

# Računalništvo04

## IS, UML, PB, SQL, ER

predavatelj: Aleksandar Lazarević

# Agenda

- UML
- **Diagrami primerov uporabe**
- Razredni diagrami
- Diagrami aktivnosti

# UML

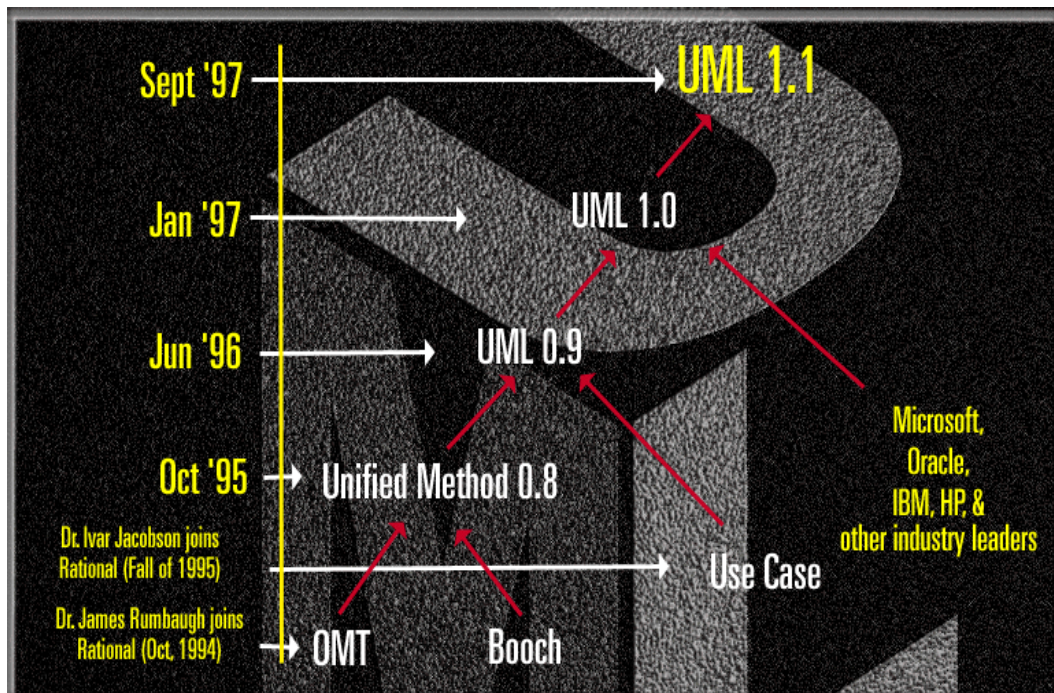
UML: diagrami primerov uporabe, diagrami aktivnosti, razredni diagrami

# Kaj je UML?

- Standard (norma) za kreiranje **modela sistema**.
  - vsebuje standardizirano notacijo s predlogami diagramov, ki jih lahko uporabimo za izgradnjo modela sistema.
- **Model sistema** je popolni opis sistema iz določenega vidika.
- UML je eden prvih korakov k **standardizaciji** programskega inženirstva.

