
RAČ04

Potek - teorija

Teorija (2 ure/teden):

- osnovni pojmi v informatiki in informacijski sistemi
- baze podatkov
 - SUPB
 - uporabniki
 - varnost
- UML modeli
- SQL
- ER modeli

Ocenjevanje:

- po vsakem sklopu test (4x)
- ustno – brez najave
- domače/seminarske naloge

Učni pripomočki

- **Andreja Šet: Načrtovanje in postavitve podatkovnih baz**
- zapiski
- računalniške revije
- spletni viri (npr. w3schools)
- knjige:
 - T. Mohorič: Uvod v podatkovne baze, Ljubljana, BI-TIM, cop. 1995
 - T. Mohorič: Načrtovanje relacijskih podatkovnih baz, Ljubljana , BI-TIM, cop. 1997
 - Tomas Connolly & Carolyn Begg: Database System, A practical approach to design implementation and management, Edison Wessly, 1999
 - ...

PODATEK - INFORMACIJA

Osnove informatike: podatek in informacija

Podatek predstavlja dejstva, pojme na formaliziran način, s simboli.

Informacija: pomen, ki ga človek predpiše podatkom s pomočjo znanih konvencij, ki so uporabljene pri njihovi predstavitvi.

Vsi podatki ne vsebujejo informacije.

Podatki, ki se jih ne da interpretirati, ne vsebujejo nobenih informacij.

Podatki: 32, 33, 32, 35, 34, 36, 35, 37, ...

Informacija: temperature v več letih

Informacija

Definicije (!):

- *je pomen, ki ga človek pripiše podatkom s pomočjo znanih konvencij, ki so uporabljene pri njihovi predstavitvi.*
- so podatki, ki so obdelani tako, da so dobili pomen. (Bocil, 1999)
- so podatki, ki so tako oblikovani, da imajo pomen in so koristni uporabnikom (uporabni za ljudi). (Laudon & Laudon, 2000)
- so podatki, katerih oblika in vsebina je primerna za določeno uporabo. (Alter, 1999)

Količina informacije

V informacijski teoriji je količina informacije lastnost, ki jo ima podatek in jo lahko izmerimo.

Merjenje količine informacij: velja, da informacija nastane, kadar se zgodi poskus, katerega izid ni vnaprej določen.

Osnovna enota informacije nastane pri poskusu z dvema enako verjetnima izidoma.

Primeri poskusov: stanje stikal, temperature, svetlost slikovne točke...

Enote za količino informacije

Bit (oznaka "b") je najmanjša enota za merjenje množine informacij.

1 bit informacije dobimo, če dobimo odgovor na vprašanje, na katerega sta možna le 2 enakovredna odgovora. Ima lahko logično vrednost 0 ali 1.

Zlog (**byte**, oznaka "B") je beseda, sestavljena iz **osmih** bitov.

Četverka (nibble)

kB = 2^{10} byte-ov = 1024 byte-ov (malo manj kot pol strani besedila)

MB = 2^{20} byte-ov = 1024 Kbyte-ov $\sim 10^6$ byte-ov (cca 500 strani debela knjiga)

GB = 2^{30} byte-ov = 1024 Mb yte-ov $\sim 10^9$ byte-ov (200m dolga knjižna polica)

TB = 2^{40} byte-ov = 1024 Gbyte-ov $\sim 10^{12}$ byte-ov (???)

Več kot bit

1 Kilobyte	=	2^{10}
1 Megabyte	=	2^{20}
1 Gigabyte	=	2^{30}
1 Terabyte	=	2^{40}
1 Petabyte	=	2^{50}
1 Exabyte	=	2^{60}
1 Zettabyte	=	2^{70}
1 Yottabyte	=	2^{80}
1 Brontobyte	=	2^{90}
1 Geopbyte	=	2^{100}

<http://www.whatsabyte.com/>

<http://www.t1shopper.com/tools/calculate/>

Kodiranje podatkov

Podatke pogosto predstavljamo z **besedami**.

Število različnih besed dobimo s pomočjo formule:

$$Q = k^n$$

Q predstavlja število različnih besed

n predstavlja dolžino besede

k predstavlja število različnih simbolov

V računalniku so besede sestavljene iz bitov, torej je število različnih simbolov **k** enako 2.

$$Q = 2^n$$

Z različnimi besedami opišemo različna stanja sistemov (npr. stanje stikal, barvo pikice na monitorju...) Število bitov v besedi predstavlja količino vsebovane informacije.

Kodiranje podatkov - primeri

Stikalo je lahko v dveh stanjih:

$$2 = 2^n$$

$$n = 1$$

bo	stanje
0	vklopljeno
1	izklopljeno

Stikalo na pečici, 4 (7) stopenjsko.

Točka na monitorju v štirih barvah:

$$4 = 2^n$$

$$n = 2$$

b1	b0	stanje
0	0	bela
0	1	svetlo siva
1	0	temno siva
1	1	črna

Dimenzije informacije

- **Količina informacije**

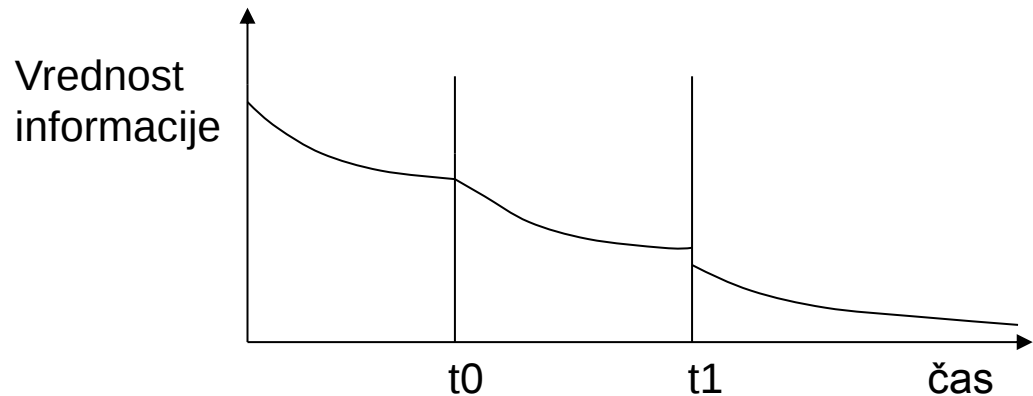
- **Kakovost**

- točnost, popolnost, relevantnost, dosegljivost, preverljivost, dostopnost, varnost, zgoščenost, ustreznost, razumljivost, objektivnost, ...

- **Vrednost**

- časovno odvisna:

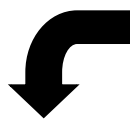
- prediktivna
- operativna
- izkušnjska



Atributi kakovostne informacije

- **Točnost** – točna informacija je brez napak.
- **Popolnost** – popolna informacija vsebuje vsa pomembna dejstva.
- **Relevantnost** – informacija mora biti zanimiva in pomembna za uporabnika.
- **Dosegljivost** – informacija je dostopna uporabnikom takrat, ko jo potrebujejo.
- **Preverljivost** – preverljivo informacijo uporabniki lahko preverijo in se prepričajo o njeni točnosti.
- **Dostopnost** – informacijo lahko pridobijo tisti uporabniki, ki jo potrebujejo.
- **Varnost** – informacija je zaščitena pred nepooblaščenimi uporabniki.

Podatek vs. informacija



Podatek in informacija



- zabeležena dejstva (predstavljena)
- so brez pomena
- surovina, ki jo predeluje IS

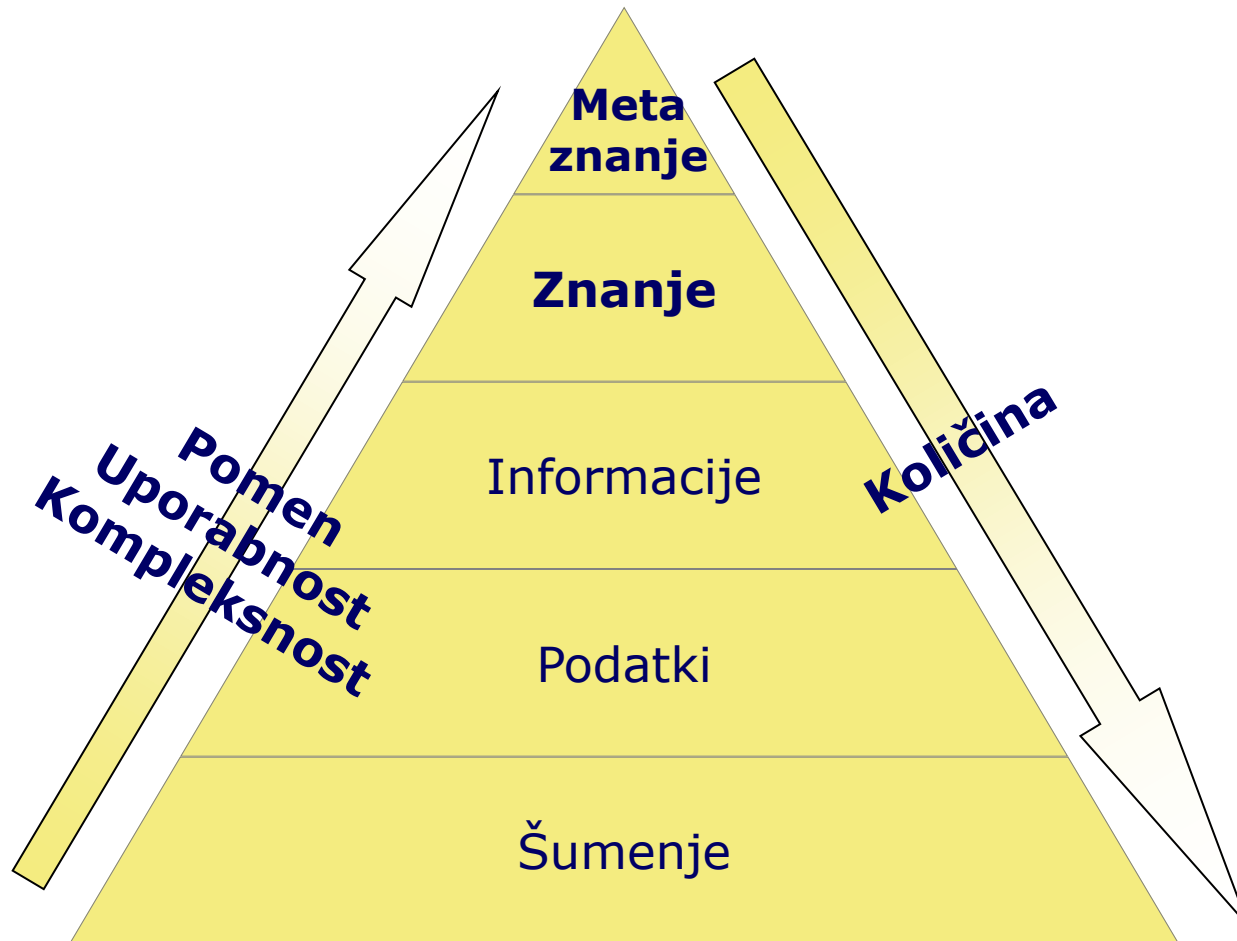
- ima pomen, pove nekaj novega
- poveča znanje prejemnika
- **vpliva na odločitve in ravnanje posameznika**

Podatek je poljubna predstavitev s pomočjo simbolov ali analognih veličin, ki ji je pripisan, ali se ji lahko pripiše nek pomen.

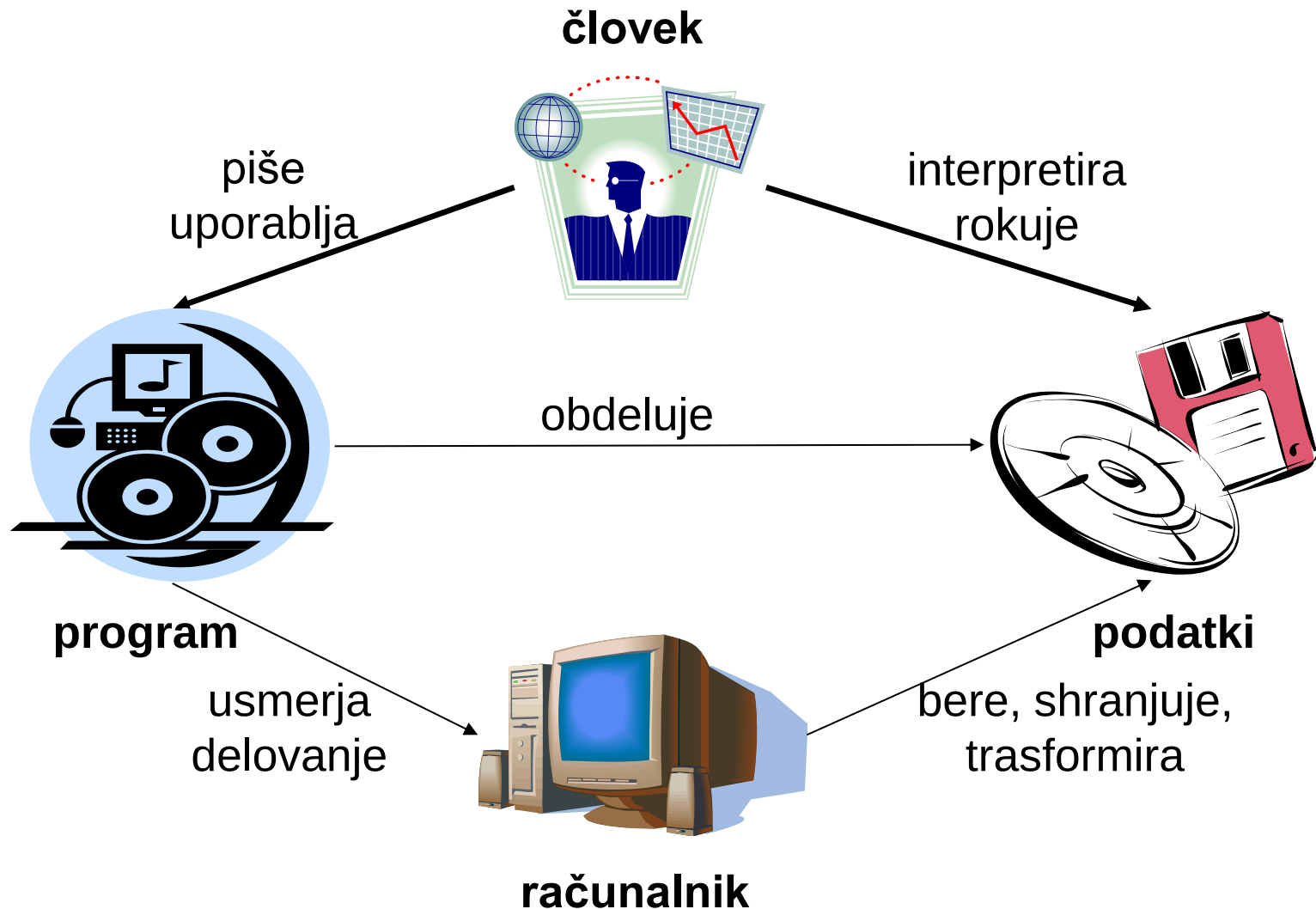
Podatek je predstavitev dejstva, koncepta ali instrukcije na formaliziran način, ki je primeren za komunikacijo, interpretacijo ali obdelavo s strani človeka ali stroja.

Informacija je pomen, ki ga človek pripiše podatkom s pomočjo znanih konvencij, ki so uporabljene pri njihovi predstavitvi.

Informacijska hierarhija



Računalniško podprt podatkovni sistem



Računalnik vs. človek

Zmaga računalnik:

- pomnjenje informacij
- izvajanje operacij brez napak
- nepretrgano delo

Računalnik ne zmore nobene operacije, ki je ne bi zmogel tudi človek sam, vendar je pri nekaterih vrstah operacij *bistveno hitrejši* in *točnejši*.

