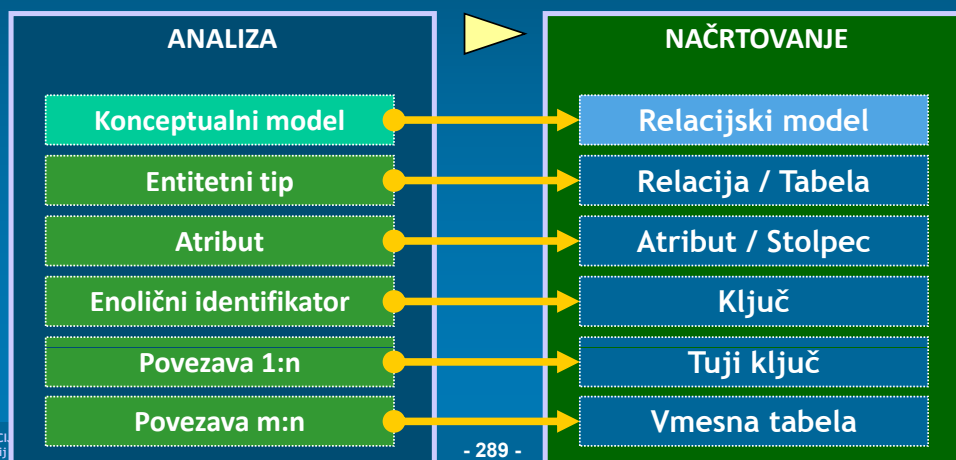
# Načrtovanje – Izdelava načrta PB

Prehod iz konceptualnega v logični model

- Prehod iz konceptualnega v logični model je navadno avtomatiziran s strani CASE orodij.

Primer: vrsta baze: relacijska

SUPB: Oracle



# Načrtovanje – Izdelava načrta PB

## Tipični koraki pri izdelavi relacijskega modela

- **Tipični koraki:**
  - Za entitetne tipe kreiraj relacije
  - Preveri relacije z normalizacijo
  - Preveri relacije s pregledom uporabniških transakcij
  - Preveri omejitve integritete
  - Preveri model z uporabnikom
  - Združi lokalne modele v globalni model (opcijsko)
  - Preveri zmožnosti modela za razširitve
- **V nadaljevanju si bomo pogledali nekaj osnov o relacijskem modelu.**



# Načrtovanje – Izdelava načrta PB

## Osnovni koncepti relacijskega modela

- **Pri relacijskem modeliranju se srečamo z naslednjimi koncepti:**
  - Relacija
  - Atribut
  - Domena
  - n-terica
  - Funkcionalna odvisnost
  - Ključ
  - Pogled
  - Normalizacija
  - ...



# Načrtovanje – Izdelava načrta PB

## Predstava relacije

- Relacijo si lahko predstavljamo kot dvodimenzionalno tabelo s stolpci in vrsticami.
  - Velja za logično strukturo podatkovne baze in ne za fizično.

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

→ Relacija

# Načrtovanje – Izdelava načrta PB

## Atribut relacije

- **Atribut** je poimenovani stolpec relacije.

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

→ Atribut relacije

## Načrtovanje – Izdelava načrta PB

### Domena relacije

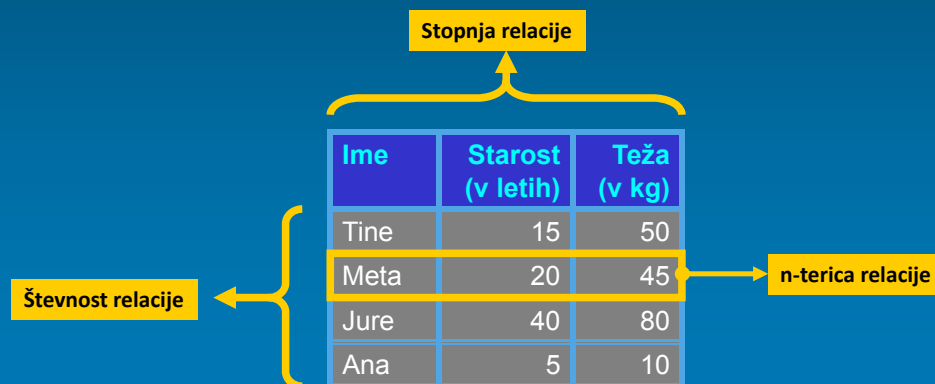
- **Domena** je množica dovoljenih vrednosti enega ali več atributov, ki so vključeni v to domeno.
- **Primeri domen:**

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

## Načrtovanje – Izdelava načrta PB

### Osnovne karakteristike relacije

- **N-terica** je ena vrstica v relaciji.
- **Števnost relacije** je število n-teric relacije.
- **Stopnja relacije** je število atributov v relaciji.



## Načrtovanje – Izdelava načrta PB

### Relacijska podatkovna baza in normalizacija

- Relacijska podatkovna baza je množica normaliziranih relacij z enoličnimi imeni.
- Normalizirane relacije so relacije, ki ustrezajo normalnim oblikam. Te določajo pravila, ki jim morajo relacije zadoščati, da pri vnašanju, spreminjanju in brisanju podatkov ne prihaja do anomalij.



## Načrtovanje – Izdelava načrta PB

### Matematična definicija relacije

- Relacijo matematično definiramo kot podmnožico kartezičnega produkta množic, ki predstavljajo domeno vrednosti atributov relacije.

$$r \subseteq D_1 \times D_2 \times \dots \times D_n$$

D1 je domena atributa A<sub>1</sub>:  
množica vrednosti, ki jih A<sub>1</sub>  
lahko zavzame!



# Načrtovanje – Izdelava načrta PB

## Relacijska shema

- Vsaki relaciji pripada **relacijska shema**.
- Relacijsko shemo sestavlja oznaka sheme R ter lista oznak atributov  $A_i$  s pripadajočimi oznakami domen  $D_i$ :
  - $R (A_1: D_1, A_2: D_2, \dots, A_n: D_n)$
- Relacijska shema predstavlja **semantiko** ali **pomen** relacije.



# Načrtovanje – Izdelava načrta PB

## Relacijska shema

Ime	Starost (v letih)	Teža (v kg)
Tine	15	50
Meta	20	45
Jure	40	80
Ana	5	10

Schema relacije

Relacija, predstavljena kot tabela

$Sh(r) = Oseba(Ime: I, Starost: C, Teža: C)$  Schema relacije

Domena, ki obsega imena:  $I \equiv \{Tine, Meta, Jure, Ana\}$   
Domena, ki obsega interval celih števil:  $C \equiv 1, 2, \dots, 200$  Domene atributov relacije





# Načrtovanje – Izdelava načrta PB

## Lastnosti relacij

- **Enoličnost:**
  - Ime relacije je enolično. V relacijski shemi podatkovne baze ni dveh relacij z enakim imenom;
  - Vsak atribut relacije ima enolično ime. V isti relaciji ni dveh atributov, ki bi imela isto ime;
  - Vsaka n-terica relacije je enolična → v relaciji ni dveh enakih n-teric.



# Načrtovanje – Izdelava načrta PB

## Lastnosti relacij

- **Atomarnost:**
  - Vsaka celica tabele, ki predstavlja relacijo, vsebuje natančno eno atomarno vrednost.
- **Zaloga vrednosti:**
  - Vrednosti nekega atributa so vse iz iste domene.
- **Nepomembnost vrstnega reda:**
  - Vrstni red atributov v relaciji je nepomemben.
  - Vrstni red n-teric v relaciji je nepomemben.



## Načrtovanje – Izdelava načrta PB

### Lastnosti relacij – primer

Ime	Starost (v letih), teža (v kg)
Tine	S15_T150
Meta	S20_T150
Jure	S40_T80
Ana	S5_T10

Celice ne vsebujejo atomarnih vrednosti

Zakonca	Leto poroke (celo število)
Tine, Meta	1995
Ana, Jure	1980

Celice vsebujejo več vrednosti

## Načrtovanje – Izdelava načrta PB

### Funkcionalne odvisnosti

- Predpostavimo, da obstaja relacijska shema  $R$  z množico atributov, katere podmnožici sta  $X$  in  $Y$ .
- V relacijski shemi  $R$  velja  $X \rightarrow Y$  ( $X$  funkcionalno določa  $Y$  oziroma  $Y$  je funkcionalno odvisen od  $X$ ), če v nobeni relaciji, ki pripada shemi  $R$ , ne obstajata dve  $n$ -terici, ki bi se ujemale v vrednostih atributov  $X$  in se ne bi ujemale v vrednostih atributov  $Y$ .

## Načrtovanje – Izdelava načrta PB

### Funkcionalne odvisnosti - primer

- Imamo relacijo s shemo
  - Izpit( VpŠt, Priimek, Ime, ŠifraPredmeta, DatumIzpita, OcenaPisno, OcenaUstno)
- z naslednjim pomenom:
  - Študent z vpisno številko VpŠt ter priimkom Priimek in imenom Ime je na DatumIzpita opravljal izpit iz predmeta s šifro ŠifraPredmeta. Dobil je oceno OcenaPisno in OcenaUstno.
- Funkcionalne odvisnosti sheme Izpit so:
  - $F \equiv \{ VpŠt \rightarrow (Priimek, Ime), (VpŠt, ŠifraPredmeta, DatumIzpita) \rightarrow (OcenaPisno, OcenaUstno) \}$



## Načrtovanje – Izdelava načrta PB

### Ključni relacije

- Ker je relacija množica n-teric, so v njej vse n-terice ločene med seboj.
- Za sklicevanje na posamezno n-terico ni potrebno poznati vseh vrednosti atributov n-terice, če v shemi nastopajo funkcionalne odvisnosti.
- Množici atributov, ki določajo vsako n-terico, pravimo **ključ relacije** oziroma ključ relacijske sheme.



## Načrtovanje – Izdelava načrta PB

### Ključni relacije

- Predpostavimo, da obstaja relacijska shema z atributi  $A_1, A_2, \dots, A_n$ , katere podmnožica je množica atributov  $X$ .
- Atributi  $X$  so ključ relacijske sheme oziroma pripadajočih relacij, če sta izpolnjena naslednja dva pogoja:
  - $X \rightarrow A_1, A_2, \dots, A_n$
  - ne obstaja  $X'$ , ki bi bila prava podmnožica od  $X$  in ki bi tudi funkcionalno določala  $A_1, A_2, \dots, A_n$



## Načrtovanje – Izdelava načrta PB

### Ključni relacije

- Poznamo več vrst ključev:
  - Kandidat za ključ (a key candidate)
  - Primarni ključ (primary key)
  - Superključ (superkey)
  - Tuji ključ (foreign key)
- Kandidat za ključ je vsaka podmnožica atributov relacije, ki relacijo enolično določa.



## Načrtovanje – Izdelava načrta PB

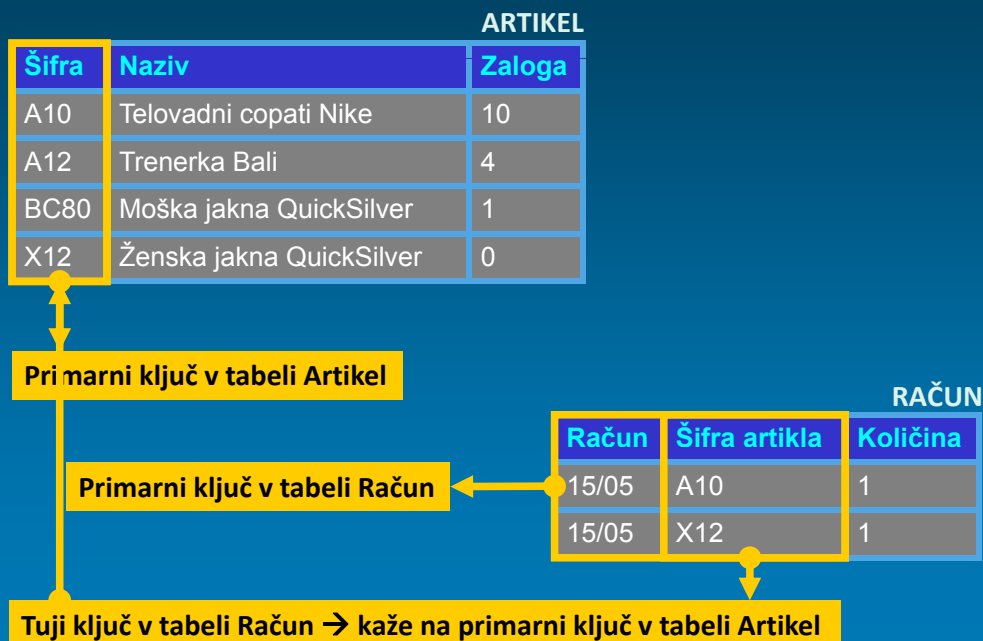
### Ključni relacije

- **Primarni ključ** je tisti kandidat za ključ, ki ga izberemo za shranjevanje relacij v fizični podatkovni bazi.
- **Superključ** je vsaka množica atributov, v kateri je vsebovan ključ → ključ je podmnožica superključa.
- **Tuji ključ** je množica atributov, v okviru ene relacije, ki je enaka kandidatu za ključ neke druge ali iste relacije.



## Načrtovanje – Izdelava načrta PB

### Ključni relacije primer



## Načrtovanje – Izdelava načrta PB

### Omejitve nad podatki

- Za celovitost ter skladnost podatkov v podatkovni bazi skrbimo s pomočjo omejitev.
- Poznamo več vrst omejitev:
  - Omejitve domene (Domain constraints)
  - Pravila za celovitost podatkov (Integrity constraints)
    - Celovitost entitet (Entity Integrity)
    - Celovitost povezav (Referential Integrity)
  - Števnost (Multiplicity)
  - Splošne omejitve (General constraints)



## Načrtovanje – Izdelava načrta PB

### Omejitve nad podatki

- **Omejitev entitete**
  - V osnovni relaciji ne sme biti noben atribut, ki je del ključa, enak Null.
- **Omejitve povezav**
  - Če v relaciji obstajajo tuji ključi, potem morajo:
    - (a) njihove vrednosti ustrezati tistim, ki so v obliki ključa zapisane v eni izmed n-teric neke druge ali iste relacije
    - (b) ali pa mora biti tuji ključ v celoti enak Null.



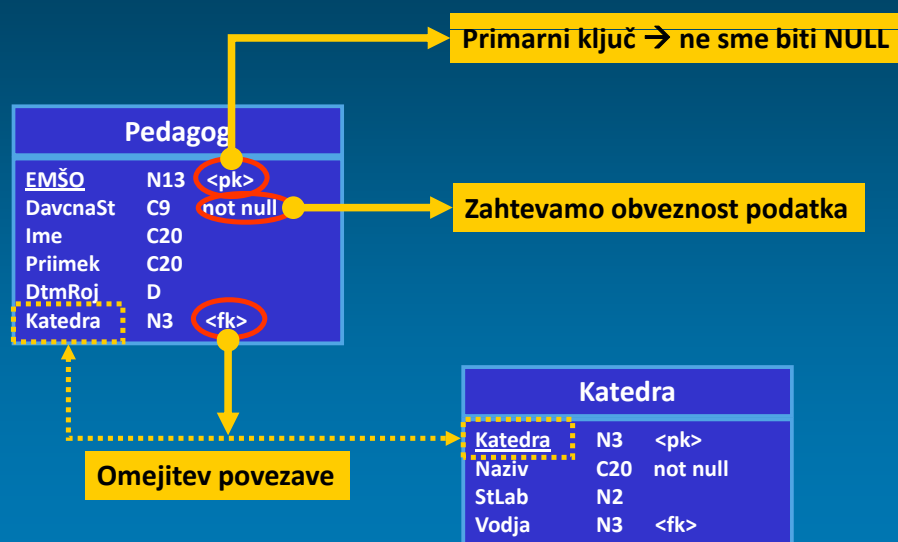
# Načrtovanje – Izdelava načrta PB

## Omejitve nad podatki

- Splošne omejitve
  - Dodatna pravila, ki jih določi uporabnik ali skrbnik podatkovne baze, ki definirajo ali omejujejo nek vidik področja, za katerega je narejena podatkovna baza.

# Načrtovanje – Izdelava načrta PB

## Omejitve nad podatki – primeri omejitev



## Načrtovanje – Izdelava načrta PB

### Pogledi

- Pogled je **navidezna relacija**, ki ne obstaja v relacijski bazi, temveč se dinamično kreira takrat, ko nekdo po njej povprašuje.
- Vsebina pogleda je definirana kot poizvedba nad eno ali več osnovnimi relacijami.
- Pogledi so **dinamični** → spremembe nad osnovnimi relacijami, katerih atributi so zajeti tudi v pogledu, so v pogledu takoj vidne.



## Načrtovanje – Izdelava načrta PB

### Namen uporabe pogledov

- **Zakaj uporabljamo poglede:**
  - **Varnost:** predstavljajo odličen mehanizem za zagotavljanje varnosti → skrivajo posamezne dele podatkovne baze pred določenimi uporabniki.
  - **Prilagojenost uporabnikom:** uporabnikom dajejo možnost, da do podatkov dostopajo na prilagojen način → isti podatki so lahko s strani različnih uporabnikov v istem času vidni na različne načine.
  - **Poenostavitev:** poenostavljajo kompleksne operacije nad osnovnimi relacijami.





## Načrtovanje – Izdelava načrta PB

### Spreminjanje pogledov

- Vse spremembe nad osnovnimi relacijami morajo biti takoj vidne tudi v pogledih nad temi relacijami.
- Če spremenimo podatke v pogledu, se morajo spremembe poznati tudi v osnovnih relacijah, na katere se te spremembe nanašajo.



## Načrtovanje – Izdelava načrta PB

### Spreminjanje pogledov

- V pogledih niso možne vse spremembe. Veljajo naslednje omejitve:
  - Nad pogledom so možne spremembe, če pogled zajema eno samo osnovno relacijo ter vključuje attribute, ki so kandidat za ključ relacije.
  - Če pogled zajema več relacij, spremembe niso možne.
  - Če je pogled pridobljen z agregacijo ali grupiranjem n-teric, spremembe niso možne.



## Načrtovanje – Izdelava načrta PB

### Primer pogleda

ARTIKEL			RAČUN		
Šifra	Naziv	Zaloga	Račun	Šifra artikla	Količina
A10	Telovadni copati Nike	10	15/05	A10	1
A12	Trenerka Bali	4	15/05	X12	1
BC80	Moška jakna QuickSilver	1	16/05	A10	3
X12	Ženska jakna QuickSilver	0	17/05	A10	1

```
SELECT A.sifra, A.naziv, sum(R.kolicina) AS Prodanih
FROM artikel A, racun R
WHERE A.sifra = R.sifra
GROUP BY A.sifra, A.naziv
```

Šifra	Naziv	Prodanih
A10	Telovadni copati Nike	5
X12	Ženska jakna QuickSilver	1
...		



## Načrtovanje – Izdelava načrta PB

### Normalizacija

- **Normalizacija** je postopek, s katerim pridemo do množice primernih relacij, ki ustrezajo potrebam poslovne domene.
- **Nekaj lastnosti primernih relacij:**
  - Relacije imajo minimalen nabor atributov → zgolj tiste, ki so potrebni za pokritje potreb poslovnega sistema;
  - Atributi, ki so logično povezani, so zajeti v isti relaciji;
  - Med atributi relacij je minimalna redundanca → vsak atribut (razen tujih ključev) je predstavljen samo enkrat.



# Načrtovanje – Izdelava načrta PB

## Normalizacija

- Prednosti uporabe podatkovnih baz, ki jih sestavljajo množice primernih relacij, so:
  - Enostavnejša dostop do podatkov ter vzdrževanje podatkov;
  - Večja učinkovitost;
  - Boljša izraba diskovnih kapacitet.



# Načrtovanje – Izdelava načrta PB

## Normalizacija – ažurne anomalije

- Relacije, ki vsebujejo odvečne podatke lahko povzročajo anomalije pri spreminjanju podatkov → govorimo o **ažurnih anomalijah**.
- Poznamo več vrst anomalij:
  - Anomalije pri dodajanju n-teric v relacijo
  - Anomalije pri brisanju n-teric iz relacije
  - Anomalije pri spreminjanju n-teric



## Načrtovanje – Izdelava načrta PB

### Normalizacija – ažurne anomalije pri dodajanju

- **Primeri anomalij:**
  - Če želimo dodati podatke o novih članih (staff) za neko organizacijsko enoto (branch) moramo vpisati tudi vse podrobnosti o organizacijski enoti.
  - Če želimo dodati podatke o novi organizacijski enoti, ki še nima nobenega člana, moramo v vsa polja, ki člane opisujejo, vpisati Null.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

## Načrtovanje – Izdelava načrta PB

### Normalizacija – ažurne anomalije pri brisanju

- **Primeri anomalij:**
  - Če iz relacije zbrisemo n-terico, ki predstavlja zadnjega člana v neki organizacijski enoti, zgubimo tudi podatke o tej organizacijski enoti.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

## Načrtovanje – Izdelava načrta PB

### Normalizacija – ažurne anomalije pri spreminjanju

- **Primeri anomalij:**
  - Če želimo spremeniti vrednost nekega atributa določene organizacijske enote (npr. naslov), moramo popraviti vse n-terice, v katerih takšna vrednost atributa nastopa.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London



## Načrtovanje – Izdelava načrta PB

### Normalne oblike

- Postopku preoblikovanja relacij v obliko, pri kateri do ažurnih anomalij ne more priti, pravimo **normalizacija**.
- **Obstaja več stopenj normalnih oblik:**
  - 1NO – Prva normalna oblika
  - 2NO – Druga normalna oblika
  - 3NO – Tretja normalna oblika in
  - 4PNO – Četrta poslovna normalna oblika
  - BCNO – Boyce Codova normalna oblika
  - 4NO – Četrta normalna oblika
  - 5NO – Peta normalna oblika



# Načrtovanje – Izdelava načrta PB

## Normalizacija – prva normalna oblika

- **Relacija je v prvi normalni obliki, če:**
  - Nima ponavljajočih atributov → ne obstajajo atributi ali skupine atributov, ki bi imele več vrednosti pri isti vrednosti ostalih atributov (na presečišču ene vrstice in enega stolpca je več vrednosti)
  - Ima definiran primarni ključ in določene funkcionalne odvisnosti
- **Koraki:**
  - Odstranimo ponavljajoče attribute
  - Določimo funkcionalne odvisnosti
  - Določimo primarni ključ



# Načrtovanje – Izdelava načrta PB

Indeks( VŠ, priimek, ime, pošta, kraj, ( šifra predmeta, naziv, ocena ) )

Skupina ponavljajočih se atributov.