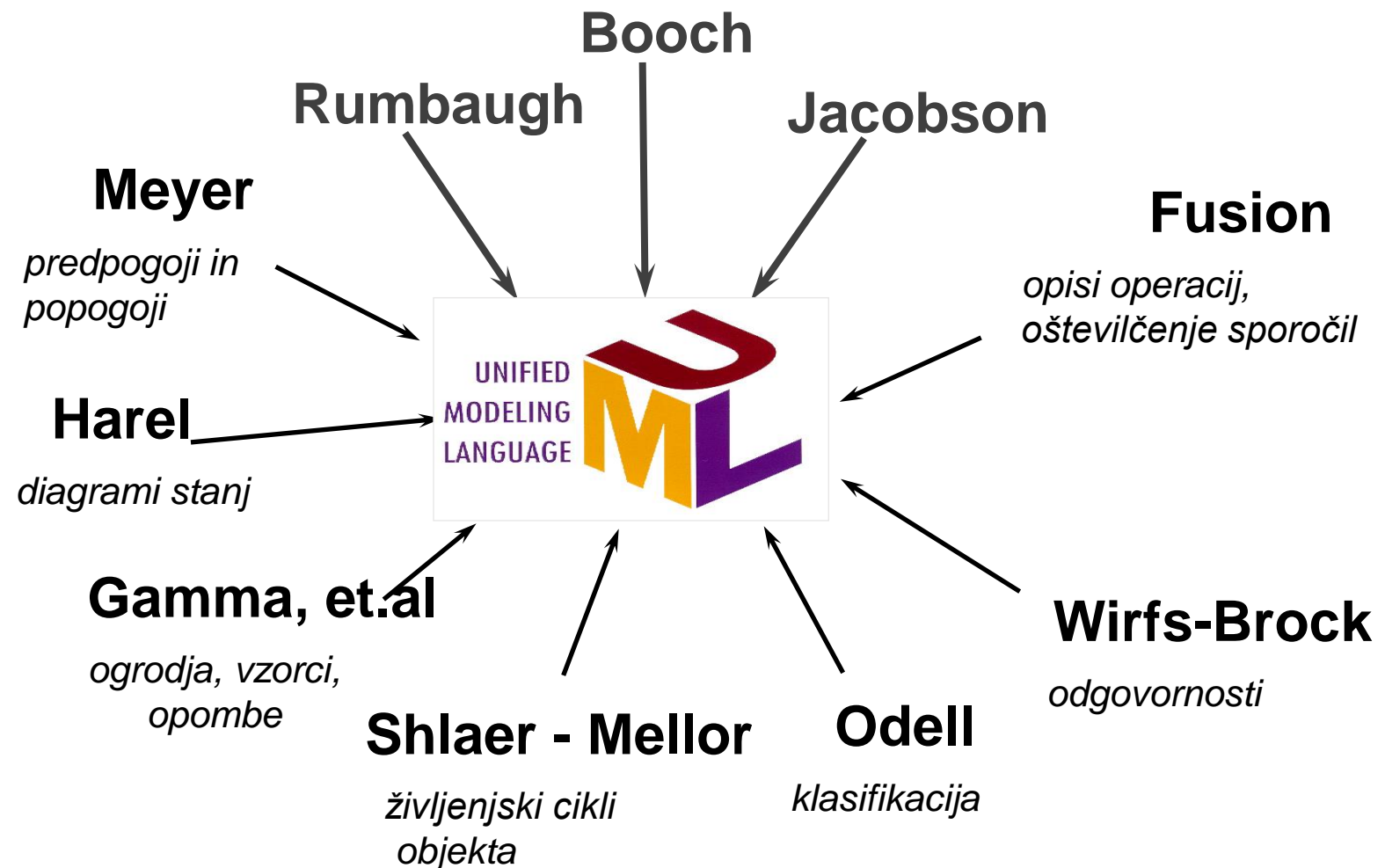
# UML

- o Unified Modeling Language (UML) je jezik za specifikacijo, vizualizacijo, konstrukcijo in dokumentacijo izdelkov programsko intenzivnega sistema.



# UML

- Razvoj UML vključuje ideje in prispevke številnih avtorjev
- Izziv, s katerim so se soočali pri nastajanju UML:
  - jezik naj bo po eni strani enostaven,
  - po drugi strani pa robusten in bo omogočal modeliranje široke palete različnih sistemov.