Zmaga človek:

- znajde se v novi, neznani situaciji
- učenje na podlagi poskusov in napak

Za človeka je značilno *hevristično sklepanje*: to pomeni, da je pri reševanju problemov zmožen med zelo velikim številom možnosti v naprej presoditi, katere bolj verjetno vodijo k rešitvi in katere ne.

Računalnik vs. človek

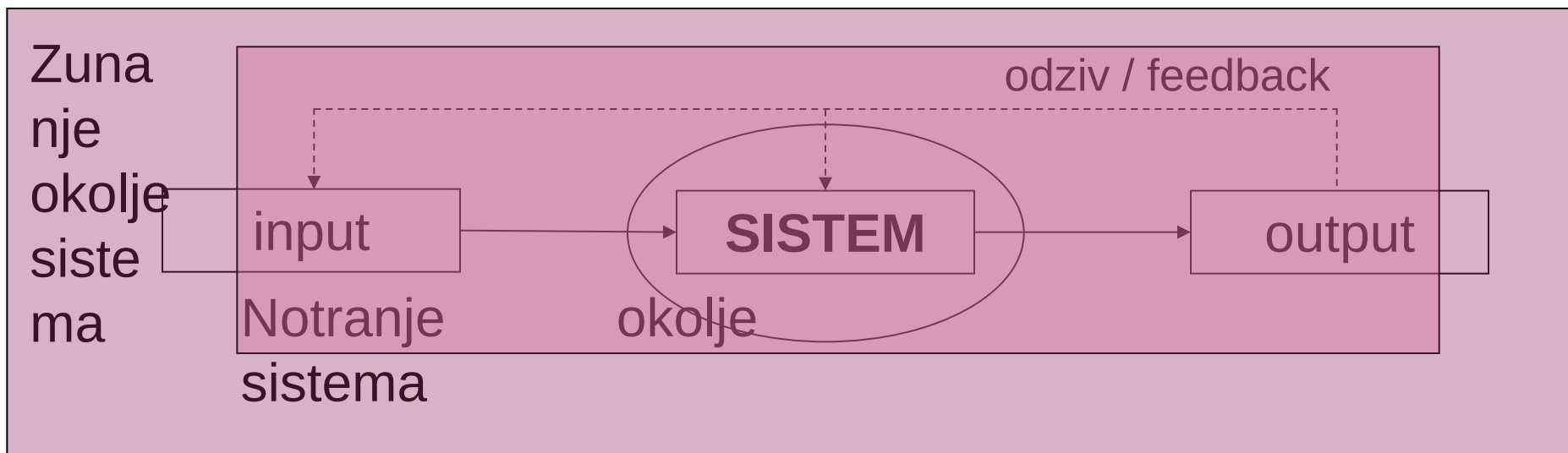
Povzetek:

Človek je sposobnejši v mišljenju, sklepanju in odkrivanju novega, računalnik pa v računanju in manipuliranju s podatki.

INFORMACIJSKI SISTEMI

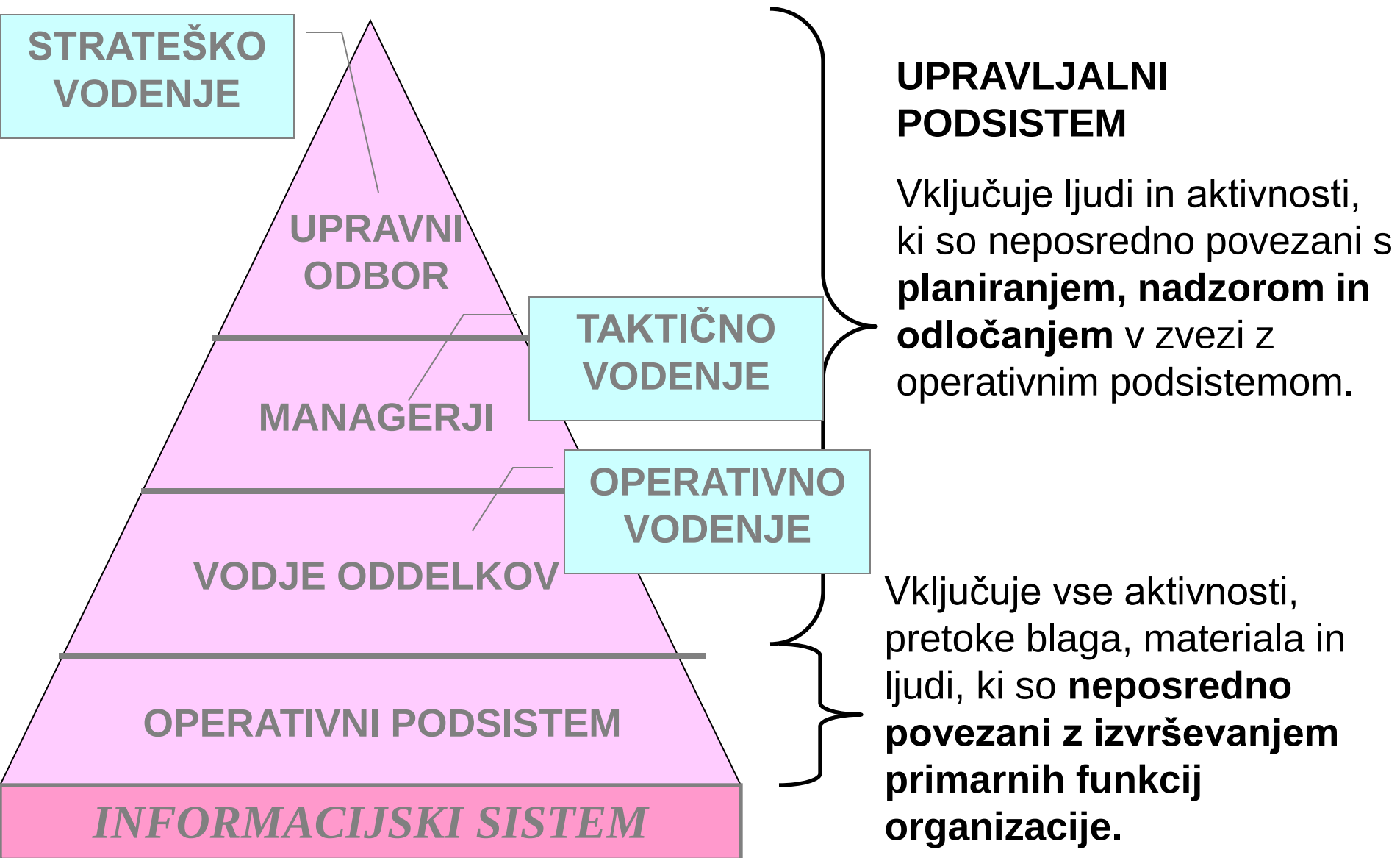
Sistem

- *Sistem* = katerakoli skupina povezanih komponent (funkcije, ljudje, aktivnosti), z namenom doseganja cilja / izvajanja določenih funkcij
- Sistem je vpet v *okolje*.
- Okolje *vpliva* na sistem, sistem pa na okolje.
- *Uspešnost* – zmožnost (pravočasnega) prilagajanja spremembam.



- **Primeri sistemov:** proizvodno podjetje, trgovsko podjetje, gospodinjstvo, izobraževalni sistem, pravosodni sistem, zdravstveni sistem, ...
- **Kompleksnejši sistemi so sestavljeni iz več manjših podsistemov.**

Podsistemi organizacije: upravljalni + operativni + informacijski



Če nimamo informacij,

- je odločanje slabo (neustrezno):
 - neažurne, nepopolne in ne povezane informacije prispevajo k slabim odločitvam, ki vodijo k neuspehu (izgubi finančnih sredstev, izgubi trga, lahko tudi do uničenja podjetja).
- zapravljamo resurse:
 - podvojene informacije največkrat kažejo na zapravljanje resursov (če nekdo iz dveh oddelkov podjetja dobi iste informacije, to pomeni da različni oddelki opravljajo isto delo).
- zapravljamo priložnosti:
 - če informacije niso pravočasne ali so nepopolne, se lahko zamudijo priložnosti.
- izgubljammo stranke / trg:
 - slabe storitve in posledično izguba strank so pogosto posledica neučinkovite distribucije informacij.

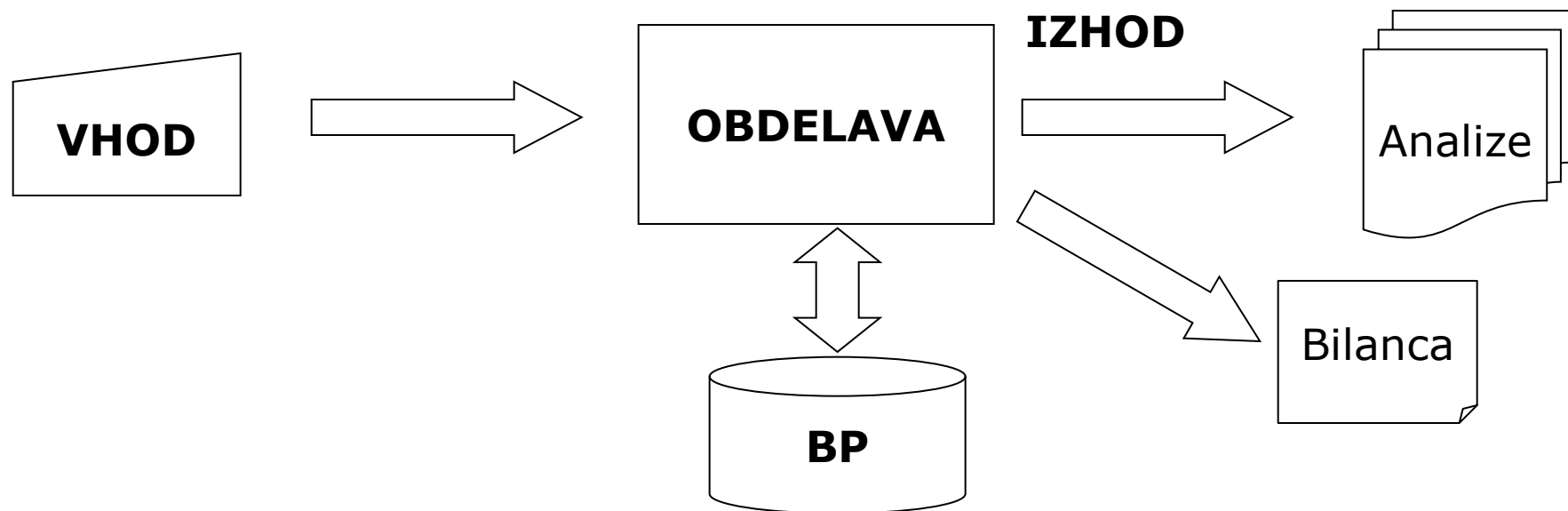
Da bi bili uspešni, moramo imeti kakovostne informacije!

Informacijski sistem

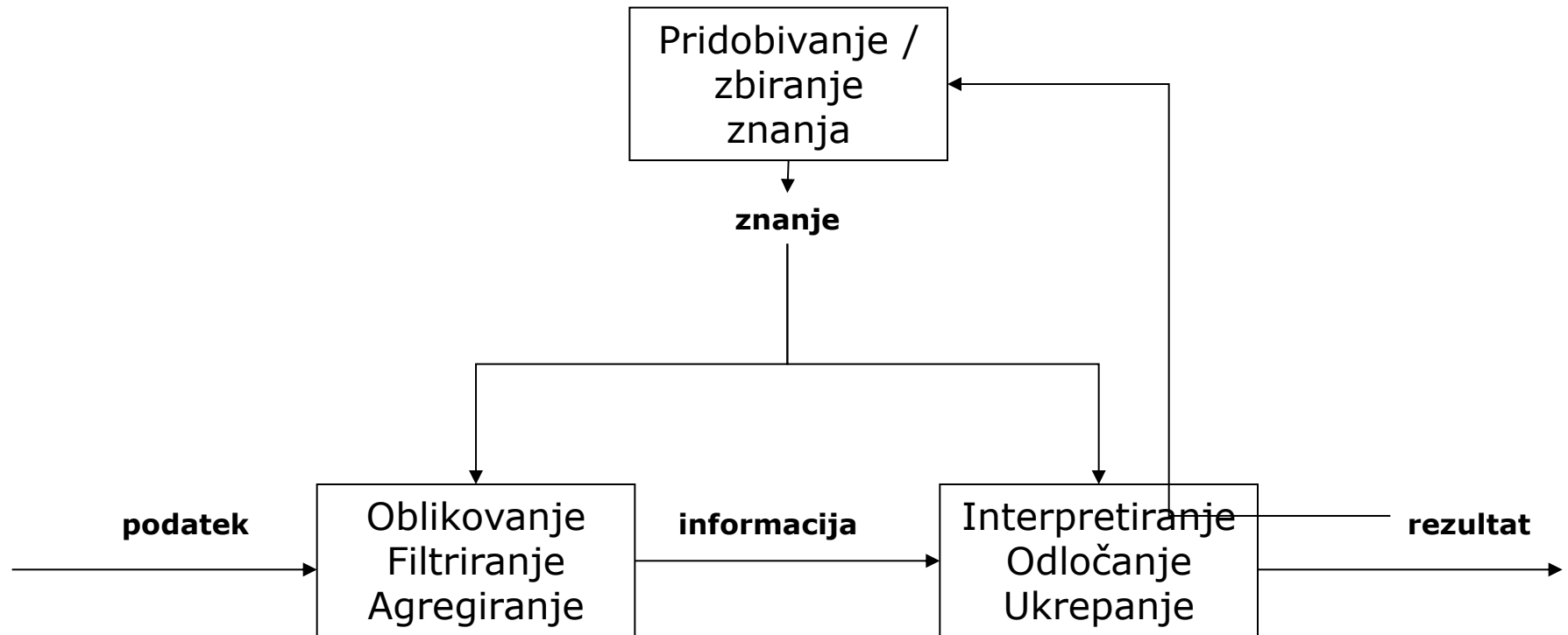
- **IS** je množica aplikacij in aktivnosti, ki so potrebne za upravljanje, obdelavo in uporabo informacij kot virov organizacije.
- omogoča zbiranje, shranjevanje, obdelavo in analizo **podatkov** na način, ki zagotavlja zadovoljevanje informacijskih zahtev na vseh ravneh upravljanja.

