

Računalništvo04

IS, UML, PB, SQL, ER

predavatelj: Aleksandar Lazarević

Agenda

- **Potek predmeta**
- **Osnovni pojmi v informatiki**
 - podatek
 - informacija
 - količina informacije
 - atributi kakovosti informacije
 - informacijska hierarhija
- **Informacijski sistemi**
 - sistem
 - informacijski sistemi (IS)
 - delitve IS

Potek predmeta

Teorija (2 uri/teden):

- osnovni pojmi v informatiki in informacijski sistemi
 - UML
- sistemi za upravljanje s podatkovnimi bazami (SUPB) in SQL
 - SQL DDL
 - SQL DML
- ER-modeli

Ocenjevanje:

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

Potek predmeta

LAV (2 uri/teden) / RAČ vaje (2 uri/na dva tedna):

- sistemi za podporo odločanju (DEXi)
- UML (Visual Paradigm)
- SQL (SQLite/MySQL oz. MariaDB)
- ER modeliranje (Visual Paradigm)
- *seminarska naloga*

Ocenjevanje:

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

Učni pripomočki

- učbenika:
 - Informatika, učbenik (več avtorjev)
 - Načrtovanje in postavitve podatkovnih baz, učbenik (Andreja Šet)
- zapiski
- spletni viri (npr. w3schools)
- računalniške revije
- knjige ...

Osnovni pojmi v informatiki

Podatek, informacija, količina informacije ...

Osnovni pojmi v informatiki

Podatek, informacija

2024/2025

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

Osnovni pojmi v informatiki

Količina informacije

2024/2025

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 ...

Osnovni pojmi v informatiki

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)

- **kiB** = 2^{10} byte-ov = 1024 byte-ov (malo manj kot pol strani besedila)
- **MiB** = 2^{20} byte-ov = 1024 Kbyte-ov $\sim 10^6$ byte-ov (cca 500 strani debela knjiga)
- **GiB** = 2^{30} byte-ov = 1024 Mb yte-ov $\sim 10^9$ byte-ov (200m dolga knjižna polica)
- **TiB** = 2^{40} byte-ov = 1024 Gbyte-ov $\sim 10^{12}$ byte-ov (???)

Osnovni pojmi v informatiki

Več, kot bit

SI desetiške predpone			IEC binarne predpone	
Ime (oznaka)	SI	Binarno (narobe)	Ime (oznaka)	Vrednost
kilobit (kbit)	10^3	2^{10}	kibibit (Kibit)	2^{10}
megabit (Mbit)	10^6	2^{20}	mebibit (Mibit)	2^{20}
gigabit (Gbit)	10^9	2^{30}	gibibit (Gibit)	2^{30}
terabit (Tbit)	10^{12}	2^{40}	tebibit (Tibit)	2^{40}
petabit (Pbit)	10^{15}	2^{50}	pebibit (Pibit)	2^{50}
exabit (Ebit)	10^{18}	2^{60}	exbibit (Eibit)	2^{60}
zettabit (Zbit)	10^{21}	2^{70}	zebibit (Zibit)	2^{70}
yottabit (Ybit)	10^{24}	2^{80}	yobibit (Yibit)	2^{80}

Osnovni pojmi v informatiki

Več, kot bit

- | | | | |
|----------------------------|----------|--|--------|
| □ Binarna odločitev | 1 bit | □ Potiskan papir narejen iz 50,000 dreves | 1 TB |
| □ Znak besedila | 1 byte | □ Univerzitetna knjižnica | 2 TB |
| □ Tipična beseda | 10 bytov | □ Tiskani izvodi kongresne knjižnice (ZDA) | 10 TB |
| □ Tipkana stran | 2-4 kB | □ Podatki centra za klimatske spremembe | 400 TB |
| □ Slika nizke ločljivosti | 100 kB | □ Vse akademske knjižnice v ZDA | 2 PB |
| □ Kratek roman | 1 MB | □ Kapaciteta vseh trdih diskov leta 1995 | 20 PB |
| □ Vsebina diskete 1.44" | 1.44 MB | □ Ves tiskan material na svetu | 200 PB |
| □ Slika visoke ločljivosti | 5 MB | □ Količina podatkov tvorjenih leta 1999 | 2 EB |
| □ Ena minuta Hi-Fi zvoka | 10 MB | □ Vse kadarkoli izgovorjene besede | 5 EB |
| □ En meter knjig | 100 MB | | |
| □ Vsebina CD-ROM-a | 650 MB | | |
| □ Tovornjaček knjig | 1 GB | | |
| □ Vsebina DVD | 4,5 GB | | |
| □ Zbirka del Beethovna | 20 GB | | |
| □ Eno nadstropje knjižnice | 100 GB | | |

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.

$$Q = 2^n$$

Kodiranje podatkov

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

Stikalo je lahko v dveh stanjih:

$$2 = 2^n$$

$$n = 1$$

b_0	stanje
0	vklopljeno
1	izklopljeno

Točka na monitorju v štirih barvah:

$$4 = 2^n$$

$$n = 2$$

b_1	b_0	stanje
0	0	bela
0	1	svetlo siva
1	0	temno siva
1	1	črna

Primer

Osnovna lastnost interneta stvari (IoT) je neprestano zajemanje podatkov na merilnih mestih in njihovo posredovanje v obdelavo storitvam v oblaku.

Na sliki vidimo merilni valj, na katerem so označene vrednosti od 10 do 100 ml. Nanj je Peter Zmeda namestil kontakte na vsakih 10 ml (od vključno 10 ml do vključno 100 ml). Ko nivo tekočine doseže določen kontakt, računalnik to zazna in podatek o odčitku pošlje v oblak.

Najmanj koliko bitov potrebujemo za zapis poslanega podatka?



Primer

V računalniški učilnici je 22 računalnikov. Enemu izmed računalnikov je nehala delovati mrežna kartica.

Najmanj koliko bitov je dolgo sporočilo sistemskemu administratorju, ki opisuje, kateremu računalniku ne deluje mrežna kartica?

Miha ima na trgu svojo stojnico, na kateri prodaja sadje in zelenjavo. Včasih je na trgu, včasih pa ga ni, pri čemer je verjetnost, da je na trgu, enaka verjetnosti, da ga ni. Včasih pa namesto njega pride žena Jožica.

Izračunajte, koliko bitov informacije dobimo, ko izvemo, da je Miha na trgu.

Ko izvemo, da stanuje Tomaž v pritličju neke zgradbe, dobimo 3 bite informacije. *Napišite, največ koliko nadstropij ima zgradba, v kateri stanuje Tomaž, in svoj odgovor utemeljite.*

Primer

Če želimo kodirati velike črke slovenske abecede z znakoma 0 in 1, porabimo za vsako črko po pet znakov. V Butalah pa so sprejeli novo zakonodajo, ki dovoljuje kodiranje črk z znaki *, + in o namesto z 0 in 1. Najmanj koliko znakov potrebujejo za kodiranje vsake velike črke slovenske abecede?

Janko in Metka sta si omislila svojo pisavo. Dogovorila sta se, da uporabljata le tri znake: zvezdico (*), krogec (o) in plus (+). Z njimi sta enolično zakodirala velike črke slovenske abecede in presledek. Za vsako črko sta uporabila enako število znakov. S koliko znaki je Janko zapisal svoje ime »JANKO«?