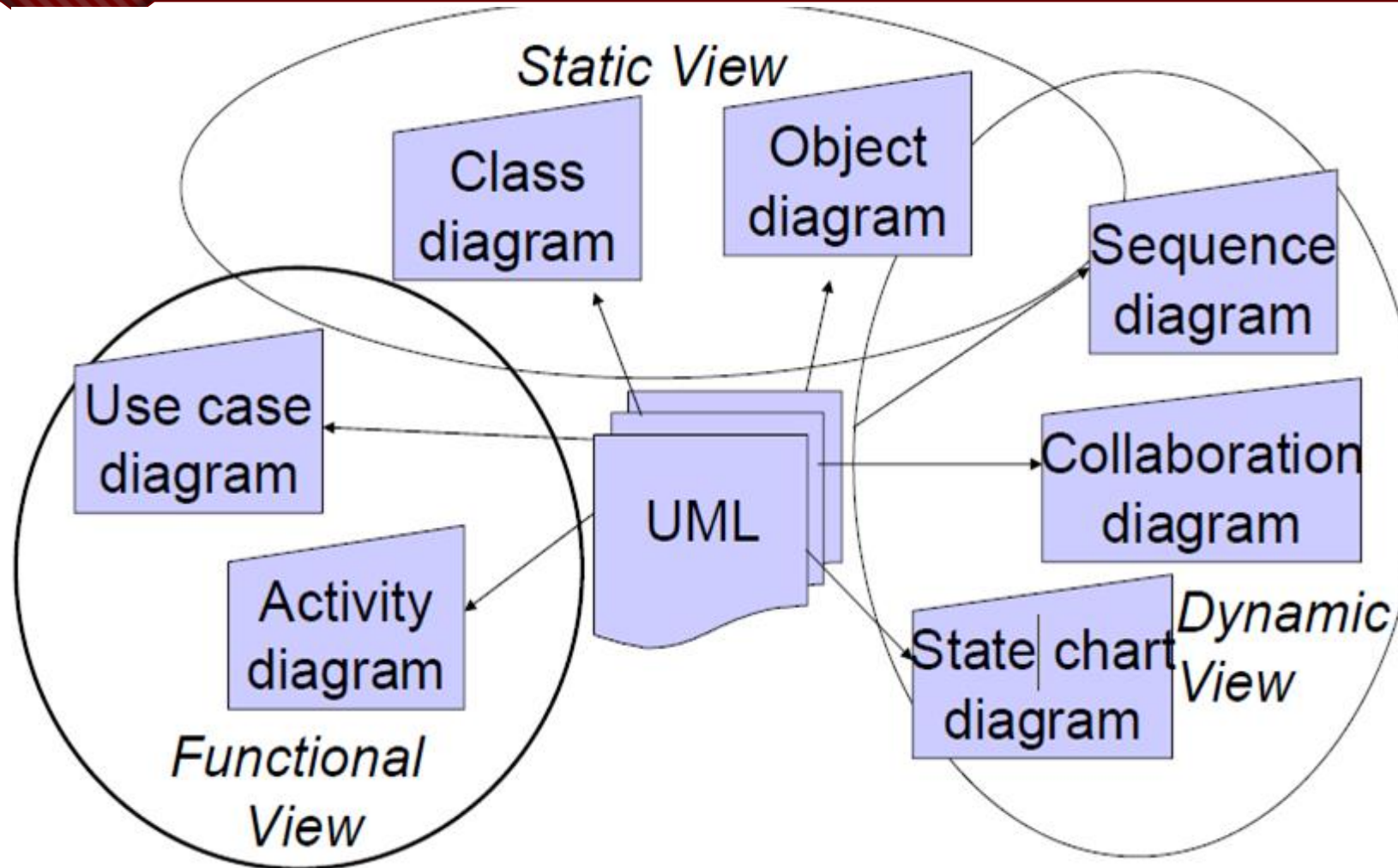




# UML komponente

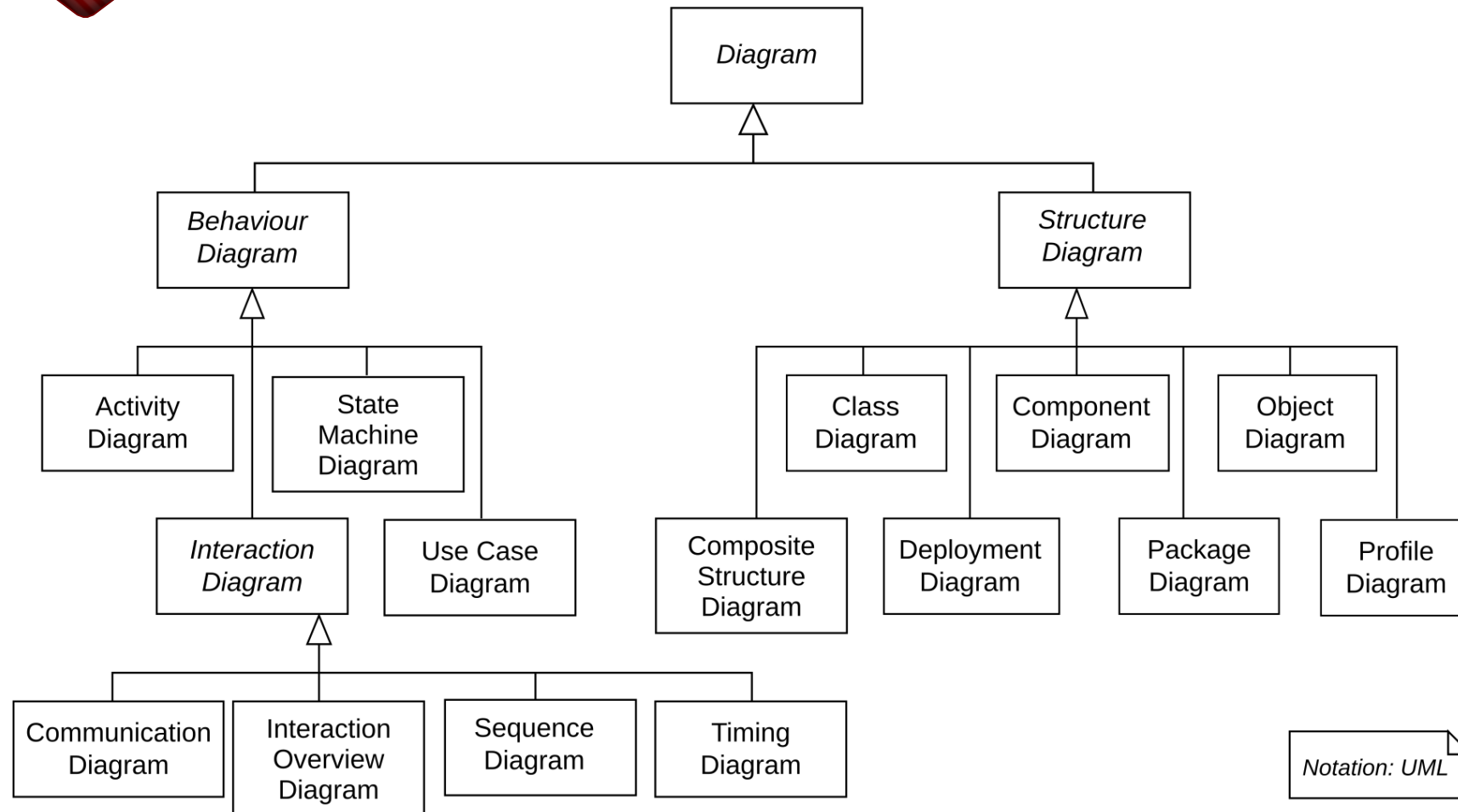
- vsebuje specifikacije 9 različnih diagramov (v najnovejši različici 2.5.1 pa 14) ki so namenjeni dokumentaciji različnih vidikov programskih produktov, v različnih fazah življenjskega cikla
  - 7 diagramov za modeliranje zahtev in načrtovanje
  - 2 za opis implementacije
- Diagram u UML-u:
  - predstavlja pogled na model sistema,
  - lahko je prikazan iz različnih perspektiv, od različnih udeležencev (ang. *stakeholder*)
  - podaja samo delni opis celotnega sistema,
  - semantično je konsistentan z ostalimi pogledi na sistem.

# UML vidiki



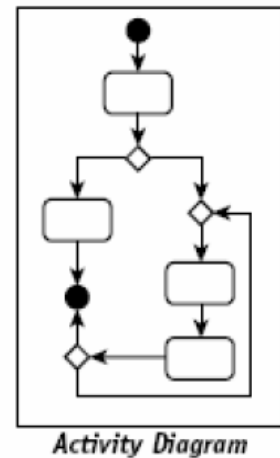
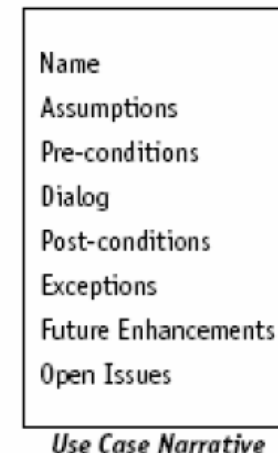
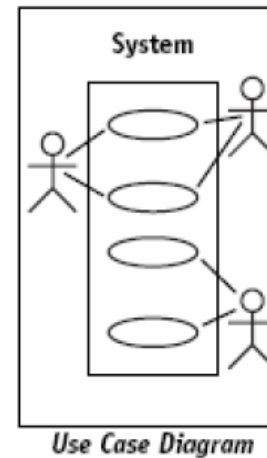


# UML 2.5.1



# Funkcionalni vidik

- Opisuje **kako** naj sistem dela.
- **Use Case diagram**
  - opisuje lastnosti sistema, ki jih pričakuje uporabnik.
- **Activity diagram**
  - opisuje procese sistema kot niz nalog, ki morajo biti opravljene, pogojev in vzporednih obdelav (podobno kot flowchart).