Aktivnosti IS

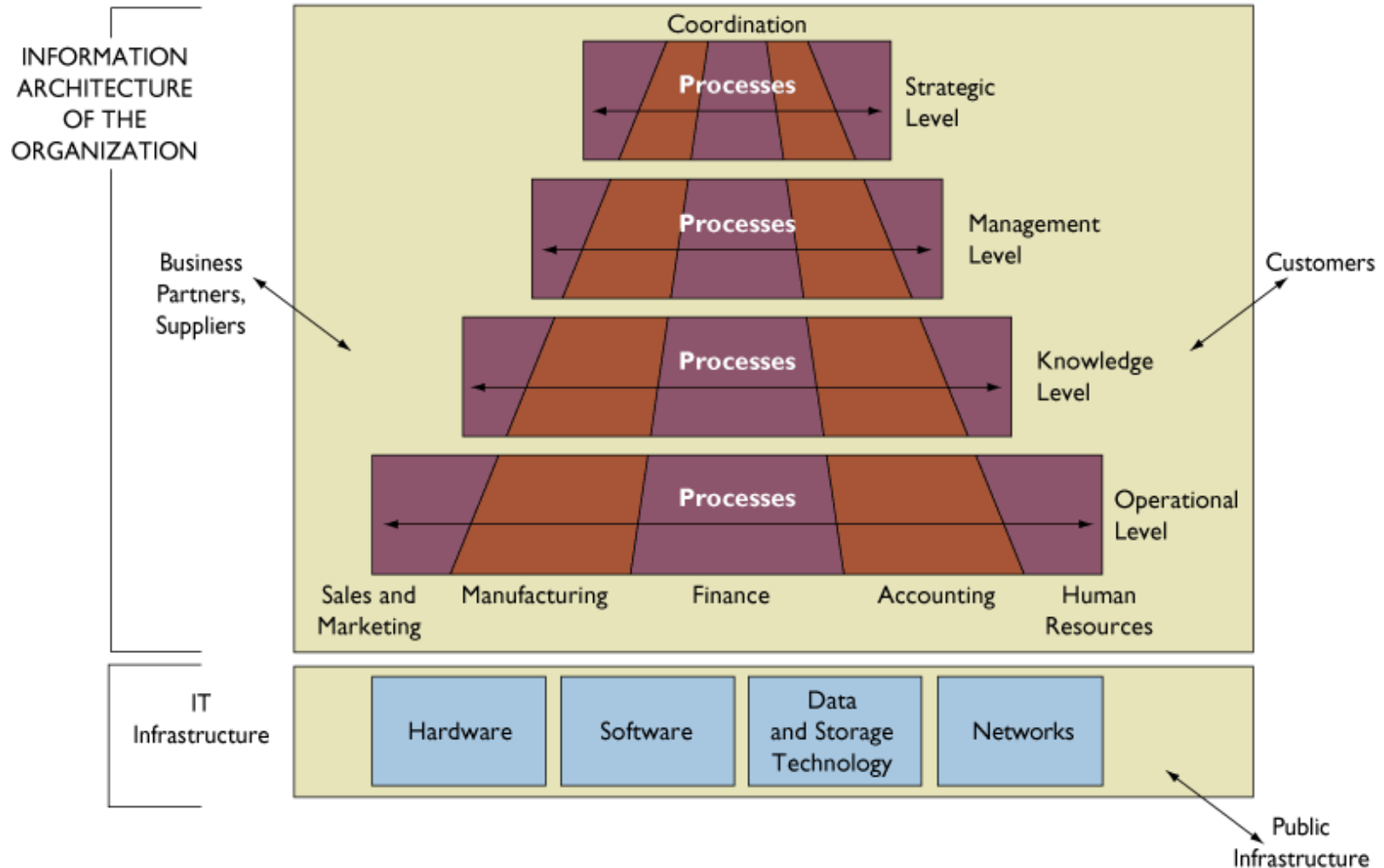
1. **sprejemanje / zajemanje** podatkov iz notranjih in zunanjih virov.
2. **obdelava** podatkov / **shranjevanje**.
3. **podajanje** informacij v zahtevani obliki.



'neskončna zanka'



Informacijska arhitektura in infrastruktura



- **INFORMACIJSKA ARHITEKTURA**
 - načrt informacijskih potreb organizacije.

Inf. sistem ≠ Inf. arhitektura ≠ Inf. infrastruktura

- **INFORMACIJSKI SISTEM**

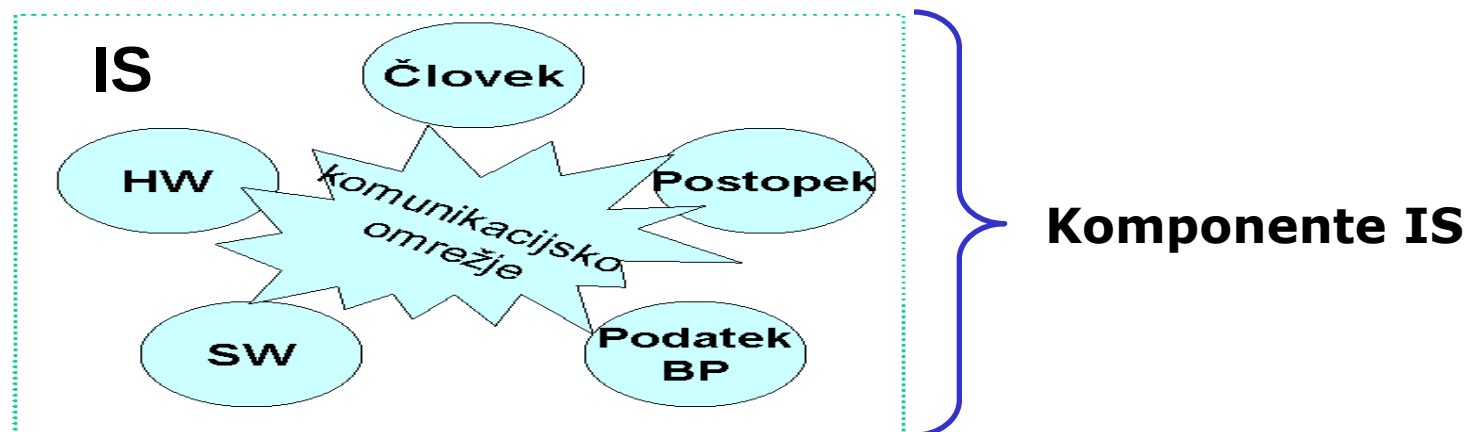
- mora ustrezati informacijski arhitekturi, podpira ga informacijska infrastruktura.

- **INFORMACIJSKA ARHITEKTURA**

- načrt informacijskih potreb organizacije.

- **INFORMACIJSKA INFRASTRUKTURA**

- sestavljena iz fizičnih zmogljivosti, storitev in ljudi, ki podpirajo računalniške resurse podjetja.
- vključuje integracijo, delovanje, dokumentacijo, vzdrževanje in vodenje HW, SW, omrežja in komunikacij, PB in inf. osebja.



Tehnološki trendi

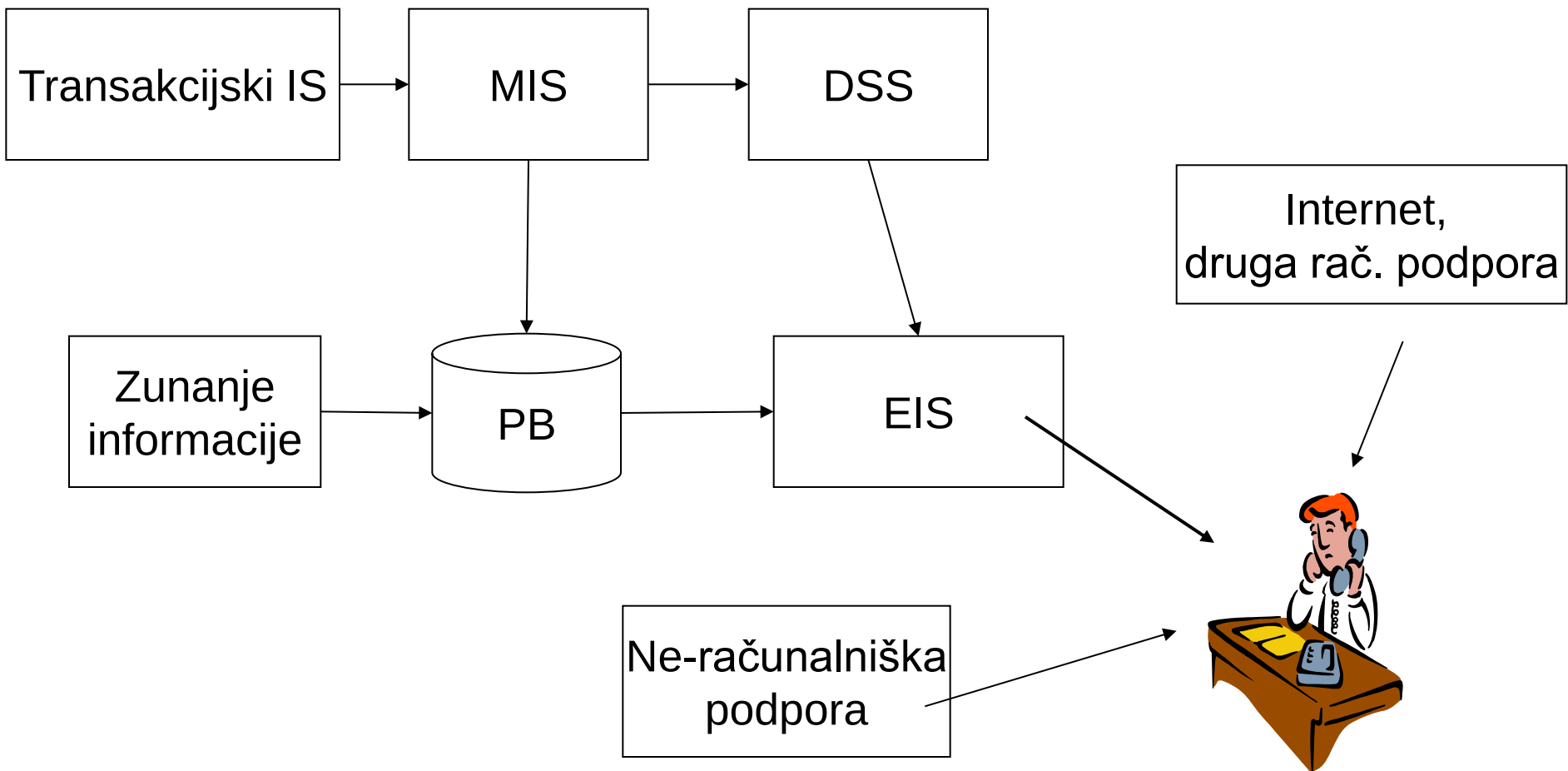
- razmerje cena / zmogljivost (ang. *cost/performance ratio*)
 - *izboljšave tudi za faktor 100* – čez 10 let bo računalnik z enako ceno kot danes vsaj 50x zmogljivejši.
 - **Moorov zakon** (Gordon Moore – soustanovitelj Intel-a) iz leta **1965(!)**: procesorska moč se bo vsakih 18 mesecev podvojila.
- objektno usmerjeno okolje in upravljanje z dokumenti
- omrežno računalništvo – Metcalfe-ov zakon (eden od začetnikov računalniških mrež): vrednost mreže (omrežja) narašča kot kvadratna funkcija števila uporabnikov
- e-poslovanje, m-poslovanje
- računalništvo v oblaku ([video](#))
- omrežena podjetja
- intraneti / ektraneti
- portali
- Večpredstavnost
- Big data
- AI

Delitve IS

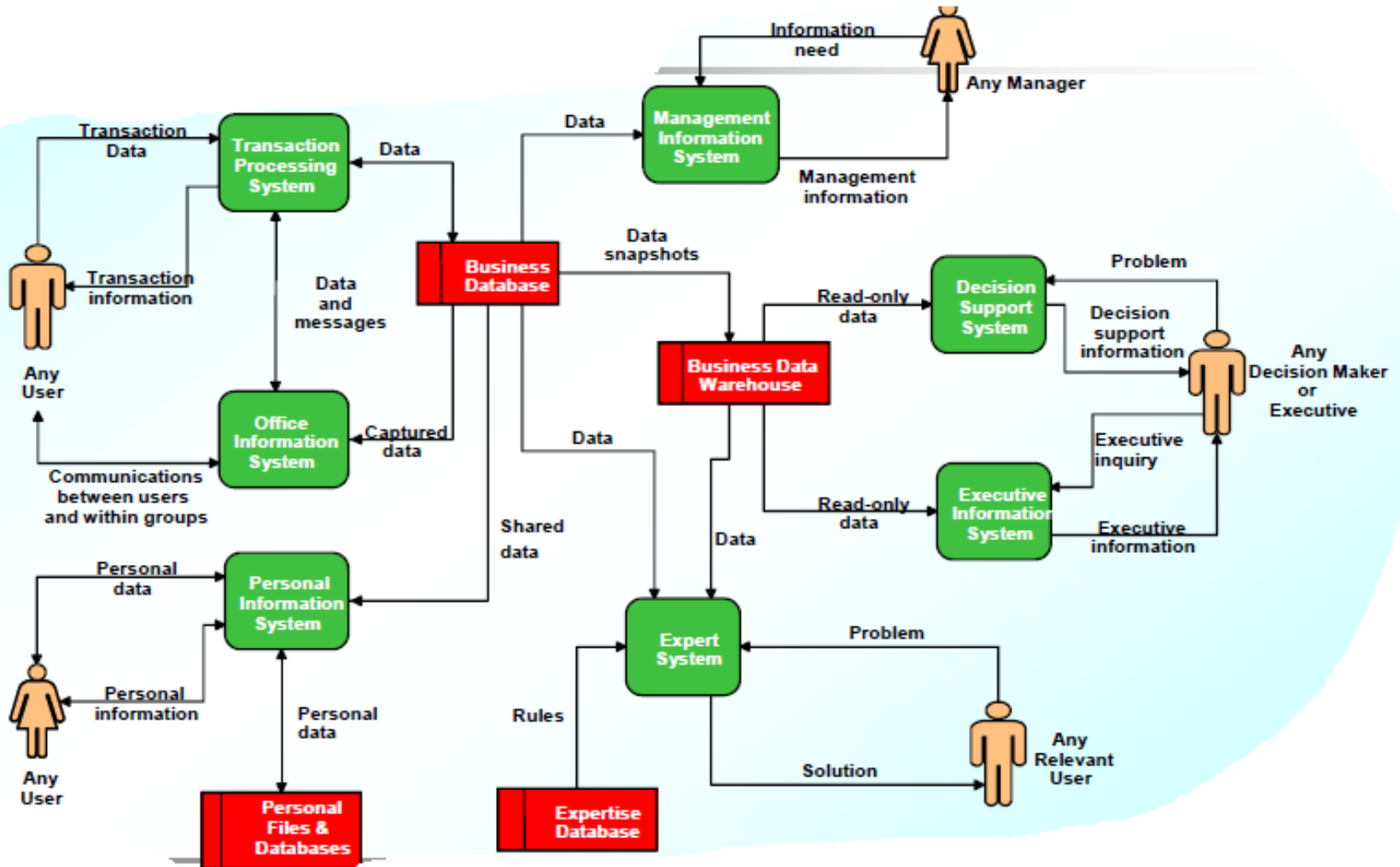
Različne delitve IS – glede na različne vidike:

- **organizacijsko strukturo** (oddelčni, divizijski, IS podjetja, medpodjetniški IS).
- **funkcijsko področje** (računovodstvo, finance, proizvodnja, marketing, ...).
- **sistemska arhitekturo** (mainframe, samostojni PC (?), distribuirani/omrežni).
- **aktivnosti, ki jih podpirajo** (operativni, managerski, strateški).
- **podporo** (transakcijski, upravljalni, vodstveni, IS za podporo odločanju, ekspertni sistemi, za podporo pisarniškemu delu, za podporo skupinskemu delu, ...).

Primer povezanosti IS v organizaciji (1)



Primer povezanosti IS v organizaciji (2)



Vrste IS

- *Transakcijski IS*
- *Managerski IS*
- *Vodstveni IS*
- *IS za avtomatizacijo pisarniškega poslovanje*
- *IS za podporo skupinskemu delu*
- *IS za podporo odločanju*
- ***Ekspertni sistemi***

Transakcijski IS

(ang. ***Transaction Processing System***)

- transakcija = dogodek, ki ima določen vpliv na poslovanje v organizaciji.
 - primer : nakup nekega izdelka, pridobljena ocena dijaka, izposoja knjige, ...
- TIS hrani in obdeluje podatke o poslovnih dogodkih.
- TIS izdelava različne analize / poročila.
- TIS posreduje informacije uporabnikom in drugim IS organizacije.
- Značilnosti:
 - velike količine podatkov.
 - enostavna (nezahtevna) obdelava.
 - namenjen je operativnem nivoju organizacije.
 - odzivni časi so pomembni.
 - veliko število uporabnikov.
 - zahteva po visoki zanesljivosti sistema.

Managerski IS

(ang. **Management Information System**)

- na osnovi podatkov, dobljenih od transakcijskega IS in na osnovi delovanja organizacije zagotavlja informacije in poročila.
- podpora odločanju v primerih, ko so informacijske zahteve vnaprej opredeljene.
- glede na transakcijske IS, managerski IS:
 - imajo manjše število uporabnikov,
 - obdelave so zahtevnejše,
 - odzivni časi so manj pomembni.

Vodstveni IS

(ang. ***Executive Information System***)

- namenjen najvišjemu vodstvu organizacije.
- omogoča pogled / nadzor nad vsemi aktivnostmi organizacije (brez podrobnosti).
- pomaga pri vodenju organizacije (iskanju novih priložnosti za večjo konkurenčnost organizacije).
- podatke črpajo iz notranjih in zunanjih virov.
- zelo pomemben vir podatkov: transakcijski IS in managerski IS.
- odzivni časi niso zelo pomembni.
- majhno število uporabnikov.