Dimenzija informacije

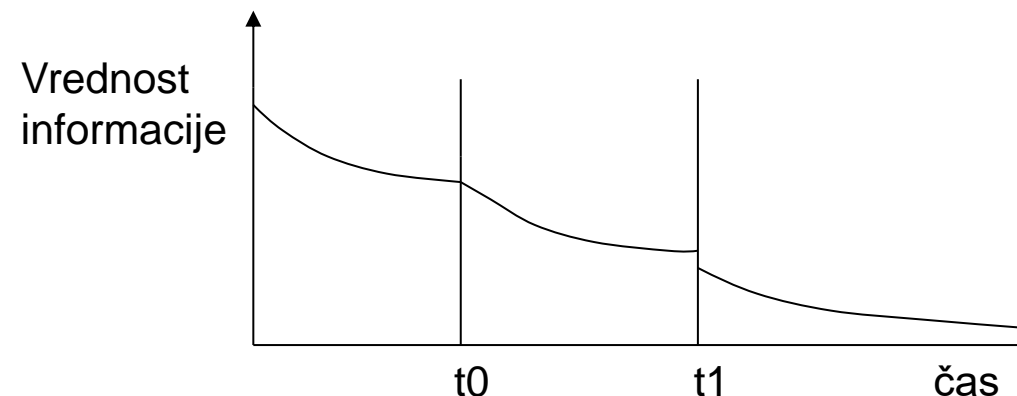
Količina informacije

Kakovost

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

○ Vrednost

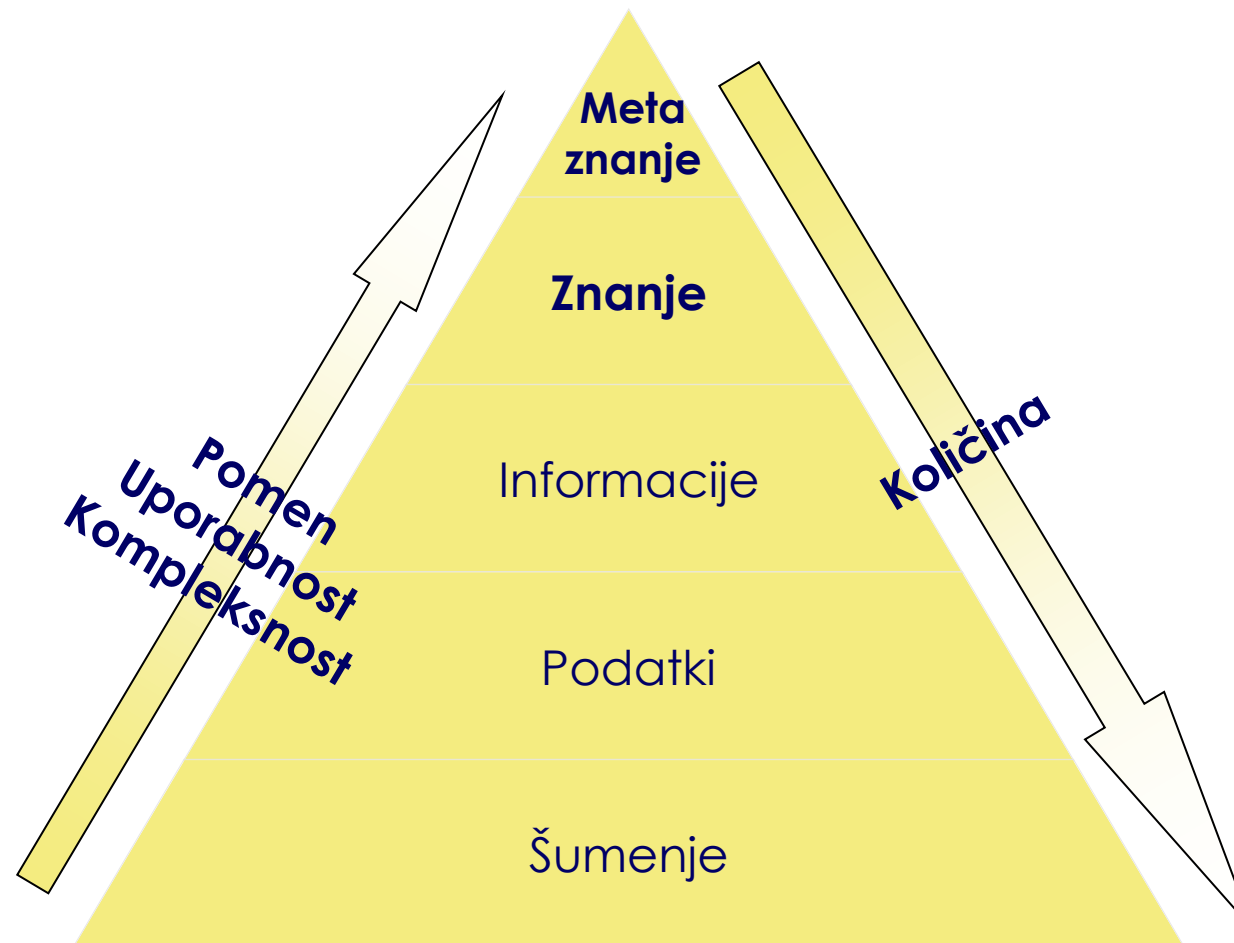
- časovno odvisna:
 - prediktivna
 - operativna
 - izkušenijska



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.

Informacijska hierarhija



Informacijski sistemi

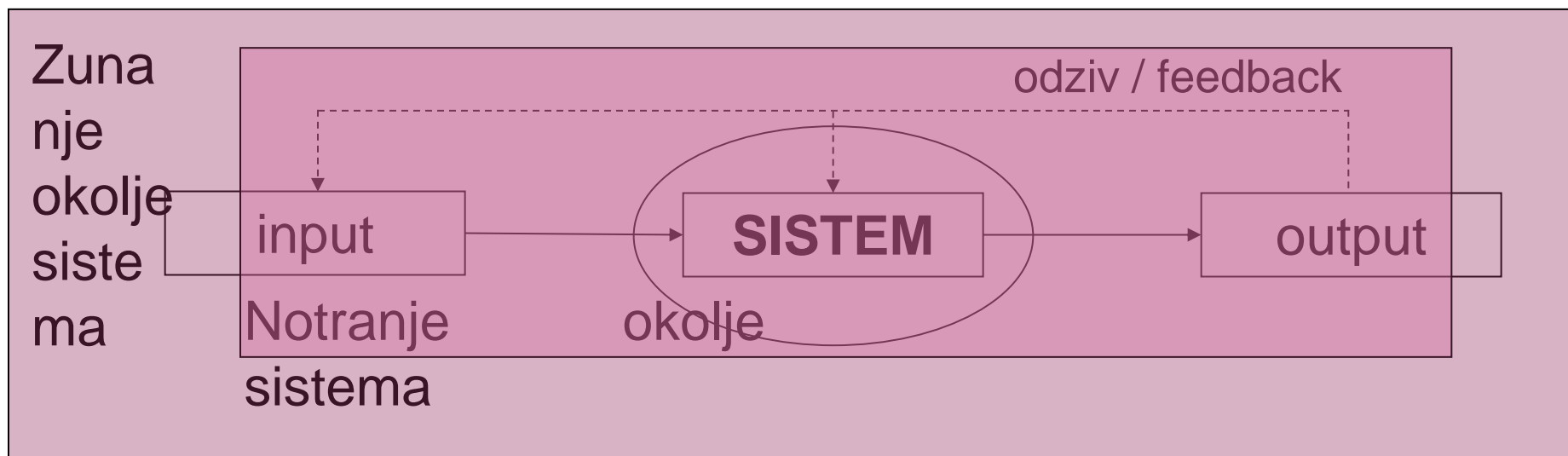
Sistemi, informacijski sistemi, vrste IS

Sistem

Definicija sistema

2024/2025

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.