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
					20021	TPO	8
					20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
					20033	IPI	6



## Načrtovanje – Izdelava načrta PB

Indeks( VŠ, priimek, ime, pošta, kraj, ( šifra predmeta, naziv, ocena ) )



Odpravimo ponavljajoče attribute

Indeks( VŠ, priimek, ime, pošta, kraj, šifra predmeta, naziv, ocena )



Identificiramo funkcionalne odvisnosti

$F \equiv \{ V\check{S} \rightarrow (\text{priimek, ime, pošta, kraj}), \text{šifra predmeta} \rightarrow \text{naziv, pošta} \rightarrow \text{kraj}, (V\check{S}, \text{šifra predmeta}) \rightarrow \text{ocena} \}$



Določimo primarni ključ

Indeks( VŠ, priimek, ime, pošta, kraj, šifra predmeta, naziv, ocena )



## Načrtovanje – Izdelava načrta PB

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
64010632	Bratina	Simon	4100	Kranj	20021	TPO	8
64010632	Bratina	Simon	4100	Kranj	20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
64016209	Bizjak	Tadeja	2250	Ptuj	20033	IPI	6



VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
					20021	TPO	8
					20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
					20033	IPI	6



# Načrtovanje – Izdelava načrta PB

Normalizacija – druga normalna oblika

- **Relacija je v drugi normalni obliki:**
  - Če je v prvi normalni obliki in
  - Ne vsebuje parcialnih odvisnosti → noben atribut, ki ni del ključa, ni funkcionalno odvisen le od dela primarnega ključa, temveč od celotnega ključa



# Načrtovanje – Izdelava načrta PB



**Relacijo razbijemo**

Študent( VŠ, priimek, ime, pošta, kraj )  
Predmet( šifra predmeta, naziv )  
Indeks( #VŠ, #šifra predmeta, ocena )





## Načrtovanje – Izdelava načrta PB

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
64010632	Bratina	Simon	4100	Kranj	20021	TPO	8
64010632	Bratina	Simon	4100	Kranj	20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
64016209	Bizjak	Tadeja	2250	Ptuj	20033	IPI	6



šifra predmeta	naziv
20020	IS
20021	TPO
20033	IPI
20060	E1
20033	IPI

VŠ	šifra predmeta	ocena
64010632	20020	10
64010632	20021	8
64010632	20033	8
64016209	20060	9
64016209	20033	6

VŠ	priimek	ime	pošta	kraj
64010632	Bratina	Simon	4100	Kranj
64016209	Bizjak	Tadeja	2250	Ptuj



## Načrtovanje – Izdelava načrta PB

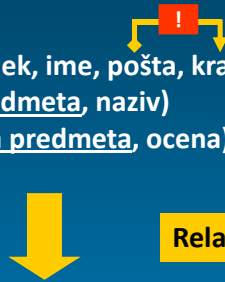
### Normalizacija – tretja normalna oblika

- **Relacija je v tretji normalni obliki:**
  - Če je v drugi normalni obliki in
  - Če ne vsebuje tranzitivnih funkcionalnih odvisnosti → med atributi, ki niso del primarnega ključa, ni odvisnosti.



## Načrtovanje – Izdelava načrta PB

Študent( VŠ, priimek, ime, pošta, kraj)  
Predmet( šifra predmeta, naziv)  
Indeks( #VŠ, #šifra predmeta, ocena)



Relacijo razbijemo

Študent( VŠ, priimek, ime, #pošta)  
Pošta(pošta, kraj)  
Predmet( šifra predmeta, naziv)  
Indeks( #VŠ, #šifra predmeta, ocena)



## Načrtovanje – Izdelava načrta PB

VŠ	priimek	ime	pošta	kraj
64010632	Bratina	Simon	4100	Kranj
64016209	Bizjak	Tadeja	2250	Ptuj



VŠ	priimek	ime	pošta
64010632	Bratina	Simon	4100
64016209	Bizjak	Tadeja	2250

pošta	kraj
4100	Kranj
2250	Ptuj



# Načrtovanje – Izdelava načrta PB

Normalizacija – četrta poslovna normalna oblika

- Relacija je v četrthi poslovni normalni obliki, če:
  - je v tretji normalni obliki in
  - v relaciji ne obstajajo atributi, ki bi bili odvisni od vrednosti primarnega ključa.



# Načrtovanje – Izdelava načrta PB

Študent( VŠ, priimek, ime, #pošta, datum plačila šolnine, rok diplome)



Za izredne študenta

Za redne študenta

Študent( VŠ, priimek, ime, #pošta)  
Redni študent( #VŠ, rok diplome)  
Izredni študent( #VŠ, datum plačila šolnine)



## Načrtovanje – Izdelava načrta PB

VŠ	Priimek	Ime	Datum plačila šolnine	Rok diplome
64010632	Bratina	Simon		15.3.2005
64016209	Bizjak	Tadeja	19.4.2002	
64010670	Berce	Marjan	12.4.2004	
64620010	Mele	Silvana		1.4.2005
65120987	Leban	Tibor		15.7.2005



VŠ	Priimek	Ime
64010632	Bratina	Simon
64016209	Bizjak	Tadeja
64010670	Berce	Marjan
64620010	Mele	Silvana
65120987	Leban	Tibor

VŠ	Datum plačila šolnine
64016209	19.4.2002
64010670	12.4.2004

VŠ	Rok diplome
64010632	15.3.2005
64620010	1.4.2005
65120987	15.7.2005



## Načrtovanje – Izdelava načrta PB

Tipični koraki pri izdelavi relacijskega modela

Ponovitev

- **Tipični koraki:**
  - Za entitetne tipe kreiraj relacije
  - Preveri relacije z normalizacijo
  - Preveri relacije s pregledom uporabniških transakcij
  - Preveri omejitve integritete
  - Preveri model z uporabnikom
  - Združi lokalne modele v globalni model (opcijsko)
  - Preveri zmožnosti modela za razširitve



# Načrtovanje – Izdelava načrta PB

Primer pretvorbe konceptualnega v relacijski model

Študent
<u>Vpisna številka</u>
Priimek
Ime
Naslov
Telefon
E-mail
Status
...

Izpit
<u>Zap. št. polaganja</u>
Ocena pisno
Ocena ustno
Datum ocene

Izpit		
ZapStPol	N2	<pk>
VpisSt	N8	<pk, fk>
OcPisno	N2,2	
OcUstno	N2,2	
DatumOc	D	

Student		
VpisSt	N8	<pk>
Priimek	C20	
Ime	C20	
Ulica	C25	
Posta	N4	
Drzava	C20	
GSM	N15	
Tel	N15	
Email	C25	null
Status	N1	
...		

# Načrtovanje – Izdelava načrta PB

Primer pretvorbe konceptualnega v relacijski model

## ▪ Dodatno poskrbimo za:

- Indekse
- Integriteto povezav
- Druge omejitve
- Pogleda
- ...

Izpit		
ZapStPol	N2	<pk>
VpisSt	N8	<pk, fk>
OcPisno	N2,2	
OcUstno	N2,2	
DatumOc	D	

Student		
VpisSt	N8	<pk>
Priimek	C20	
Ime	C20	
Ulica	C25	
Posta	N4	
Drzava	C20	
GSM	N15	
Tel	N15	
Email	C25	null
Status	N1	
...		



# Razvoj po strukturnem pristopu

Kje smo?

## Postopki strukturnega pristopa:

- Zajem in specifikacija zahtev;
- Analiza

### ➔ Načrtovanje

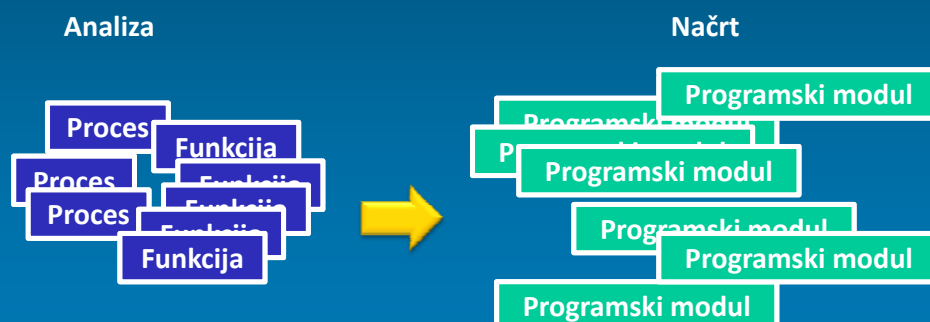
- Izdelava načrta podatkovne baze,
- ➔ Izdelava načrta programskih modulov,
- Izdelava načrta dokumentacije,
- Izdelava načrta testiranja,
- Izdelava načrta namestitve in uvedbe in
- Predstavitev načrta naročniku.
- Izvedba;
- Testiranje;
- Namestitev in uvedba.



# Načrtovanje – izdelava načrta modulov

Namen aktivnosti

- Namen aktivnosti je prikazati, kako bodo posamezne funkcije in procesi, identificirani v sklopu analize, realizirani v okviru rešitve.



# Načrtovanje – izdelava načrta modulov

## Namen aktivnosti

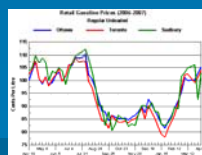
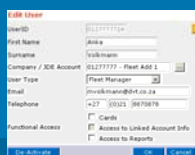
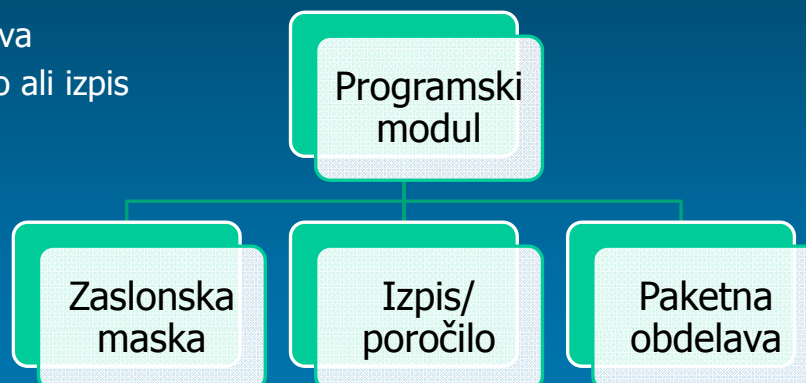
- Funkcije in procesi iz analize predstavljajo **logične sklope sistema**. V fazi načrtovanja jih pretvorimo v **fizične oziroma programske sklope ali module**.
- Pretvorba ni nujno 1:1
  - Implementacija enega logičnega sklopa je lahko izvedena z več programskimi sklopi.
  - En programski sklop lahko implementira več logičnih enot.



# Načrtovanje – izdelava načrta modulov

## Programski moduli

- Tipi programskih modulov:
  - Zaslonska maska
  - Obdelava
  - Poročilo ali izpis



# Načrtovanje – izdelava načrta modulov

## Strukturni diagram

- Strukturo programskih modulov prikažemo s pomočjo **strukturnega diagrama**.
- **Lastnosti strukturnih diagramov:**
  - Prikazuje programske module, s katerih bo sestavljena informacijska rešitev;
  - Prikazuje odvisnost med programskimi moduli ter podatke, ki se med moduli prenašajo;
  - Omogoča prikaz zaporedja, izbire in ponavljanja.



# Načrtovanje – izdelava načrta modulov

## Strukturni diagram

- **Lastnosti strukturnih diagramov:**
  - Moduli so organizirani v hierarhijo, podobno kot funkcije v funkcionalni razgradnji.
  - Na najvišjem mestu je vseobsegajoč modul ali koren. Na naslednjem nivoju so moduli, ki jih koren lahko kliče (analogno kot izbire v meniju).
  - Moduli komunicirajo med seboj s pomočjo parametrov:
    - nosilci podatkov
    - kontrolne zastavice





# Načrtovanje – izdelava načrta modulov

Strukturni diagram – primer



## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza
  - ➔ Načrtovanje
    - Izdelava načrta podatkovne baze,
    - Izdelava načrta programskih modulov,
    - ➔ Izdelava načrta dokumentacije,
    - Izdelava načrta testiranja,
    - Izdelava načrta namestitve in uvedbe in
    - Predstavitev načrta naročniku.
  - Izvedba;
  - Testiranje;
  - Namestitev in uvedba.

## Načrtovanje – načrt dokumentacije

### Izdelava načrta dokumentacije

- **Namen aktivnosti** je določiti obseg in strukturo dokumentacije ter izbrati ustrezne standarde in vzorce za dokumentacijo.
- Upoštevamo zahteve naročnika iz zajema in specifikacije zahtev.
- Določimo tudi vire za dokumentacijo.



## Načrtovanje – načrt dokumentacije

### Izdelava načrta dokumentacije

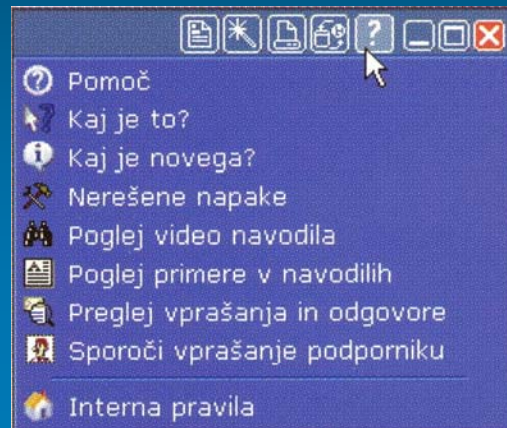
- **Potrebno upoštevati, da lahko nekatere dele dokumentacije izdelamo šele, ko je del sistema, ki ga dokumentiramo, izdelan:**
  - Npr. posnetke zaslonov za uporabniško dokumentacijo lahko izdelamo šele, ko je uporabniški vmesnik izdelan in smo prepričani, da se ne bo več spremenjal.
- **V grobem lahko dokumentacijo razdelimo na:**
  - uporabniško dokumentacijo,
  - sistemsko-tehnično dokumentacijo ter
  - navodila za operativno skrbništvo.



# Načrtovanje – načrt dokumentacije

## Izdelava načrta dokumentacije

- **Uporabniška dokumentacija** je osnovna pomoč za uporabnike oziroma uporabo rešitve.
- Možne različne oblike...



Vir: ERP sistem Pantheon

# Načrtovanje – načrt dokumentacije

## Izdelava načrta dokumentacije

- **Sistemska-tehnična dokumentacija** dokumentira informacijsko rešitev s sistemsko-tehničnega vidika. Npr.:
  - Podatkovni model;
  - Arhitektura sistema;
  - Komponente sistema;
  - Opis testnega, produkcijskega in razvojnega okolja;
  - Opis sistemske programske opreme in druge infrastrukture, ki je potrebna za delovanje sistema;
  - ...

# Načrtovanje – načrt dokumentacije

## Izdelava načrta dokumentacije

- Navodila za operativno skrbništvo zajemajo opis ključnih postopkov, ki so potrebni za operativno delovanje sistema. Npr:
  - Postopek izdelave varnostnih kopij;
  - Postopek ustavitve sistema;
  - Postopek ponovnega zagona sistema;
  - Postopek namestitve nadgradenj in popravkov;
  - Postopek vzpostavitve nadomestnega sistema;
  - ...



# Načrtovanje – načrt dokumentacije

## Izdelava načrta dokumentacije

- V okviru načrta dokumentacije izdelamo vzorce dokumentacije in jih predstaviti naročniku ter uporabniku.
- Cilj je uskladitev o obliki dokumentacije.
- Nivo podrobnosti in obseg dokumentacije je odvisen od dogovora med izvajalcem in naročnikom. Praviloma se podrobneje dokumentirajo samo pomembnejši podsistemi in kritični postopki.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza
  - ➔ Načrtovanje
    - Izdelava načrta podatkovne baze,
    - Izdelava načrta programskih modulov,
    - Izdelava načrta dokumentacije,
    - ➔ Izdelava načrta testiranja,
      - Izdelava načrta namestitve in uvedbe in
      - Predstavitev načrta naročniku.
  - Izvedba;
  - Testiranje;
  - Namestitev in uvedba.



## Načrtovanje – načrt testiranja

Izdelava načrta testiranja

- Testiranje, ki intenzivno poteka v fazi izvedbe ter v okviru namestitve in uvedbe sistema, je nujno predhodno načrtovati.
- Načrt testiranja zajema:
  - Načrt poteka testiranja
  - Načrte za izvedbo posameznih testov



# Načrtovanje – načrt testiranja

## Načrt poteka testiranja

- Osnovni potek testiranja, vrste testov in vsebino testiranja določa **postopek testiranja**.
- Z **načrtom poteka testiranja**, ki ga izdelamo v sklopu načrtovanja, podrobneje določimo zaporedje izvajanja posameznih testov po sklopih informacijske rešitve in po okoljih.
- Načrt poteka testiranja je **odvisen** od načrta programskih modulov.



# Načrtovanje – načrt testiranja

## Načrt za izvedbo testa

- Z **načrti za izvedbo posameznih testov** podrobneje določimo, kaj bomo v okviru posameznega testa, ki je predviden v načrtu poteka testiranja, testirali.
- V odvisnosti od namena testiranja, faze testiranja in okolja testiranja podrobneje določimo vsebino testiranja
  - npr. testiranje funkcionalnosti, testiranje GUI, testiranje vmesnikov, testiranje prevedbe podatkov, testiranje drugih nefunkcionalnih zahtev IR ipd.



# Načrtovanje – načrt testiranja

## Načrt za izvedbo testa

- **Načrt za izvedbo testa mora opredeliti vsaj:**
  - **Namen testiranja:** čemu je namenjena izvedba testa? (npr. testiranje programske enote »Prijava na izpit«)
  - **Faza testiranja:** v kateri fazi izvajamo testiranje? (npr. testiranje v fazi razvoja ali testiranje v fazi namestitve in uvedbe)
  - **Okolje testiranja:** v katerem okolju je potrebno test izvesti? (npr. v testnem okolju)
  - **Področje testiranja:** kaj se s testom testira? (npr. funkcionalnost sklopa, integracija sklopa, GUI,...)
  - **Način izvedbe testiranja:** podroben opis, ki pove, kako se izvede test. Opis je odvisen od področja testiranja.



# Načrtovanje – načrt testiranja

## Uporaba scenarijev

- Pri izdelavi načrtov za izvedbo posameznih testov si pomagamo s **testnimi scenariji**.
- Testni scenariji opisujejo postopke uporabe sistema in so zato dobro vodilo za testiranje funkcionalnosti IR.
- V okviru testiranja po izbranem scenariju spremljamo uporabo in spreminjanje izbranih podatkov, na osnovi česar lahko ob zaključku testa bodisi **potrdimo** ali **ovržemo** pravilnost delovanja modulov/ sklopov, ki so bili v testu vključeni.



# Načrtovanje – načrt testiranja

## Uporaba scenarijev

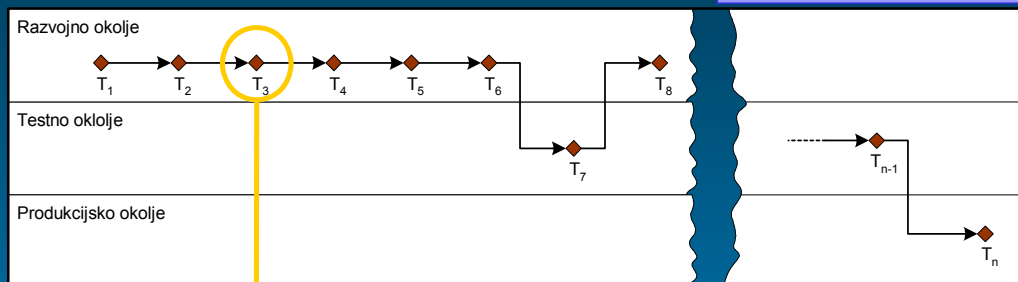
- Za izvedbo testa po nekem scenariju je potrebno določiti vhodne podatke ter pričakovane rezultate.



# Načrtovanje – načrt testiranja

## Opredelitev scenarija

## Načrt poteka testiranja



### Scenarij:

- Vhodni podatki
- Pričakovani rezultati
- Načrt za izvedbo testa

Test T<sub>3</sub>





# Načrtovanje – načrt testiranja

## Načrt testiranja – izdelek

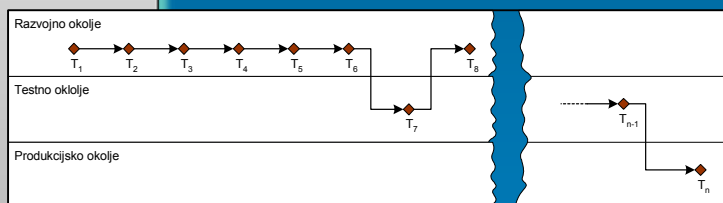
- Izdelek načrt testiranja je sestavljen iz več dokumentov:
  - Dokument, ki podaja potek testiranja.
  - Dokument, ki podaja scenarije za testiranje funkcionalnih in nefunkcionalnih zahtev sistema.
  - Dokumenti, ki podrobneje opredeljujejo izvedbo posameznih testov, ki jih načrt poteka testiranja predvideva.

# Načrtovanje – načrt testiranja

## Načrt testiranja – primer načrta poteka testiranja

### Načrt poteka testiranja

- T1:** sklop: prijava na izpit  
okolje: razvoj
- T2:** sklop: odjava iz izpita  
okolje: razvoj
- T3:** sklop: pregled števila prijavljenih kandidatov  
okolje: razvoj
- T4:** sklop: Izpis seznama prijavljenih kandidatov  
okolje: razvoj
- T5:** sklop: Vnos rezultatov  
okolje: razvoj
- T6:** sklop: Objava rezultatov  
okolje: razvoj
- T7:** sklop: Opravljanje pisnih izpitov (integracijski test)  
okolje: testno
- T8:** sklop: Vnos obvestil  
okolje: razvoj
- ....
- Tn-1:** sklop: celoten sistem (potrditveni test)  
okolje: testno
- Tn:** sklop: celoten sistem (končni test)  
okolje: produkcijsko



# Načrtovanje – načrt testiranja

## Načrt testiranja – primer načrta za izvedbo testa

### NAČRT TESTIRANJA ZA TEST T2

**Namen testiranja:** testiranje sklopa odjava iz izpita

**Faza testiranja:** testiranje v fazi razvoja

**Okolje testiranja:** razvojno okolje

**Področje testiranja:** funkcionalnost sklopa, GUI

#### Način izvedbe testiranja:

1. Testiranje funkcionalnosti: testirati po scenarijih S1, S3, S4, S8
2. Testiranje GUI: preveriti skladnost elementov GUI z načrtom programskega modula/podsistema PM3



# Načrtovanje – načrt testiranja

## Načrt testiranja – primer scenarija

### OZNAKA SCENARIJA: S3

**Kratek opis:** odjava iz izpitnega roka, če je rok za odjavo že potekel

#### Koraki scenarija:

1. v sistem se prijaviti kot študent [Š]
2. prijaviti se na razpisan rok za izbran datum in predmet [I]
3. na strežniku spremeniti sistemski datum na trenutni datum [D]
4. poskusiti se odjaviti iz izpita [I]

**Vhodni podatki:** Š, I, D

#### Pričakovani rezultati:

sistem ne dovoli odjave iz izpita [I]. Če študent izbere opcijo Odjava iz izpita, dobi sporočilo »Iz izpita se ne morete odjaviti. Rok za odjavo je že potekel!«.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza
  - ➔ Načrtovanje
    - Izdelava načrta podatkovne baze,
    - Izdelava načrta programskih modulov,
    - Izdelava načrta dokumentacije,
    - Izdelava načrta testiranja,
    - ➔ Izdelava načrta namestitve in uvedbe in
      - Predstavitev načrta naročniku.
  - Izvedba;
  - Testiranje;
  - Namestitev in uvedba.



## Načrtovanje – načrt namestitve in uvedbe

Načrt namestitve in uvedbe

- Naloga aktivnosti je izdelati **načrt namestitve in uvedbe IR** v razvojno, testno in produkcijsko okolje ter uvedbo uporabnikov in skrbnikov za delo z IR.
- Načrt namestitve in uvedbe v razvojnem okolju pripravijo člani projektne skupine razvijalca, pri izdelavi načrta namestitve in uvedbe v testnem oz. produkcijskem okolju pa sodelujejo tudi predstavniki končnih uporabnikov.