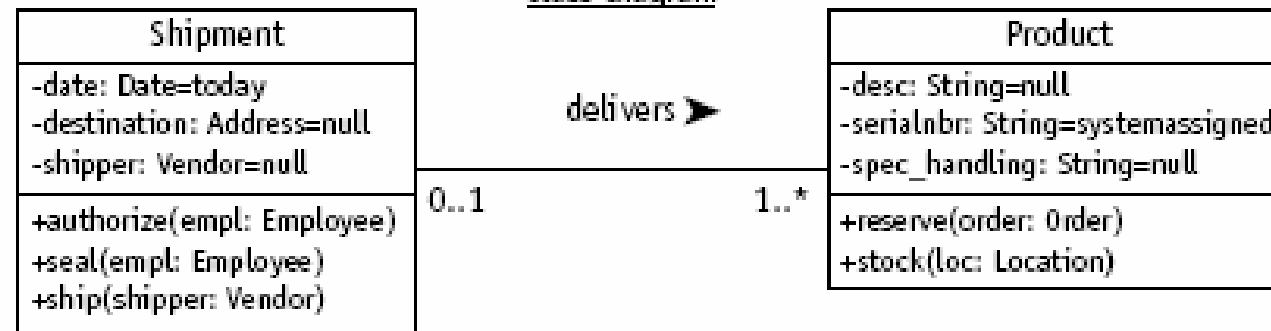


# Statični vidik

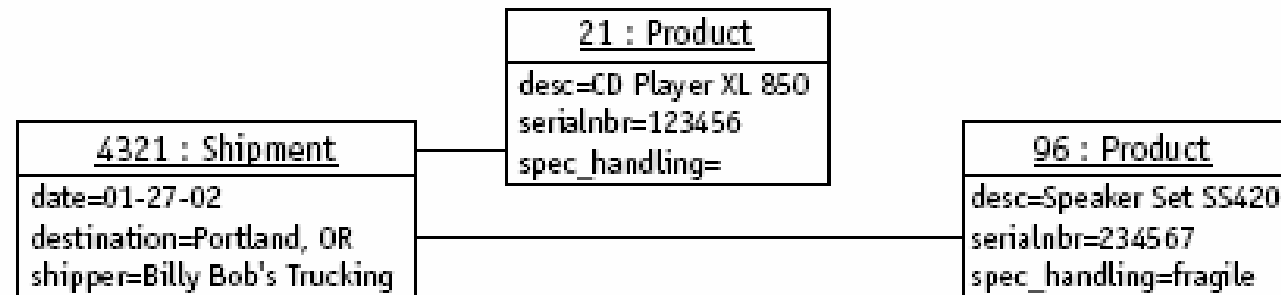
- Opisuje gradbene elemente sistema ampak ne način kako med seboj sodelujejo.
- Tipični predstavniki so:
  - *Class diagram*
    - pogled na vse vire (razrede) in njihove lastnosti. Skoraj vedno se uporablja za generiranje kode in reverzni inženiring.
  - *Object diagram*
    - konkretizira splošno sliko sistema prikazano z razrednim diagramom.

# Statični vidik

Class diagram



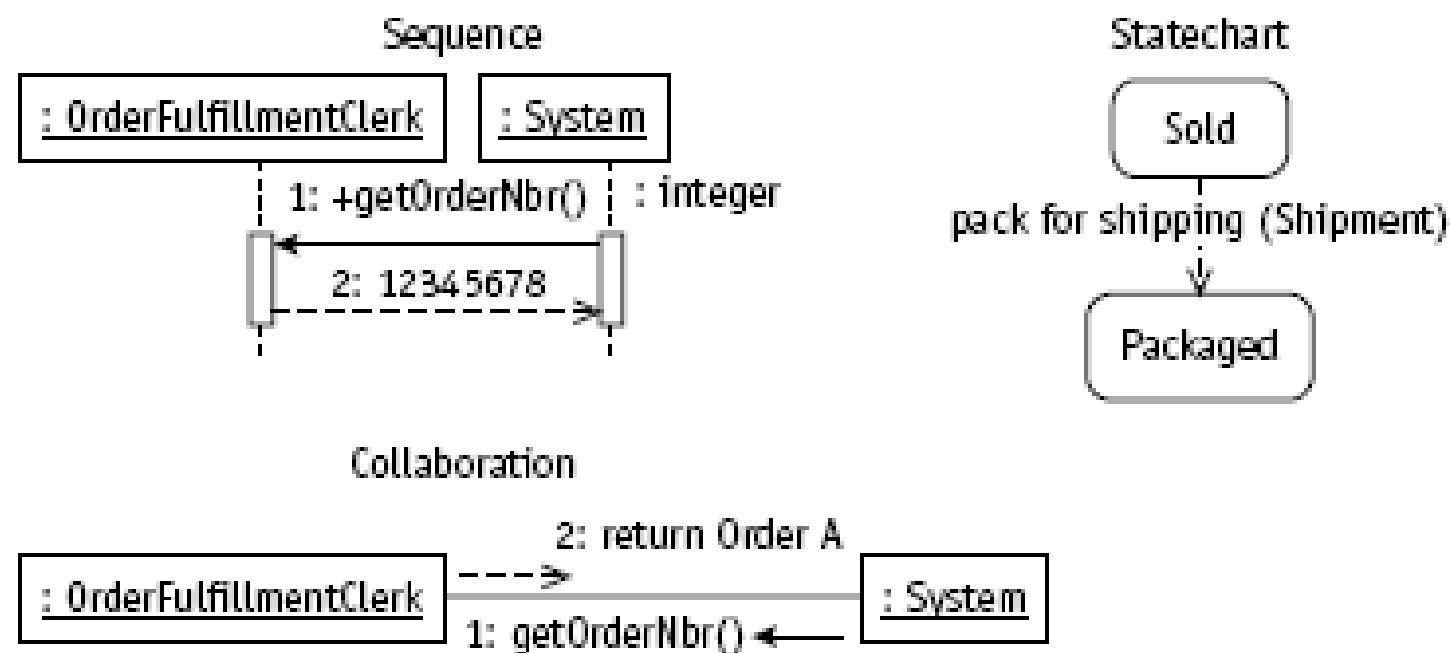
Object diagram



# Dinamični vidik

- o Določa **obnašanje sistema** – medsebojno delovanje elementov sistema, kot odgovor na spodbude iz okolja.
- o **Sequence and collaboration** diagrami
  - o opisujejo interakcije.
- o **State chart** diagram
  - o kako i zakaj se objekti spreminjajo, kot odgovor na spodbude iz okolja.