Informacijski system

Definicija informacijskega sistema

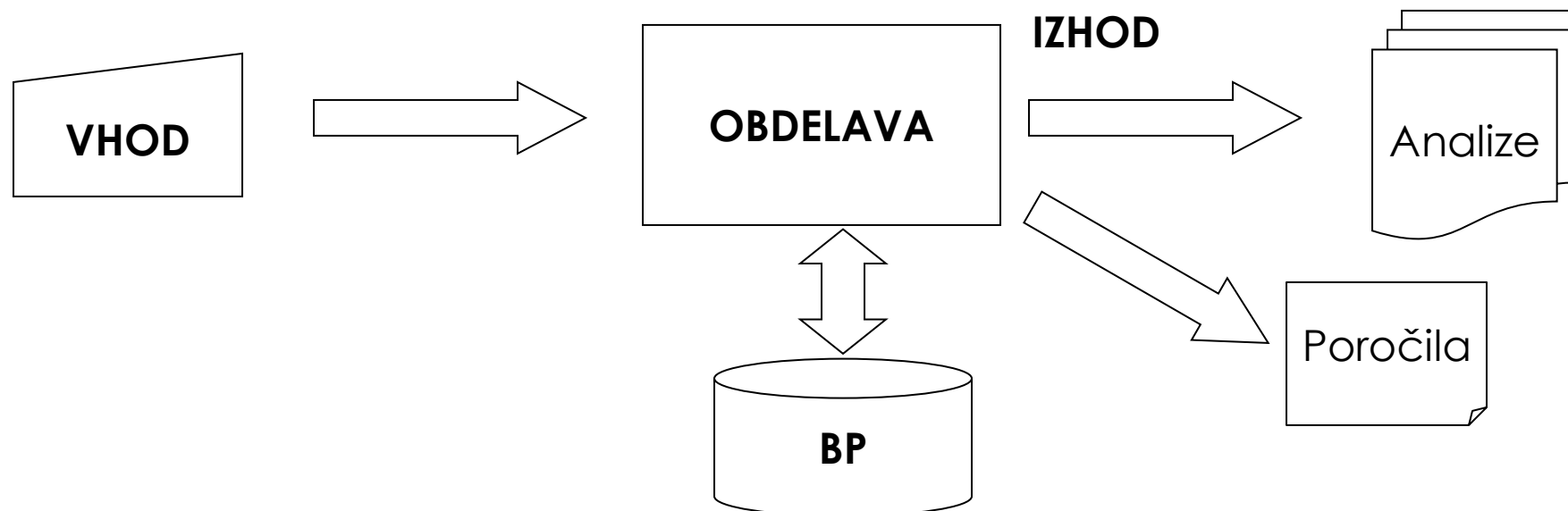
2024/2025

Informacijski sistem (IS) je množica aplikacij in aktivnosti, ki so potrebne za upravljanje, obdelavo in uporabo informacij kot virov organizacije.

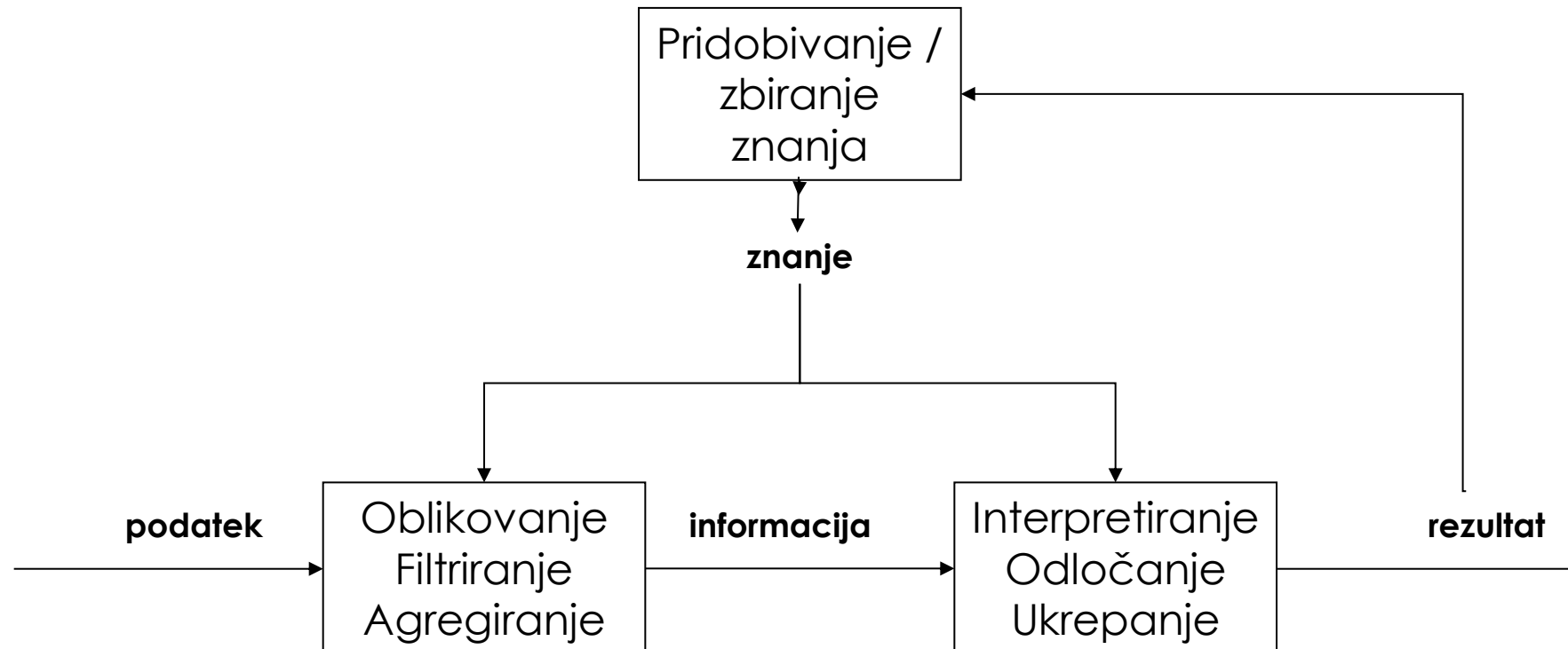
IS mogoča zbiranje, shranjevanje, obdelavo in analizo **podatkov** na način, ki zagotavlja zadovoljevanje **informacijskih zahtev** na vseh ravneh upravljanja.

Aktivnosti IS

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



'neskončna zanka'