Hierarhija uporabe IS

Narašča:

- odgovornost
- pomembnost odločanja
- stopnja agregacije podatkov

Vodstveni
IS

Managerski
IS

Transakcijski IS

Narašča:

- število uporabnikov
- potreba po hitrih odzivnih časih
- potreba po zanesljivosti sistema
- število elementarnih podatkov

Sistemi za avtomatizacijo pisarniškega poslovanja

(ang. **Office Automation System**)

- za povečanje učinkovitosti (pisarniškega) dela in za izboljšanje komunikacije.
- veliko število uporabnikov.
- pomembni odzivni časi, dostopnost sistema, zanesljivost delovanja.
- primer: Lotus Notes.

Sistemi za podporo odločanju - SPO

(ang. **Decision Support System – DSS**)

- nudijo podporo vodstvu v edinstvenih situacijah sprejemanja strateških odločitev.
- bistvo: za dani primer je treba ugotoviti katere dejavnike je potrebno upoštevati in v kolikšni meri.
- zelo zahtevne obdelave.
- odzivni časi so lahko tudi daljši.

Odločanje

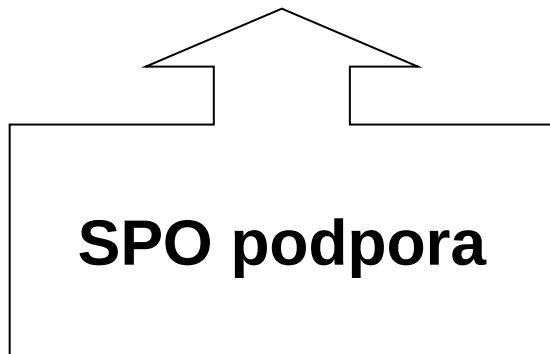
- del procesa reševanja problemov.
- je proces pri katerem izbiramo med več možnostmi – **variantami**; izbrana varianta naj čimbolj ustreza danim ciljem.
- rezultat je odločitev, t.j. izbira najboljše (najoptimalnejše) možnosti (variante) iz množice razpoložljivih.
- elementi: variante, kriteriji, odločitvena (preferenčna) funkcija, ocene variant.

Težave pri odločanju

- ne poznamo vseh faktorjev, ki vplivajo na odločitev.
- variante, med katerimi se odločamo, niso natančno definirane.
- veliko števila dejavnikov, ki vplivajo na odločitev.
- velike količine variant ➔ za natančno študijo odločitvenega problema in variant bi zmanjkalo časa.
- vsi podatki niso vedno dosegljivi.
- celo cilji različnih odločevalcev so lahko različni.
- nepoznavanja odločitvenega problema in ciljev odločitve.
- omejena sredstva (čas, denar, strokovnjaki, ...).
- nesoglasje med ljudmi, ki sodelujejo pri odločanju.

Načini odločanja

- **Naključno** (ecika-pecika) (primer: vegovc rešuje kontrolko 😊)
- **Intuitivno** (ne da bi vedeli, kako to počnemo, se nam 'zdi')
- **Sistematično** (organiziranje podatkov in znanja)



Sistemi za podporo odločanju

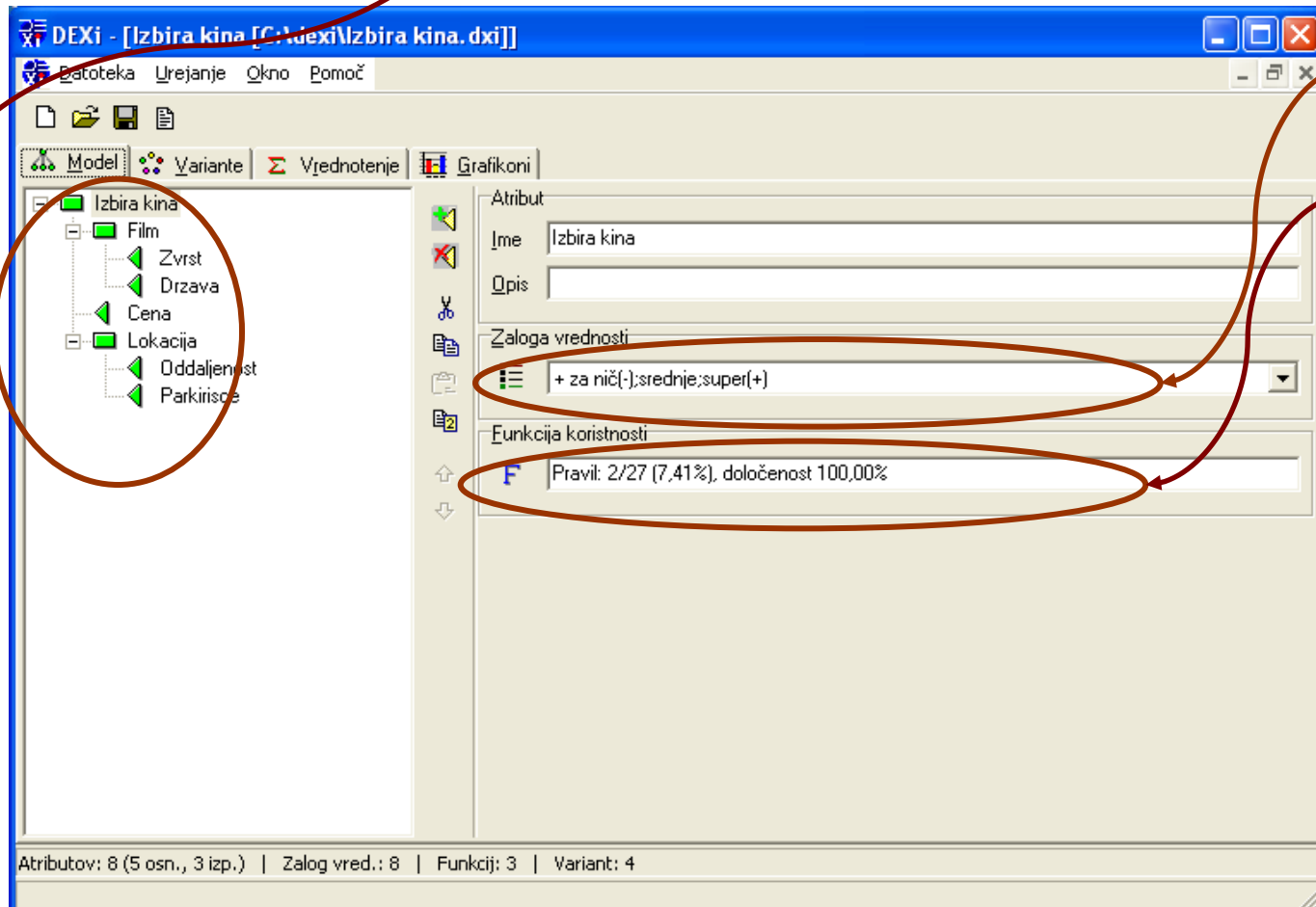
- olajšajo trud v odločitvenem procesu.
- pomagajo preseči človekove omejitve pri procesiranju informacij.
- omogočajo sistematizirano odločanje.

Faze odločitvenega procesa

1. Opredelitev problema
 2. Določitev kriterijev
 3. Strukturiranje kriterijev
 4. Merske lestvice
 5. Funkcije koristnosti
- Računalnik ne more odločati namesto nas.
 - Kljub temu pa nam je lahko IKT v procesu odločanja v **bistveno pomoč. Končna odločitev ostaja v rokah človeka, ki odloča!**
 - Primer orodja **DEXi**:
 - lupina odločitvenega sistema.
 - osnova DEX (Decision Expert).

Postopek (DEXi)

1. Opredelitev problema
2. Določitev kriterijev in strukturiranje kriterijev



3. Merske lestvice

4. Funkcije
koristnosti

5. Variante

6. Vrednotenje

7. Prikaz in
interpretacija
rezultatov

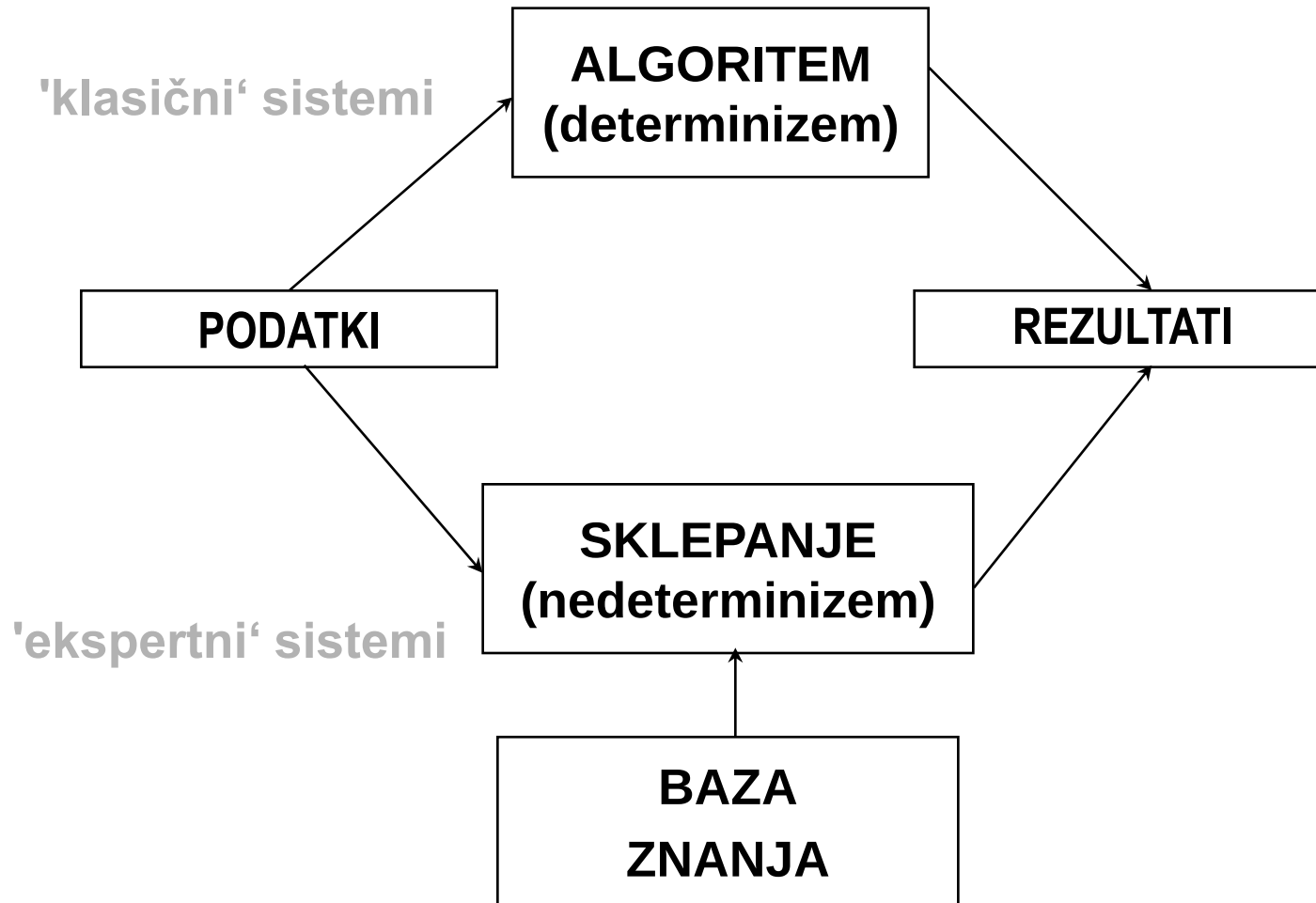
Ekspertni sistemi

- **Ekspertiza** – poglobljeno znanje na nekem ozkem področju.
- **Expert systems:** *a computer application that employs a set of rules based on human knowledge to solve problems that require human expertise!*
- **Ekspertni sistem:**
 - inteligentni računalniški sistemi, ki uporabljajo znanje in procedure sklepanja za reševanje problemov na načeloma ozkem strokovnem področju.
 - na določenem omejenem področju lahko emulirajo sposobnost odločanja in reševanja problemov človeških izvedencev.
 - aplikacije, ki se obnašajo kot strokovnjaki.

Ekspertni sistemi (nad.)

- Zagotavljajo neposredno uporabo ekspertiz.
- **ES NI NADOMESTEK** za eksperte, vendar:
 - omogoča širšo uporabo (dostopnost) njihovega znanja in ekspertiz.
 - omogoča boljše delo ne-ekspertom.
 - pomaga pri delu ekspertov.

Primerjava klasični $\leftrightarrow$ ekspertni sistem



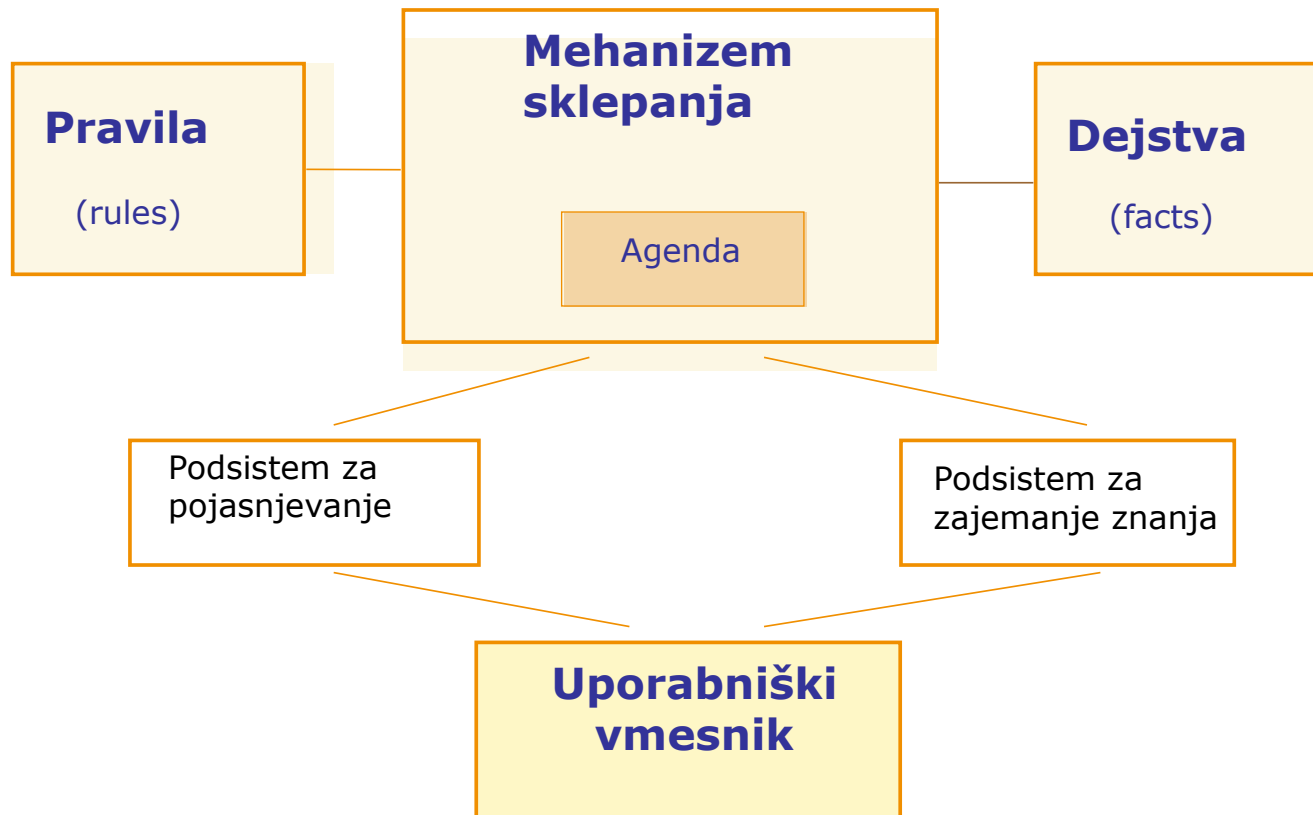
Zmožnosti ES

- sklepanje.
- presoja.
- delovanje pri nepopolnih oziroma nezanesljivih podatkih.
- pojasnjevanje delovanja in predlaganih rešitev (sledenje nazaj):
 - ES je za uporabnika 'prosojna' škatla, klasična aplikacija pa 'črna' škatla.