# Načrtovanje – načrt namestitve in uvedbe

## Načrt namestitve in uvedbe

- **V načrtu potrebno opredeliti:**
  - namen in cilj namestitve in uvedbe,
  - zahteve okolja za namestitev in uvedbo (pogoji za izvedbo namestitve in uvedbe, potrebna strojna oprema, potrebna sistemska programska oprema, povezovanje z ostalo aplikativno opremo, potrebna dokumentacija),
  - naloge namestitve in uvedbe (funkcionalne naloge, administrativne naloge),
  - udeležence namestitve in uvedbe, njihove odgovornosti ter vključitev uporabnikov in skrbnikov,
  - opis sestave paketa za namestitev IR (določitev in označitev distribucijskih medijev - CD-ROM, diskete, spletne strežniške datoteke, priložena dokumentacija),



# Načrtovanje – načrt namestitve in uvedbe

## Načrt namestitve in uvedbe

- **V načrtu potrebno opredeliti (nadaljevanje):**
  - opis procesa namestitve in uvedbe (način izvedbe faz namestitve, dodelitev pravic za delo, opis prevedbe podatkov, opis načina uvajanja uporabnikov in skrbnikov, opis izvedbe potrditvenega testa IR, opis prehoda na nov sistem),
  - merila za uspešno namestitev in uvedbo (ključne točke za uspešno opravljeno namestitev, seznam ali opis pričakovanih rezultatov namestitve in uvedbe in dovoljenih odstopanj),
  - vrednotenje ugotovljenih napak ali pomanjkljivosti pri postopku namestitve in uvedbe,
  - potrditev oz. odobritev rezultatov namestitve in uvedbe.



# Načrtovanje – načrt namestitve in uvedbe

## Načrt namestitve in uvedbe

- Načrt namestitve in uvedbe se sestoji iz:
  - načrta namestitve IR;
  - načrta dodelitve pravic;
  - načrta prevedbe podatkov;
  - načrta uvajanja
  - načrta za izvedbo potrditvenega ter končnega testa IR
  - načrta prehoda na nov sistem



# Razvoj po strukturnem pristopu

## Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza
  - ➔ Načrtovanje
    - Izdelava načrta podatkovne baze,
    - Izdelava načrta programskih modulov,
    - Izdelava načrta dokumentacije,
    - Izdelava načrta testiranja,
    - Izdelava načrta namestitve in uvedbe in
  - ➔ Predstavitev načrta naročniku.
  - Izvedba;
  - Testiranje;
  - Namestitev in uvedba



## Načrtovanje – predstavitev načrta

### Predstavitev načrta naročniku

- **Osnovni namen** predstavitve je pridobitev potrditve s strani naročnika o ustreznosti načrta.
- Pri predstavitvi je potrebno upoštevati, da naročnik navadno nima strokovnih znanj, ki bi mu omogočala poglobljeno razumevanje načrta, hkrati pa to tudi ni njegova naloga.
- Naročniku predstavimo le osnovne elemente načrta.



## Načrtovanje – predstavitev načrta

### Predstavitev načrta naročniku

- Primeren za podrobno predstavitev je **načrt uporabniškega vmesnika** → posredno razkriva tudi druge elemente načrta.
- Na osnovi predstavitve lahko naročnik opozori na pomanjkljivosti oz. izrazi dodatne želje.
- Aktivnost se zaključi s **potrditvijo** naročnika (navadno podpis dokumenta), da načrt ustreza zahtevam.





## Razvoj po strukturnem pristopu

Kje smo?

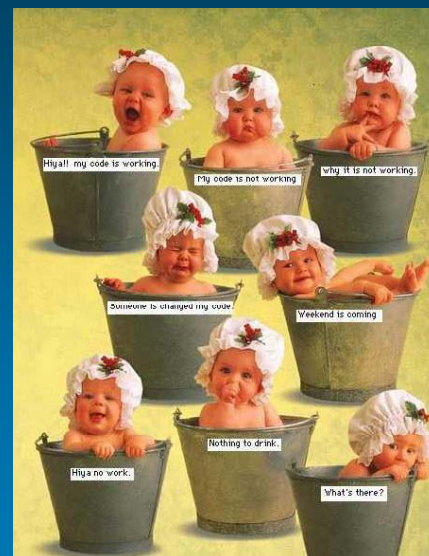
- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - ➔ Testiranje;
  - Namestitev in uvedba.



## Testiranje

Opredeitev in namen

- Glavni namen testiranja je zagotoviti, da IR deluje tako, kot smo načrtovali.
- Postopek testiranja se prepleta skozi ves življenjski cikel razvoja IR: analiza, načrtovanje, izvedba in uvedba.



Podroben opis aktivnosti testiranja je podan po posameznih postopkih.



# Testiranje

## Končni izdelek 📦

- Končen izdelek testiranja je preverjena in delujoča IR.
- Skozi potek testiranja nastane tudi več drugih izdelkov.
  - Opisani so pri aktivnostih, kjer nastanejo (glej postopke analiza, načrtovanje in uvedba).



# Testiranje

## Vloge 👤

- Testiranje se izvaja na različnih ravneh.
- Prvo testiranje izvaja že sam razvijalec neposredno ob razvoju. Sledi sistematično testiranje s strani preizkuševalca, zatem pa še testiranje s strani končnega uporabnika.





# Testiranje

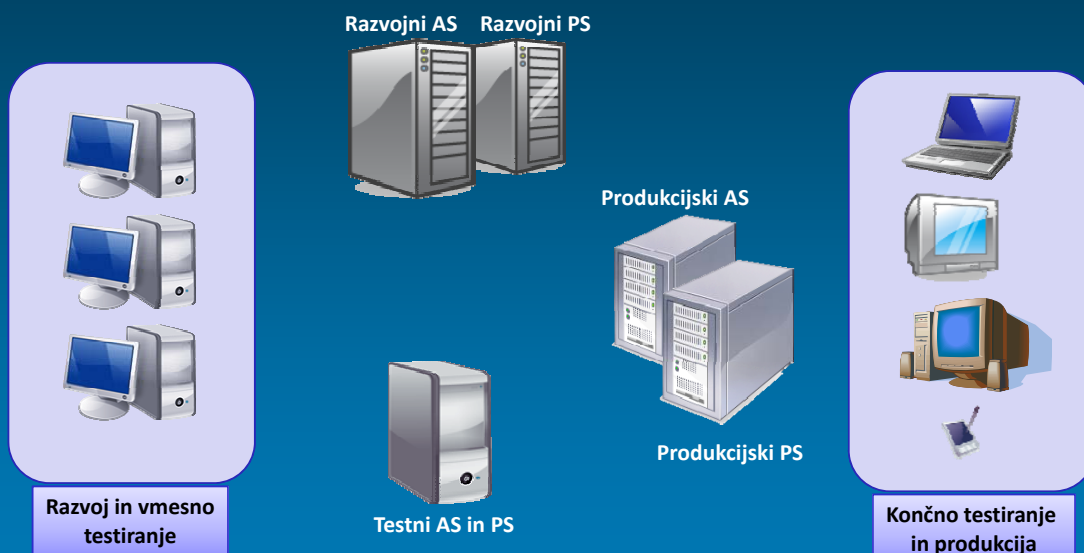
## Okolje za testiranje

- Za zagotovitev ustrezne ravni varnosti potrebno za testiranje in produkcijo zagotoviti **ločena okolja**.
- Ločena okolja pogosto težko (drago) zagotoviti.
- Najboljši pristop je uporaba ločene strojne opreme.
- Producersko okolje vedno namestimo na ločeno strojno opremo.



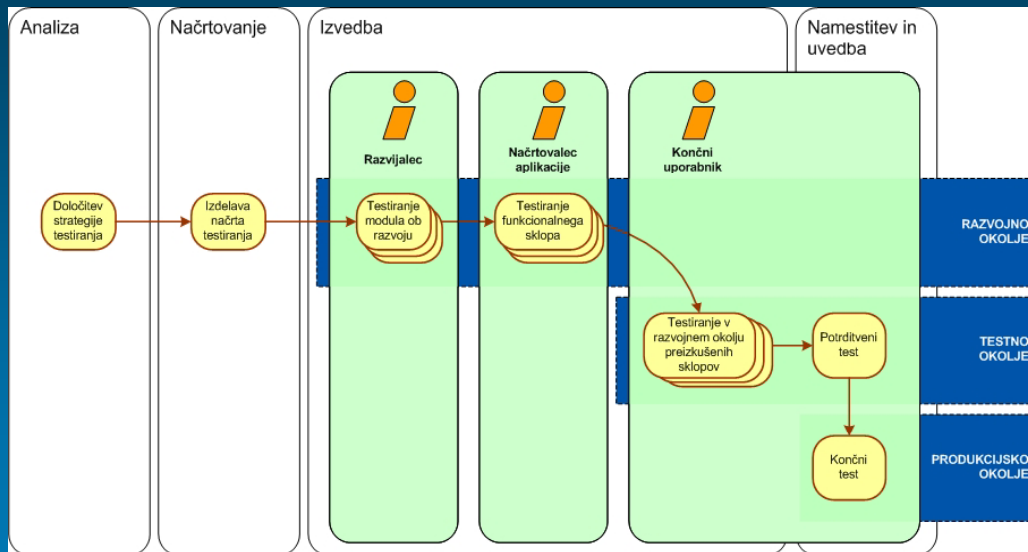
# Testiranje

## Primer okolja za testiranje



# Testiranje

## Schema postopka ▶



# Testiranje

## Vrste in potek testiranja

- Testiranje poteka na različnih ravneh, v različnih okoljih in s strani različnih vlog.
- Vrste testov:
  - Test programskih enot
  - Test integracije
  - Sistemski test
  - Test sprejemljivosti - Potrditveni test
  - Test sprejemljivosti - Končni test

# Testiranje

## Vrste in potek testiranja

Vrsta testa	Vsebina testiranja	Okolje testiranja	Vloge
Test programskih enot	Uporabniški vmesnik, funkcionalnost programske enote, skladnost z zahtevami in standardi	Razvojno okolje, Testno okolje	Razvijalec, Preizkuševalec
Test integracije	Test integracije programskih enot, test integracije v okolje (vmesniki)	Testno okolje	Integrator, Preizkuševalec, Ključni uporabnik
Sistemi test	Obnašanje v okolju, zmogljivosti, dostopnost, nefunkcionalne zahteve	Testno okolje	Skrbnik podatkovne baze, Sistemski administrator, Skrbnik aplikacije, Preizkuševalec, Ključni uporabnik
Test sprejemljivosti - Potrditveni test	Preverjanje delovanja celotne funkcionalnosti, preverjanje nefunkcionalnih zahtev	Testno okolje	Ključni uporabnik
Test sprejemljivosti - Končni test	Test funkcionalnosti v omejenem obsegu v produkcijskem okolju	Produkcijsko okolje	Ključni uporabnik

# Testiranje

## Vrste in potek testiranja

- **Testiranje programski enot**
  - Testiranje programskih enot je osnovno testiranje, osredotočeno na ustreznost uporabniškega vmesnika in funkcionalnosti programske enote glede na podane zahteve in standarde.
  - Izvaja v razvojnem okolju. Najprej se s testiranjem ukvarjajo razvijalci, ko sami testirajo module oziroma programske enote, ki so jih razvili.
  - Testiranje s strani razvijalcev je navadno nesistematično in se ne izvaja po načrtu.

# Testiranje

## Vrste in potek testiranja

- **Testiranje programski enot (nadaljevanje)**
  - Ko razvitih več programskih enot (funkcionalni sklop), testiranje prevzame preizkuševalec. Skladno z načrtom testiranja preveri pravilnost funkcionalnega sklopa, preden se le-ta namesti v testno okolje.
  - Testiranje programskih enot se izvaja tudi v testnem okolju. Sodelujeta preizkuševalec in ključni uporabnik. Ustreznost sklopa potrди ključni uporabnik.
  - Če je moč testiranje v testnem in razvojnem okolju izvajamo vzporedno.



# Testiranje

## Vrste in potek testiranja

- **Testiranje integracije**
  - Testiranje integracije namenjeno testiranju povezav med programskimi enotami ter testiranju vmesnikov - povezav IR z okoljem.
  - Testiranje integracije poteka v več korakih, vzporedno z razvojem. Testiranje vmesnikov z okoljem navadno izvedemo na koncu za celoten sistem.
  - Testiranje integracije poteka v testnem okolju. Pri tem sodelujejo integrator, preizkuševalec in ključni uporabniki.
  - Testiranje se izvaja po pripravljenem načrtu in se zaključí z ustrežno potrditvijo.



# Testiranje

## Vrste in potek testiranja

- **Testiranje sistema**
  - Testiranje sistema namenjeno testiranju obnašanja sistema kot celote ter njegovega delovanja v okolju. Poudarek na testiranju nefunkcionalnih zahtev kot na primer: zmogljivost, dostopnost, test pod obremenitvijo ipd.
  - Test se izvaja v testnem okolju. Izvede se enkrat, in sicer takrat, ko je v testnem okolju nameščen celoten sistem in povezan z okoljem. Pri izvedbi testa sodelujejo: skrbnik podatkovne baze, sistemski administrator, skrbnik aplikacije, preizkuševalec in ključni uporabniki.



# Testiranje

## Vrste in potek testiranja

- **Testiranje sprejemljivosti**
  - Test sprejemljivosti se deli na **potrditveni test** in **končni test**.
  - Potrditveni test namenjen testiranju celotne IR, tako z vidika funkcionalnih kot nefunkcionalnih zahtev, z namenom pridobitve potrdila o njegovi ustreznosti. Test se izvaja v testnem okolju v sklopu postopka namestitve in uvedbe IR. Test izvaja ključni uporabnik po pripravljenem načrtu.
  - Končni test je zadnji test IR. Izvedemo ga v produkcijskem okolju z namenom, da se prepričamo, da pri postopku namestitve ni prišlo do kakršnihkoli napak. Končni test izvedemo s pomočjo produkcijskih oziroma pravih primerov. Izvaja ga ključni uporabnik.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
  - ➔ Namestitev in uvedba.



## Namestitev in uvedba

Opredelitev in namen

- Glavni namen namestitve in uvedbe je
  - namestitev izbrane IR ali njenih delov v testno ali produkcijsko okolje ter;
  - izvedba potrditvenega in končnega testa IR;
  - uvedba uporabnikov, skrbnikov in drugih, ki bodo z IR delali, za delo z novo IR.
- Z izvedbo postopka zagotovimo, da lahko uporabniki nemoteno uporabljajo novo IR.



## Namestitev in uvedba

### Končni izdelek 📦

- Končen izdelek namestitve in uvedbe sta nameščena IR v produkcijsko okolje in uvedeni uporabniki.
- Ko je nova IR nameščena v produkcijskem okolju in so uporabniki usposobljeni za uporabo nove IR, se uporaba nove IR lahko prične.



## Namestitev in uvedba

### Vloge in koraki 👤

- Aktivnosti v okviru postopka namestitve in uvedbe IR izvajajo načrtovalec podatkovne baze, uvajalec, skrbnik podatkovne baze, končni uporabnik, sistemski administrator, skrbnik aplikacije, postavitveni inženir, informacijski varnostni inženir in poslovni lastnik.
- Po potrebi sodelujejo tudi ostale vloge.



# Namestitev in uvedba

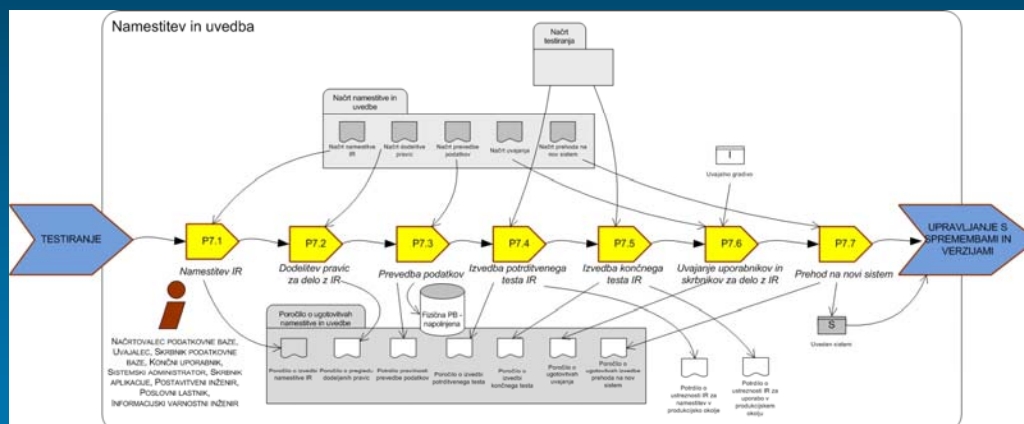
## Vloge in koraki

- Postopek namestitve in uvedbe lahko razdelimo na sedem aktivnosti:
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.

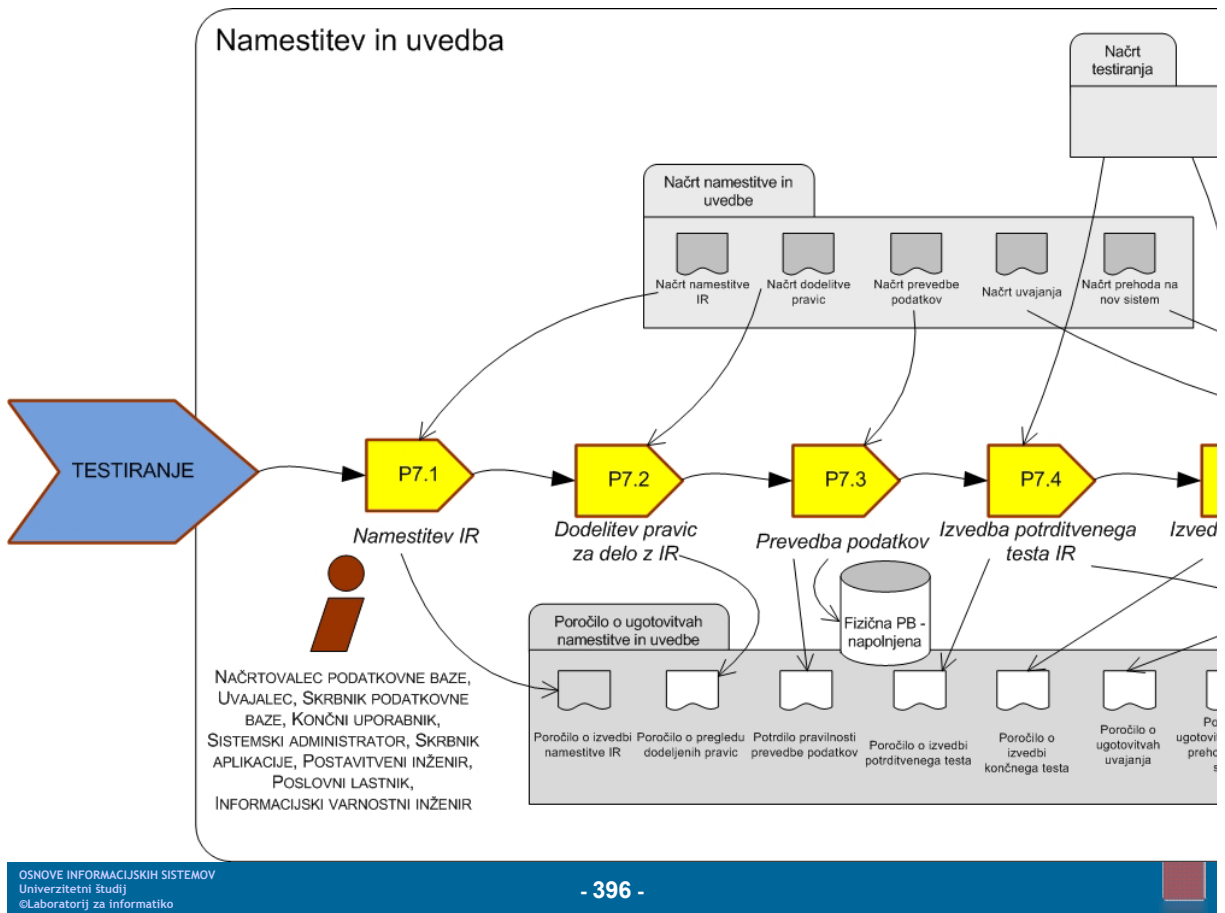


# Namestitev in uvedba

## Shema postopka







## Razvoj po strukturnem pristopu

Kje smo?

### Postopki strukturnega pristopa:

- Zajem in specifikacija zahtev;
- Analiza;
- Načrtovanje;
- Izvedba;
- Testiranje;

### ➔ Namestitev in uvedba:

- ➔ Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.

# Namestitev in uvedba – Namestitev IR

## Namestitev IR

- Naloga namestitve je vključitev nove IR v testno ali produkcijsko okolje. V okviru tega je potrebno namestiti:
  - strojno opremo,
  - sistemsko programsko opremo in
  - aplikativno programsko opremo - programske module.
- Namestitev poteka na podlagi načrta namestitve IR, postopki nameščanja opreme pa morajo biti v čim večji meri avtomatizirani.



# Namestitev in uvedba – Namestitev IR

## Namestitev IR

- Namestitev IR v testno okolje poteka v več fazah (med razvojem) ali v celoti (ob zaključku).
- Namestitev IR v produkcijsko okolje se izvede, ko je aplikacija razvita in potrjena s potrditvenim testom v testnem okolju.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ Namestitev in uvedba:
  - Namestitev IR,
  - ➔ Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.



## Namestitev in uvedba – Dodelitev pravic

Dodelitev pravic za delo z IR

- V skladu z načrtom namestitve in uvedbe – načrt dodelitve pravic **definiramo vloge, izvedemo vključitev oseb** – uporabnikov v ustrezne vloge in jim dodelimo gesla.
- Upoštevamo tudi **varnostno politiko**. Pravice za dostop do podatkov in uporabo programskih modulov dodelimo naenkrat celotni posamezni skupini uporabnikov, ki izvajajo določeno vlogo.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ Namestitev in uvedba:
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
- ➔ Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.



## Namestitev in uvedba – Prevedba

Prevedba podatkov

- Prevedba podatkov pomeni vzpostavitev začetnega stanja podatkov.
- Prevedba podatkov je lahko zelo kompleksen proces, saj se poleg prenosa podatkov pogosto izvaja tudi čiščenje, agregacija, reorganizacija podatkovnih struktur itd.
- Podatki v obstoječih sistemih so pogosto pomanjkljivi ali nepopolni, zapisani v obliki, ki je razumljiva zgolj programerjem itd.



# Namestitev in uvedba – Prevedba

## Prevedba podatkov

- Načrt za prevedbo podatkov je narejen v sklopu načrta namestitve in uvedbe. Programski moduli, ki so potrebni za prevedbo podatkov, so narejeni v sklopu izvedbe.
- Prevedba podatkov v sklopu namestitve in uvedbe poteka v **testno in produkcijsko okolje**.



# Namestitev in uvedba – Prevedba

## Prevedba podatkov

- **Prevedba podatkov v testno okolje se izvede v sklopu namestitve testnega okolja. Po potrebi se lahko izvede večkrat.**
  - Zagotavljanje testnih podatkov ter ponovljivosti testov je lahko zelo kompleksna naloga...
- **Prevedba podatkov v produkcijsko okolje se izvede v sklopu namestitve IR v produkcijsko okolje.**
  - Navadno gre za ponovitev že preverjenih in utečenih postopkov prevedbe, ki zagotovijo pravilen prenos podatkov.





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ Namestitev in uvedba:
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - ➔ Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.