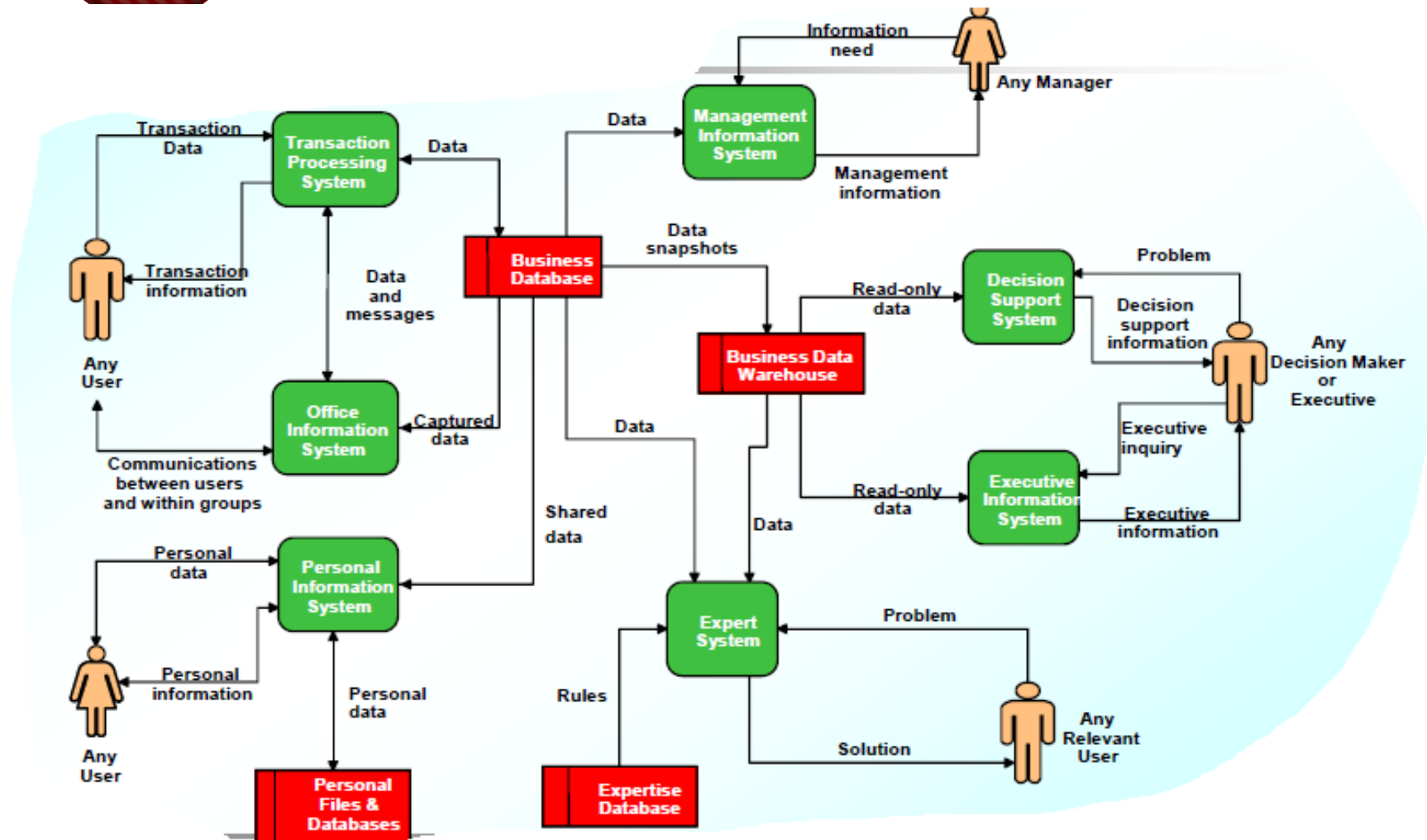


Delitve IS

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



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.

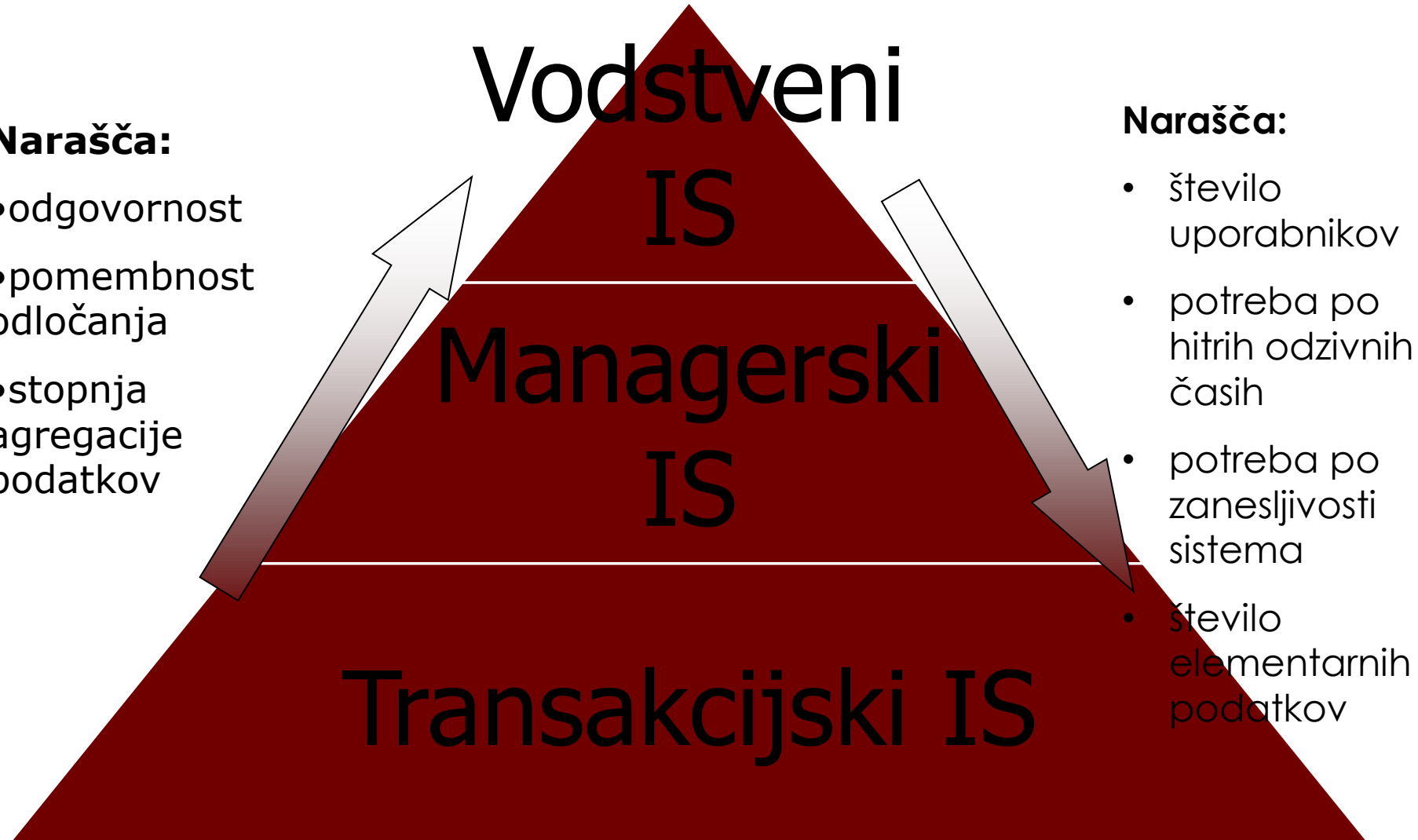
Vodstveni IS

(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.

Naraščča:

- odgovornost
- pomembnost odločanja
- stopnja agregacije podatkov



Naraščča:

- 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, O365

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

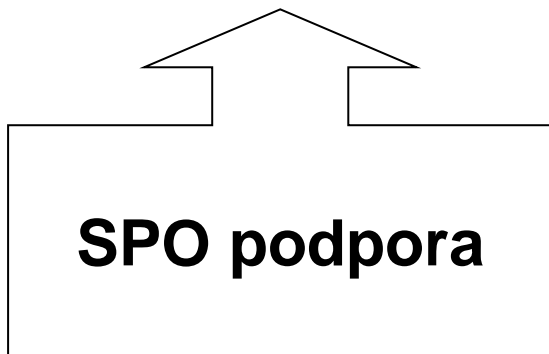
- del procesa reševanja problemov.
- je proces pri katerem izbiramo med več možnostmi – **variantami**; izbrana varianta naj čimbolj ustreza danim ciljem.
- rezultat je odločitev, t.j. izbira najboljše (najoptimalnejše) možnosti (variante) iz množice razpoložljivih.
- elementi: variante, kriteriji, odločitvena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

- o **Naključno** (ecika-pecika) (primer: vegovc rešuje kontrolko 😊)
- o **Intuitivno** (ne da bi vedeli, kako to počnemo, se nam 'zdi')
- o **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).