Uporaba ES

- **interpretacija**
 - pomoč pri dojemanju oziroma spoznavanju pomena ali vsebine česa.
- **napovedovanje**
 - vnaprejšnje določanje poteka oziroma pojavitev česa.
- **diagnosticiranje**
 - ugotavljanje in določanje vzroka bolezni ali okvar na osnovi simptomov, ki jih je mogoče opazovati.
- **načrtovanje**
 - snovanje in izdelovanje načrta za določen objekt; določanje ustreznih ukrepov, s katerimi bo mogoče doseči zadani cilj.
- **razhroščevanje**
 - predpisovanje in implementacija rešitev za določene pomanjkljivosti.
- **instrukcije**
 - pomoč pri obvladovanju določene učne snovi.

Struktura ES

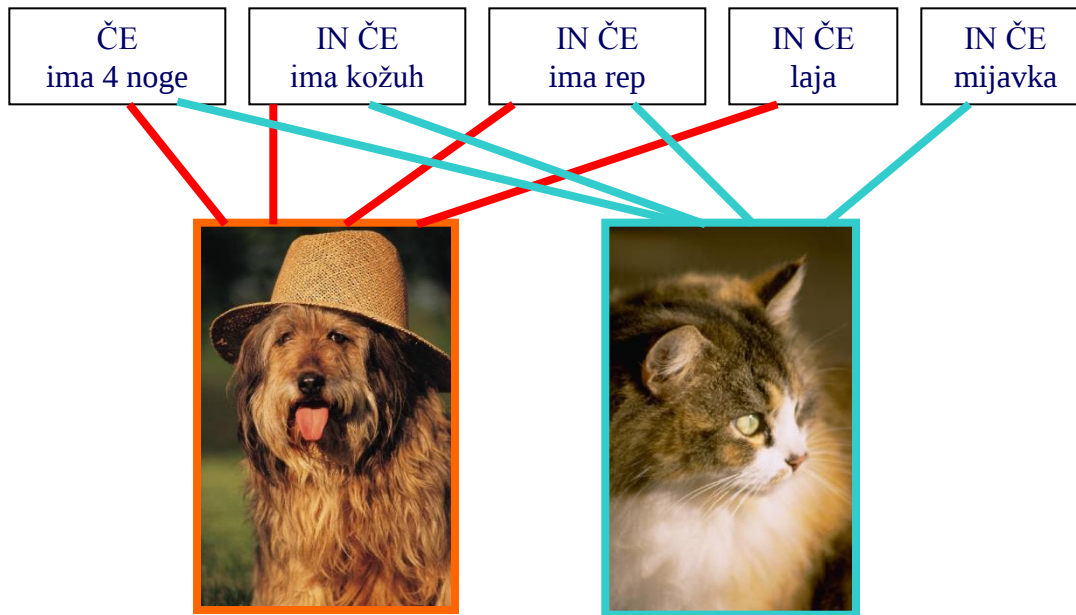


Struktura ES - baza znanja

- osrednji del ekspertnega sistema, vsebuje:
 - dejstva iz določene domene,
 - pravila, ki opisujejo relacije med dejstvi in
 - metode za reševanje problemov v tej domeni.
- **baza znanja <> baza podatkov:**
 - baza znanja hrani kompleksne opise,
 - baza znanja hrani tudi vzročno posledične informacije,
 - baza znanja hrani tudi nenatančne ter verjetnostne informacije.
- znanje v bazi znanja je mogoče predstaviti z različnimi predstavitvenimi formalizmi:
 - okvirji,
 - semantične mreže,
 - **produkcijska pravila** (pravila oblike če-potem),
 - ...

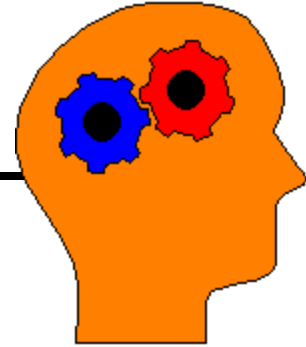
Produksijska pravila

Predpostavke



Sklep

Mehanizem sklepanja

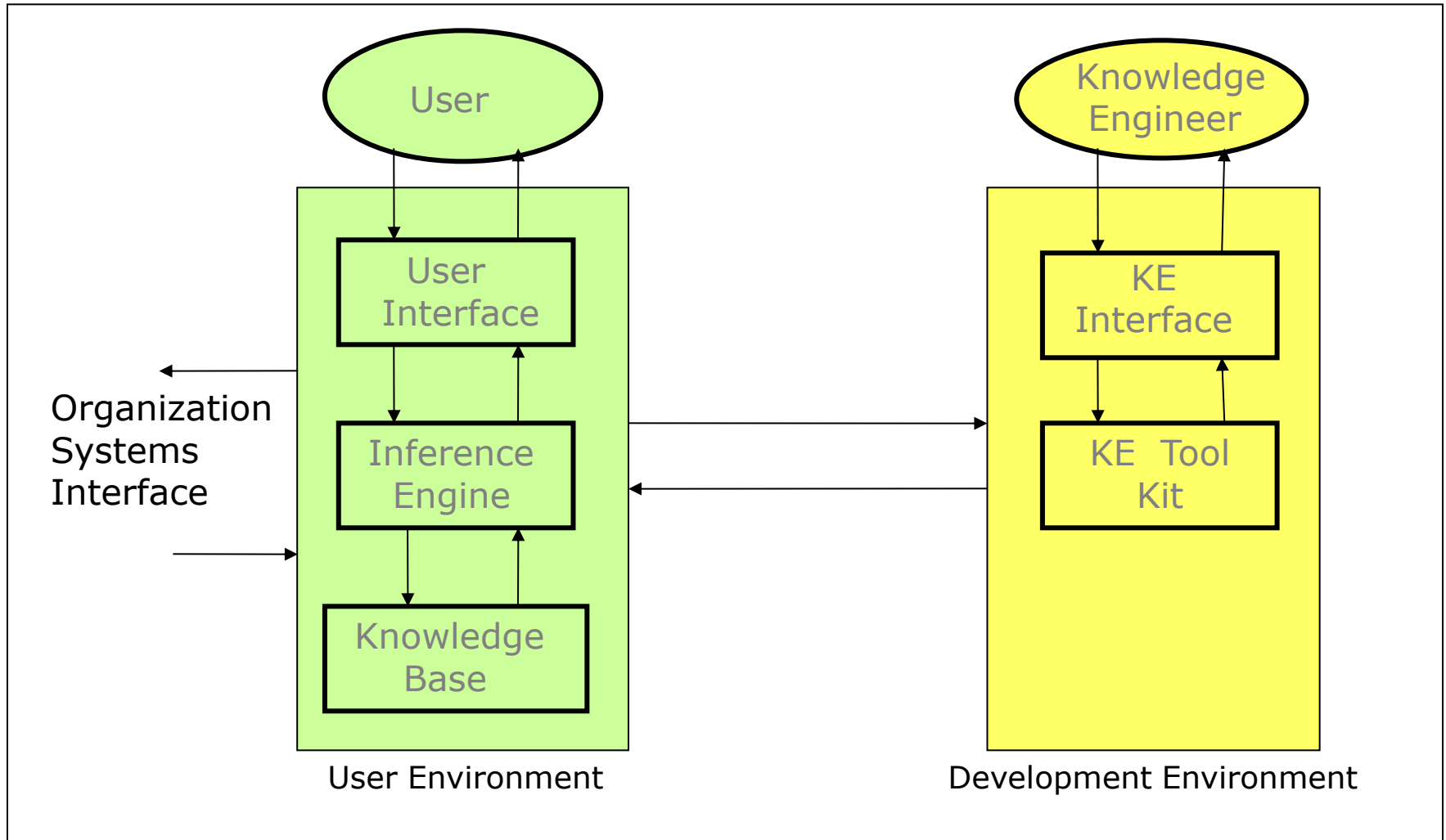


- podsystem, ki upravlja in nadzoruje delovanje celotnega ekspertnega sistema.
- zadolžen je za aktivno uporabo znanja iz baze znanja manipuliranje s podatki, ki vstopajo v sistem, in izpeljevanje ustreznih sklepov.
- 2 najpogostejše uporabljeni tehniki sklepanja:
 - **veriženje naprej (forward chaining)** - sistem sklepa naprej od množice znanih dejstev (induktivno sklepanje) in skuša priti do določenega sklepa oziroma cilja.
 - **veriženje nazaj (backward chaining)** - deluje sistem nazaj od množice možnih sklepov oziroma ciljev in skuša najti dokaze (dejstva v bazi znanja), s katerimi bi podprl in preveril njihovo pravilnost (deduktivno sklepanje).

Primera veriženja naprej in veriženja nazaj

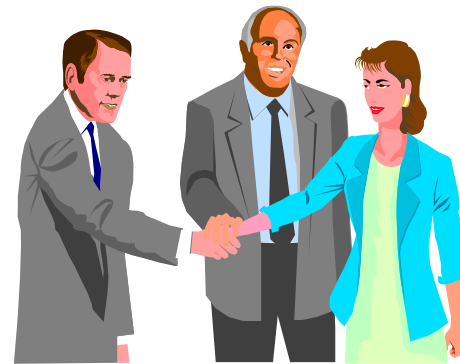
- Preprost primer veriženja naprej
 - Denimo, da imamo 3 pravila:
 - R1: Če 'je Medi priden' in 'je nedelja', potem 'Medi dobi sladoled'.
 - R2: Če 'je nedelja', potem 'gremo na Bled'.
 - R3: Če 'Medi dobi sladoled' in 'gremo na Bled', potem 'slaščičar Miki ima dobiček'.
 - Če najdemo dejstvi, da je Medi priden in da je nedelja, sklepamo, da Medi dobi sladoled (R1) in da se gre na Bled (R2). S pomočjo dobljenih sklepov sklepamo, da bo slaščičar Miki (ki je domenvno na Bledu) imel dobiček (R3).
- Preprost primer veriženja nazaj
 - Denimo, da imamo zgornja 3 pravila.
 - Če najdemo dejstvo, da slaščičar Miki ima dobiček (R3), predpostavljamo, da sta resnični trditvi, da je Medi dobil sladoled in da smo šli na Bled. Iz dejstva, da smo šli na Bled sklepamo, da je nedelja (R2). Iz dejstva, da je Medi dobil sladoled pa sklepamo, da je bil Medi priden (R1).

Okolja, potrebna za razvoj in uporabo ES



Človeški viri, vključeni v razvoj in uporabo ES

- Eksperti
- Inženirji znanj
- Uporabniki
- Sistemski analitiki
- Izdelovalci SW
- Dobavitelji SW



Prednosti ES

- Poveča produktivnost
- Zmanjša čas, potreben za odločanje
- Prilagodljivost
- Poveča dostopnost znanja
- Integrira mnenje večjega števila ekspertov (medicina – 'drugo mnenje')
- Obvladuje situacije z nepopolnimi / negotovimi podatki
- Omogoča učenje (pojasnjevanje)
- Izboljša kakovost odločitev
- Omogoča prenos znanja na oddaljene lokacije

Problemi (omejitve) ES

- Zahtevno pridobivanje ekspertiz strokovnjakov.
- ES dela le na ozkem strokovnem področju.
- Slovar ekspertov je omejen in visoko strokoven.
- Inženirji znanj: maloštevilčni in dragi.
- Pomanjkanje zaupanja s strani uporabnikov.
- ES lahko proizvaja tudi napačna priporočila.

INFORMACIJSKA DRUŽBA

Informacijska družba

- <https://www.youtube.com/watch?v=xWC0FvRRHhA>
- ID: večji del narodnega gospodarstva je odvisen / temelji na uporabi informacijsko komunikacijske tehnologije (IKT)
- Temelji na znanju
 - iniciativna družba – ustvarja osnovna znanja in omogoča uporabo v praksi
 - imitativna družba – nadgrajuje tuja znanja, razvija lastno znanje (prestižne tehnologije)
 - inovativna družba – temelji na podlagi znanj iz preteklosti in sedanjosti – usmerjena v prihodnost
- Sinonimi:
 - Digitalna družba
 - Družba znanja
 - Nova ekonomija

IS v današnji družbi (ali tudi v naši??)

- Razširitev dosega inf. družbe:
 - preteklost:
 - Tehnologija je posel (opravilo) nekaj posameznikov
 - Razmeroma šibko povezani (medsebojno odvisni) poslovni procesi
 - danes:
 - Tehnologija je vpeta v delo (slehernega) posameznika
 - Narašča medsebojna odvisnost med poslovnimi procesi
 - e-trgovina, e-poslovanje, m-poslovanje, ...
- Nove organizacijske oblike:
 - krčenje števila upravljalskih nivojev,
 - povečanje fleksibilnosti / prilagodljivosti na spremembe v okolju, (npr. spremembe trga)
 - mreženje podjetij
- Primer: Estonija: <https://www.youtube.com/watch?v=IIVT4SVnIIM>

Značilnosti informacijske družbe

- teledelo (delo na daljavo)
- učenje na daljavo
- e-bančništvo
- e-trgovina
- e-poslovanje
- e-državna uprava
- e-zabava
- e-????

Učinki uporabe IKT

- Očitni učinki:
 - transakcije in pretok denarja
 - sledenje blaga in ljudi
 - sledenje industrijskim procesom in proizvodom
 - avtomatizacija pisarniškega poslovanja
 - komunikacije in internet
- Inf. tehnologija in ekonomija
 - pospešitev komunikacij in dostopa do informacij
 - spreminjanje osnovne strukture svetovne ekonomije
- Globalizacija
 - selitev proizvodnje v kraje, kjer so stroški proizvodnje minimalni
 - geografska porazdeljenost - centraliziran nadzor
 - koordinacija dobaviteljev (v celotni dobavni verigi)
 - uničujoč učinek na ekonomije držav tretjega sveta

Učinki uporabe IKT (nad.)

- Hitrejši razvoj izdelkov
 - Hitrejši razvoj in distribucija novih izdelkov
 - Hitrejše zastaranje izdelkov
 - Poslovni modeli se morajo hitro prilagajati spremembam
 - 'outsourcing' storitev, specializacija
- Dematerializacija
 - skladišča (just-in-time)
 - denarja (elektronske transakcije, plastični denar)
 - ciljev (cilj: znanje)
 - produktov
 - ljudi (zamenja jih tehnologija)

Učinki uporabe IKT (nad.)