## Namestitev in uvedba – Potrditveni test

Izvedba potrditvenega testa

- Potrditveni test izvedemo v testnem okolju.
- Predstavlja **zaključni test** pravilnega delovanja celotne IR v testnem okolju in zajema vse funkcionalnosti sistema, potrebne za delovanje v testnem okolju.
- Poteka na osnovi načrta testiranja in se zaključi s **potrdilom o ustreznosti IR za namestitev v produkcijsko okolje.**





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ Namestitev in uvedba:
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - ➔ Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.



## Namestitev in uvedba – Končni test

Izvedba končnega testa

- **Končni test** izvedemo v produkcijskem okolju.
- Predstavlja **končni test** pravilnega delovanja IR v produkcijskem okolju in poteka na produkcijskih primerih - na "živih" oz. produkcijskih podatkih v produkcijskem okolju.
- V končni test običajno ne moremo zajeti vse funkcionalnosti sistema. Poteka na osnovi načrta testiranja in se zaključi s **potrdilom o ustreznosti IR za uporabo v produkcijskem okolju.**





## Razvoj po strukturnem pristopu

Kje smo?

- Postopki strukturnega pristopa:
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ Namestitev in uvedba:
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - ➔ Uvajanje uporabnikov in skrbnikov za delo z IR,
  - Prehod na novi sistem.



## Namestitev in uvedba – Uvajanje

Uvajanje uporabnikov in skrbnikov za delo z IR

- Naloga uvajanja je naučiti uporabnike uporabljati in skrbeti za IR.
- Uvajanje izvaja **uvajalec**, ki je v primeru razvoja IR običajno član razvojne ekipe, v primeru nakupa IR pa predstavnik proizvajalca.
- Osnovni cilj uvajanja je vsako skupino uporabnikov naučiti uporabljati tiste sklope IR, ki jih uporabniki pri svojem delu potrebujejo.





# Namestitev in uvedba – Uvajanje

## Uvajanje uporabnikov in skrbnikov za delo z IR

- V okviru obravnavane aktivnosti je potrebno poleg običajnih uporabnikov uvesti tudi **skrbnike IR**.
- Poleg uvajanja samega je potrebno poskrbeti tudi za pripravo **uvajalnih gradiv in testnih podatkov** v podatkovni bazi, ki bodo služili za potrebe uvajanja.



# Namestitev in uvedba – Uvajanje

## Uvajanje uporabnikov in skrbnikov za delo z IR – Smernice

- **Smernice in priporočila za uvajanje:**
  - uvajanje uporabnikov za uporabo novega sistema naj poteka ločeno za običajne uporabnike in za skrbnike novega sistema;
  - najprej naj predstavniki izvajalca ali prodajalca IR izvedejo uvajanje skrbnikov, nato pa naj se izvede uvajanje uporabnikov – možen pristop: train the trainer;
  - pri uvajanju je potrebno upoštevati velikost organizacije in uporabnike razdeliti v skupine glede na njihovo področje dela in podobnost načina uporabe novega sistema;



# Namestitev in uvedba – Uvajanje

## Uvajanje uporabnikov in skrbnikov za delo z IR – Smernice

- **Smernice in priporočila za uvajanje (nadaljevanje):**
  - uvajanje naj poteka v testnem okolju; če se uvajanje izvaja na produkcijski podatkovni bazi, je potrebno paziti, da ne izvajamo aktivnosti, kjer bi nastali podatki, ki jih ne bi mogli povrniti v prejšnje stanje oz. v stanje, ki odraža dejansko stanje v organizaciji,
  - uporabniki morajo imeti dostop do sistema pomoči uporabnikom (raznovrstnih navodil), hkrati pa tudi vedeti, na koga se lahko obrnejo, če naletijo na težave, ki jih kljub navodilom ne znajo sami rešiti.



# Razvoj po strukturnem pristopu

## Kje smo?

- **Postopki strukturnega pristopa:**
  - Zajem in specifikacija zahtev;
  - Analiza;
  - Načrtovanje;
  - Izvedba;
  - Testiranje;
- ➔ **Namestitev in uvedba:**
  - Namestitev IR,
  - Dodelitev pravic za delo z IR,
  - Prevedba podatkov,
  - Izvedba potrditvenega testa IR,
  - Izvedba končnega testa IR,
  - Uvajanje uporabnikov in skrbnikov za delo z IR,
- ➔ **Prehod na novi sistem.**



# Namestitev in uvedba – Prehod

## Prehod na nov sistem

- **Prehod na nov sistem** predstavlja trenutek, ko je IR primerna za uporabo v produkcijskem okolju.
- O trenutku začetka uporabe nove IR mora biti vnaprej obveščena vsa organizacija.
- Najprimernejši termini za začetek uporabe nove IR so **delovno neintenzivna obdobja**. Pogoji, ki mora biti izpolnjeni za možnost začetka uporabe, so uspešno opravljene aktivnosti namestitve, testiranja, prevedbe podatkov in uvajanja.



# Namestitev in uvedba – Prehod

## Strategije uvedbe

- V praksi se za uvedbo IR uporabljajo različne strategije. V grobem jih delimo v tri skupine:
  - **Fazni pristop** (Phased strategy);
  - **Zamenjava ali vse naenkrat** (Replacement strategy, BigBang);
  - **Vzporedno delovanje** (Parallel strategy).
- Poleg omenjenih strategij uvedbe se lahko uporabljajo tudi različne kombinacije.



# Namestitev in uvedba – Prehod

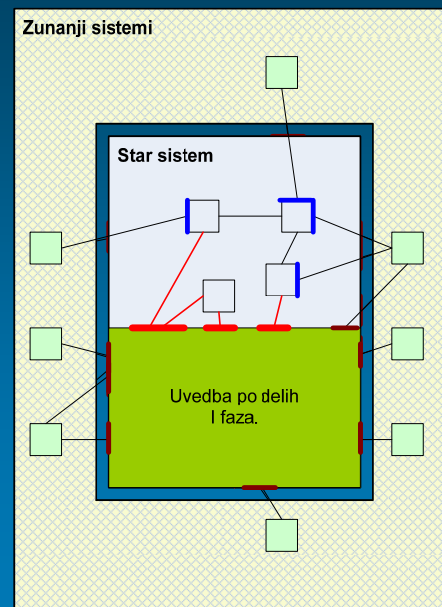
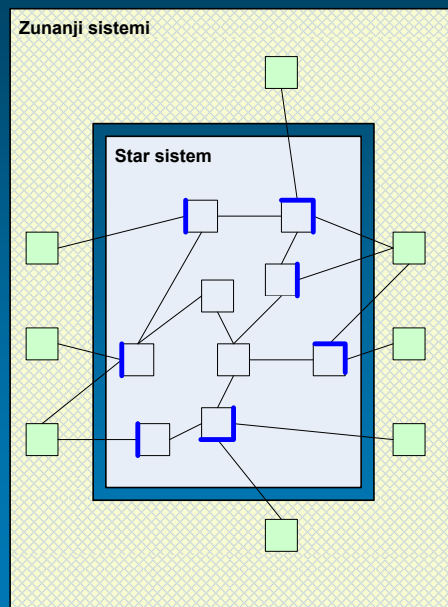
## Strategije uvedbe – Fazni pristop

- **Fazni pristop:**
  - Star/obstoječ sistem nadomestimo z novim v več korakih oziroma postopoma.
  - Vsebina posameznega koraka ali faze je lahko različna; npr. najprej eno poslovno področje, potem drugo itd., ali najprej en funkcionalni sklop, zatem drugi itd.
  - Prednosti postopne uvedbe oziroma postopne zamenjave starega z novim sistemom so številne, vendar takšen pristop ni vedno izvedljiv, saj pogosto zahteva začasne vmesnike za komunikacijo med deli novega in starega sistema.



# Namestitev in uvedba – Prehod

## Strategije uvedbe – Fazni pristop – problem začasnih vmesnikov



# Namestitev in uvedba – Prehod

## Strategije uvedbe – Zamenjava ali vse naenkrat

- **Zamenjava ali vse naenkrat:**
  - pri tej strategiji gre za enkratno zamenjavo obstoječega sistema z novim. Na izbran (in ustrezno načrtovan) trenutek se stare aplikacije ugasne in nadomesti z novimi.
  - Takšen pristop zmanjša potrebo po virih, vendar je bolj tvegan, saj pomeni, da nimamo več dostopa do starega sistema v primeru, da gre kaj narobe.



# Namestitev in uvedba – Prehod

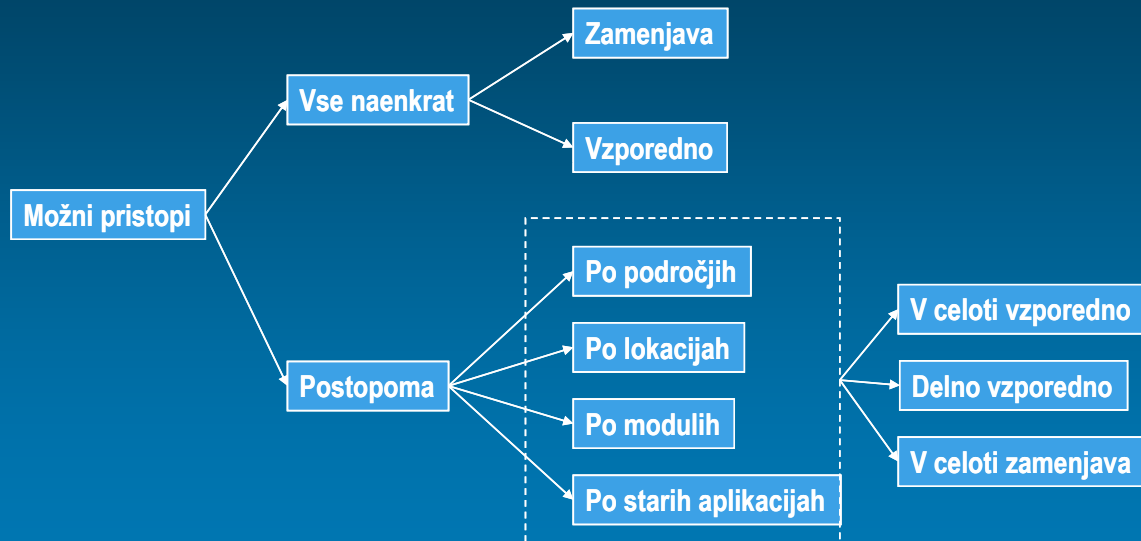
## Strategije uvedbe – Vzporedno delovanje

- **Vzporedno delovanje:**
  - strategija vzporednega delovanja temelji na vzporedni uporabi starega in novega sistema.
  - Med uvajanjem novega sistema v produkcijo starega ne ugašamo, temveč uporabljamo oba vzporedno.
  - Prednost takšne strategije je v zmanjšanju tveganja (če gre kaj narobe, imamo še vedno na voljo star sistem), ključna slabost pa v zahtevnosti po virih ter potencialni neskladnosti podatkov zaradi dvojnega zajema.
  - V praksi se strategija vzporednega delovanja pogosto uporablja v okrnjeni obliki, kjer star sistem ohranimo za vpogled, medtem ko so vi vnosni izvedeni v novem sistemu.



# Namestitev in uvedba – Prehod

## Strategije uvedbe



## Poglavje V Objektni razvoj

- Osnovni principi objektne usmerjenosti
- Osnove modelirnega jezika UML in procesa RUP
- Objektna analiza in načrtovanje



# Objektni razvoj

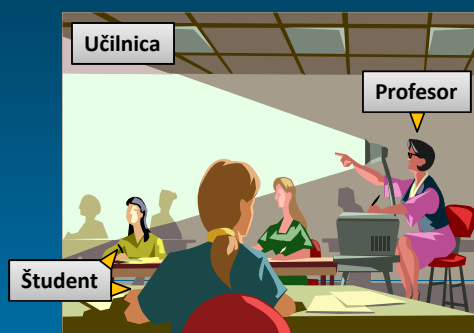
## Osnovne značilnosti objektnega pristopa

- **Objektni pristop** k razvoju IR se pojavi kot posledica uveljavitve objektnih programskih jezikov in objektnih tehnologij...
- V 90-ih letih nastane več deset **metod objektivne analize in načrtovanja IR**.
- Objektna analiza in načrtovanje se od strukturnega ločuje predvsem v **predstavitvi realnega sveta**: ne ločuje med podatki in aktivnostmi temveč vse modelira z objekti.



# Objektni razvoj

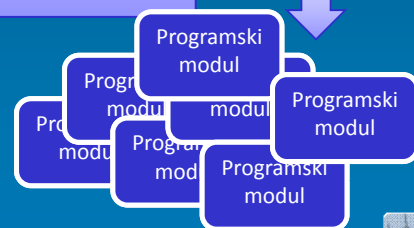
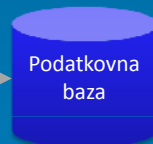
## Osnovne značilnosti objektnega pristopa



### Strukturni pristop

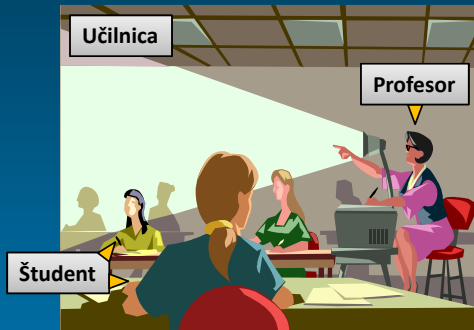
1. Podatki
  - Študent
  - Profesor
  - Učilnica
  - Predmet
  - ...
2. Procesi
  - Vzdrževanje izpitnih rokov
  - Opravljanje izpita
  - Vodenje izpitne evidence
  - ...

Programski modul:  
Pregled zasedenosti  
učilnice



# Objektni razvoj

## Osnovne značilnosti objektnega pristopa



### Objektni pristop

- **Objekt tipa ŠTUDENT**
  - ☐ Lastnosti:
    - Priimek,
    - Ime,
    - Vpisna številka,
    - ...
  - ☐ Funkcije:
    - Prijava na izpit
    - Odjava iz izpita
    - Pregled elektronskega indeksa
    - ...
- **Objekt tipa UČILNICA**
  - ☐ Lastnosti:
    - Velikost,
    - Število sedežev,
    - ...
  - ☐ Funkcije:
    - Pregled predavanj v učilnici po dnevih
    - Izpis naziva profesorja P, ki na določen dan D predava v učilnici
    - ...

# Objektni razvoj

## Primerjava postopkov strukturnega in objektnega razvoja IR





# Objektni pristop

## Vsebina

- Sledi:
  - Osnovni principi objektne usmerjenosti
  - Osnove modelirnega jezika UML in procesa RUP
  - Objektna analiza in načrtovanje



# Razvoj po objektnem pristopu

## Kje smo?

- Objektni pristop:
  - ➔ Osnovni principi objektne usmerjenosti;
  - ➔ Objekt in razred
    - Enkapsulacija ali skrivanje podatkov
    - Dedovanje in hierarhija
  - Jezik UML in proces RUP;
  - Objektna analiza in načrtovanje;



# Objektni razvoj

## Definicija objekta

- Objekt lahko predstavlja fizično entiteto ali konceptualni pojem.
  - Učitelj, učenec,...
  - Predmet, test,...
- S stališča razvoja IR je objekt koncept, abstrakcija ali stvar z natančno določenimi mejami in pomenom za IR.



# Objektni razvoj

## Definicija objekta

- Objekti lahko predstavljajo tudi konkretne koncepte s področja računalniških jezikov in rešitev:
  - Seznam
  - Indeks
  - ...



# Objektni razvoj

## Abstrakcija

- Objekt je **abstrakcija** neke entitete, ki je pomembna za IR...



**Vesna: Profesor**

Ime: Vesna Jug

Telefon: 01/4768 814

Prijavi (int pid): boolean



# Objektni razvoj

## Stanje, obnašanje in identiteta objekta

- Objekt ima svoje **stanje, obnašanje in identiteto**.
- Stanje objekta določajo njegove **lastnosti**:
  - **Flomaster** ima barvo, proizvajalca, debelino,...
  - **Knjiga** ima vsebino, število strani, založnika,...



# Objektni razvoj

Stanje, obnašanje in identiteta objekta

- **Objekt se zaveda svojega stanja!** Ve kakšne vrednosti imajo lastnosti, ki ga opisujejo.
  - Flomaster je rumene barve, debeline 3mm, proizvajalca Pilot,...
  - Knjiga opisuje razvoj IS, ima 432 strani, izdaja jo založba Pasadena,...
- Vsakokrat ko se spremeni vrednost neke lastnosti, ki nas o objektu zanima, rečemo se je spremenilo **stanje objekta**.



# Objektni razvoj

Stanje, obnašanje in identiteta objekta

- **Objekti imajo svoje obnašanje.**
  - Študent študira, se prijavi na izpit, opravlja izpit, posluša predavanja,...
  - Pisalo piše??? Kdo v resnici piše?
- **Objekt se zaveda, kaj lahko počne in kaj se lahko z njim počne!**



# Objektni razvoj

## Razred

- Objekte istega tipa združujemo v **razrede**.
- Razred je
  - opis skupine objektov z enakimi lastnostmi, enakim obnašanjem, povezavami in semantiko.
  - abstrakcija, ki poudarja pomembne karakteristike in izpusti ostale nepomembne karakteristike.
- Objekt je **primerek razreda**.



# Objektni razvoj

## Razred

- Odnos med objektom in razredom je podoben odnosu med entiteto in entitetnim tipom.

### Objekti



### Razred

Ime razreda: Profesor

Lastnosti:

- Ime
- Priimek
- ...

Obnašanje

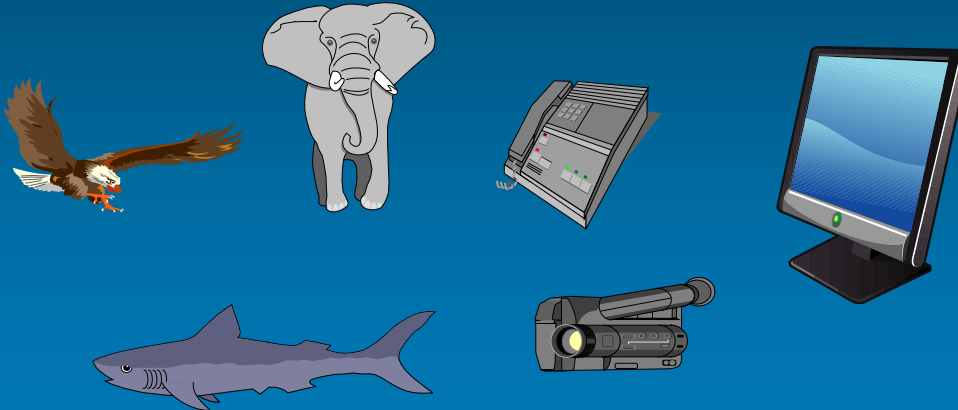
- ...



# Objektni razvoj

Razred

- Koliko razredov je na sliki?



# Razvoj po objektnem pristopu



Kje smo?

- **Objektni pristop:**
  - ➔ Osnovni principi objektne usmerjenosti;
    - Objekt in razred
    - ➔ Enkapsulacija ali skrivanje podatkov
      - Dedovanje in hierarhija
  - Jezik UML in proces RUP;
  - Objekta analiza in načrtovanje;



# Objektni razvoj

## Enkapsulacija ali skrivanje podatkov

- **Enkapsulacija** govori o organizaciji podatkov, ki jih vemo o objektih, na način, da jih bo moč učinkovito uporabljati in vzdrževati?
- Kar vemo o objektu, razdelimo v dve skupini:
  - Kar moramo vedeti, da objekt lahko uporabimo,
  - Kar moramo vedeti, da bo objekt pravilno deloval.



# Objektni razvoj

## Enkapsulacija ali skrivanje podatkov

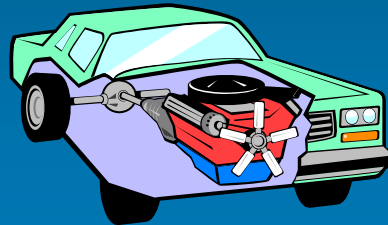
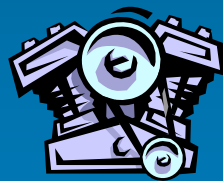
- Želimo se naučiti voziti avtomobil. Kaj moramo o avtomobilu vedeti?
  - Kako se upravlja z volanom, kako se zavira in p pospešuje,...
  - Potrebno vedeti, kako deluje motor? Kako pride do vžiga? Kaj natančno se zgodi, ko pritisnemo na plin?
- Za delo z objektom ne potrebujemo vseh podatkov o objektu, temveč samo določene – **vmesnik** (interface).



## Objektni razvoj

### Enkapsulacija ali skrivanje podatkov

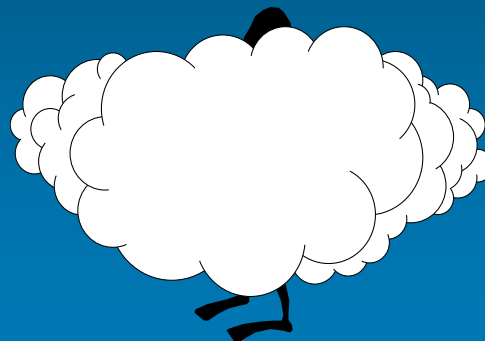
- **Kaj mora veljati, da objekt deluje pravilno?**
  - Vmesnik nam omogoča uporabo objekta. Za njegovo pravilno delovanje pa morajo biti vse funkcije, ki jih vmesnik daje na voljo, tudi dejansko implementirane!
  - Za delovanje objekta moramo priskrbeti mehanizem, ki se zna odzivati na vmesnik...



## Objektni razvoj

### Enkapsulacija ali skrivanje podatkov

- **Enkapsulacija objekta zahteva, da skrijemo:**
  - Implementacijo obnašanja, ki je na voljo prek vmesnika;
  - Podatke znotraj objekta, ki so potrebni za implementacijo obnašanja in beležijo stanje objekta v vsakem trenutku njegovega obstoja.







## Razvoj po objektnem pristopu

Kje smo?

- **Objektni pristop:**
  - ➔ Osnovni principi objektne usmerjenosti;
    - Objekt in razred
    - Enkapsulacija ali skrivanje podatkov
  - ➔ Dedovanje in hierarhija
  - Jezik UML in proces RUP;
  - Objekta analiza in načrtovanje;



## Objektni razvoj

Hierarhija in dedovanje

- **Dedovanje je ključen koncept objektnega razvoja.**
  - Pove, da ima objekt v času svojega kreiranja dostop tudi do lastnosti drugih razredov poleg dostopa do razreda, kateremu pripada.
  - Podedovan razred združi vse lastnosti (lastne in podedovane) v svojo definicijo.
- **Primer:**
  - **Profesor** je poseben primer **pedagoškega delavca**.
  - Objekt tipa profesor ima lastnosti pedagoškega delavca ter svoje lastne lastnosti.