# Dinamični vidik





# Diagrami primerov uporabe

(ang. *use case diagrams*)

Osnovni gradniki diagramov primerov uporabe: akterji, primeri uporabe, povezave

# Primer uporabe (use case)

- **Primer uporabe (ang. use case)** predstavlja neko funkcionalnost sistema oz. opisuje sistem iz vidika uporabe sistema zunanjega uporabnika.
- Npr.: primeri uporabe spletne banke:
  - plačilo položnice
  - prenos denarja iz računa na račun
  - pregled stanja računa
  - ...

# Diagrami primerov uporabe

○ Diagram primerov uporabe je **grafična predstavitev** uporabnikov, sistema in primerov uporabe.

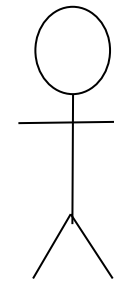
■ Osnovni gradniki diagramov primerov uporabe so:

- **uporabnik, agent, akter** (ang. actor),
- **sistem**,
- **primer uporabe** (ang. use case),
- **povezave**.



# Agent, akter

- ... je tisti, ki uporablja sistem.
- Nekaj kaj:
  - stimulira sistem, da reagira - **primarni akter** (*ang. primary actor*)
  - odgovarja na zahteve sistema – **sekundarni akter** (*ang. secondary actor*)
- Nekaj, **kar ne moremo kontrolirati**.
- Ni nujno, da je človek – lahko je **drug sistem**!



# Iskanje akterjev

- Kdo uporablja sistem?
- Kdo zagotavlja podatke sistemu?
- Kdo pridobiva informacije iz sistema?
- kateri drugi sistemi uporabljajo sistem?
- Kdo namešča, nastavlja in vzdržuje sistem?

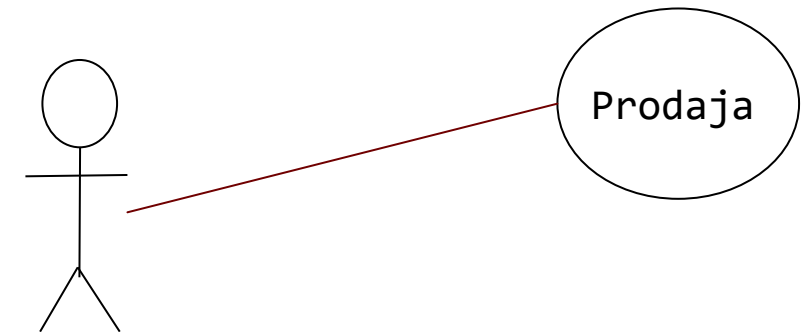
# Imenovanje akterjev

- Grupirati uporabnike glede na njihovo uporabo sistema. Ugotoviti vloge, ki jo imajo, ko uporabljajo sistem.
- Vsaka **vloga** je potencialni akter.
- Imenovati posamezno vlogo in definirati njihove lastnosti.
  - Uporabiti znana imena v obstoječem sistemu; ne izmišljati nova imena.
  - Ne enačiti naziv delovnega mesta z uporabniško vlogo.
  - Ne izgubljati preveč časa pri iskanju imen akterjev.



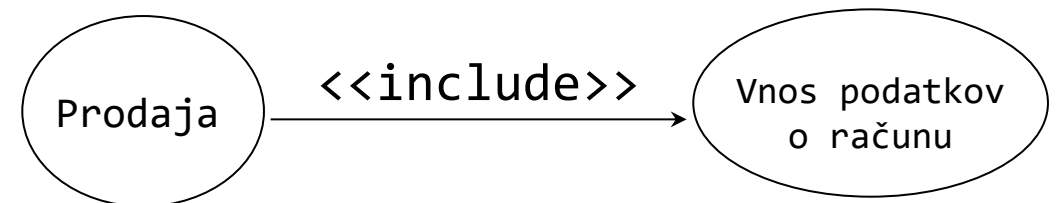
# Povezave med primeri uporabe

- Osnovni mehanizmi:
  - akter-primer uporabe:
    - osnovna povezava - uporaba
    - include - vključitev
    - extends – razširitev
    - inheritance - dedovanje



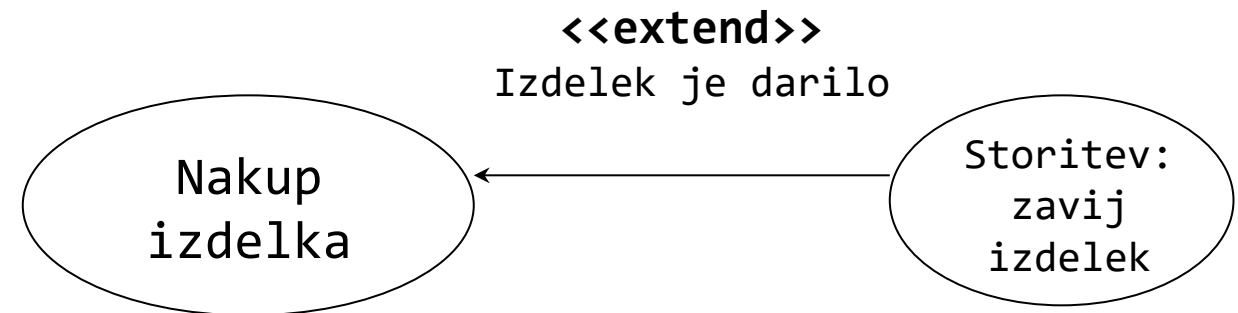
# Include

- **include** – uporablja se ko želimo izvesti:
  - dekompozicijo zapletenega obnašanja sistema,
  - centralizacijo splošnega obnašanja sistema.
- Pomeni, da osnovni primer uporabe **obvezno** vključuje drugi primer uporabe.