1. Opredelitev problema

2. Določitev kriterijev in strukturiranje kriterijev

3. Merske lestvice

4. Funkcije koristnosti

5. Variante

6. Vrednotenje

7. Prikaz in interpretacija rezultatov

Attributov: 8 (5 osn., 3 izp.) | Zalog vred.: 8 | Funkcij: 3 | Variant: 4

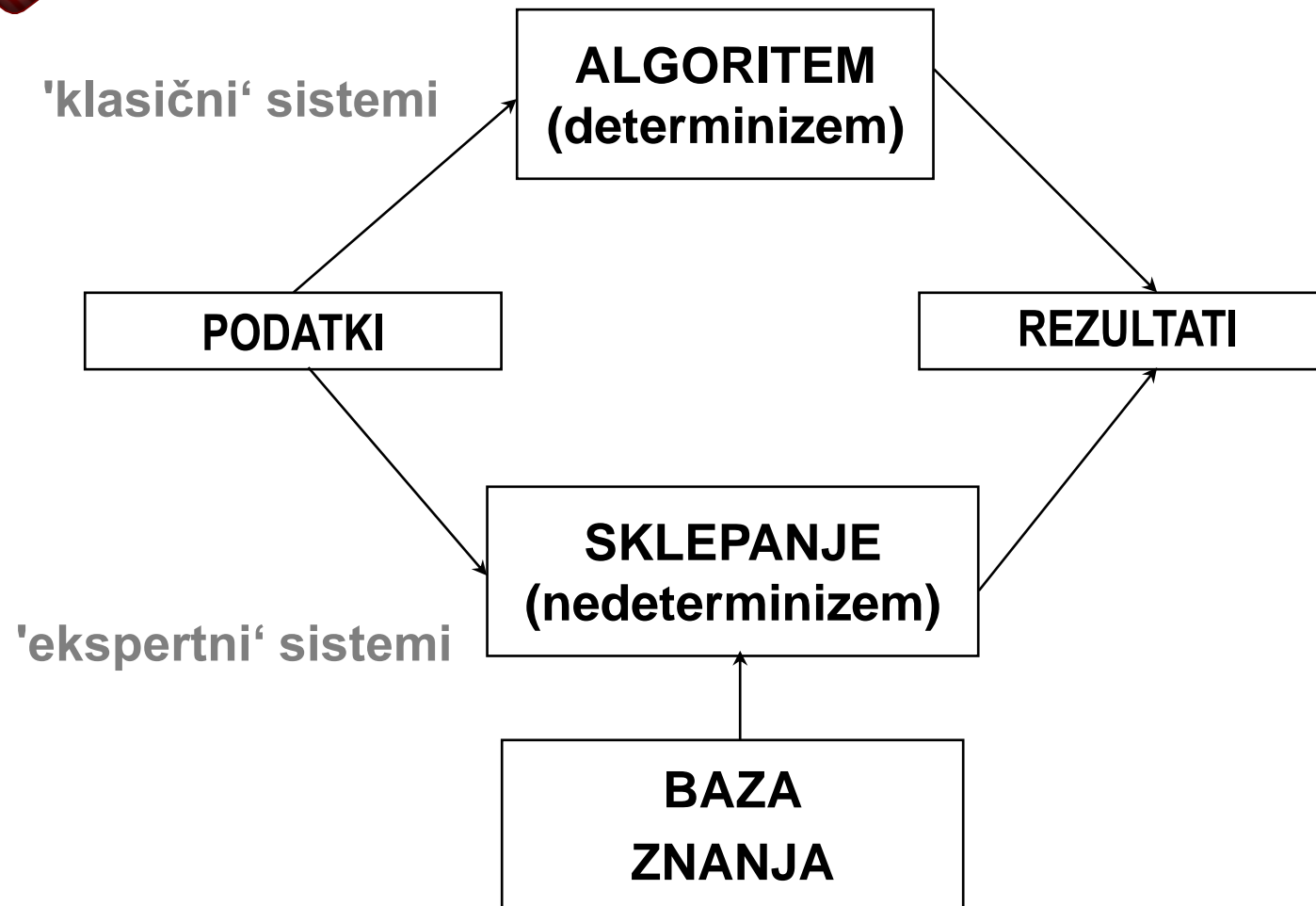
Ekspertni sistemi

- **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

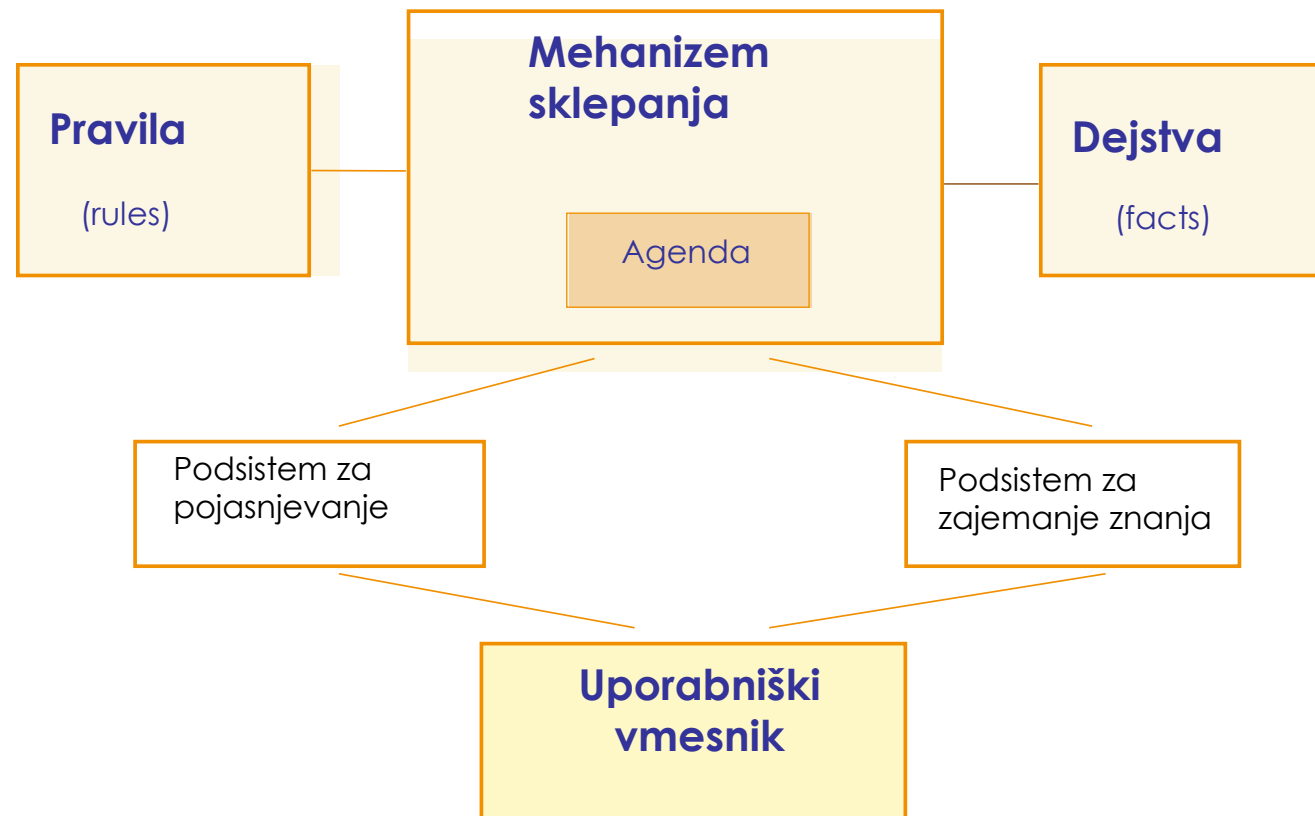
- sklepanje.
- presoja.
- delovanje pri nepopolnih oziroma nezanesljivih podatkih.
- pojasnjevanje delovanja in predlaganih rešitev (sledenje nazaj):
 - ES je za uporabnika 'prosojna' škatla, klasična aplikacija pa 'črna' škatla.