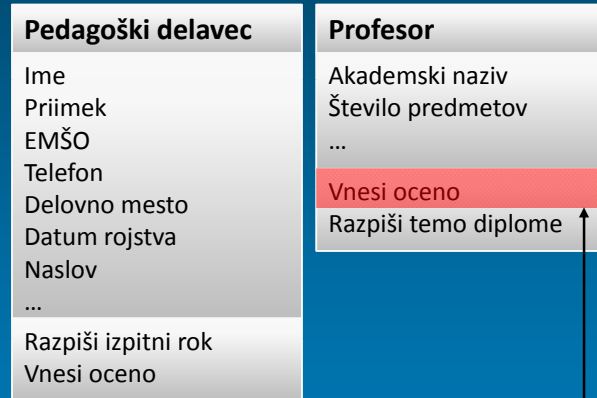
# Objektni razvoj

## Hierarhija in dedovanje

- Primer razreda **Profesor in Pedagoški delavec**

- Opis:

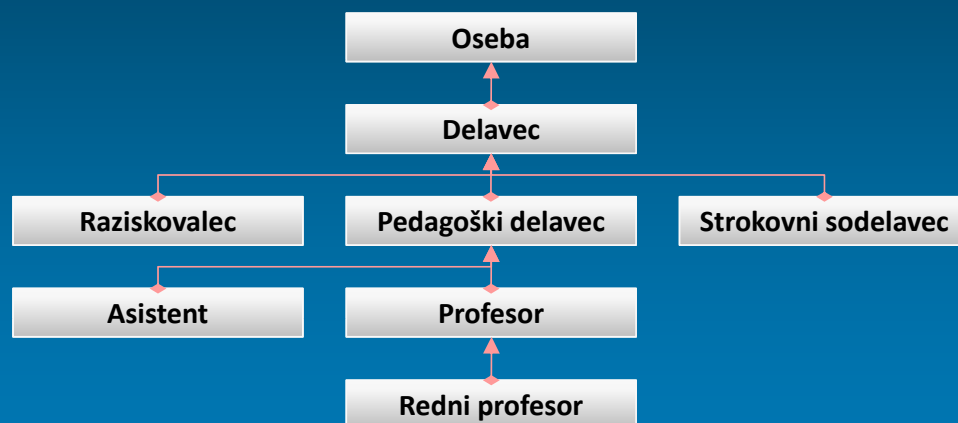
- P ima enake lastnosti kot PD.
- Za P nas zanima tudi akademski naziv ter št. predmetov, ki jih poučuje.
- P lahko razpisuje tudi teme za diplome.
- P ima pri vnašanju ocen več pravic kot PD. Lahko vpisuje tudi ocene ustno...



# Objektni razvoj

## Hierarhija in dedovanje

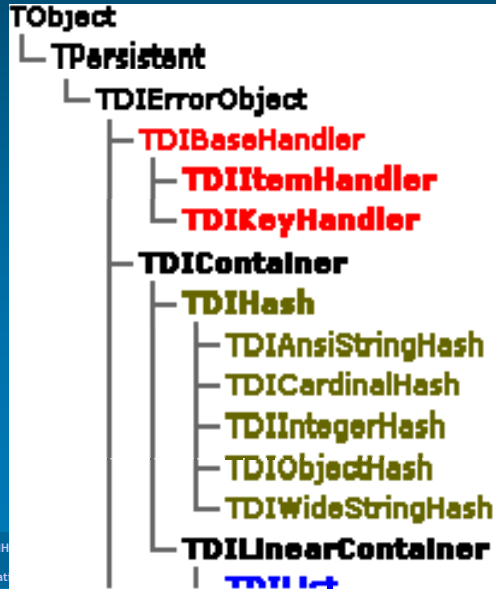
- Razrede lahko zapišemo v hierarhijo.



# Objektni razvoj

## Hierarhija in dedovanje

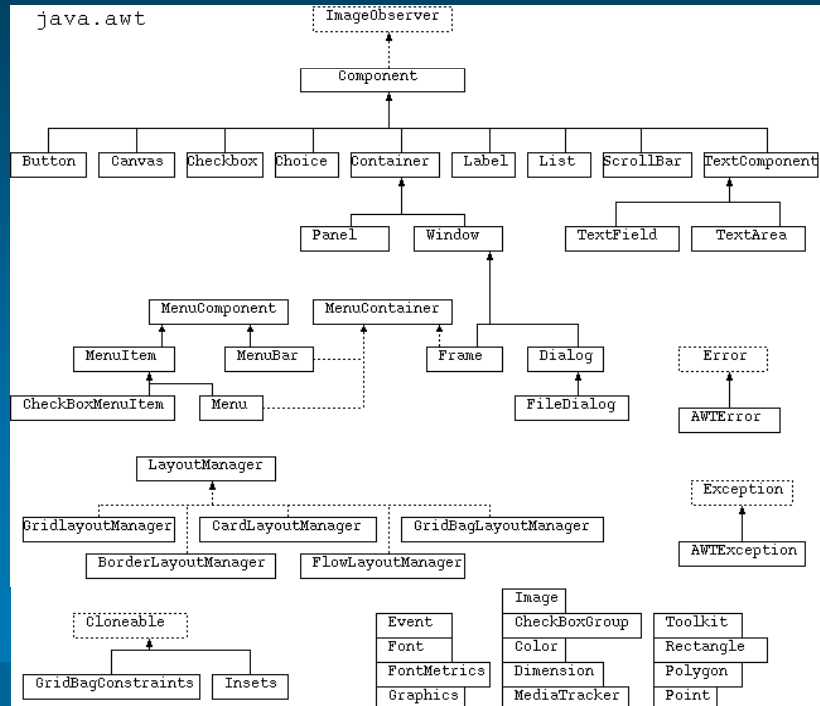
- Hierarhija razredov v Delphi razvojnem okolju...



# Objektni razvoj

## Hierarhija in dedovanje

- Delček razredne hierarhije java.awt...





# Razvoj po objektnem pristopu

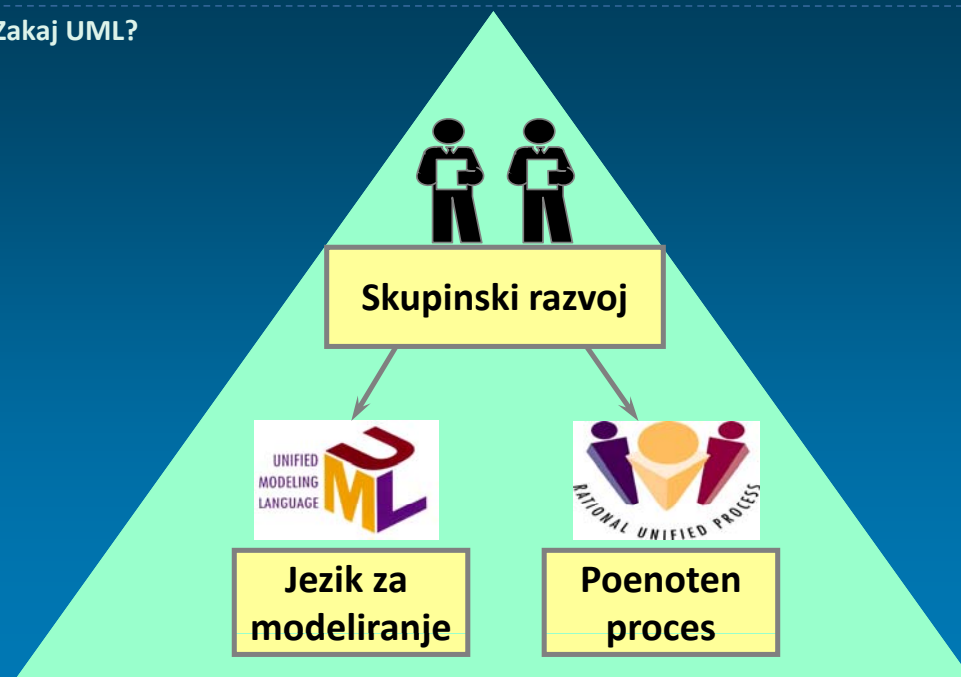
Kje smo?

- **Objektni pristop:**
  - Osnovni principi objektne usmerjenosti;
- ➔ **Jezik UML in proces RUP:**
  - ➔ O jeziku UML
    - Osnove procesa objektnega razvoja RUP;
  - Objekta analiza in načrtovanje;



# Objektni razvoj

Zakaj UML?



# Objektni razvoj

UML – univerzalni modelirni jezik



## Univerzalni modelirni jezik

Univerzalni modelirni jezik (UML) je jezik za specifikacijo, vizualizacijo, konstrukcijo in dokumentacijo izdelkov v okviru objektnega razvoja informacijskih rešitev.



# Objektni razvoj

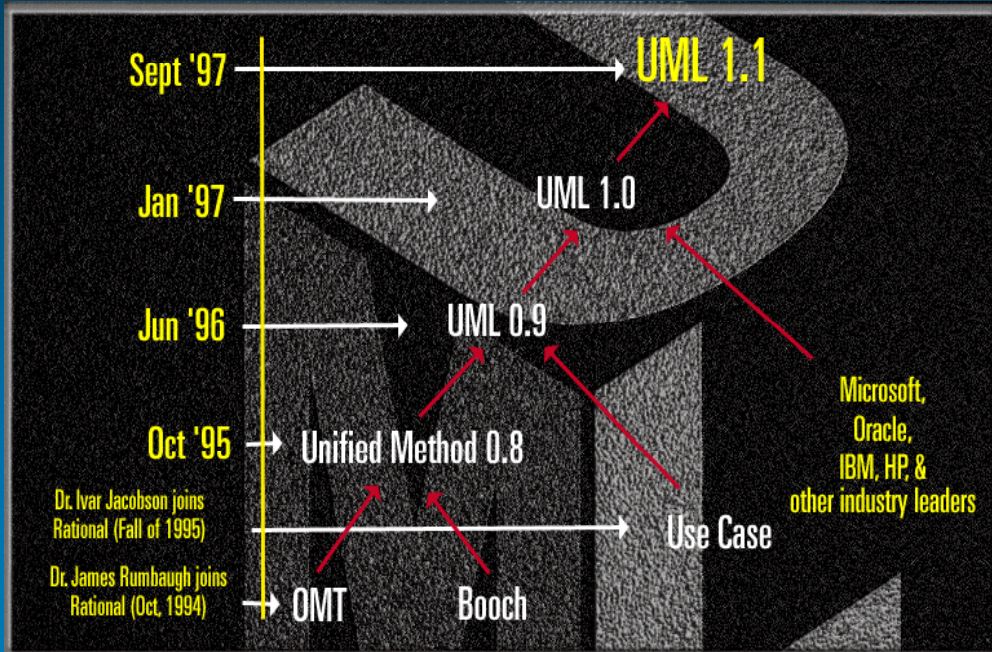
Razlogi za nastanek UML

- V začetku 90-ih več kot 50 OO metod!
  - Fusion, Shlaer-Mellor, ROOM, Wirfs-Brock, Coad-Yourdon, MOSES, BOOM, OOSD, OSA, BON, Catalysis, HOOD, Ooram, DOORS ...
- Problem:
  - Presek in konflikti v meta-modelih...
  - Različne grafične notacije...
  - Procesi različni ali manjkajo
  - Gospodarstvo potrebuje standard!!!



# Objektni razvoj

UML – Zgodovina do 1997



# Objektni razvoj

Drugi viri in prispevki



# Objektni razvoj

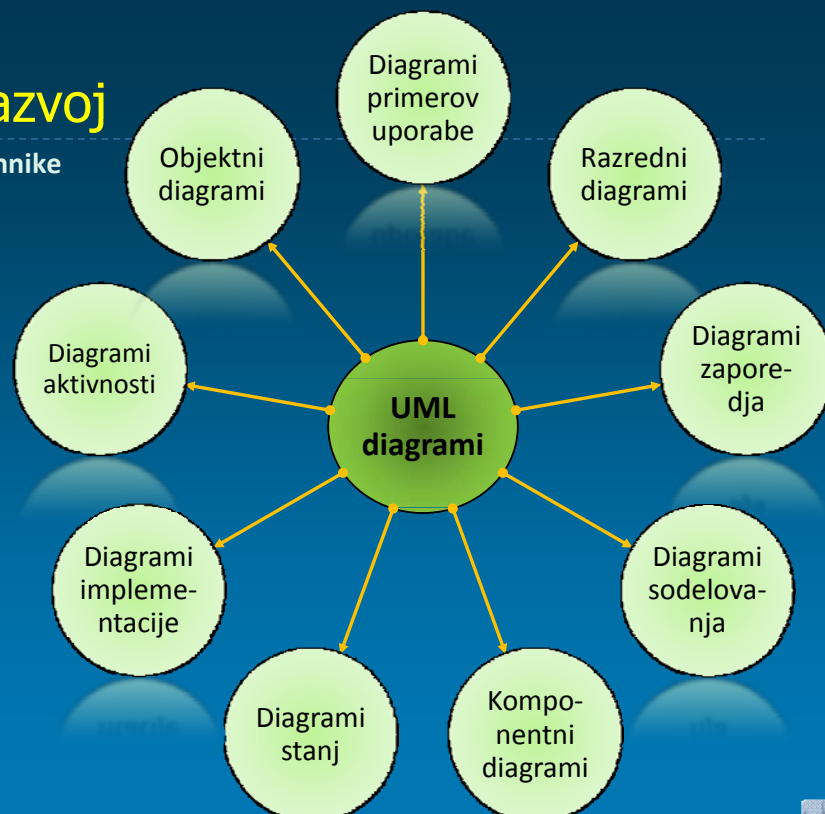
## UML danes

- **UML 1.1 specifikacija** predstavlja integracijo različnih metod/pristopov → problem slaba semantična integracija.
  - Številne revizije: UML 1.3, 1.4, 1.5
  - **UML 2.0** največja revizija – odpravlja semantične neskladnosti... sprejet v dveh delih: oktobra 2004 prvi del, novembra 2005 drugi del.
  - V postopku sprejemanja **UML 2.1**
- Na voljo številna **orodja**, ki podpirajo UML 2.0 specifikacijo...



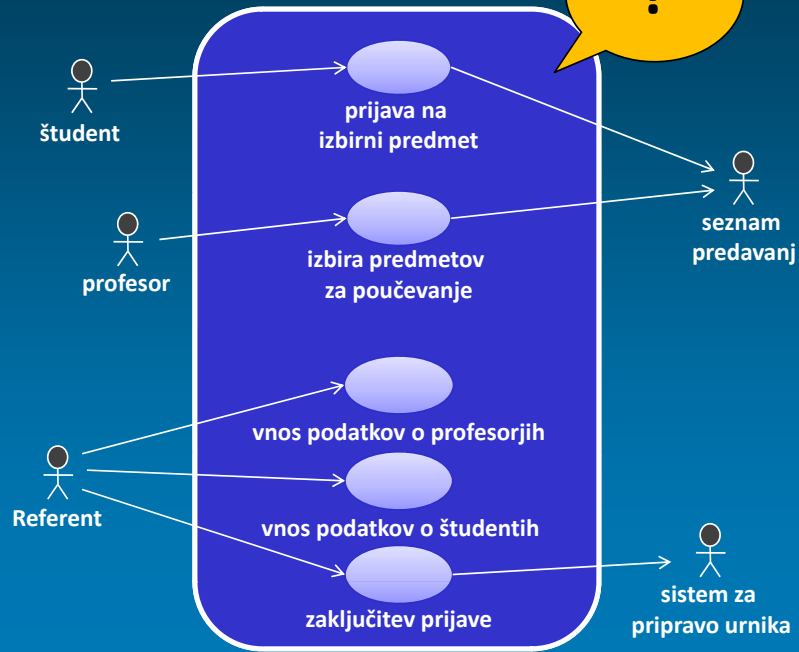
# Objektni razvoj

## UML diagramске tehnike



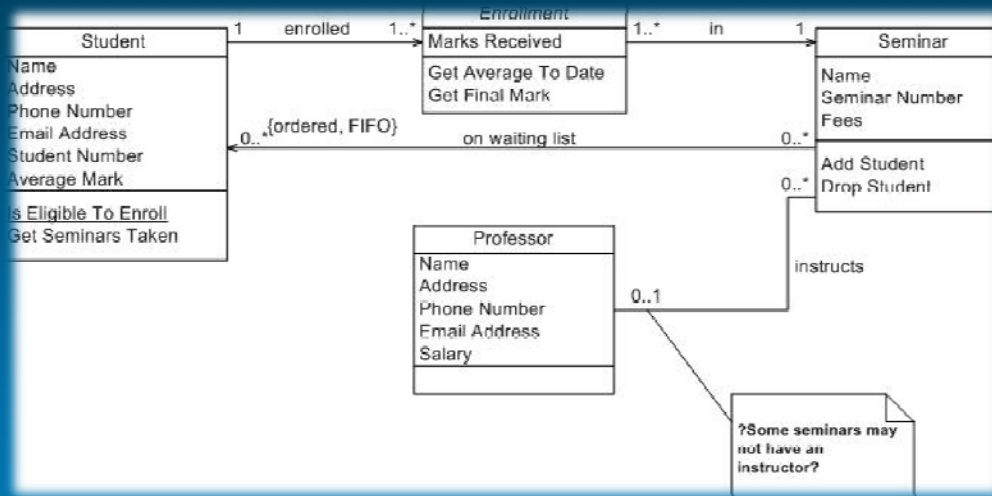
# Objektni razvoj

## UML diagramske tehnike – enostavnost



# Objektni razvoj

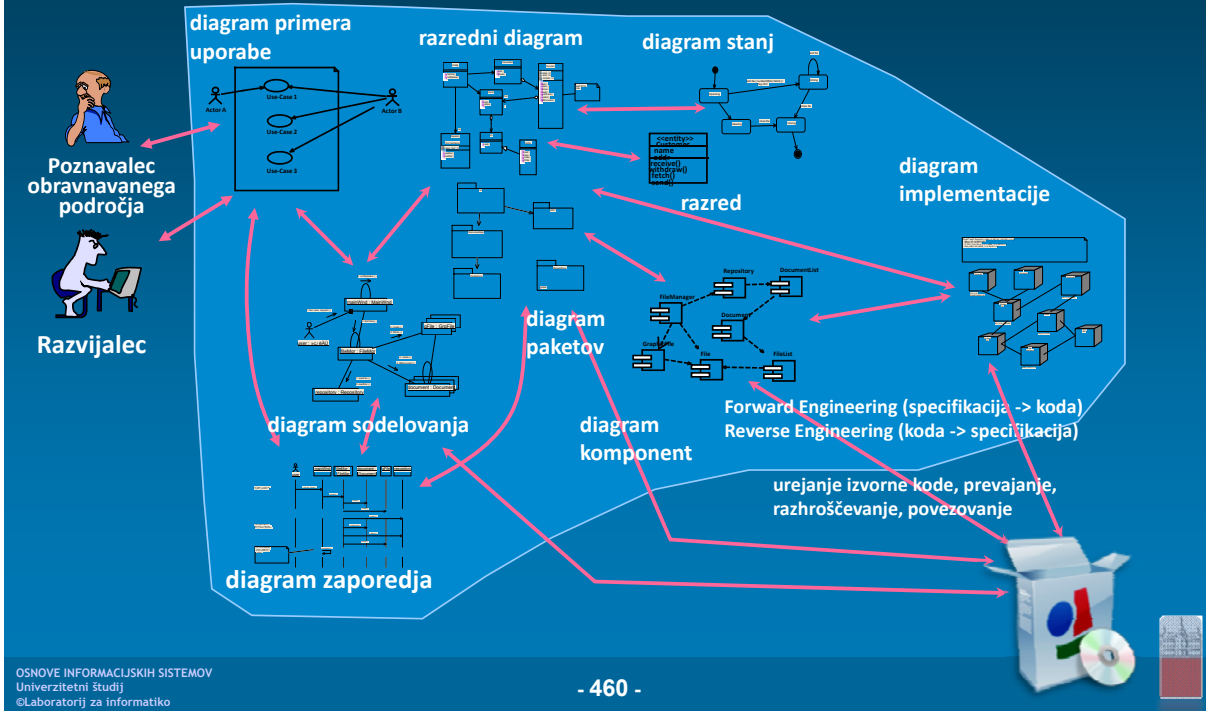
## UML diagramske tehnike – kompleksnost





# Objektni razvoj

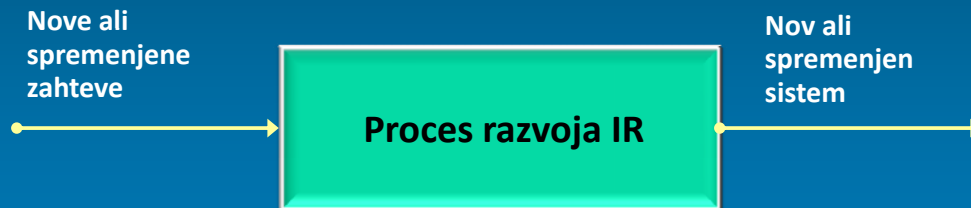
## UML diagramske tehnike – hrbtnica razvoja



# Objektni razvoj

## Proces razvoja

- **Proces** določa kdo dela kaj, kdaj in kako za doseganje določenega cilja.
- Cilj razvoja IR je njena izgradnja ali izboljšava.

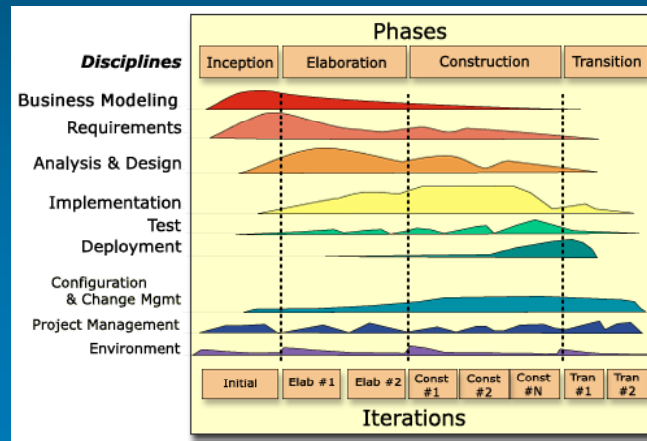


# Objektni razvoj

## Proces razvoja – RUP



- **RUP** – Rational Unified Process
  - primer procesa objektnega razvoja
  - avtor: podjetje Rational



# Objektni razvoj

## Proces razvoja – RUP

- **Glavne lastnosti RUP:**
  - Daje smernice za učinkovit razvoj kakovostne programske opreme
  - Zmanjšuje tveganje in povečuje predvidljivost
  - Zajema in vpeljuje najboljše izkušnje
    - učenje iz izkušenj drugih
    - mentorstvo v elektronski obliki
    - razširitev izobraževalnega gradiva
  - Pospešuje skupno vizijo in kulturo
  - Vpeljuje načrt za uvedbo orodij
  - Omogoča enostaven in hiter dostop do informacij v elektronski obliki (spletna stran,...)

# Objektni razvoj

## Proces razvoja – RUP

- RUP opisuje, kako učinkovito uporabiti **šest najboljših izkušenj** s področja razvoja IR.



# Objektni razvoj

## Proces razvoja – RUP

- Primeri uporabe so osnova mnogim aktivnostim procesa:
  - Izdelava in potrditev razvojnega modela
  - Določitev preizkusnih primerov in postopkov za model preizkušanja
  - Načrtovanje iteracij
  - Izdelava uporabniške dokumentacije
  - vpeljava sistema
  - Primeri uporabe pripomorejo k uskladitvi vsebine različnih modelov



# Objektni razvoj

Proces razvoja – RUP

- Primeri uporabe so **hrbtenica** procesa RUP.