# Extend

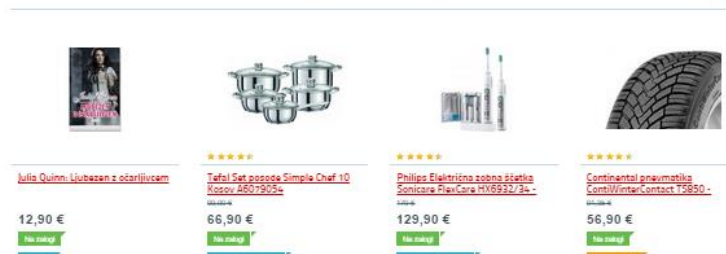
- **extend** pomeni, da osnovno obnašanje primera uporabe lahko **razširimo** z drugim primerom uporabe.
- Smer je iz drugega v osnovni: osnovni primer uporabe ne zna s katerim primerom je razširjen.



## 2024/2025

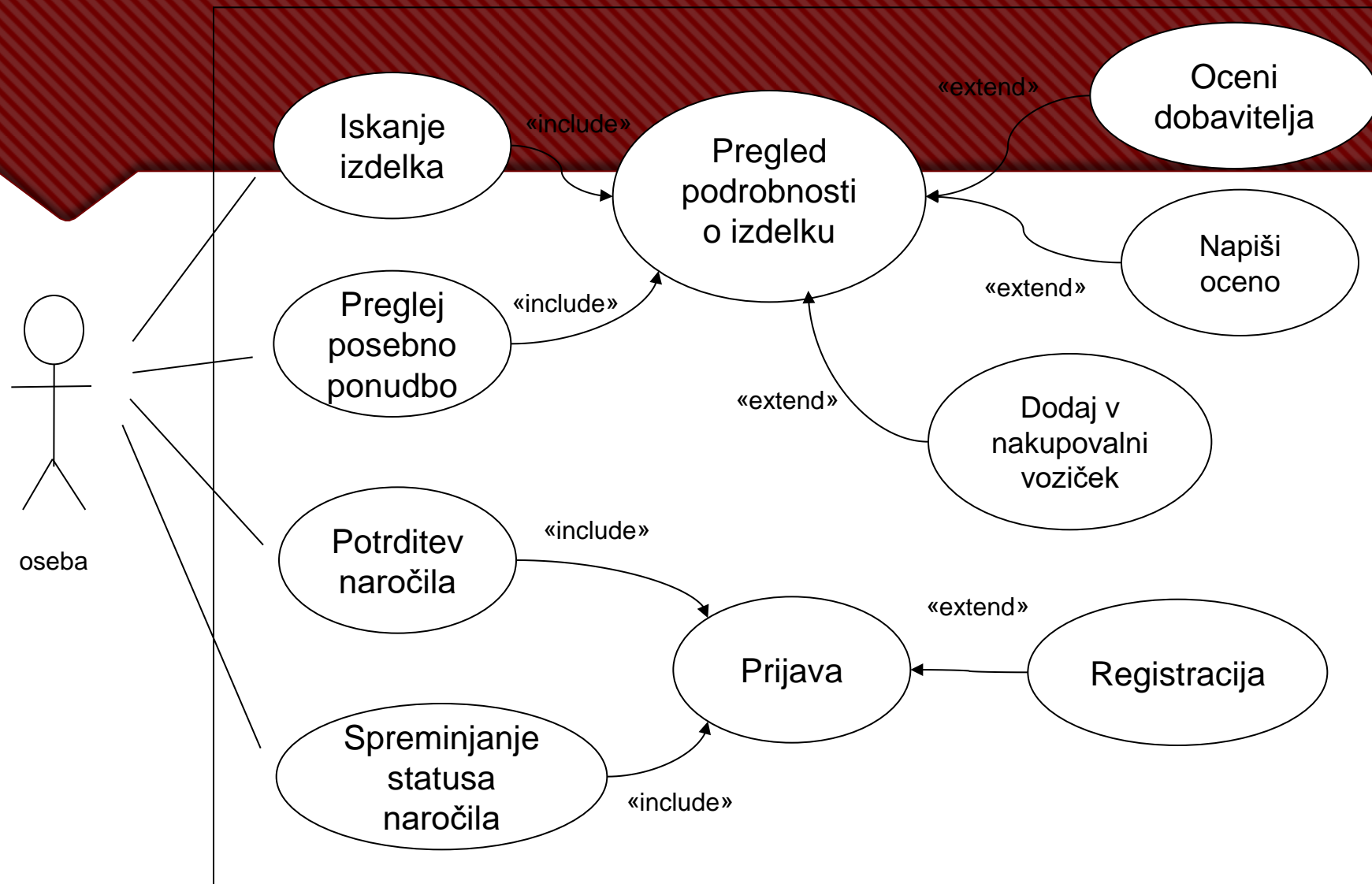


## PRIJUBLJENE KATEGORIJE

**NAJBOLJŠA PONUDBA DNEVA****IZBRANO ZA VAS**

# Primer: spletna trgovina

2024/2025



# Primeri

- profesor sprašuje študenta, pri čemer mu vedno mora dati oceno.
- žena in mož pripravljata večerjo, pri čemer mož vedno pripravi koktel.
- oseba gre na kosilo in včasih mora dvigniti denar iz bankomata.
- serviser opravlja servis avtomobila; v primeru rednega servisa včasih mora zamenjati zavore.
- vodja laboratorija vedno opravlja testiranje skupaj z asistentom, pri čemer asistent včasih opravi dodatno testiranje, vodja pa vedno pripravi poročilo.
- študent in profesor morata biti prisotna na izpitu. Če je izpit ustni, profesor mora biti prisoten, kot drugi ocenjevalec.