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

2024/2025

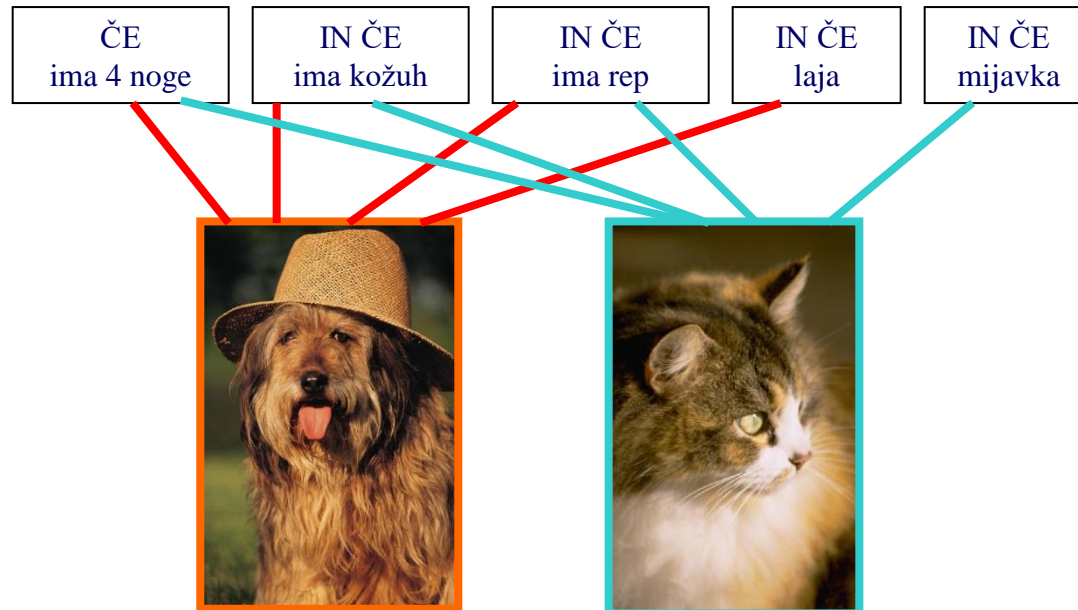


Struktura ES - baza znanja

- 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

- podsistem, 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).

Primeri 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 domnevno 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).

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.

Življenjski cikel

2024/2025

(ang. *SDLC – Software Development Life Circle*)

V teoriji naletimo na bolj ali manj podrobno razdelane faze v “življenju” nekega programa. Govorimo o **življenjskem ciklu programa**. Ena od možnih opredelitev življenjskega cikla programa:

1. definicija in analiza

- opredelitev osnovne ideje in ciljev, analiza problema

2. načrtovanje

- razbijanje problema na podprobleme, izbira najboljših algoritmov, planiranje vhodnih in izhodnih podatkov ter njihovega izgleda, uporaba raznih diagramskih tehnik, npr. diagramov poteka
- v tej fazi problem v bistvu že dokončno rešimo

3. implementacija (programiranje)

- pretvorba rešitve v izbrani programski jezik (kodiranje)

4. preverjanje in testiranje

- testiranje pravilnosti in učinkovitosti in po potrebi vračanje na zgodnejše faze

5. vzdrževanje

- odpravljanje neodkritih napak, posodabljanje

6. (ukinitiv)

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:

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

Študija izvedljivosti

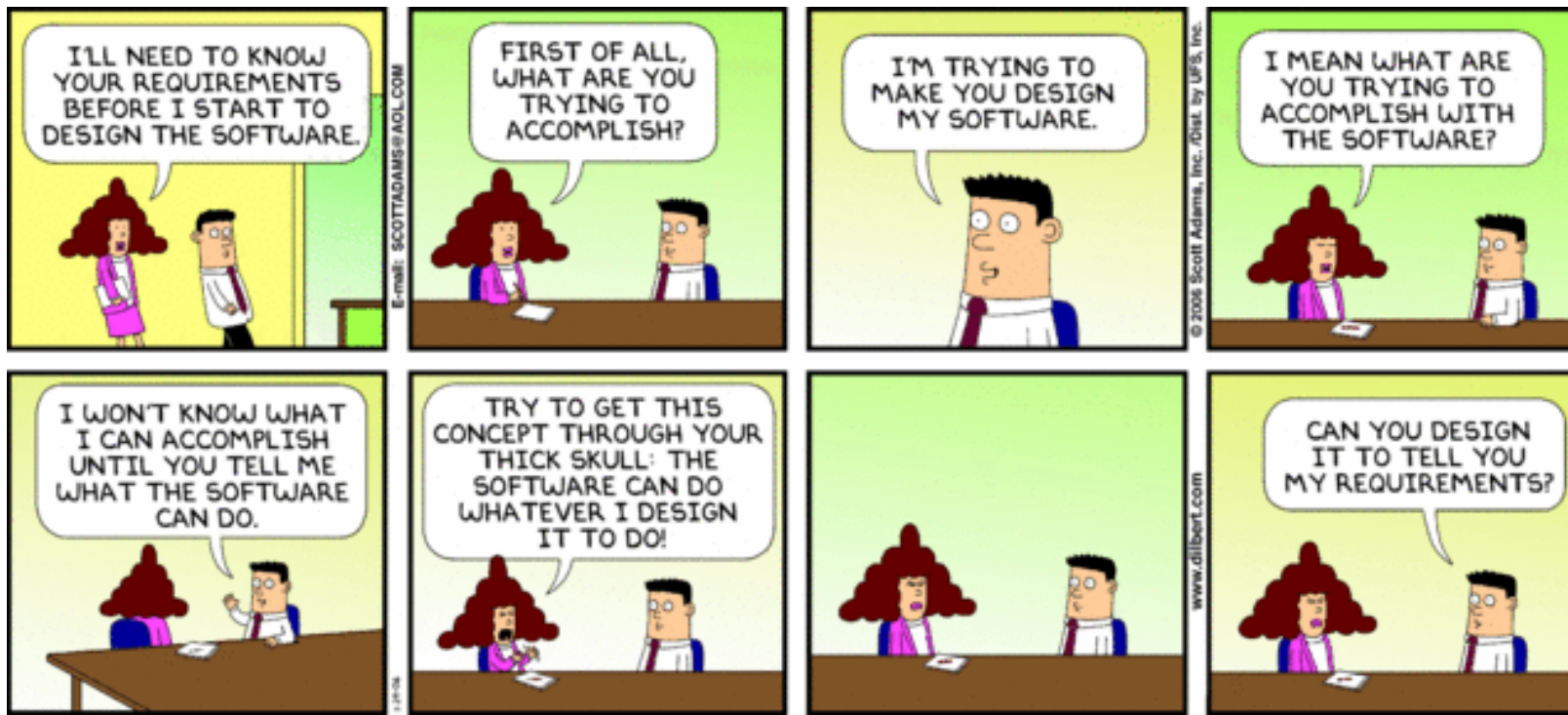
Študija izvedljivosti predhodi odločitvi o začetku projekta razvoja:

- 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

Testing cannot show the absence of defects, it can only show that software defects are present.

Dijkstra

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.