# Objektni razvoj – RUP

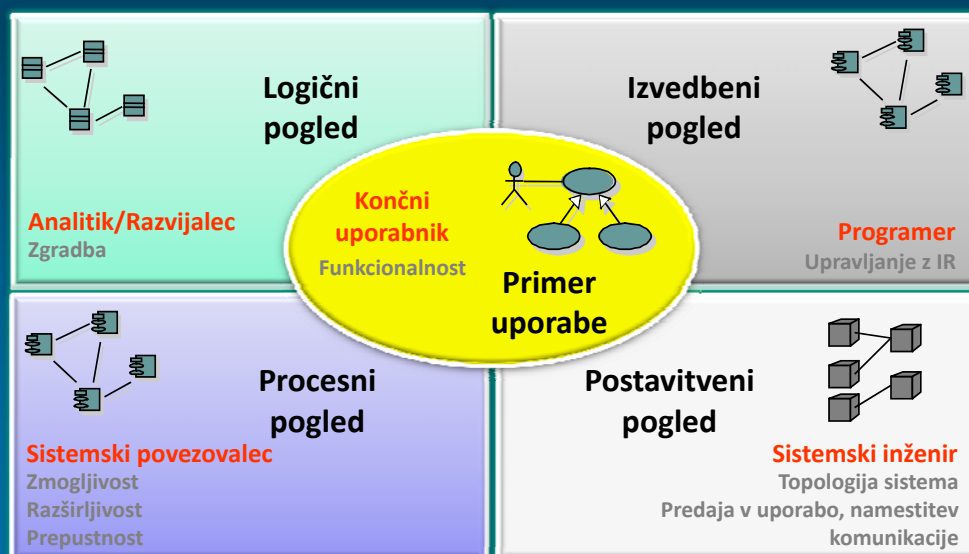
RUP – arhitekturna usmerjenost

- Na arhitekturo se osredotočimo v začetnih iteracijah
  - Zasnova in potrditev arhitekture je eden primarnih ciljev razvoja IR.
- Dokument o arhitekturi IR pomeni primarni izdelek, ki opisuje izbrano arhitekturo.
- Drugi izdelki, ki izhajajo iz arhitekture
  - Smernice razvoja
  - Zgradba izdelka
  - Sestava skupine



# Objektni razvoj – RUP

RUP – arhitekturna usmerjenost – 3+1 pogledi na arhitekturo



# Objektni razvoj – RUP

RUP – prednosti arhitekturno usmerjenega pristopa

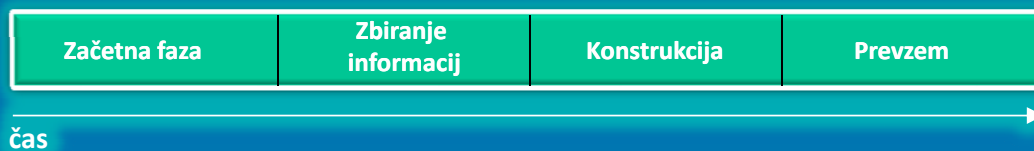
- **Omogoča**
  - vzpostavitev in ohranitev nadzora nad projektom,
  - obvladovanje kompleksnosti in
  - vzdrževanje celovitosti sistema.
- Razširja možnosti **ponovne uporabe izdelkov**.
- Pospesuje **komponentno usmerjen razvoj**.



# Objektni razvoj – RUP

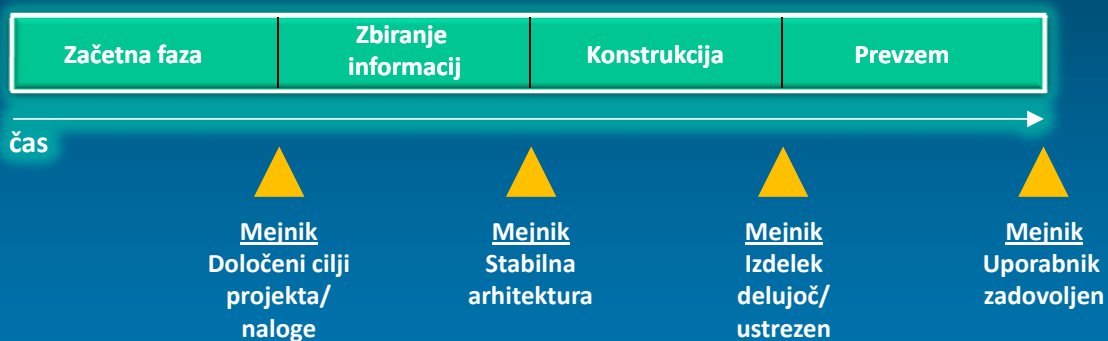
RUP – faze življenjskega cikla razvoja po RUP

- RUP zajema štiri faze:
  - **Začetna faza** – vzpostavitev projekta, opredelitev okvirjev obravnavanega področja, načrtovanje virov,...;
  - **Zbiranje informacij** – zbiranje informacij o obravnavanem področju, specifikacija značilnosti, načrtovanje arhitekture;
  - **Konstrukcija** – konstrukcija izdelka;
  - **Prevzem** – predaja izdelka v uporabo končnemu uporabniku.



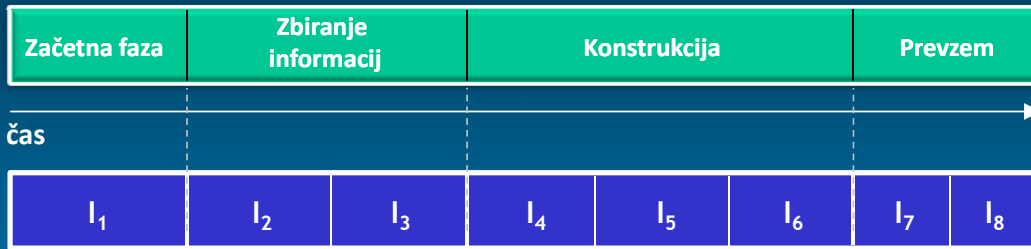
# Objektni razvoj – RUP

RUP – makro mejniki



# Objektni razvoj – RUP

RUP – mikro mejniki

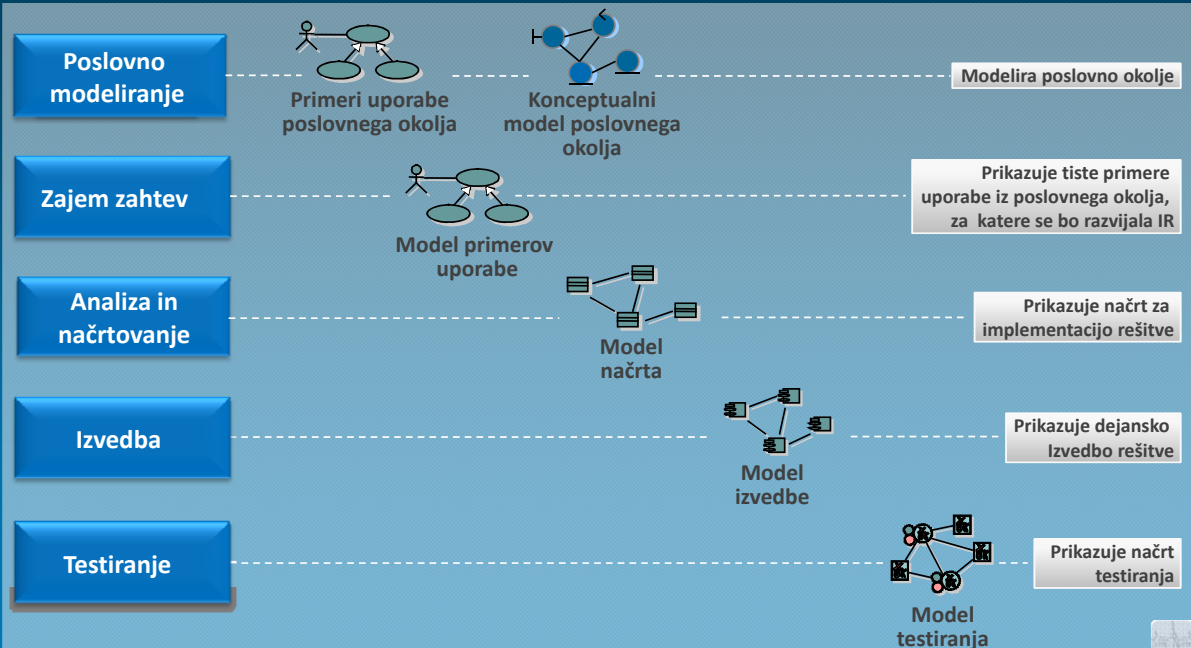


## Iteracija

Iteracija je specifično zaporedje aktivnosti izvedenih na osnovi načrta in z določenim kriterijem vrednotenja, ki se konča z izdajo izdelka.

# Objektni razvoj – RUP

RUP – gradnja modela sistema prek postopkov RUP



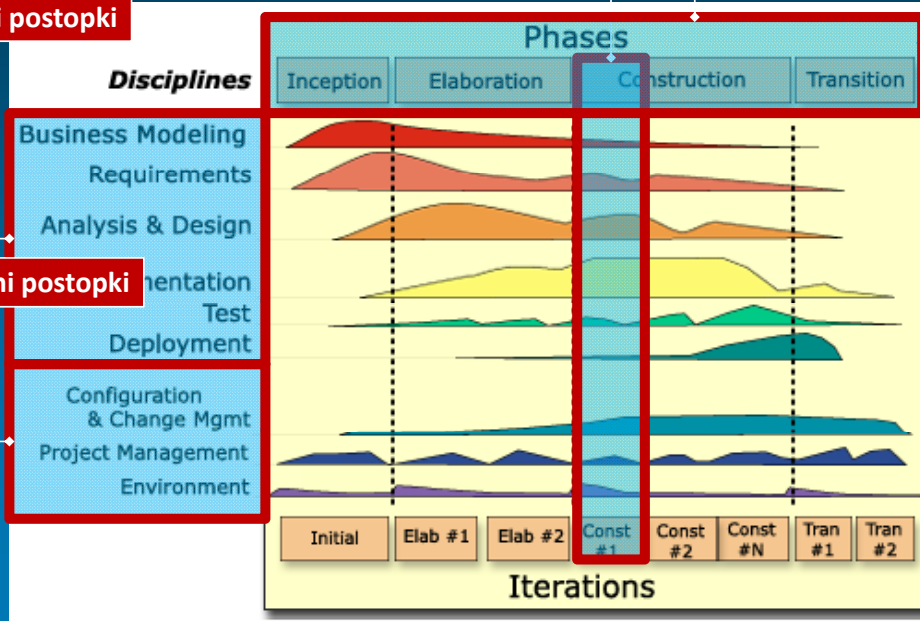
Razvoj IR konvergira od vzpostavitve projekta in zbiranja zahtev/podatkov proti konstrukciji in uvedbi

V vsaki iteraciji gremo čez vse postopke

RUP – prednosti arhitekturno usmerjenega pristopa

Osnovni postopki

Podporni postopki



## Objektni razvoj – RUP

RUP – osnovni in podporni postopki

### ▪ RUP definira več postopkov:

- Osnovni postopki:
  - Poslovno modeliranje
  - Zajem zahtev
  - Analiza in načrtovanje
  - Izvedba
  - Testiranje
  - Uvedba
- Podporni postopki:
  - upravljanje s konfiguracijami
  - projektno vodenje
  - obvladovanje okolja





## Razvoj po objektnem pristopu

Kje smo?

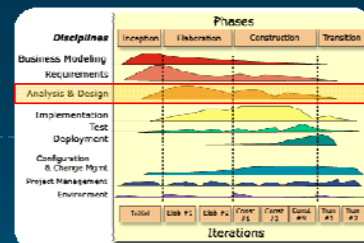
- **Objektni pristop:**
  - Osnovni principi objektne usmerjenosti;
  - Jezik UML in proces RUP;
- ➔ Objekta analiza in načrtovanje:



## Objektni razvoj – RUP

RUP – namen analize in načrtovanja

- **Namen analize in načrtovanja je:**
  - Pretvoriti zahteve v načrt sistema;
  - Razviti robustno arhitekturo sistema;
  - Prilagoditi načrt izvedbenemu okolju.
- **Povezava z drugimi postopki:**
  - Poslovno modeliranje postavi organizacijski kontekst sistema.
  - Postopek zajema zahtev je primarni vhod v postopek analize in načrtovanja.
  - Postopek izvedbe realizira sistem na osnovi načrta,
  - ...



# Objektni razvoj – RUP

RUP – Analiza in načrtovanje – primerjava analize in načrtovanja

## Analiza

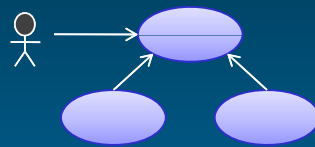
- Se osredotoča na razumevanje problema
- Idealizirano načrtovanje
- Obnašanje
- Struktura sistema
- Funkcijske zahteve
- Majhen model

## Načrtovanje


- Se osredotoči na razumevanje rešitve
- Operacije in atributi
- Zmogljivost rešitve
- Blizu programski kodi
- Življenjski cikel objektov
- Nefunkcionalne zahteve
- Velik model

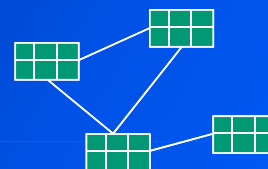
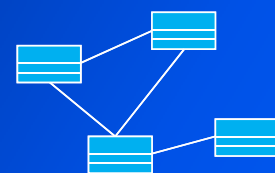
# Objektni razvoj – RUP

RUP – Analiza in načrtovanje – vhodni in izhodni



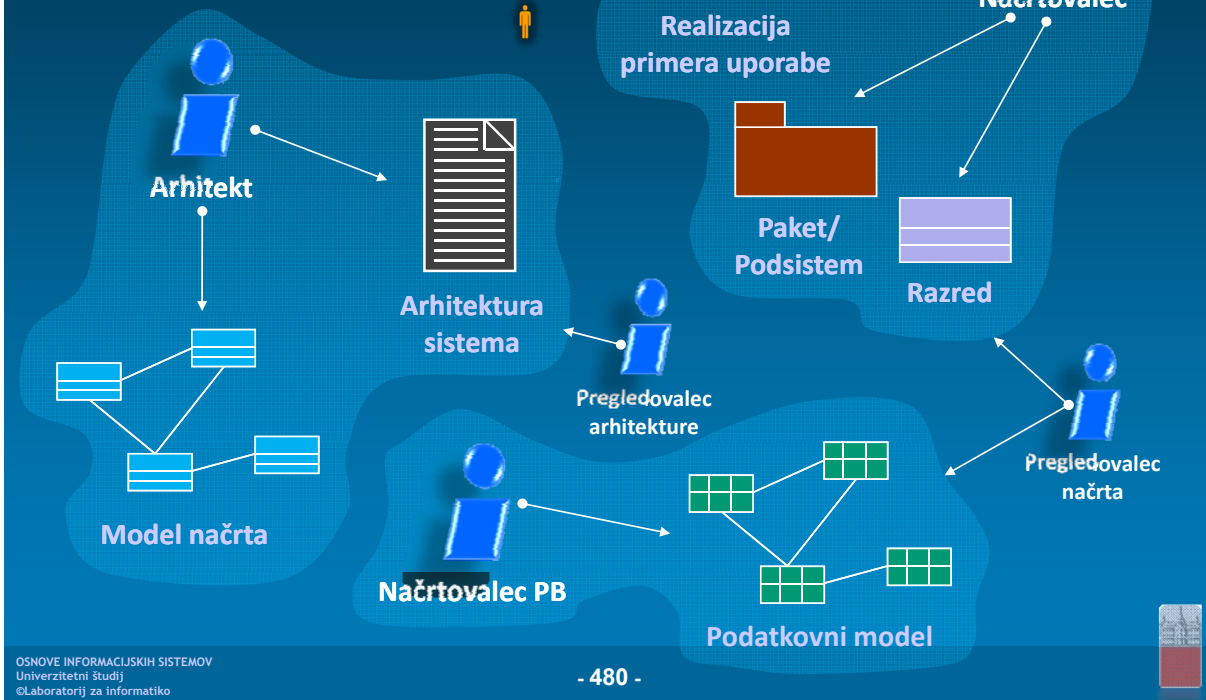
Analiza in načrtovanje

 Končen izdelek analize in načrtovanja



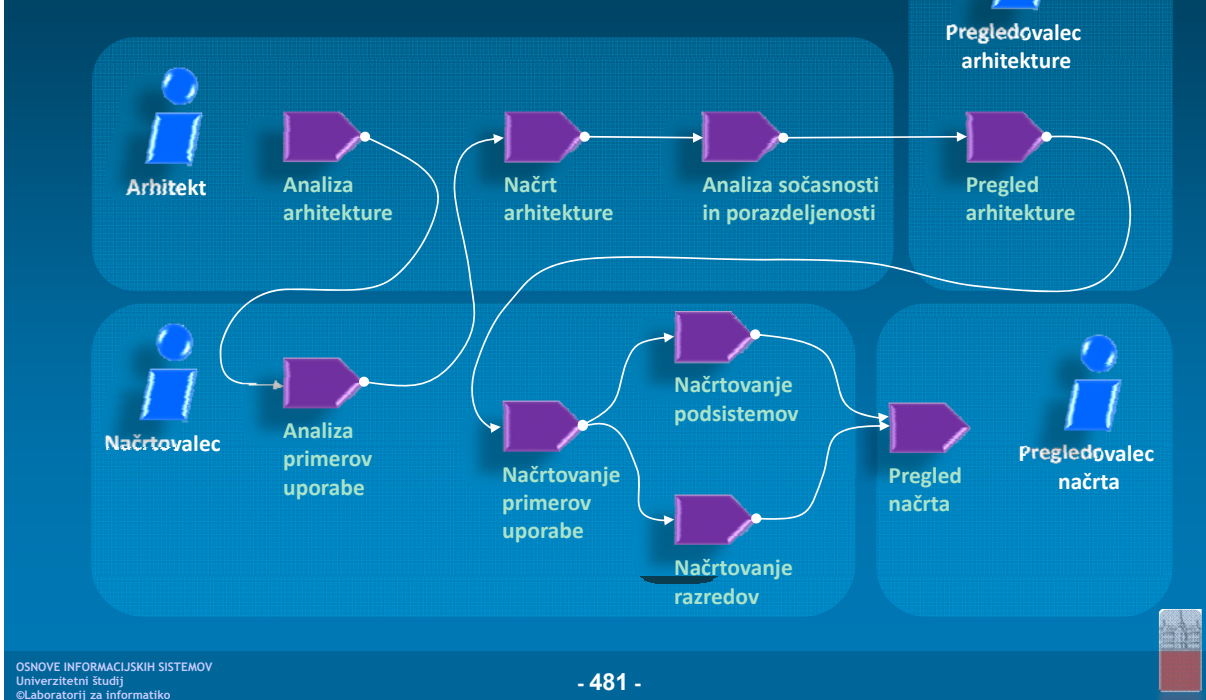
# Objektni razvoj – RUP

RUP – Analiza in načrtovanje – vloge



# Objektni razvoj – RUP

RUP – Analiza in načrtovanje – koraki



## Poglavje VI

# Načrtovanje podatkovnih baz

- Tri-nivojsko načrtovanje
- Konceptualno načrtovanje
- Osnove relacijskega modela in logično načrtovanje
- Normalizacija

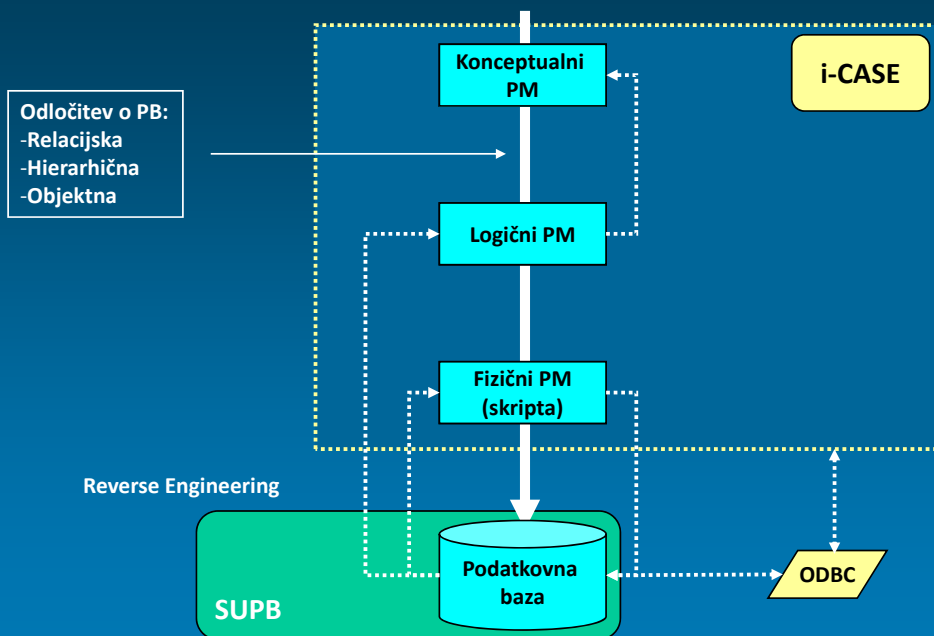


## Trije nivoji načrtovanja – trije modeli...

- Konceptualno načrtovanje – konceptualni oz. semantični podatkovni model
- Logično načrtovanje – logični podatkovni model
- Kreiranje fizične podatkovne baze – fizična podatkovna baza oz. fizični podatkovni model



## Trije nivoji načrtovanja – trije modeli



## Konceptualno načrtovanje

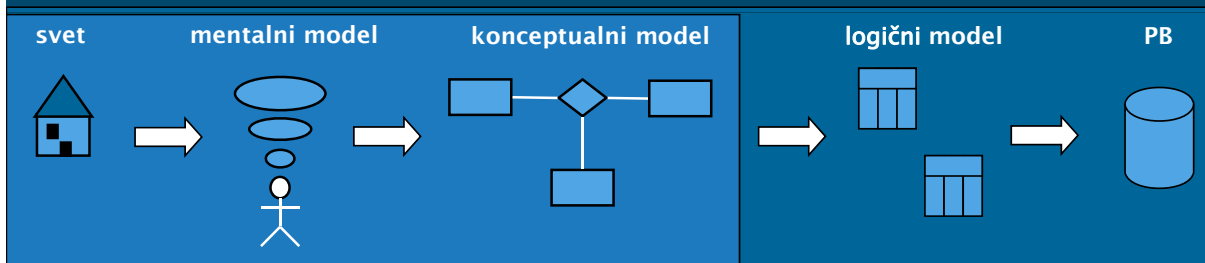
- **Konceptualno načrtovanje** je opredelitev podatkovnih potreb oz. zahtev poslovne domene s pomočjo konceptualnega modela.
- Konceptualno načrtovanje preko konceptualnega modela poskrbi za opis pomena podatkov, potrebnih za poslovno domeno.
- Konceptualnega načrtovanja ne moremo avtomatizirati, za njegovo izvedbo je odgovoren analitik. Gre za prenos semantike v model.

## Pomen konceptualnega načrtovanja

- Je najbolj kritično – napake se prenašajo naprej na naslednje modele.
- Zahteva sodelovanje uporabnikov. Uporabniki so nosilci znanja o poslovni domeni, so poznavalci semantike.
- Konceptualno načrtovanje mora upoštevati tudi poslovna pravila.