# Naloga

Z diagramom primerov uporabe želimo takole opisati restavracijo takole: stranka naroči hrano. Natak ar sprejme naročilo in hrano prinese, ko je pripravljena. Ko stranka konča obrok, natak ar prinese račun, ki ga stranka plača, natak ar pa sprejme plačilo.

Obkrožite črko pred akterjema diagrama.

A Natak ar.

B Hrana.

C Stranka.

D Plačilo.

Narišite diagram primerov uporabe.

# Naloga

Spletno borzo uporabljajo kupci in prodajalci. Oboji se morajo pred pričetkom trgovanja prijaviti. Postopek prijave obsega tudi preverjanje uporabniških podatkov.

Kupec na borzi lahko opravlja analizo tveganja, s prodajalcem pa še pogajanje in sklepanje pogodbe.

Pri pogajanju in sklepanju pogodb se pri prikazu cene uporablja pretvornik valut.

Ko kupec izvaja analizo tveganja, ima na voljo možnost analize valutnih tveganj in analize trga surovin.

# Naloga

V spletni trgovini stranke naročajo in plačajo blago. Stranka se mora za naročanje blaga registrirati.

Naročilo se preveri v oddelku za naročanje. Uslužbenci oddelka, ki preverjajo registracijo strank, se prav tako morajo prijaviti.

Stranke plačajo blago s kreditno kartico ali prek sistema PayPal.

Izdelajte model uporabe (Use Case) za spletno trgovino. Na modelu morajo biti razvidni akterji, meje sistema, posamezni primeri uporabe (use case) in povezave med njimi.

# Naloga

Treba je narediti sistem, ki bo profesorjem omogočal vnos posameznih izostankov dijakov. Pri tem je potrebna avtorizacija (preverjanje pristopnih dovoljenj) profesorja.

Želimo, da bi se samodejno po vnosu izostanka dijaka spremenila tudi statistika njegovih izostankov.

Sistem mora omogočati razrednikom vpogled v statistiko izostankov. Tudi za dostop do nje je potrebna avtorizacija razrednika.

Z diagramom UseCase narišite model sistema.

# Naloga

Šolski informacijski sistem uporabljajo učitelji, dijaki in vodstvo šole. Vodstvo šole se mora prijaviti in potem ima vpogled v vse ocene dijakov, lahko tudi objavlja okrožnice.

Dijaki imajo le možnost vpogleda v lastne podatke (ocene in izostanke) in pregled objav. Za pregledovanje ocen in izostankov se morajo prijaviti, objave so prosto dostopne.

Učitelji lahko vpisujejo ocene in izostanke. Šolski pravilnik opredeljuje, da se v nekaterih primerih po vnosu izostankov izrekajo tudi vzgojni ukrepi. Za vnos ocen in izostankov morajo biti učitelji prijavljeni, objave vodstva lahko pregledujejo prosto dostopno.

# Formalni zapis primera uporabe

**Naziv, ID**

**Akterji**

**Sprožilec**

**Predpogoji**

**Tok podatkov**

**Rezultat**

**Alternativni tok(ovi)**



# Zapisovanje primerov uporabe

**Primer uporabe:** Ažuriraj polico

**ID:** PU2

**Akter:** Zavarovalniški uradnik (ZU)

**Predpogoj:** Polica že obstaja

**Sprožilec:** Stranka želi spremeniti enega od atributov upravičenca

**Tok dogodkov:**

- ZU (zavarovalniški uradnik) odpre obstoječo polico
- ZU vpiše novo vrednost atributa
- Sistem vpraša za potrditev spremembe
- IF ZU potrdi spremembo
  - THEN sistem shrani novo vrednost
  - ELSE sistem ne shrani nove vrednosti

**Rezultat:** Podatek na polici je ažuriran.

**Alternativni tok:**

1. zavarovalniški uradnik (zavarnik) je zavarovalen

# Zapis primera uporabe (tok)

- Opisna oblika (*ang. narrative*),
- Oblika scenarija (*ang. scenario*),
- Oblika dialoga (*ang. conversation dialog*).

# Opisna oblika

- o Oblika **prostega spisa**.
- o Opisuje **namen uporabnika** v konkretnem primeru uporabe.
- o Opisuje **aktivnosti** uporabnika na **višjem nivoju** pri konkretnem primeru uporabe.
- o Nanašajo se na **ključne koncepte problemskega področja** primera uporabe.

Primer: plačilno položnice

Uporabnik lahko preko spletne banke izvaja plačila položnic. Pri plačilu lahko uporablja zneske v višini svojega limita. Zneski položnic se lahko nakazujejo na račune pravnih in fizičnih oseb.

# Oblika scenarija

- Opisuje neko **določeno pot** skozi primer uporabe, zapisan iz vidika uporabnika sistema.
- Opisuje **zaporedje dogodkov** oz. **seznam korakov**, ki jih je potrebno doseči.
- Vsak korak je **enostavna izjava brez vejanja**.
- Lahko opiše:
  - uporabnike in njihove namene,
  - aktivnosti in naloge sistema.

Primer: prijava v spletno banko

1. Uporabnike vnese podatke za prijavo (upo.ime, PIN, geslo).
2. Sistem preveri ali je geslo ustrezno.
3. Sistem preveri pravice uporabnika.
4. Glede na pravice, sistem ponudi uporabniku ustrezne opcije spletne banke.