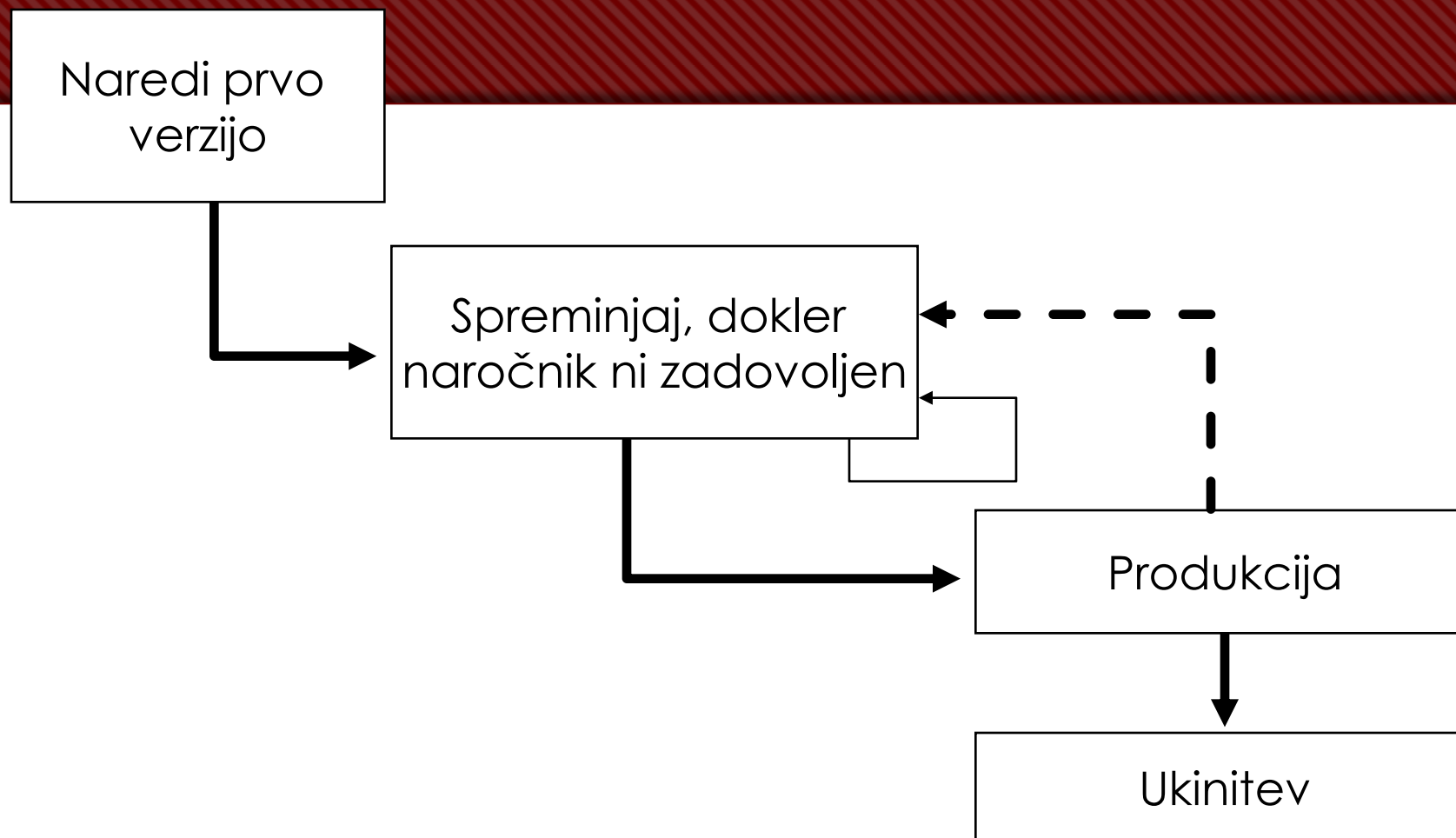
Cilj te faze je **zagotoviti nemoteno izvajanje** programske opreme pri uporabniku.