## Umestitev konceptualnega načrtovanja



## Predstavitev konceptualnih modelov

- Najpogosteje uporabljena tehnika za predstavitev konceptualnih podatkovnih modelov sta **diagram entiteta-razmerje** ter **razredni diagram**.  
Obravnavali bomo prvega!
- Nazivi, ki se uporabljajo:
  - Konceptualni podatkovni model
  - Podatkovni model
  - Entitetni model
  - ER model
- Obstaja tudi razširjeni model entiteta razmerje



## Gradniki entitetnega modela

- Entitetni tip
- Atribut
- Razmerje
- Enolični identifikator



## Entitetni tip – entiteta...

- Entitete so posamezni primerki tipov objektov iz poslovne domene: dogodki, predmeti, osebe, pravila, dejstva
- O entitetah obstaja določena predstava o tem:
  - kakšne lastnosti dejansko imajo
  - kakšne lastnosti jim moramo določiti (morajo imeti), da bodo izpolnjevale poslanstvo entitetnega modela
- Na osnovi predstave o tem in percepcije, lahko entitete klasificiramo v entitetne tipe: vse entitete, ki ustrezajo določeni predstavi, pripadajo posameznemu entitetnemu tipu.



## Entitetni tip – entiteta

- Vsak trenutek pripada posameznemu entitetnemu tipu množica entitet tega entitetnega tipa, ki jo imenujemo **entitetna množica**
- Entitetna množica je **časovno spremenljiva**: entitete nastajajo, se spreminjajo in tudi izginjajo (izstopajo iz množice).
- Entitetna množica je v nekem trenutku lahko tudi prazna.





## Predstavitev entitetnega tipa

Ime entitetnega tipa

Ime entitetnega tipa

Prostor za attribute



## Atribut...

- Entitete imajo določene lastnosti, posamezne entitete (istega entitetnega tipa) se med seboj razlikujejo po vrednosti njihovih lastnosti
- Entiteta ima praviloma veliko lastnosti, le del teh lastnosti je zanimiv oz. pomemben za opazovano poslovno domeno
- Lastnosti, ki so pomembne za opazovano poslovno domeno, vključimo v konceptualni model tako, da jih kot attribute določimo entiteti (entitetnemu tipu)



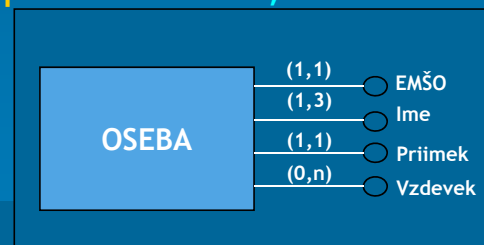
## Atribut...

- Govorimo lahko o več vrstah lastnosti:
  - Entitetna imena: naziv, ime, opis
  - Prave entitetne lastnosti: višina, teža, cena, vrednost
  - Lastnosti, ki jih določimo za potrebe poslovnih procesov, poslovnih funkcij in poslovnih pravil: statusi
- Atribut določimo za tisto lastnost, ki je za poslovno domeno pomembna

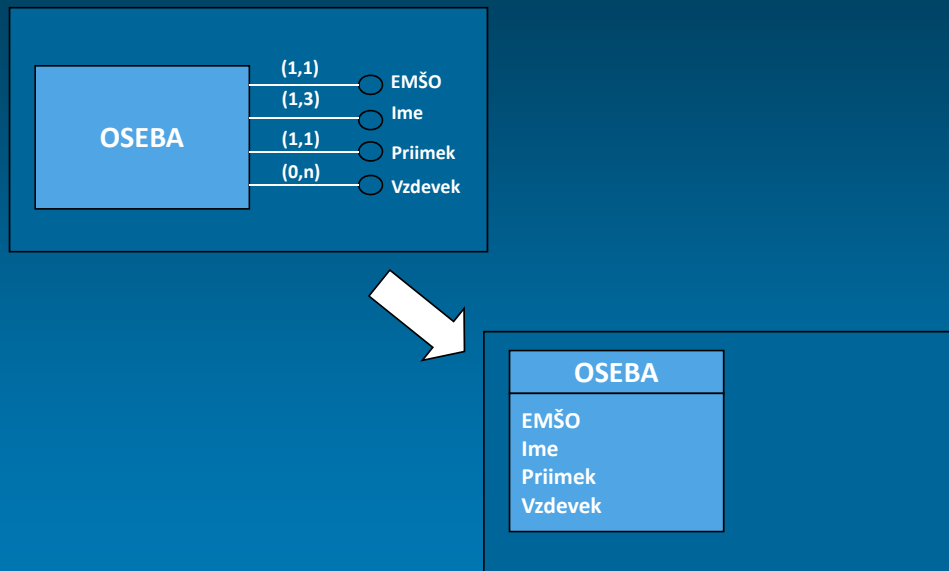


## Atribut

- Kardinalnost atributa je **minimalna in maksimalna**  
Glede na kardinalnost atributa ločimo:
  - Totalni atribut  $(1,n)$ , kjer je  $n \geq 1$
  - Parcialni atribut  $(0,n)$ , kjer je  $n \geq 1$
  - Enovrednostni atribut  $(m,1)$ , kjer je  $m \in \{0,1\}$
  - Večvrednostni atribut  $(m,n)$ , kjer je  $m \in \{0,1\}$  in  $n > 1$
- Minimalna števnost 0 pomeni, da je atribut lahko brez vrednosti (ni obvezen).
- Atribut pripada določenemu **tipu**: numerični, znakovni,...
- Za večino tipov je potrebno določiti tudi **dolžino**.



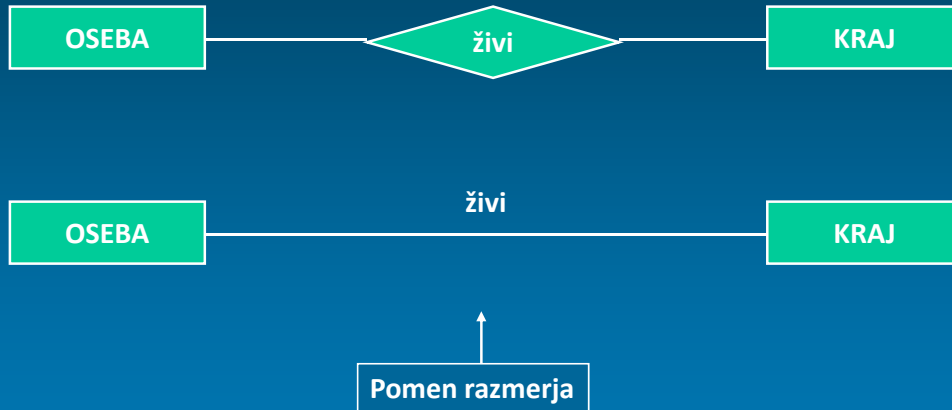
## Predstavitev atributa



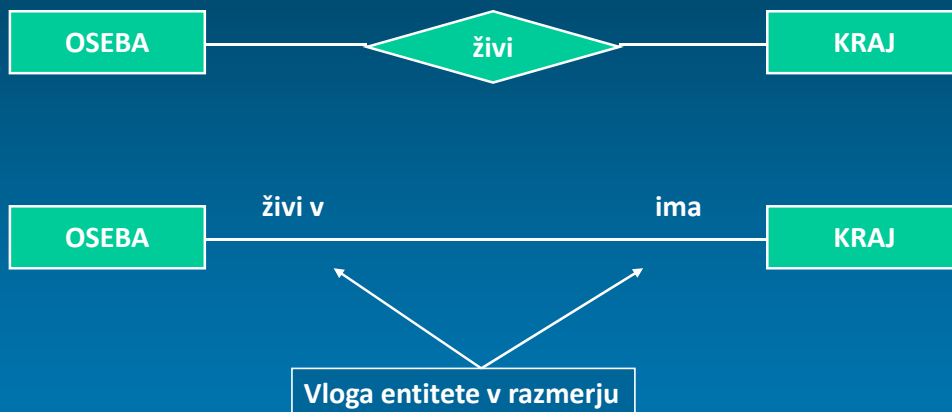
## Razmerja med entitetami

- Entitete niso svet zase, medsebojno se povezujejo preko razmerij, povezav
- Razmerje ima določen pomen
- Predstavitev razmerja v modelu entiteta-razmerje je povezava.
- Med opazovanim parom (v splošnem podmnožici) entitet je lahko več razmerij: OSEBA, KRAJ – stalno bivališče, začasno bivališče

## Predstavitev razmerja...



## Predstavitev razmerja

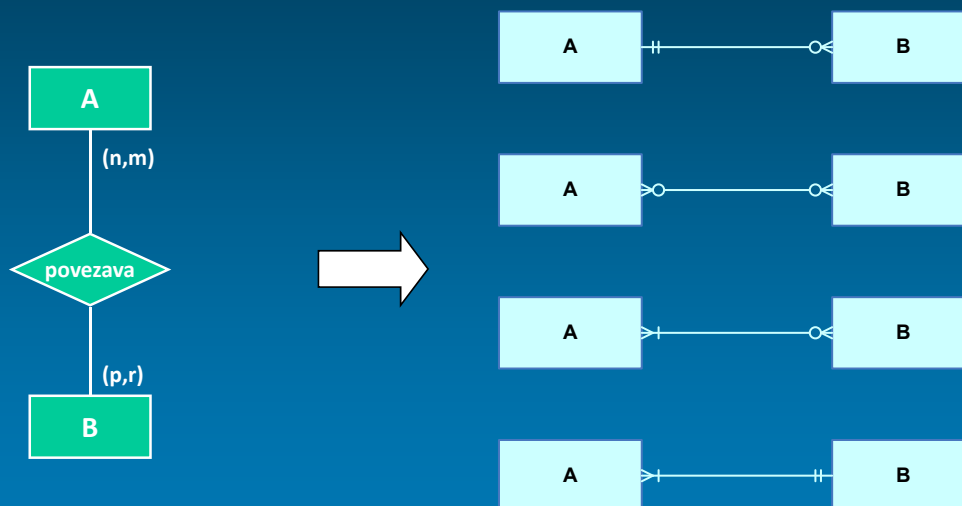


## Kardinalnost razmerja...

- Kardinalnost (števnost) predstavlja število entitet entitetnega tipa, ki so v razmerju glede na pomen razmerja.
- Vsaka entiteta ima svojo kardinalnost v razmerju – kardinalnost glede na vlogo. Entiteti OSEBA, POŠTA:
  - Ena (naključno izbrana) oseba ima stalno bivališče v enem kraju
  - V enem (naključno izbranem) kraju ima stalno bivališče več oseb

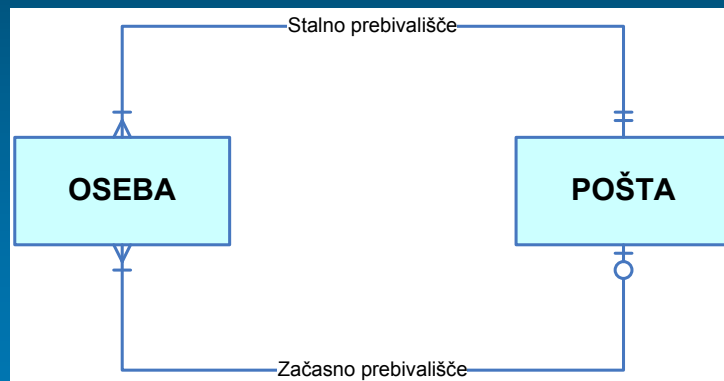


## Kardinalnost razmerja...



## Kardinalnost razmerja

- Razmerji med entitetama OSEBA in POŠTA



## Obveznost razmerja

- **Obveznost** pove, ali sta dve entiteti vedno v razmerju ali lahko tudi nista v razmerju: obvezno, neobvezno razmerje
- Obveznost lahko obravnavamo pod okriljem števности, zaradi česar dodatno uvedemo števnost 0



## Razmerje tudi opisuje lastnost entitete

- Razmerje tudi opisuje lastnost entitete
- Primer: OSEBA, POŠTA
- Razmerje ima atributiven značaj



## Enolični identifikator entitete...

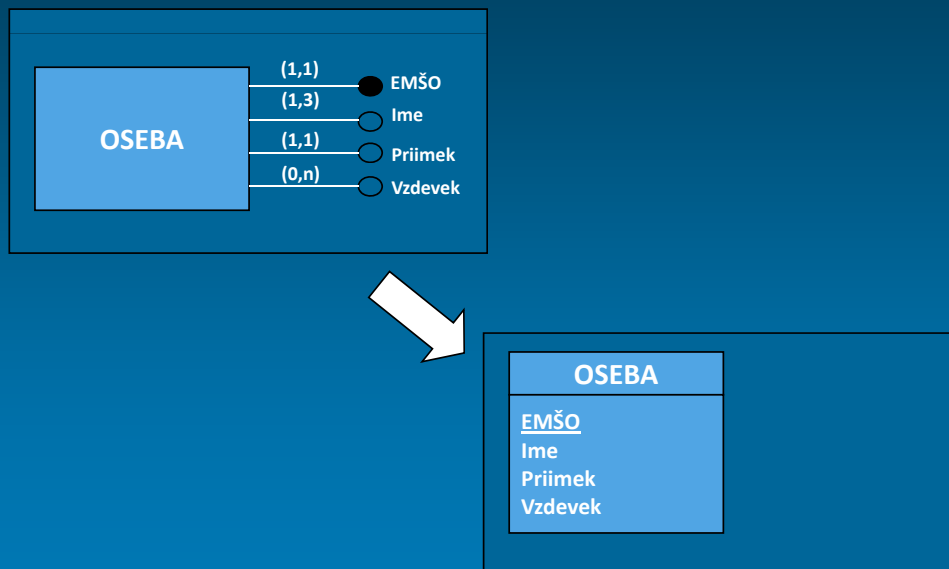
- **Enolični identifikator entitete** je podmnožica lastnosti entitete (atributov in razmerij – drugih entitet), ki enolično razlikujejo posamezno instanco entitete znotraj entitetne množice
- Z ozirom na to, ali tvorijo enolični identifikator entitete le atributi entitete ali pa je v enoličnem identifikatorju tudi kakšno razmerje, ločimo med **močnim entitetnim tipom** in **šibkim entitetnim tipom**



## Enolični identifikator entitete

- Imamo lahko več enoličnih identifikatorjev, vendar moramo enega izbrati – določiti
- Izbrani – določeni enolični identifikator je podlaga za ključ v relacijskem modelu

## Predstavitev enoličnega identifikatorja





## Močni entitetni tip

- Enolični identifikator sestavljajo le atributi entitete (identifikacijski atributi)
- $\{a_1, \dots, a_k\}$  je enolični identifikator entitete  $A$ , če ustreza naslednjim pogojem:
  - a)  $a_1, \dots, a_k$  so vsi **totalni enovrednostni atributi**, kar zagotavlja, da imajo vsi identifikacijski atributi definirano natanko eno vrednost (eno dimenzijo)
  - b)  $T: V_1 \times \dots \times V_k \rightarrow E_T$  je **totalna** ali **parcialna enovrednostna** funkcija, kar zagotavlja, da se vsak element kartezijskega produkta vrednostnih množic, ki so območja identifikacijskih atributov, preslika v največ eno entiteto tipa  $A$
  - c) Je **minimalna podmnožica**, ne obstaja **prava podmnožica**, za katero bi tudi veljal pogoj b)



## Šibki entitetni tip

- Enolični identifikator ni sestavljen le iz lastnih atributov, temveč tudi iz razmerij oz. drugih entitet v razmerju oz. njenih identifikatorjev.
- $\{a_1, \dots, a_k\} \cup I_{T_1} \cup \dots \cup I_{T_n}$  je enolični identifikator entitete  $A$ , če ustreza naslednjim pogojem:
  - a)  $a_1, \dots, a_k$  so vsi **totalni enovrednostni atributi**,  $I$  pa **identifikatorji entitetnih tipov**
  - b)  $T: V_1 \times \dots \times V_k \times E_{T_1} \times \dots \times E_{T_n} \rightarrow E_T$  je **totalna** ali **parcialna** enovrednostna funkcija, kar zagotavlja, da se vsak element kartezijskega produkta vrednostnih množic, ki so območja identifikacijskih atributov, preslika v največ eno entiteto tipa  $A$
  - c) Je **minimalna podmnožica**, ne obstaja **prava podmnožica**, za katero bi tudi veljal pogoj b)

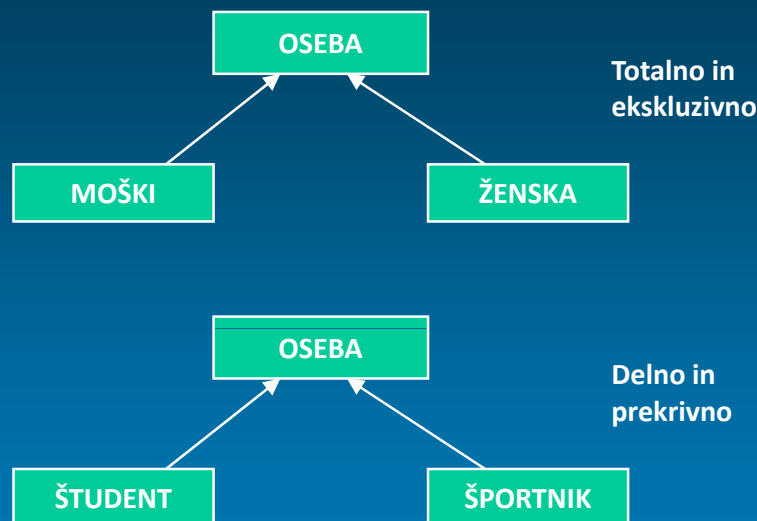


## Generalizacija in specializacija...

- Entitetni tip A s podtipoma B in C
- B in C pokrivata A **totalno in ekskluzivno**, če velja:  $EB \cup EC = EA$  in  $EB \cap EC = \{\}$
- B in C pokrivata A **totalno in prekrivno**, če velja:  $EB \cup EC = EA$  in  $EB \cap EC \neq \{\}$
- B in C pokrivata A **delno in ekskluzivno**, če velja:  $EB \cup EC \subset EA$  in  $EB \cap EC = \{\}$
- B in C pokrivata A **delno in prekrivno**, če velja:  $EB \cup EC \subset EA$  in  $EB \cap EC \neq \{\}$



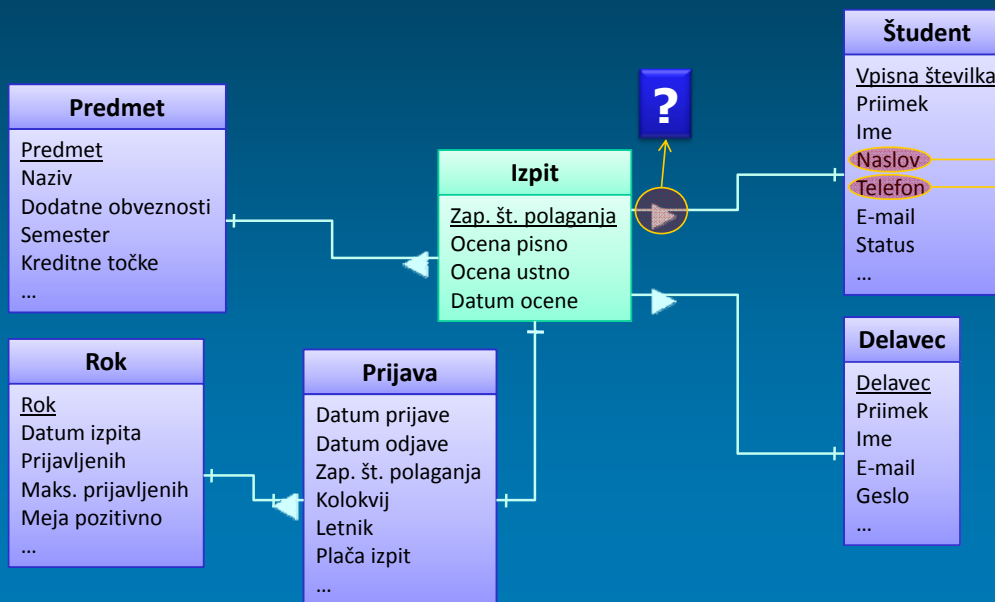
## Generalizacija in specializacija



# Konceptualni model

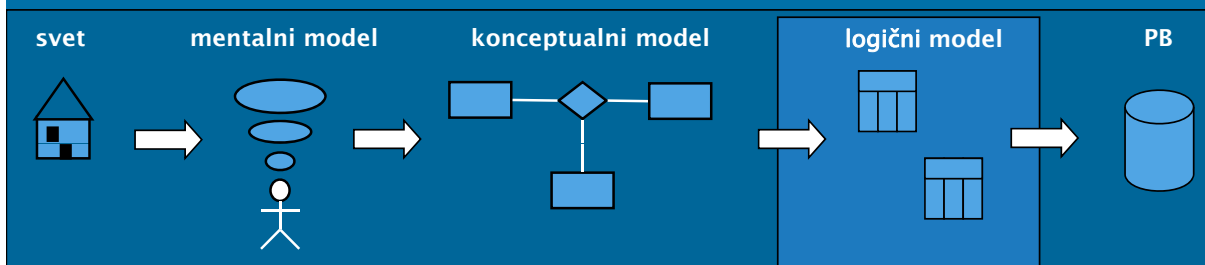
Primer konceptualnega modela

V konceptualnem modelu lahko nastopajo tudi **sestavljeni** in **večvrednostni** atributi!

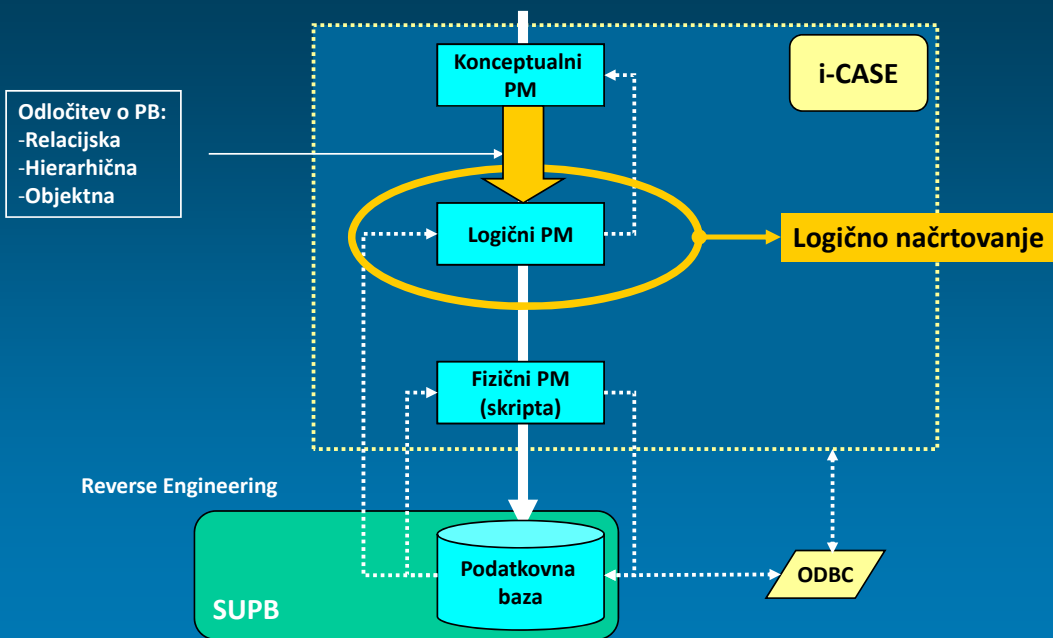


## Logično načrtovanje podatkovne baze...

- Logično modeliranje podatkovne baze nastopi za konceptualnim modeliranjem.
- Osnova logičnega modela je jezik, ki je razumljiv ciljnemu SUPB.
- Če izberemo **relacijski SUPB**, potem govorimo o **relacijskem modelu**.