- standardizacija
 - V globalni ekonomiji prevladuje nekaj ogromnih multinacionalk
 - Stroški inovacij zahtevajo velika vlaganja → s tem se 'manjši' konkurenti izrinejo iz tekmovanja
 - Sodelovanje manjših firm v večjih omrežjih
 - Standardizacija blaga, storitev in življenjskega sloga
 - Osebna izbira široka
- avtomatizacija
 - Industrijska revolucija: zamenjava fizičnega dela človeka s stroji
 - Informacijska revolucija: zamenjava odločanja in presoje človeka z IS
 - Spremenjeni načini zaposlovanja

Vplivi IS



Pozitivni

- **Avtomatizacija poslovanja** → hitrejša izvedba poslovnih funkcij
- **Razbremenitev ljudi** – lahko se bolje posvečajo vsebini svojega dela (manj birokracije)
- **Razvoj znanosti** (medicine, kemije, fizike, ...)
- **Internet:** povezovanje celega sveta, večja dostopnost/pretočnost informacij, novi trgi

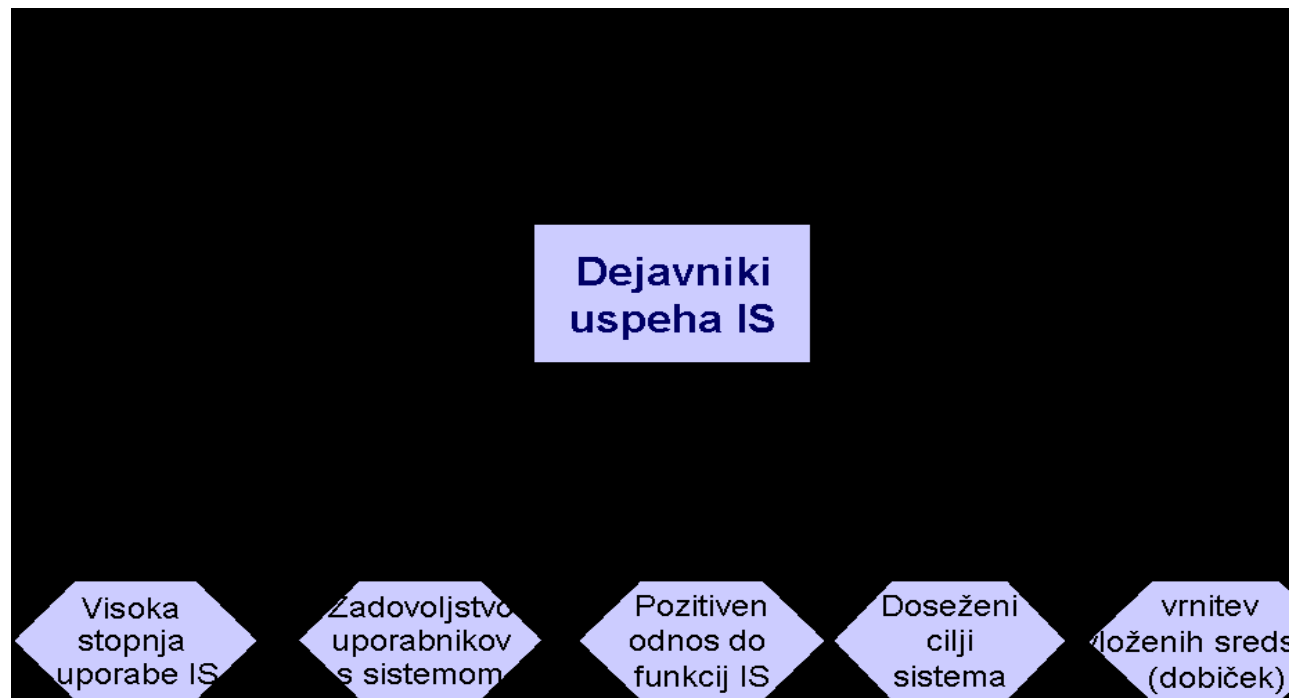


Negativni

- Manjše število delovnih mest za nezahtevna dela
- Manjša (vprašljiva) zasebnost
- Pretirana uporaba računalnikov => težave z zdravjem
- Internet: težave z avtorskimi pravicami, nelegalne transakcije, socialni konflikti, terorizem, ...

Nekaj izzivov inf. družbe

- Kako naj IT pomaga, da bo poslovanje učinkovitejše in bolj konkurenčno?
- Izzivi globalizacije
- Kako doseči dober IS?
- Upravljanje / ravnanje s spremembami
- Varnostni problemi / nadzor
- Etični problemi
- Uporaba interneta
- E-poslovanje
- M-poslovanje

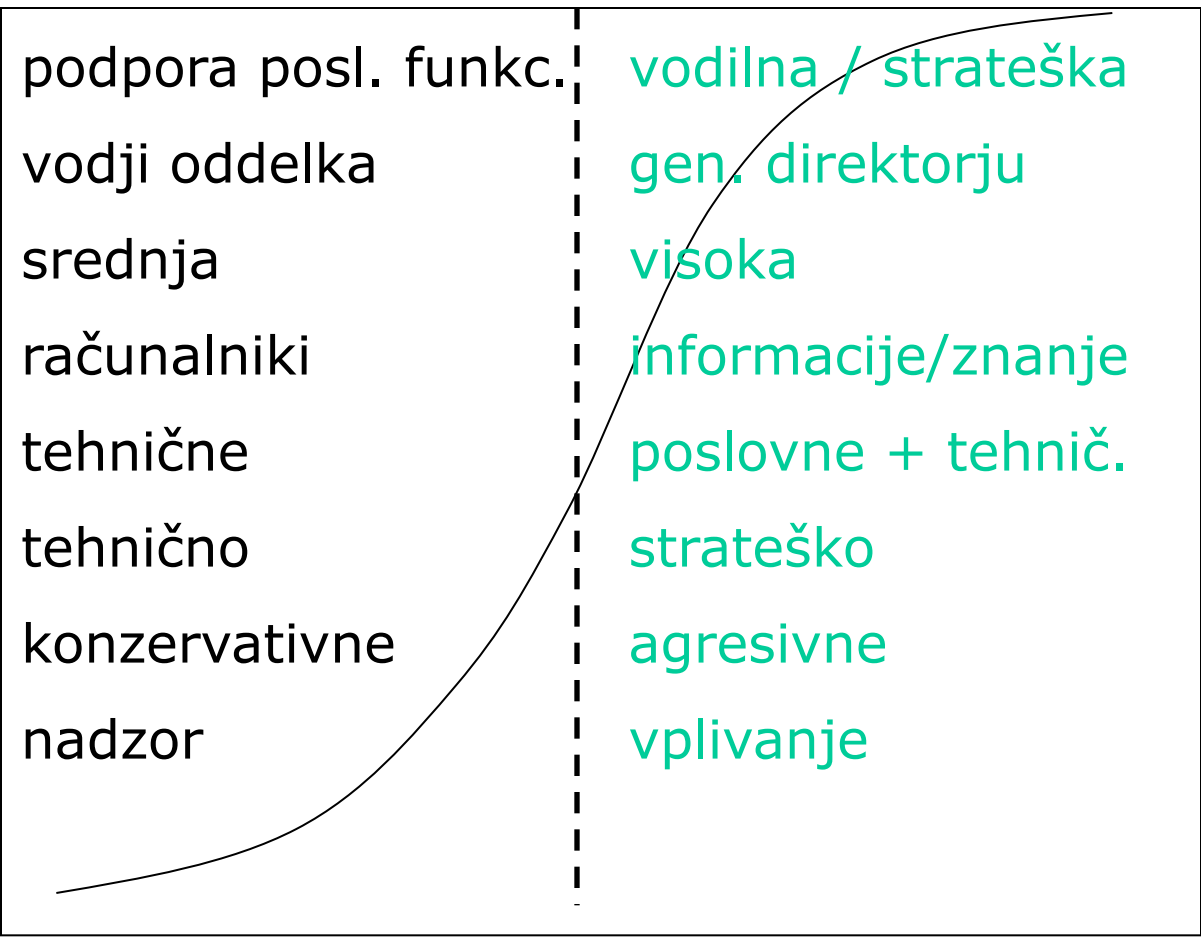


Informacijski poklici

- vodja informatike (CIO)
- sistemski administrator
- administrator PB
- sistemski analitik
- inženir znanja
- programer
- operater
- α tester, β tester
- vzdrževalec strojne / komunikacijske opreme
- + vsi ostali poklici

Primerjava: računalniška doba – informacijska doba

vloga informatike
podrejenost
upravljalna raven
glavni vir
veščine
načrtovanje
investicije
način vodenja



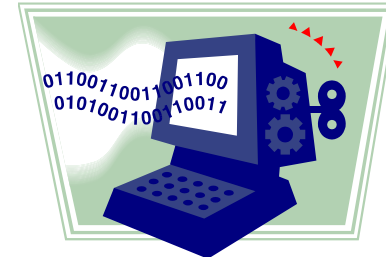
Informacijska družba - povzetek

- temelji na informacijah
- tekmovanje med konkurenti; cilj: biti hitrejši, učinkovitejši, uspešnejši, ...
- kako? – uvajamo spremembe, posodobitve, inovacije; pri tem pa se odločamo.
- odločamo se na osnovi znanja.
- znanje pridobimo z akumulacijo in interpretacijo informacij.
- informacijo pridobimo tako, da pripišemo pomen podatkom.

PREDSTAVITEV PODATKOV V RAČUNALNIKU

Predstavitev podatkov v računalniku

- dvojiška oblika
- znaki – kodne tabele
- števila – **pravila za pretvorbo**



- $(10) \rightarrow (n)$
 - celo število deliš z osnovo sestava, ostanke izpišeš v obrnjenem vrstnem redu; postopek končaš, ko je število 0.
 - realni del števila množiš z osnovo sestava, izpišeš cele dele rezultata; postopek končaš ko dobiš želeno št. decimalk oz., ko je število = 0.

Pretvorbe

- $(n) \rightarrow (10)$
 - vsako števko zmnožiš z ustrežno potenco osnove in rezultate sešteješ; pozor: prvo mesto pred decimalno vejico ima potenco 0;
 - primer: $216,6(8) = 142,75(10)$

2	1	6	,	6	Osmiško število
8^2	8^1	8^0		8^{-1}	potence

$$2 * 64 + 1 * 8 + 6 * 1 + 6 * 0,125 = 142,75 (10)$$

Pretvorbe

- $(n) \rightarrow (m)$
 - s pomočjo desetiškega sestava: $(n) \rightarrow (10) \rightarrow (m)$
 - neposredne pretvorbe med potenčnimi sestavi; primer:
 $123(4) = 011011(2)$

1	2	3	Štiriško število
01	10	11	Dvojiško število

- $(-10) \rightarrow (2)$ // negativna števila
- postopek:
- $(10) \rightarrow (2) \rightarrow$ eniški komplement $\rightarrow$ dvojiški komplement
(obrni bite) (prištej 1)

PROGRAMSKO INŽENIRSTVO

Zakaj potrebujemo programsko inženirstvo?

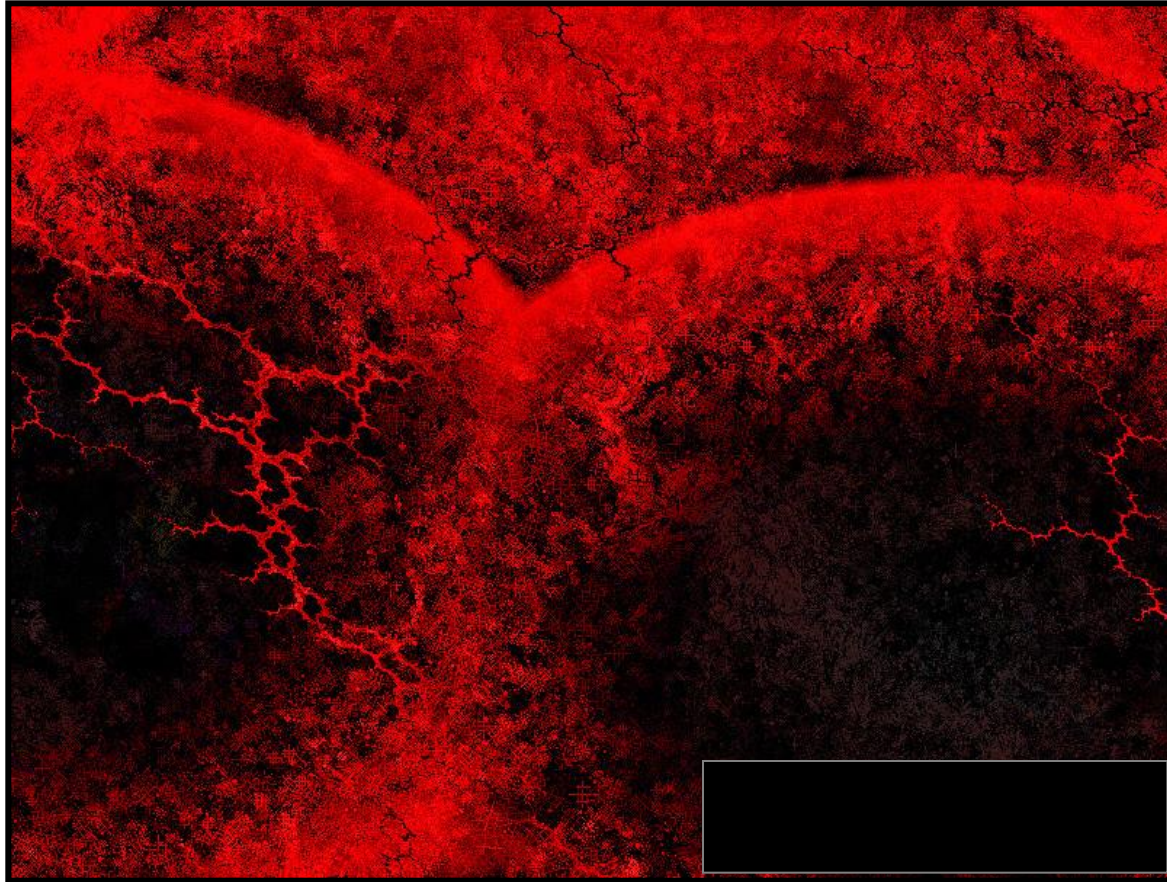
Preprosti pogled na razvoj programov:

specifikacija problema → **končni program**

ampak ...

- od kje pridejo *specifikacije*?
- kako vemo, da specifikacije ustrezajo *uporabnikovim potrebam*?
- kako bomo *strukturirali* naš program?
- kako vemo ali naš program *ustreza specifikacijam*?
- kako vemo ali bo naš program *vedno deloval pravilno*?
- kaj narediti, če uporabnik *zahteva spremembo*?
- kako *razdeliti naloge*, če imamo več kot enega razvijalca?
- ...

Kaj je na spodnji sliki?



... to je zadnja slika, ki jo je zajela sonda Mars Polar Lander

- " ... programska napaka je povzročila prezgodnjo izključitev motorjev, kar je povzročilo, da je sonda Mars Polar Lander treščila v rdeči planet."
- Uradno sporočilo: " Načrtovanje in testiranje programske opreme ni bilo ustrezno. Programska oprema bi morala biti načrtovana tako, da onemogoči prezgodnje ugašanje motorjev. V vesolju je ena napaka usodna. "
- Izguba je iznašala cca \$165M.

Programsko inženirstvo (DEF.)

Programsko inženirstvo (*ang. Software Engineering*) je uporaba sistematičnih, discipliniranih, kvantificiranih pristopov h razvoju, izvajanju in vzdrževanju programskih izdelkov, oziroma uporaba inženirstva na programski izdelek.

- po IEEE (*ang. Institute of Electrical and Electronics Engineers*),
1990

Programsko inženirstvo (DEF.)

Več-osebni razvoj **več-verzijskega** programskega izdelka.

» Parnas, 1978

Programsko inženirstvo je znanost in umetnost izgradnje kompleksnih programskih izdelkov:

- pravočasno,
- s predvidenimi stroški,
- s sprejemljivimi performansami,
- s pravilnimi funkcionalnostmi.

» Somerville, 1995

Programsko inženirstvo vs. programiranje

- Programiranje je **samo ena od aktivnosti** razvoja programskih izdelkov, kot so:
 - definiranje, kaj se bo razvijalo (*specifikacija zahtev*).
 - definiranje, kako se bo razvilo (*načrt*).
 - odpravljanje nepravilnosti in testiranje.
 - vzdrževanje.
- Programsko inženirstvo razen programiranja zahteva matematični, psihološki in managerski pristop.