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

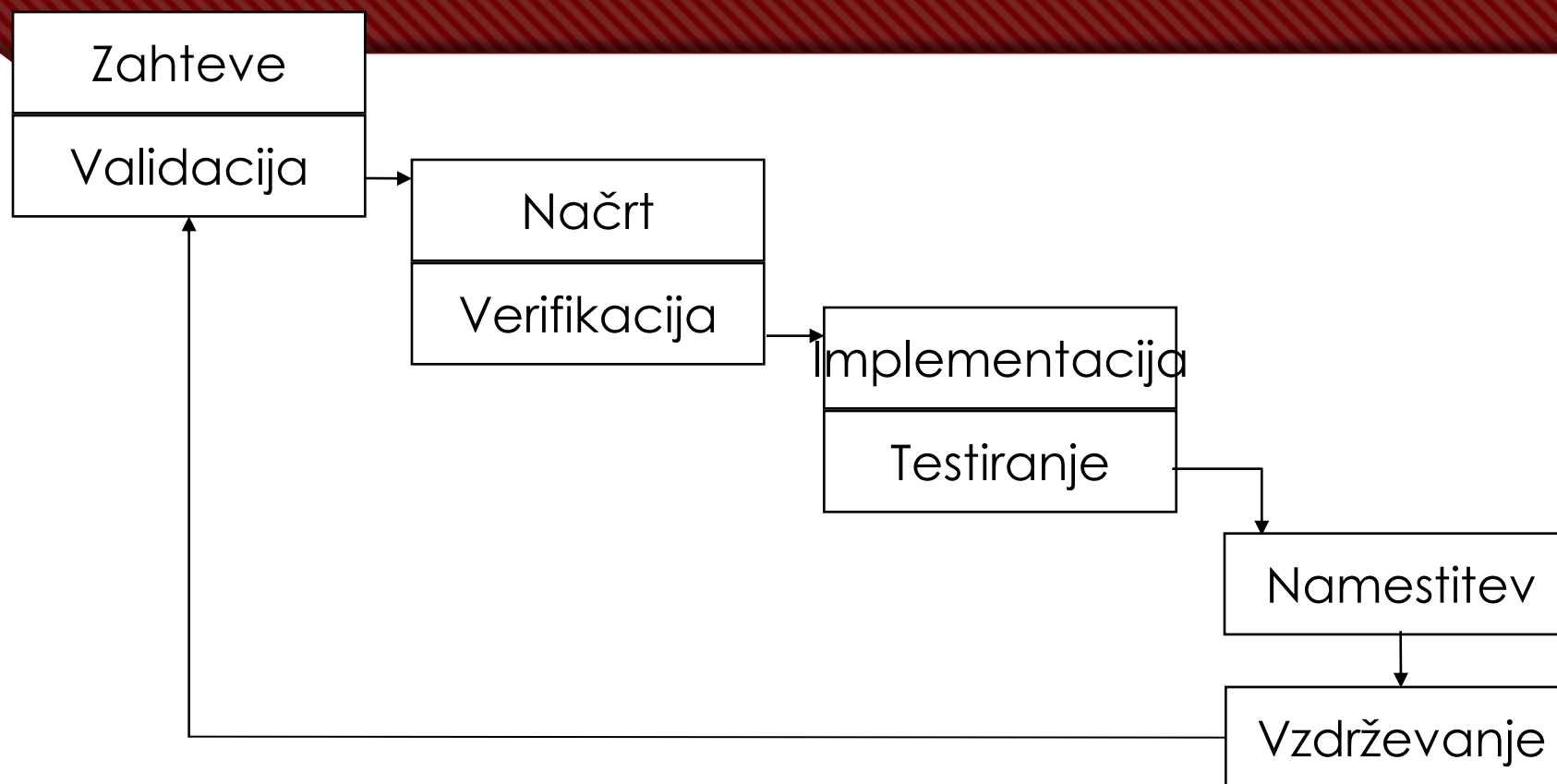


Code&Fix – prednost in pomanjkljivosti

- 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

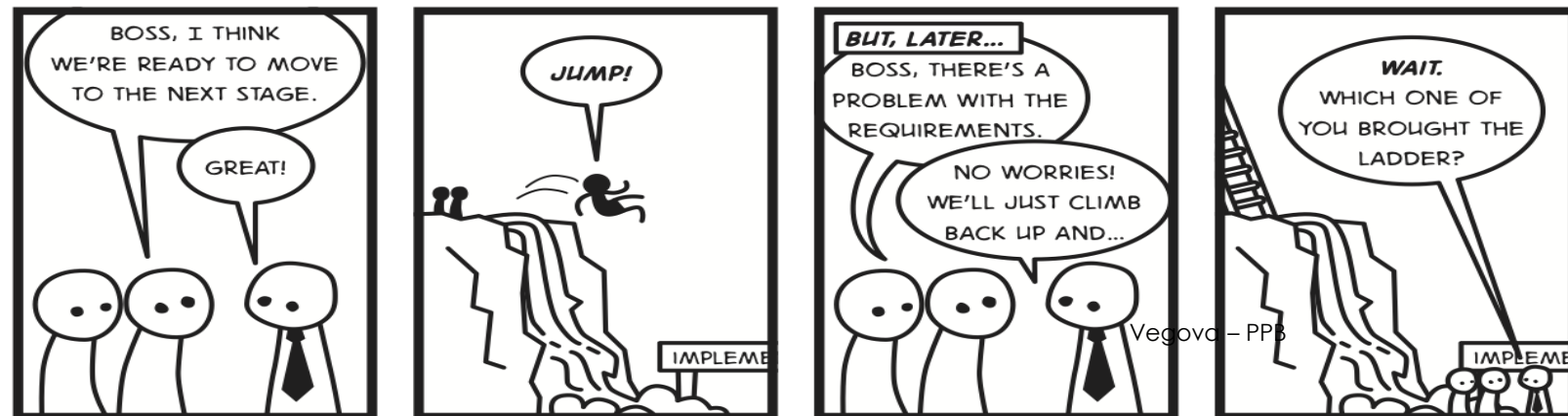


Model slapa – prednosti in pomanjkljivosti

- 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

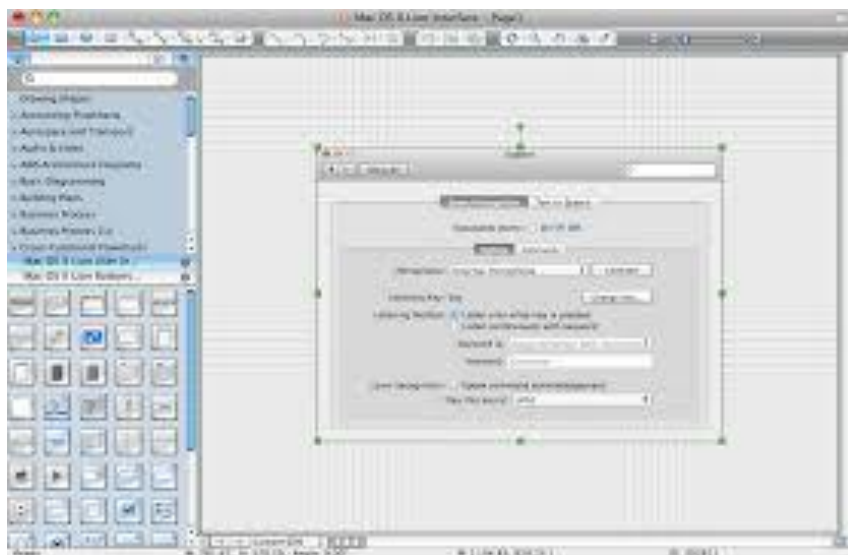
2024/2025

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

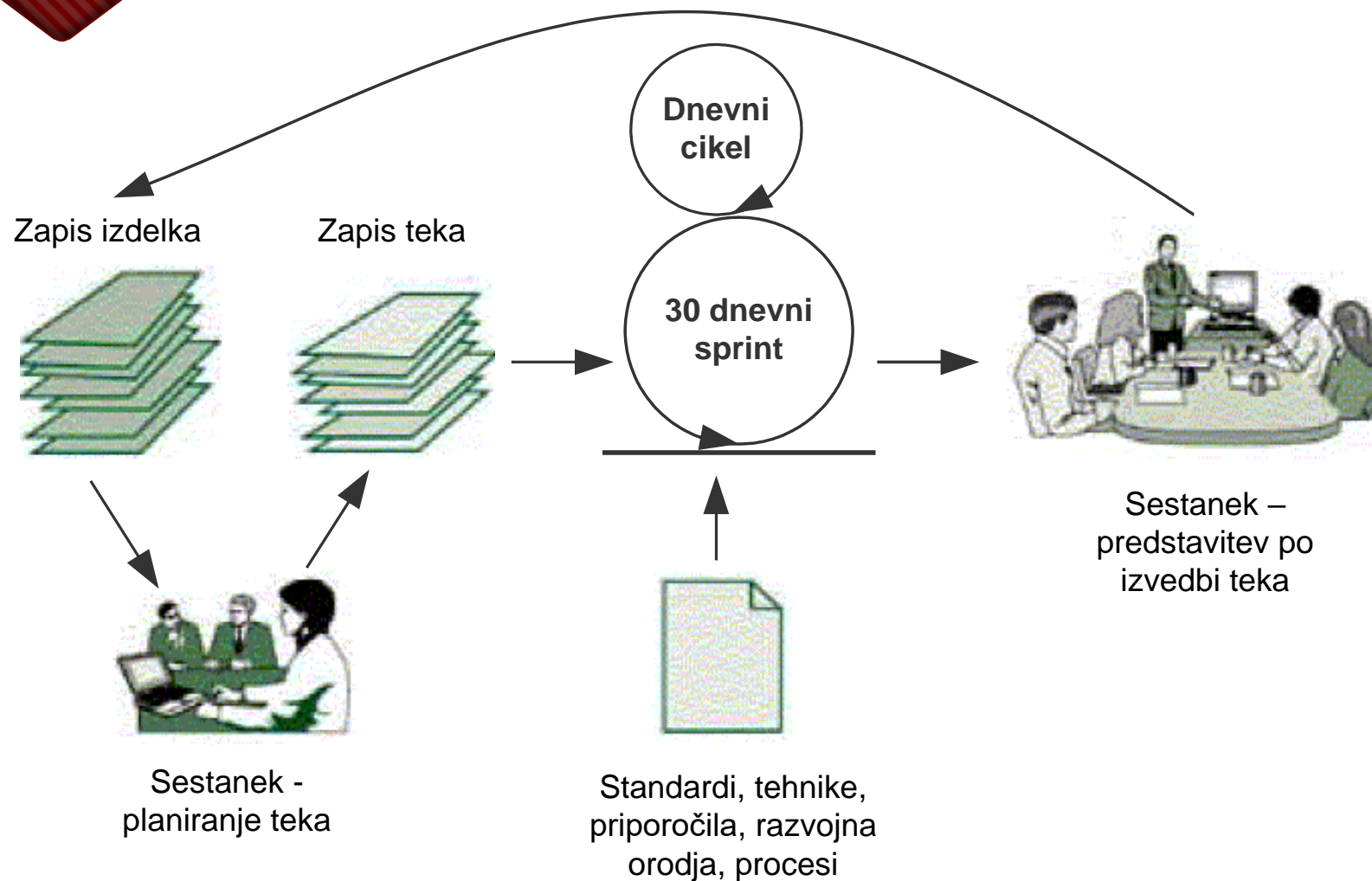
- Vsaka ponovitev je mini-projekt:
 - vsaka ponovitev ni prototip ampak delujoča verzija PO.
 - razumevanje tveganj vs. vrednosti tekom planiranja.
 - predati nekaj delujočih funkcionalnosti naročniku čimprej.
- **Timeboxing:**
 - določanje datuma za dobavo ponovitve.
 - datum se ne sme spreminjati.
 - 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

2024/2025



- **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 doseg cilja,
 - vsi sodelujejo na dnevnih 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