# Oblika dialoga

- Ena pot skozi primer uporabe s poudarkom na **interakcijah** med uporabniki in sistemom.
- Lahko prikaže **opcijske** in **vzoredne** aktivnosti.
- Vsaka aktivnost se lahko opiše skozi en ali več **delnih korakov**.

- Lahko opiše:

- aktivnosti uporabe
- aktivnosti in naloge

Uporabnik	Sistem
	Ponudi predlogo splošne položnice.
V obrazec vnese podatke o prejemniku, znesku in vrsti plačila.	
	Doda novo plačilo na seznam plačil. Ponovno prikaže predlogo splošne položnice
Izbere pregled opravljenih transakcij	
	Pojdi na primer uporabe: pregled opravljenih transakcij.

# Primerjava treh oblik

Oblika	Prednosti	Pomanjkljivosti
Opisna	Dobra za povzetke in visoko-nivojski vpogled.	Previsoki ali prenizki nivo. Ni uporabna za zapletene opise. Lahko dvoumna.
Scenarij	Dober za prikaz zaporedja korakov.	Težko prikaže vzporednost, naključno urejenost ali izbirnost. Lahko je monoton.
Dialog	Dober za prikaz interakcij med uporabnikom in sistemom. Lahko prikaže vzporedne in opsijske aktivnosti.	Težko prikaže ponavljanja.
Vse oblike	Neformalna	Neformalna

# Prednosti primerov uporabe

- Primeri uporabe opisujejo sistem iz vidika zunanjega uporabnika.
- Lahko se uredijo glede na relevantnost, pogostost uporabe, pričakovano vrednost za uporabnika.
- Lastnosti sistema se lahko povežejo z uporabo v primerih uporabe.
- Lahko se analizira vpliv dodajanja/odvzemanja funkcionalnosti sistema.

# Razumevanje primerov uporabe

- Zajem informacij v naravni obliki.
  - Uporabnik: *“ali to pomeni, da bomo morali narediti to ...”*
- Odkrivanje “lukenj” v razumevanju sistema.
  - Naročnik: *“tukaj ste izpustili eno zadevo ...”*
- Organiziranje dela, ki ga podpira sistem.



# Primeri uporabe v praksi

- Primeri uporabe se lahko razlikujejo glede na **nivo abstrakcije** (*npr. za študij, za program, za predmet, ...*).
- Primeri uporabe se lahko razlikujejo glede na **obseg** (*na katero mejo sistema smo mislili – komponenta, aplikacija, organizacija?*).
- Primeri uporabe se lahko razlikujejo glede na **nivo podrobnosti** (*ali opisujemo splošne aktivnosti ali podrobnosti*).

# Kaj primeri uporabe ne obsegajo?

- Primeri uporabe opisujejo sistem iz vidika usmerjenosti na naloge in **ne obsegajo**:
  - opise uporabniških vmesnikov,
  - sistemske zmogljivosti,
  - sistemsko arhitekturo,
  - ostale ne-funkcijske zahteve.

# Iskanje primerov uporabe

- Opis funkcionalnosti, ki jih uporabnik želi od sistema.
- Opis aktivnosti, ki ustvarjajo, berejo, spreminjajo ali brišejo podatke.
- Opis aktivnosti kako uporabniki sporočajo sistemu informacije, ki jih ta potrebuje za delovanje.
- Opis aktivnosti kako so uporabniki obveščeni o spremembah notranjega stanja sistema.

# Imenovanje primerov uporabe

- Primer uporabe se poimenuje s frazo sestavljeno iz **glagola in samostalnika**, ki določata uporabnikov **cilj**.
- Uporabiti “močne” glagole namesto splošnih:
  - Močni: ustvari, združiti, izračunati, prenesti, sprejeti, zabeležiti, aktivirati, ...
  - Šibkejši: poročati, uporabiti, kopirati, organizirati, iskati, najti, vzdrževali, ...
- Biti ekspliciten. Uporabiti specifične izraze, le ti so močnejši:
  - Močnejši: lastnost, plačilo, račun, ...
  - Šibkejši: podatek, papir, poročilo, sistem, obrazec, ...

# Kreiranje primerov uporabe

Priporočen postopek [Cockburn]:

- določi akterje in njihove cilje.
- zapiši glavni (uspešni) tok podatkov.
- ugotovi seznam možnih neuspešnih tokov.
- opiši kako sistem ravna v primeru neuspešnih tokov.

# Pogoste napake

- Kompleksni diagrami. Po Cockburnu 5 nivojev:
  - za obseg načrtovanja:
    - organization black-box,
    - organization white-box,
    - **system black-box**,
    - system white-box,
    - component
  - za cilje:
    - very high summary,
    - summary,
    - **user goal**,
    - subfunction,
    - too low

# Pogoste napake

- Ni sistema.
- Ni akterja.
- Preveč podrobnosti (npr. o delovanju uporabniškega vmesnika).
- Podrobni ciljni nivo.

# Domača naloga

Šolski informacijski sistem uporabljajo učitelji, dijaki in vodstvo šole. Vodstvo šole se mora prijaviti in potem ima vpogled v vse ocene dijakov, lahko tudi objavlja okrožnice.

Dijaki imajo le možnost vpogleda v lastne podatke (ocene in izostanke) in pregled objav. Za pregledovanje ocen in izostankov se morajo prijaviti, objave so prosto dostopne.

Učitelji lahko vpisujejo ocene in izostanke. Šolski pravilnik opredeljuje, da se v nekaterih primerih po vnosu izostankov izrekajo tudi vzgojni ukrepi. Za vnos ocen in izostankov morajo biti učitelji prijavljeni, objave vodstva lahko pregledujejo prosto dostopno.