Nekaj aktivnosti programskega inženiringa

- Sistemski inženiring
 - Izbira razvojnega procesa in izobraževanje
- Zahteve
 - zbiranje
 - analiza
 - oblikovanje
- Izbira tehnologije in izobraževanje
- Načrtovanje - oblikovanje
 - arhitektura
 - komponente
 - moduli
- Kodiranje - programiranje
 - test modulov
 - odprava napak
- Integracija
 - izgradnja
 - integracijski test
 - management konfiguracij
- Test celotnega sistema
 - performančni test in optimizacija
 - test sprejemljivosti
 - beta testiranje
- Namestitev
 - dobava
 - inštalacija
- Produkcija
 - vzdrževanje
 - nadgradnje
- Podporne aktivnosti
- Projektno vodenje
 - stikli z naročnikom
 - izboljševanje procesov
 - izobraževanje
 - dokumentacija

Življenjski cikel razvoja programske opreme

Življenjski cikel razvoja programske opreme (*ang. software development life cycle SDLC*) je zaporedje razpoznavnih korakov skozi katere gre programski izdelek:

- študija izvedljivosti (*ang. feasibility study*),
- analiza in specifikacija zahtev (*ang. requirements analysis and specification*),
- oblikovanje (*ang. design*),
- izgradnja (*ang. implementation*),
- testiranje (*ang. testing*),
- namestitev in vzdrževanje (*ang. installation and maintenance*).

Študija izvedljivosti

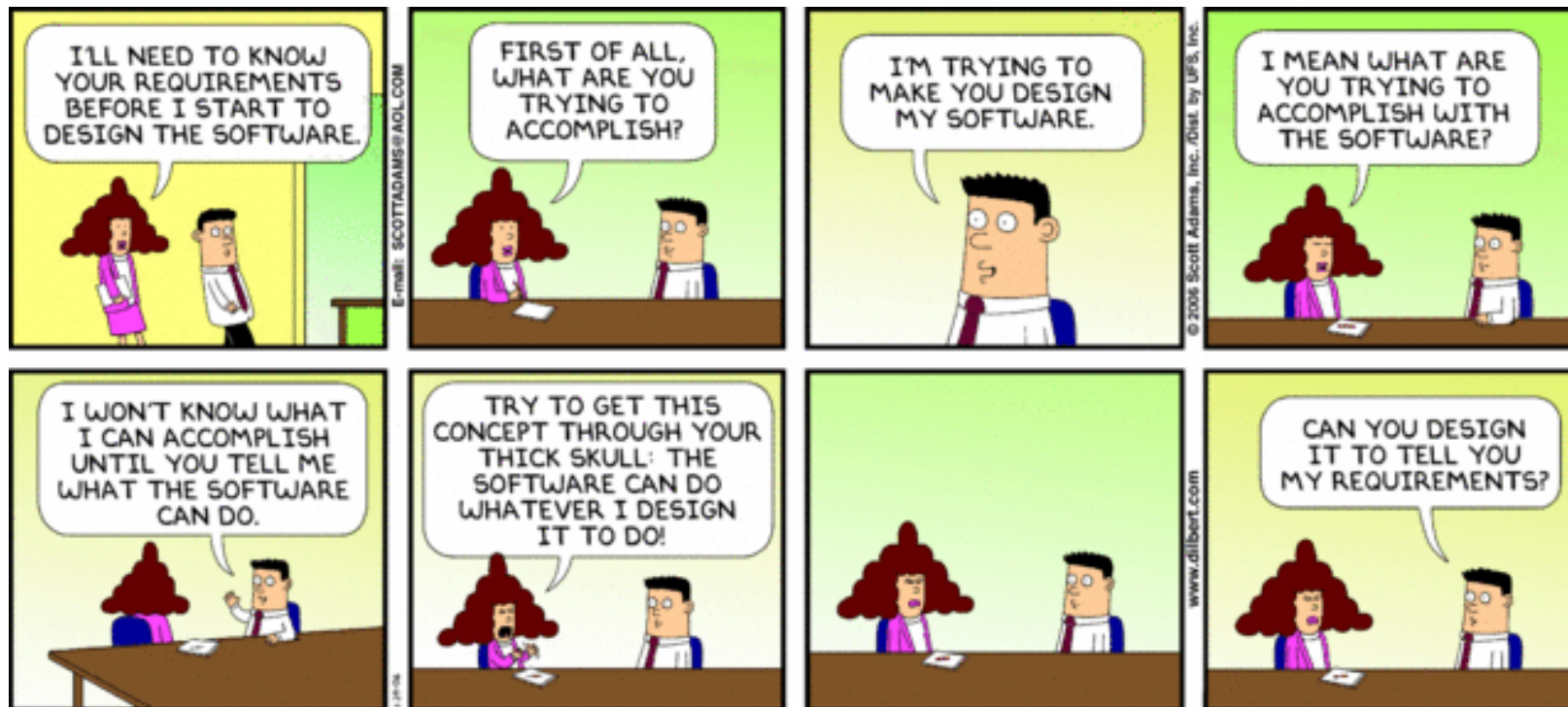
Študija izvedljivosti predhodi odločitvi o začetku projekta razvoja:

- kaj je obseg predlaganega projekta?
- ali je projekt tehnično izvedljiv?
- kakšne bodo koristi od projekta?
- kakšni so stroški, časovnica?

Študija izvedljivosti pelje k odločitvi: *gremo* ali *ne gremo* v izvedbo projekta.

Analiza in specifikacija zahtev

Zahteve definirajo funkcionalnosti sistema iz vidika naročnika: *kaj naročnik potrebuje in ne kaj je povedal, da potrebuje!*



Analiza in specifikacija zahtev

Osnovni cilj je **opis problema**, ki ga je potrebno rešiti in definiranje zahtev skladno s:

- **podanim problemom**,
- **okoljem**, katerem je sistem namenjen (npr. strojna oprema, programska oprema, uporabniki, zakonodaja, ...).

Osnovni dokument te faze je **specifikacija zahtev**. Zahteve se lahko določijo v samostojni študiji ali lahko nastanejo inkrementalno.

Načrtovanje

Načrt (*ang. design*) opisuje sistem iz vidika razvijalca:

- kako naj sistem dela tisto, kar bi moral?

Načrt sistema: vzpostavlja arhitekturo, ki ustreza zahtevam po strojni in programski opremi.

Načrt programa: predstavlja programske funkcionalnosti v takšni obliki, da se lahko pretvorijo v enega ali več izvršljivih programov.

Osnovni dokument je načrt sistema oz. **tehnična specifikacija (načrt)**.

Izgradnja

To je faza pri kateri se **programirajo** moduli sistema.

Osnovni cilj je narediti **berljivo, zanesljivo, za nadgradnje enostavno ter dobro dokumentirano programsko kodo.**

Poudarek ni na učinkovitosti, čeprav je tudi ta pomemben vidik programiranja.

Rezultat te faze je program, ki se **lahko izvaja** (in ne prevaja, kakor menijo eni razvijalci :)

Testiranje pomeni **izvajanje programa z namenom iskanja pomanjkljivosti** v programski opremi.

V novejših modelih razvoja se vse več prepleta s fazo izgradnje.

Načrtovanje testnih aktivnosti začne s fazo specifikacije zahtev.

Pravilo: **vsako pomanjkljivost je bolj poceni odpraviti, če je najdemo čim bolj zgodaj v razvojnem ciklu.**

Namestitev in vzdrževanje

Po dobavi/namestitvi v produkcijsko okolje naročnika sledi faza vzdrževanja.

Osnovni namen vzdrževanja je:

- odprava vseh pomanjkljivosti skritih napak (odkritih po dobavi).
- prilagajanje sistema spremenjenim zahtevam.
- izboljševanje sistema.

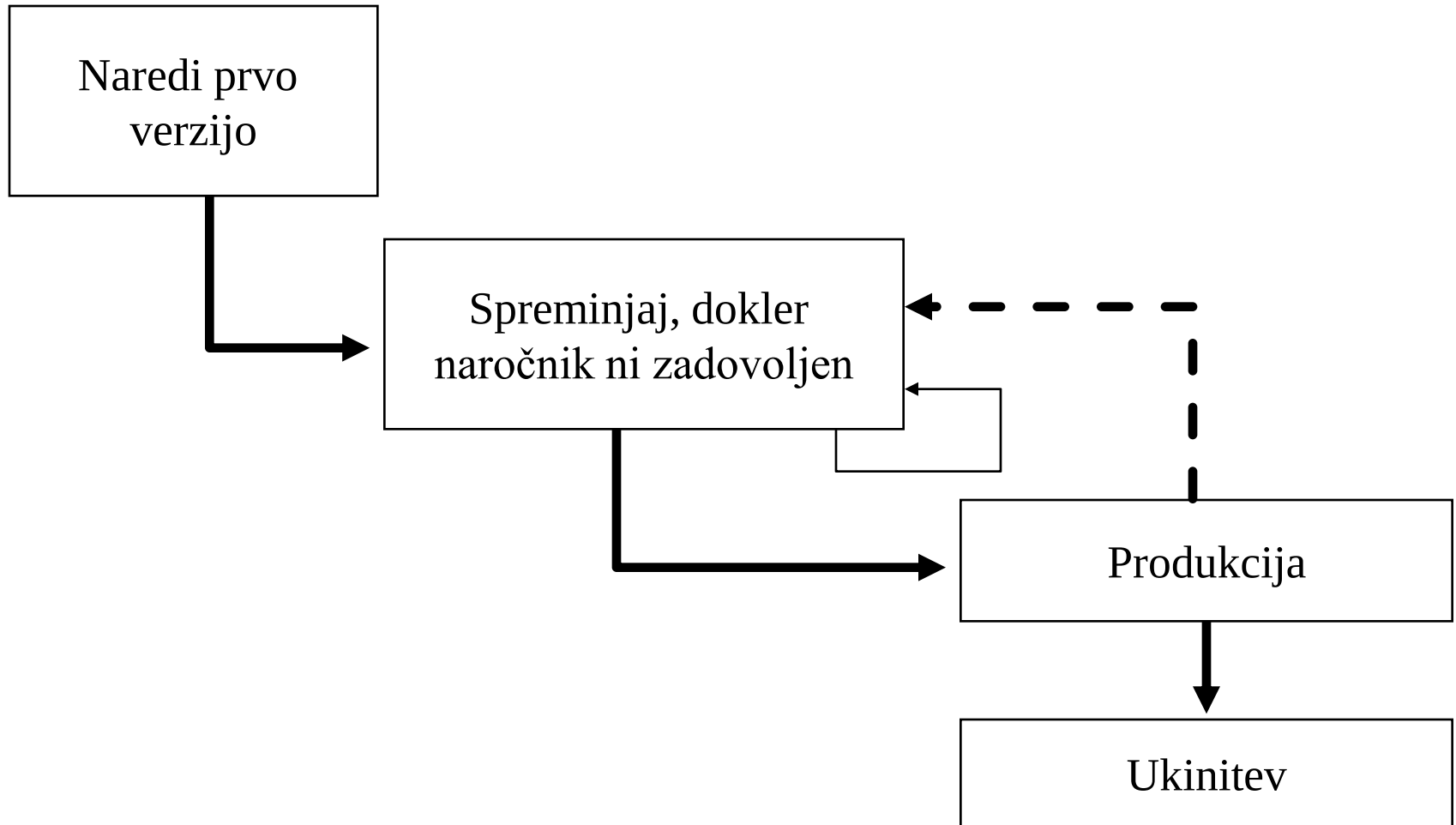
Cilj te faze je **zagotoviti nemoteno izvajanje** programske opreme pri uporabniku.

Dodatne nujne aktivnosti

Dodatne aktivnosti, ki so nujne za funkcioniranje življenjskega cikla:

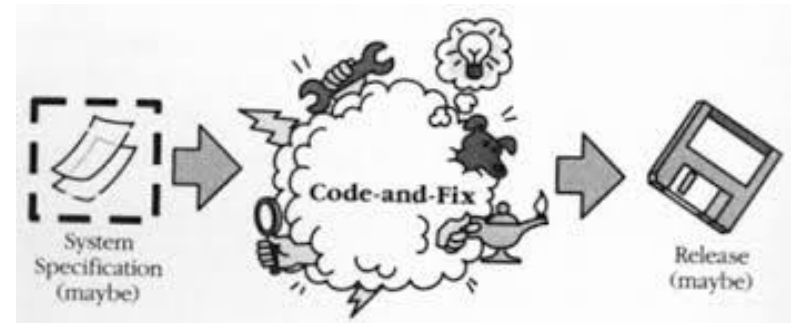
- **upravljanje** z namenom zaključevanja projektov v dogovorjenih obsegih (rok, stroški, kakovost).
- **dokumentiranje** z namenom upravljanja s kompleksnostjo programske opreme (tehnična dokumentacija) in timskega dela (projektna dokumentacija).

Code&Fix

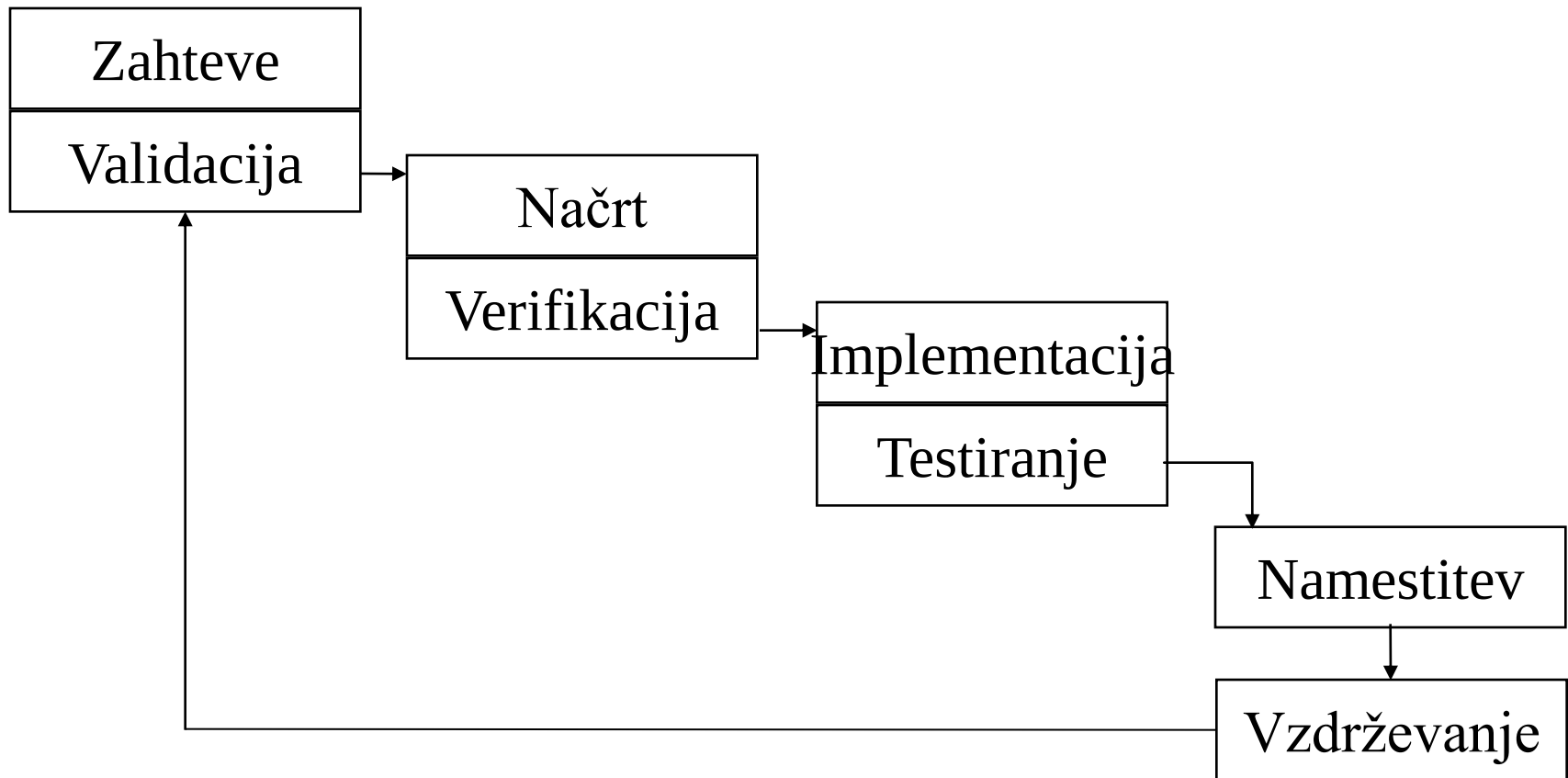


Code&Fix – prednost in pomanjkljivosti

- Zares slaba
- Zares pogosta
- Prednosti:
 - ni dodatnega dela.
 - ni potrebno posebno znanje.
- Pomanjkljivosti:
 - ni mogoče oceniti proces.
 - težko je koordinirati več razvijalcev.
- Uporabno za razvoj enouporabniških programov: začni s prazno kodo in debugiraj, dokler ne začne delati prav.