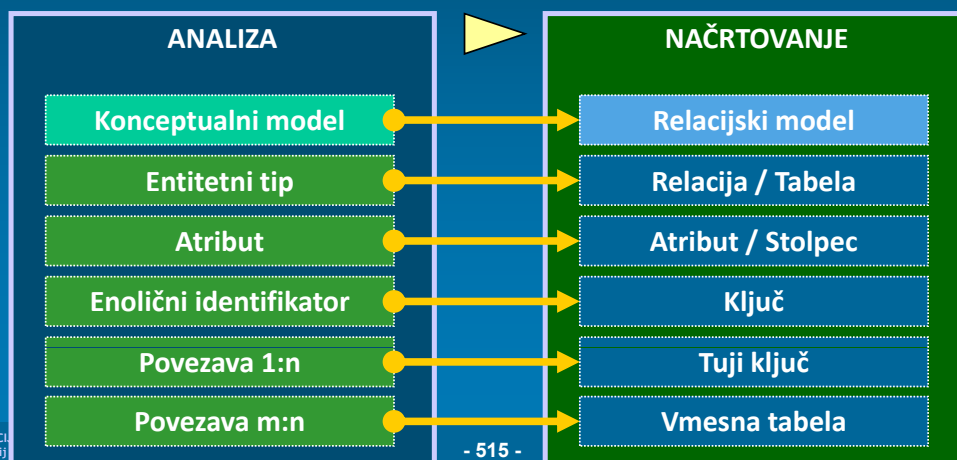
# Podpora orodij CASE



# Prehod iz konceptualnega v logični model

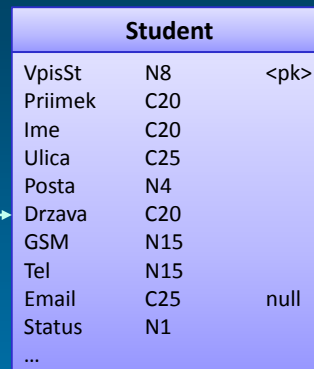
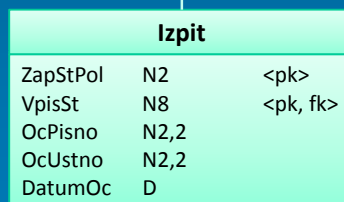
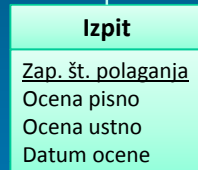
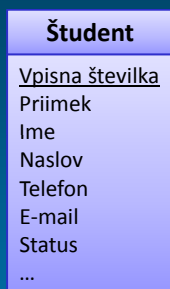
- Prehod iz konceptualnega v logični model je navadno avtomatiziran s strani CASE orodij.

Primer: vrsta baze: relacijska, SUPB: Oracle



# Relacijski podatkovni model

Primer pretvorbe konceptualnega v relacijski model



## Funkcionalne odvisnosti...

- Predpostavimo, da obstaja relacijska shema  $R$  z množico atributov, katere podmnožici sta  $X$  in  $Y$ .
- V relacijski shemi  $R$  velja  $X \rightarrow Y$  ( $X$  funkcionalno določa  $Y$  oziroma  $Y$  je funkcionalno odvisen od  $X$ ), če v nobeni relaciji, ki pripada shemi  $R$ , ne obstajata dve  $n$ -terici, ki bi se ujemale v vrednostih atributov  $X$  in se ne bi ujemale v vrednostih atributov  $Y$ .

## Funkcionalne odvisnosti

- Množico funkcionalnih odvisnosti, ki veljajo med atributi funkcionalne sheme R in v vseh njenih relacijah, označimo s F

$$X \rightarrow Y \in F \Leftrightarrow \forall r ( \text{Sh}(r) = R \Rightarrow \forall t, \forall u (t \in r \wedge u \in r \wedge t.X = u.X \Rightarrow t.Y = u.Y) )$$

kjer

t.X, u.X, t.Y in u.Y označujejo vrednosti atributov X oziroma Y v n-tericah t oziroma u.



## Primeri funkcionalnih odvisnosti

- Imamo relacijo s shemo
  - Izpit( VpŠt, Priimek, Ime, ŠifraPredmeta, Datum izpita, OcenaPisno, OcenaUstno)
- z naslednjim pomenom:  
Študent z vpisno številko VpŠt ter priimkom Priimek in imenom Ime je na DatumIzpita opravljal izpit iz predmeta s šifro ŠifraPredmeta. Dobil je oceno OcenaPisno in OcenaUstno.
- Funkcionalne odvisnosti relacijske sheme Izpit so:  
$$F \equiv \{ \text{VpŠt} \rightarrow (\text{Priimek}, \text{Ime}), (\text{VpŠt}, \text{ŠifraPredmeta}, \text{DatumIzpita}) \rightarrow (\text{OcenaPisno}, \text{OcenaUstno}) \}$$



## Ključni relacije...

- Ker je relacija množica n-teric, so v njej vse n-terice ločene med seboj.
- Za sklicevanje na posamezno n-terico ni potrebno poznati vseh vrednosti atributov n-terice, če v shemi nastopajo funkcionalne odvisnosti.
- Množici atributov, ki določajo vsako n-terico, pravimo ključ relacije oziroma ključ relacijske sheme.



## Ključni relacije...

- Predpostavimo, da obstaja relacijska shema z atributi  $A_1 A_2 \dots A_n$  katere podmnožica je množica atributov  $X$ .
- Atributi  $X$  so ključ relacijske sheme oziroma pripadajočih relacij, če sta izpolnjena naslednja dva pogoja:
  - $X \rightarrow A_1 A_2 \dots A_n$
  - ne obstaja  $X'$ , ki bi bila prava podmnožica od  $X$  in ki bi tudi funkcionalno določala  $A_1 A_2 \dots A_n$



## Ključni relacije...

- Poznamo več vrst ključev:
  - Kandidat za ključ (a key candidate)
  - Primarni ključ (primary key)
  - Superključ (superkey)
  - Tuji ključ (foreign key)
- Kandidat za ključ je vsaka podmnožica atributov relacije, ki relacijo enolično določa.



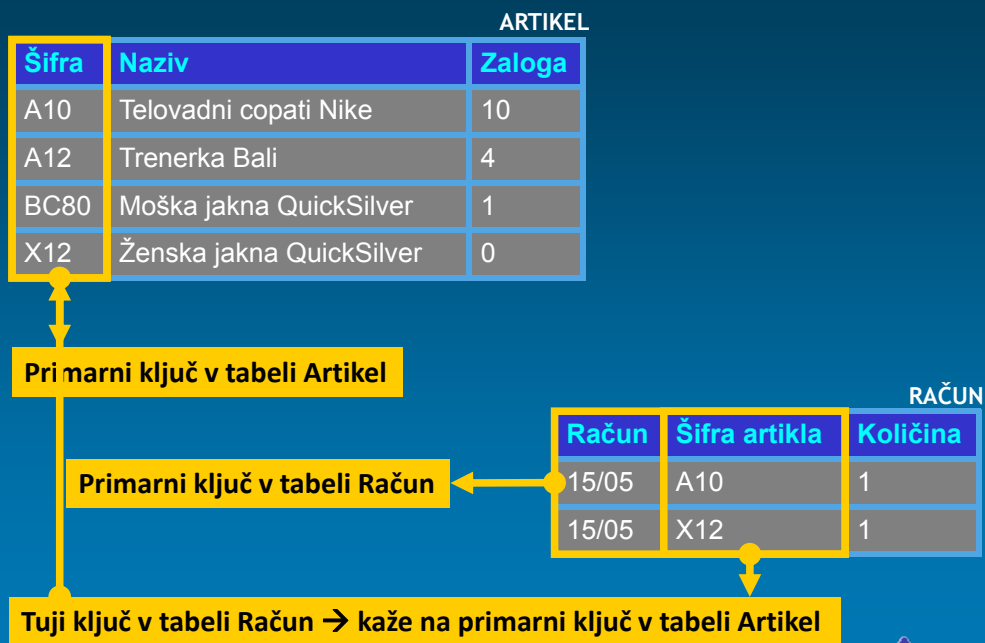
## Ključni relacije

- Primarni ključ je tisti kandidat za ključ, ki ga izberemo za shranjevanje relacij v fizični podatkovni bazi.
- Superključ je vsaka množica atributov, v kateri je vsebovan ključ → ključ je podmnožica superključa.
- Tuji ključ je množica atributov, v okviru ene relacije, ki je enaka kandidatu za ključ neke druge ali iste relacije.





## Primeri ključev



## Normalizacija...

- Kaj si bomo pogledali?
  - Namen normalizacije.
  - Uporaba normalizacije pri načrtovanju relacijske podatkovne baze.
  - Problemi zaradi redundance podatkov v osnovnih relacijah.
  - Postopek normalizacije.
  - Osnovne normalne oblike:
    - I. normalna oblika,
    - II. Normalna oblika,
    - III. Normalna oblika
    - IV. Poslovna normalna oblika



## Namen normalizacije...

- Normalizacija je postopek, s katerem pridemo do množice primernih relacij, ki ustrezajo potrebam poslovne domene.
- Nekaj lastnosti primernih relacij:
  - Relacije imajo minimalen nabor atributov → zgolj tiste, ki so potrebni za pokritje potreb poslovnega sistema;
  - Atributi, ki so logično povezani, so zajeti v isti relaciji;
  - Med atributi relacij je minimalna redundanca → vsak atribut (razen tujih ključev) je predstavljen samo enkrat.



## Prednosti pravilnega načrtovanja

- Osnovni cilj načrtovanja relacijske podatkovne baze je grupirati attribute v relacije tako, da bo čim manj redundance med podatki.
- Potencialne koristi pravilnega načrtovanja so:
  - Spremembe podatkov v podatkovni bazi dosežemo z minimalnim številom operacij → večja učinkovitost; manj možnosti za podatkovne nekonsistentnosti.
  - Manjše potrebe po diskovnih kapacitetah za shranjevanje osnovnih relacij → manjši stroški.



## Primer

- Relacija StaffBranch ima odvečne podatke.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Atribut z odvečnimi (ponavljajočimi) podatki

## Ažurne anomalije

- Relacije, ki vsebujejo odvečne podatke lahko povzročajo anomalije pri spreminjanju podatkov → govorimo o **ažurnih anomalijah**.
- Poznamo več vrst anomalij:
  - Anomalije pri dodajanju n-teric v relacijo
  - Anomalije pri brisanju n-teric iz relacije
  - Anomalije pri spreminjanju n-teric

## Anomalije pri dodajanju

### Primeri anomalij:

- Če želimo dodati podatke o novih članih (staff) za neko organizacijsko enoto (branch) moramo vpisati tudi vse podrobnosti o organizacijski enoti.
- Če želimo dodati podatke o novi organizacijski enoti, ki še nima nobenega člana, moramo v vsa polja, ki člane opisujejo, vpisati Null.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

## Anomalije pri brisanju

### Primeri anomalij:

- Če iz relacije zberemo n-terico, ki predstavlja zadnjega člana v neki organizacijski enoti, zgubimo tudi podatke o tej organizacijski enoti.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

## Anomalije pri spreminjanju

- Primeri anomalij:
  - Če želimo spremeniti vrednost nekega atributa določene organizacijske enote (npr. naslov), moramo popraviti vse n-terice, v katerih takšna vrednost atributa nastopa.

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London



## Postopek normalizacije

- Postopku preoblikovanja relacij v obliko, pri kateri do ažurnih anomalij ne more priti, pravimo **normalizacija**.
- Obstaja več stopenj normalnih oblik. Obravnavali bomo:
  - 1NO – Prva normalna oblika
  - 2NO – Druga normalna oblika
  - 3NO – Tretja normalna oblika in
  - 4PNO – Četrta poslovna normalna oblika



## 1NO – prva normalna oblika

- **Relacija je v prvi normalni obliki, če:**
  - Nima **ponavljajočih atributov** → ne obstajajo atributi ali skupine atributov, ki bi imele več vrednosti pri isti vrednosti ostalih atributov (na presečišču ene vrstice in enega stolpca je več vrednosti)
  - Ima definiran primarni ključ in določene funkcionalne odvisnosti
- **Koraki:**
  - Odstranimo ponavljajoče attribute
  - Določimo funkcionalne odvisnosti
  - Določimo primarni ključ



## Primer – relacija v nenormalizirani obliki

Indeks( VŠ, priimek, ime, pošta, kraj, ( šifra predmeta, naziv, ocena ) )

Skupina ponavljajočih se atributov.

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
					20021	TPO	8
					20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
					20033	IPI	6



## Primer – pretvorba v 1NO...

Indeks( VŠ, priimek, ime, pošta, kraj, ( šifra predmeta, naziv, ocena ) )



Odpravimo ponavljajoče attribute

Indeks( VŠ, priimek, ime, pošta, kraj, šifra predmeta, naziv, ocena )



Identificiramo funkcionalne odvisnosti

$F \equiv \{ V\check{S} \rightarrow (\text{priimek, ime, pošta, kraj}), \text{šifra predmeta} \rightarrow \text{naziv, pošta} \rightarrow \text{kraj}, (V\check{S}, \text{šifra predmeta}) \rightarrow \text{ocena} \}$



Določimo primarni ključ

Indeks( VŠ, priimek, ime, pošta, kraj, šifra predmeta, naziv, ocena )



## Primer – pretvorba v 1NO

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
					20021	TPO	8
					20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
					20033	IPI	6



VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
64010632	Bratina	Simon	4100	Kranj	20021	TPO	8
64010632	Bratina	Simon	4100	Kranj	20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
64016209	Bizjak	Tadeja	2250	Ptuj	20033	IPI	6



## 2NO – druga normalna oblika

- **Relacija je v drugi normalni obliki:**
  - Če je v prvi normalni obliki in
  - Ne vsebuje **parcialnih odvisnosti** → noben atribut, ki ni del ključa, ni funkcionalno odvisen le od dela primarnega ključa, temveč od celotnega ključa
- **Druga normalna oblika je odvisna predvsem od ključa relacije. Relacija je avtomatsko v drugi normalni obliki, če:**
  - Je njen primarni ključ sestavljen le iz enega atributa,
  - Je njen primarni ključ sestavljen iz vseh atributov relacije ali
  - Je njen primarni ključ sestavljen iz vseh razen enega atributa relacije



## Primer – pretvorba v 2NO...



Relacijo razbijemo

Študent( VŠ, priimek, ime, pošta, kraj )  
Predmet( šifra predmeta, naziv )  
Indeks( #VŠ, #šifra predmeta, ocena )





## Primer – pretvorba v 2NO

VŠ	priimek	ime	pošta	kraj	šifra predmeta	naziv	ocena
64010632	Bratina	Simon	4100	Kranj	20020	IS	10
64010632	Bratina	Simon	4100	Kranj	20021	TPO	8
64010632	Bratina	Simon	4100	Kranj	20033	IPI	8
64016209	Bizjak	Tadeja	2250	Ptuj	20060	E1	9
64016209	Bizjak	Tadeja	2250	Ptuj	20033	IPI	6

VŠ	priimek	ime	pošta	kraj
64010632	Bratina	Simon	4100	Kranj
64016209	Bizjak	Tadeja	2250	Ptuj

šifra predmeta	naziv
20020	IS
20021	TPO
20033	IPI
20060	E1
20033	IPI

VŠ	šifra predmeta	ocena
64010632	20020	10
64010632	20021	8
64010632	20033	8
64016209	20060	9
64016209	20033	6

## 3NO – tretja normalna oblika

- **Relacija je v tretji normalni obliki:**
  - Če je v drugi normalni obliki in
  - Če ne vsebuje **tranzitivnih funkcionalnih odvisnosti** → med atributi, ki niso del primarnega ključa, ni odvisnosti.
- **Relacija je avtomatsko v tretji normalni obliki, če:**
  - Je njen ključ sestavljen iz vseh atributov relacije
  - Je njen ključ sestavljen iz vseh razen enega atributa relacije.

## Primer – pretvorba v 3NO...

Študent( VŠ, priimek, ime, pošta, kraj)  
Predmet( šifra predmeta, naziv)  
Indeks( #VŠ, #šifra predmeta, ocena)



Relacijo razbijemo

Študent( VŠ, priimek, ime, #pošta)  
Pošta(pošta, kraj)  
Predmet( šifra predmeta, naziv)  
Indeks( #VŠ, #šifra predmeta, ocena)



## Primer – pretvorba v 3NO

VŠ	priimek	ime	pošta	kraj
64010632	Bratina	Simon	4100	Kranj
64016209	Bizjak	Tadeja	2250	Ptuj



VŠ	priimek	ime	pošta
64010632	Bratina	Simon	4100
64016209	Bizjak	Tadeja	2250

pošta	kraj
4100	Kranj
2250	Ptuj

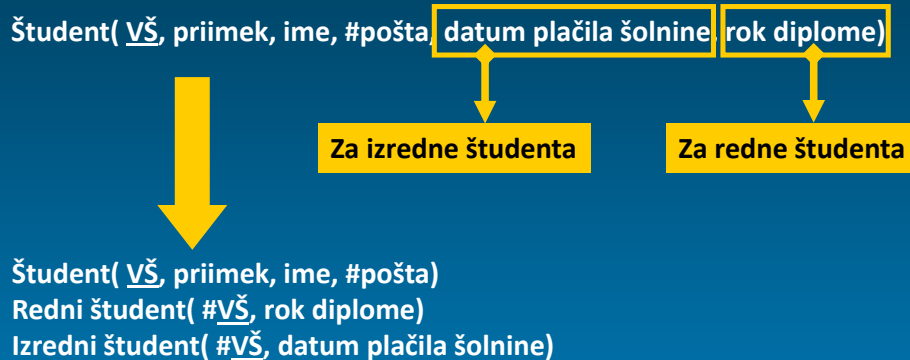


## 4PNO – četrta poslovna normalna oblika

- Relacija je v četrthi poslovni normalni obliki, če:
  - je v tretji normalni obliki in
  - v relaciji ne obstajajo atributi, ki bi bili odvisni od vrednosti primarnega ključa.



## Primer – pretvorba v 4PNO...



## Primer – pretvorba v 4PNO

VŠ	Priimek	Ime	Datum plačila šolnine	Rok diplome
64010632	Bratina	Simon		15.3.2005
64016209	Bizjak	Tadeja	19.4.2002	
64010670	Berce	Marjan	12.4.2004	
64620010	Mele	Silvana		1.4.2005
65120987	Leban	Tibor		15.7.2005



VŠ	Priimek	Ime
64010632	Bratina	Simon
64016209	Bizjak	Tadeja
64010670	Berce	Marjan
64620010	Mele	Silvana
65120987	Leban	Tibor

VŠ	Datum plačila šolnine
64016209	19.4.2002
64010670	12.4.2004

VŠ	Rok diplome
64010632	15.3.2005
64620010	1.4.2005
65120987	15.7.2005



## Uporaba nenormaliziranih relacij...

- Včasih zavestno uporabljamo relacije, ki ne ustrezajo najvišjim normalnim oblikam.
- Prve in druge normalne oblike nikoli ne kršimo.
- Višjim normalnim oblikam se včasih odrečemo na račun doseganja boljše učinkovitosti.



## Uporaba nenormaliziranih relacij

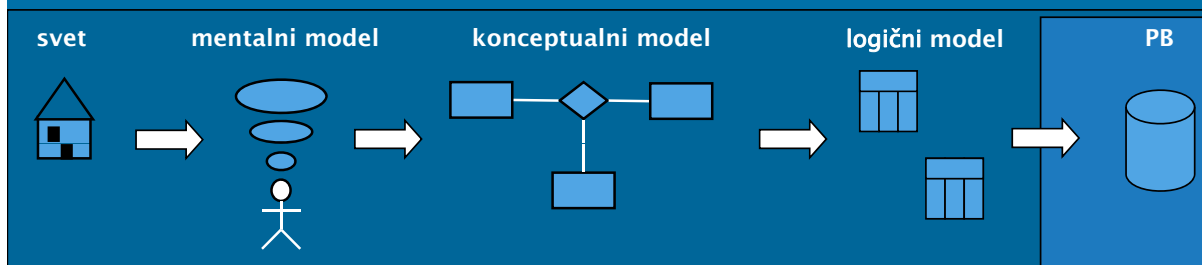
### Primer:

- Rezultat (športnik, tekmovanje, čas prvega teka, čas drugega teka, čas skupaj)
- Relacija ni v tretji normalni formi.
- Čas skupaj je izpeljan atribut → ni odvisen od ključa, temveč je seštevek časov obeh tekov.
- Skupen čas računamo ob vpisu v bazo, zato izboljšamo učinkovitost pri nadaljnji obdelavi podatkov.

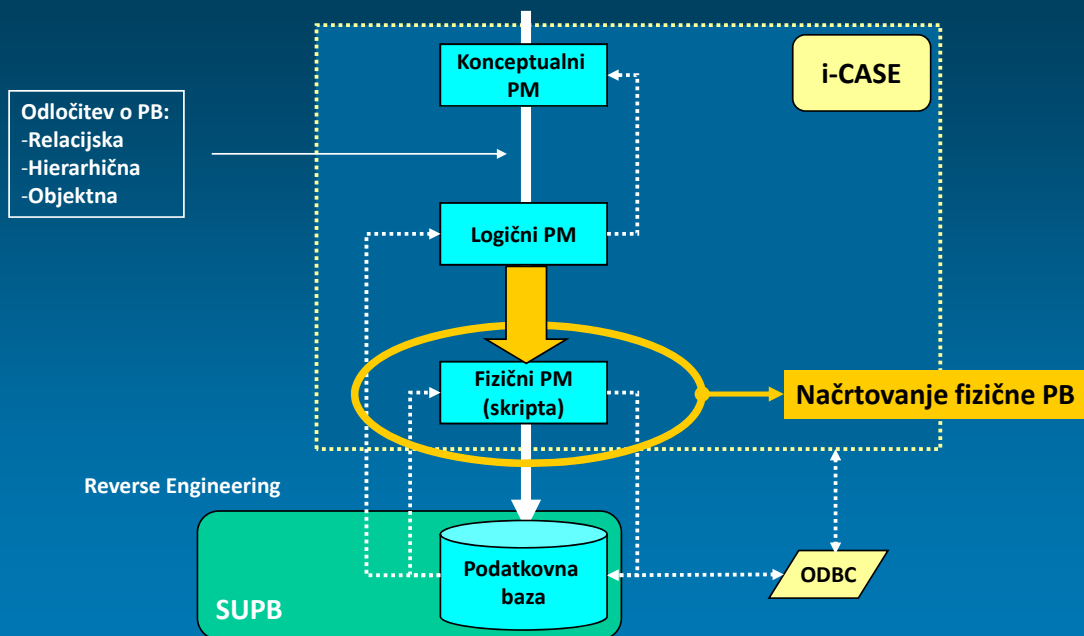


## Načrtovanje fizične podatkovne baze...

- Načrtovanje fizične PB je korak, ki sledi logičnemu načrtovanju.
- Vhod v načrtovanje fizične PB so:
  - Logični podatkovni model,
  - Relacijska shema
  - dokumentacija



## Načrtovanje fizične podatkovne baze...



## Načrtovanje fizične podatkovne baze

- Fizično načrtovanje PB opredeljuje proces, s katerim izdelamo opis implementacije PB na sekundarnem pomnilnem mediju.
- Opisuje
  - osnovne relacije,
  - datotečno organizacijo,
  - indekse za doseg učinkovitega dostopa do podatkov,
  - povezane omejitve in
  - varnostne mehanizme.