Model slapa

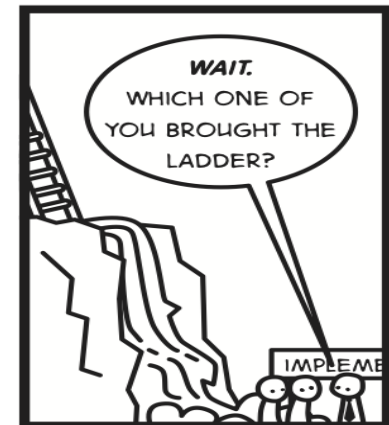
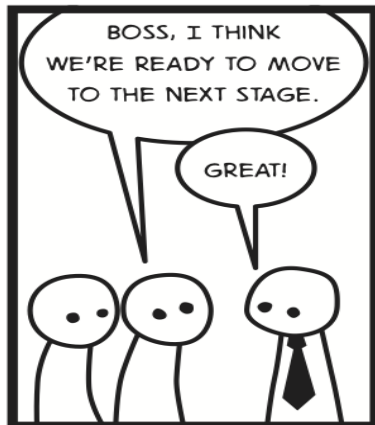


Model slapa – prednosti in pomanjkljivosti

- Prvi opisal Royce I. 1970 (DoD), danes razširjena uporaba.
- Vsaka faza v zaporedju življenjskega cikla mora biti povsem zaključena, pred začetkom naslednje.
- Ob zaključku vsake faze kontrole glede na zastavljene cilje (roki, kakovost, obseg).
- Prednosti:
 - merljivost napredka.
 - pridobljene izkušnje iz predhodnih projektov se lahko uporabijo za oceno napora podobnih korakov pri bodočih projektih.
 - izgradnja programskih modulov, ki se lahko ponovno uporabijo pri drugih projektih.
 - enostaven za naročnika.

Model Slapa – prednosti in pomanjkljivosti

- Originalni model slapa (kako mnogi interpretirajo) ne omogoča iteracijo:
 - neprilagodljiv.
 - zahteve se v resnici spreminjajo s časom.
 - ne ustreza dejanskemu razvoju PO.
 - vsiljuje strukturo projektnega vodenja – ni jasno kako izdelki iz ene faze prehajajo v drugo.
 - razvoj ne vidi kot reševanje problemov.
 - vzdrževanje ni dobro podprto.



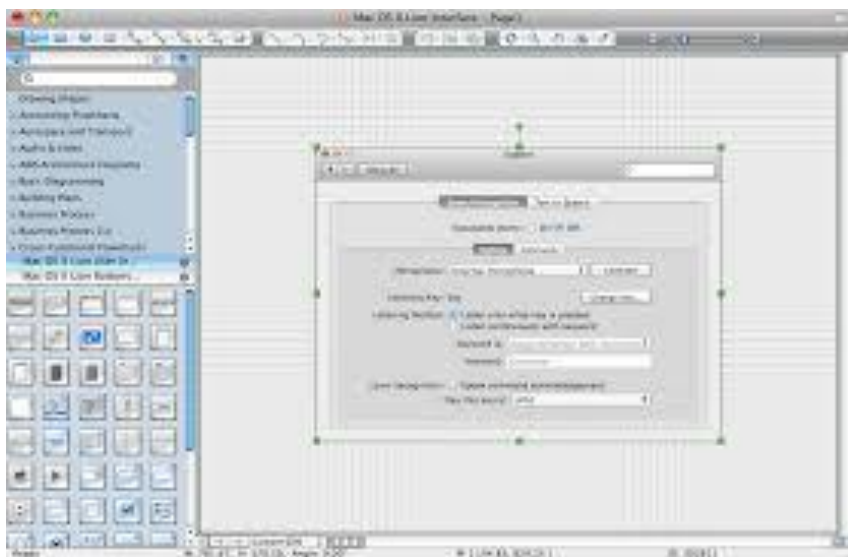
Prototipni model

Začetni koncept

Načrt in izgradnja
začetnega koncepta
Prvi prototip

Natančnejši prototip
do sprejema

Zaključni
prototip



Prototipni model

- Prototip omogoča uporabnikom, da vizualizirajo aplikacijo, ki še ni narejena.
- Uporaba pri specificiranju zahtev:
 - koristna pri hitro se spreminjajočih zahtevah.
 - ko se naročnik ne zavezuje pri specifikacijah.
- Enkrat, ko so zahteve znane, se uporabi model slapa.
- Ko se začne načrtovanje, se prototip zavrže:
 - prototipi ne bi smeli biti osnova za izgradnjo (avtomatično generiranje kode).
 - naročnik mora biti poučen o naravi prototipa (prototip ni % končnega izdelka).

Prototipni model – prednosti in pomanjkljivosti

- Prednosti:

- sposobnost prepoznavanja uporabnikovih informacijskih potreb na podlagi testnega sistema,.
- omogoča realni prikaz pomembnih vidikov sistema že v zgodnji fazi razvoja.
- izboljšuje vključenost končnih uporabnikov v projekt in komunikacijo med vsemi vpletenimi stranmi.
- omogoča identifikacijo nejasnih in pogrešanih funkcionalnosti.

- Slabosti:

- manj strog nadzor nad posameznimi fazami projekta.
- nevarnost izdelave nepopolne ali neustrezne problemske analize pred izdelavo prve različice prototipa.
- pogosto prihaja do večjih sprememb zahtev.
- lahko vodi do slabo načrtovanih sistemov.

Agilne metodologije

- Skupina udeležencev - agilna koalicija, definira štiri ključne vrednote agilnih pristopov:
 - Posamezniki in interakcije pred procesi in orodji.
 - Delujoča programska oprema pred obsežno dokumentacijo.
 - Sodelovanje z uporabniki pred pogodbenimi pogajanjmi.
 - Odgovarjanje na spremembe pred sledenjem planu.



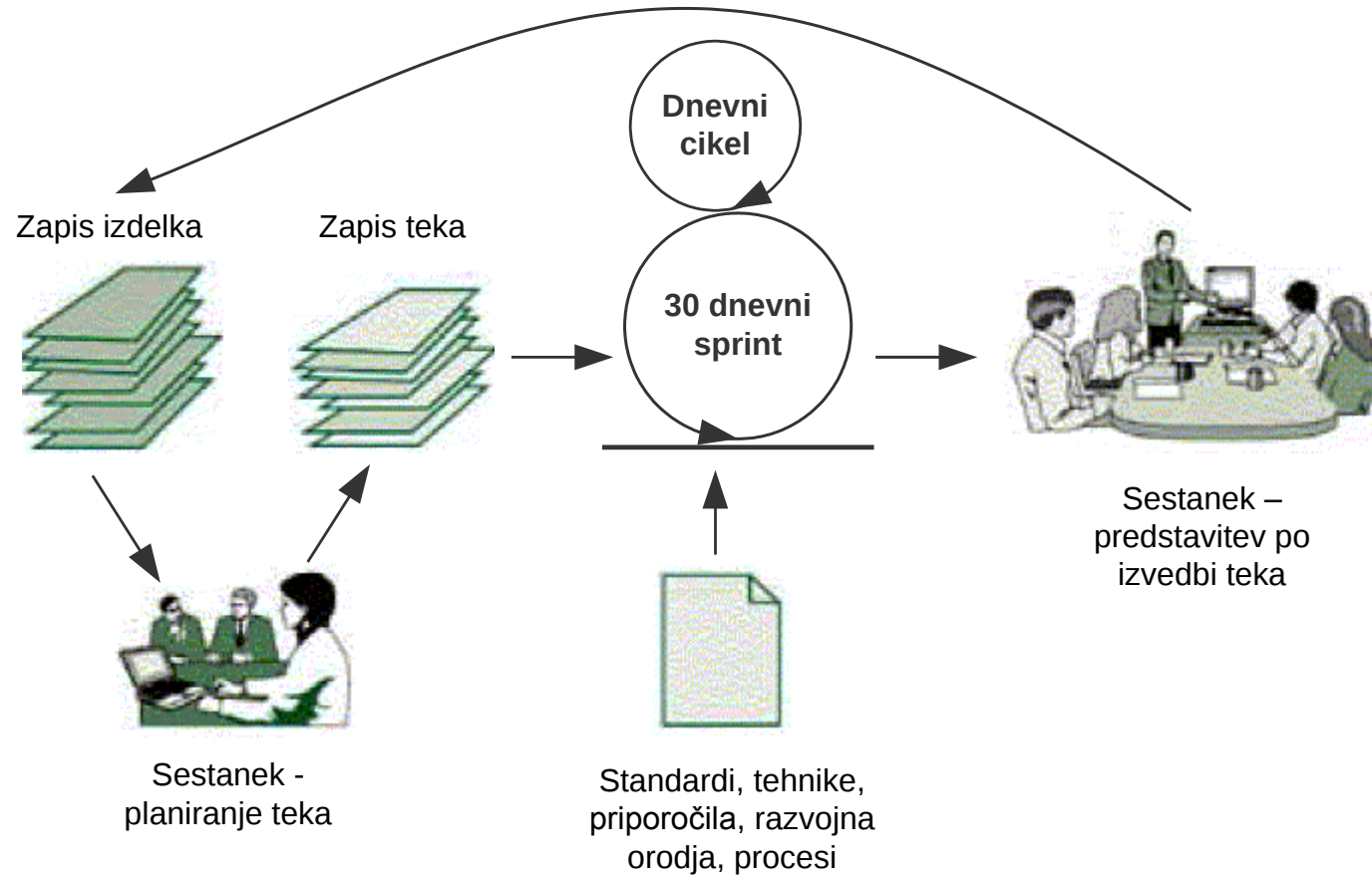
Agilni model

- Vsaka ponovitev je mini-projekt:
 - vsaka ponovitev ni prototip ampak delujoča verzija PO.
 - razumevanje tveganj vs. vrednosti tekom planiranja.
 - predati nekaj delujočih funkcionalnosti naročniku čimprej.
- Timeboxing:
 - določanje datuma za dobavo ponovitve.
 - datum se ne sme spreminjati.
 - dovoljena je samo sprememba zahtev (obsega).
 - kratko trajanje ponovitev (tedni in ne meseci).

Scrum

- Metodologija Scrum je zasnovana na prepričanju, da ima večina metodologij za razvoj informacijskih sistemov napačno filozofsko osnovo.
- Razvoj programskih rešitev ni definiran temveč empiričen proces, ki ga ni možno dosledno ponavljati in zato potrebuje sproten nadzor ter prilagajanje.
- Scrum predpostavlja, da živimo v zapletenem svetu in da je zatorej nemogoče napovedati ali natančno planirati končno obliko izdelka, časovni okvir, stroške in kvaliteto.
- Nadzorovanje kaotičnih (empiričnih) procesov in nadzorovanje predvidljivih (definiranih procesov) zahteva različne tehnike.
- Klasična praksa upravljanja projektov - projekti so predvidljivi in odkloni od plana so v prvi vrsti posledica človeških napak, neznanja ali slabe motivacije.

Scrum



Scrum

- **Pred sprint:** planiranje pred sprintom– izdelajo se 3 zapisi:
 - zapis izdelkov (opisujejo poslovne in tehnološke funkcionalnosti izdelka),
 - zapis izdaje (podmnožica zapisa izdelka – funkcionalnosti izdaje),
 - zapis sprinta (podmnožica zapisa izdaje) - identificira in definira delo, ki ga mora razvojna skupina opraviti v okviru posameznega sprinta,
 - določitev cilja oziroma poslovnega namena sprinta.
- **30 dnevni sprint** - pravila:
 - člani skupine si razdelijo naloge,
 - vsi delajo po svojih najboljših močeh za dosego cilja,
 - vsi sodelujejo na dnevni sestankih,
 - prioritete, saj razen v izjemnih primerih, med posameznim sprintom ne smejo spreminjati,
 - v okviru sprinta ne obstaja noben podroben in dovršen plan,
 - dnevni sestanki: povečujejo opaznost dela posameznega člana skupine, poenostavljajo izmenjavo znanja, preprečujejo podvajanje aktivnosti in zagotavljajo integracijo njihovega dela.
- **Po sprint** – sestanek:
 - ocena napredka, uporabnikom se predstavijo nove ali dopolnjene funkcionalnosti, revizija projekta s tehničnega vidika.

Dejanski procesi v podjetjih

Velika razvojna podjetja imajo **lastne** procese, ki so **prilagojeni njihovim posebnim potrebam**. Npr.:

- **Amazon.com** (e-commerce) razbija razvoj na faze s 4 tedenskim trajanjem.
- **Lockheed Martin** (DoD) sledi procesom, ki jih ZDA določa za njihove pogodbenike.
- **SAP** (poslovna PO) dajo poudarek na funkcionalnosti, ki jih določajo njihovi naročniki.
- **Microsoft** (OS) dajo poudarek na testiranju na različni strojni opremi in na kompatibilnosti za nazaj.

DODATNE PROSOJNICE

Povzetek (1)

- Podatki = elementarni opisi dejstev
- Informacije = Podatki + Pomen
- Znanje = Informacije + Kontekst
(kontekst predstavlja kombinacijo razumevanja, izkušenj, pridobljenih na osnovi učenja (teoretičnega / praktičnega) in uporabe informacij)

Povzetek (2)

- *IS zagotavlja ustrezno informiranje uporabnikov. (na vseh nivojih upravljanja organizacije!)*
- *IS omogoča vnos, hranjenje, obdelavo in posredovanje podatkov / informacij.*
- *Za delovanje IS sta potrebna tehnologija in ljudje.*
- *IS mora ustrezati opredeljeni informacijski arhitekturi, deluje pa s pomočjo informacijske infrastrukture.*

Projektni teden

Naloga: izdelava majhne aplikacije (*velja za koderje!*)

- uporaba znanja iz PPB in RPA:
 - PB
 - aplikacija
 - dokumentacija
 - predstavitev
- skupinsko delo

Cilj IS

- Izboljšati učinkovitost ljudi v organizaciji s pomočjo informacijske tehnologije – **prenova poslovnih procesov** (ang. *business process reengineering*).
- Omogoči pravočasen dostop do pravih informacij.
- Izboljšati kakovost odločanja.

Primeri:

Prodaja kino vstopnic. Karte se prodajajo v Ljubljani v kinematografih in v Koloseju. Mogoč je nakup preko mobitela. Rač. sistemi so povezani in omogočajo nakup za posamezno karto samo enkrat. Če ne bi bilo povezave,...

Šola ...

Krožek/raziskovalne naloge

- Izdelava didaktičnega pripomočka (Android, Java, SQLite, PHP, MySQL)
- Polarni tiskalnik (RPi, Python, C/C++, PHP)
- Vizualizacija senzorskih podatkov (RPi, PHP, MySQL, C/C++/C#)
- Lokalizacija storitev (RPi, PHP, C/C++/C#, JavaScript)
- Spletni strežnik na sončne celice (RPi, PHP, MySQL, Wordpress)
- Sistem za priporočanje (PHP, MySQL, SQLite)
- Merilec hitrosti (RPi/Arduino, MySQL, PHP/C++)

Potek - vaje

Vaje (2 uri/teden):

- Sistemi za podporo odločanju – DEXi
- Microsoft Access
- SQL (za RDBMS: Microsoft Access/SQLite/MySQL/Oracle XE)
- ER modeliranje: Case Studio, Toad Data Modeler + RDBMS
- seminarska naloga

Ocenjevanje:

- za vsak sklop 1 izdelek/test (+zagovor)
- bolj podrobna navodila na vajah (za vsak izdelek posebej)