

3

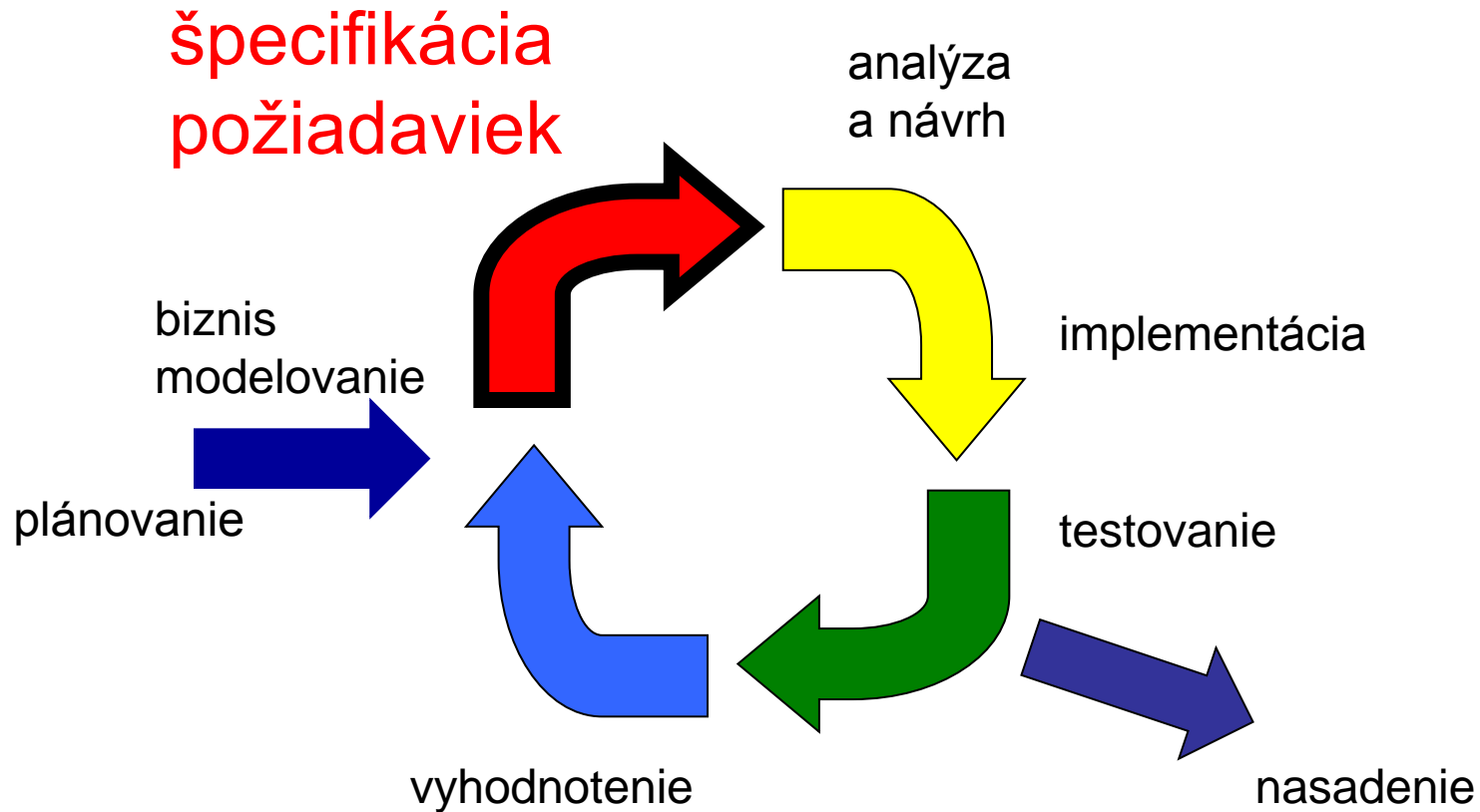
Špecifikácia požiadaviek

- Špecifikácia požiadaviek
- Delenie požiadaviek
- Spôsob ich špecifikácie
- UML a špecifikácia požiadaviek

RUP



Špecifikácia požiadaviek



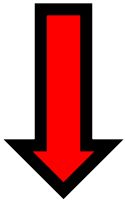
Špecifikácia (zber) požiadaviek

Skutočný systém



Biznis modelovanie

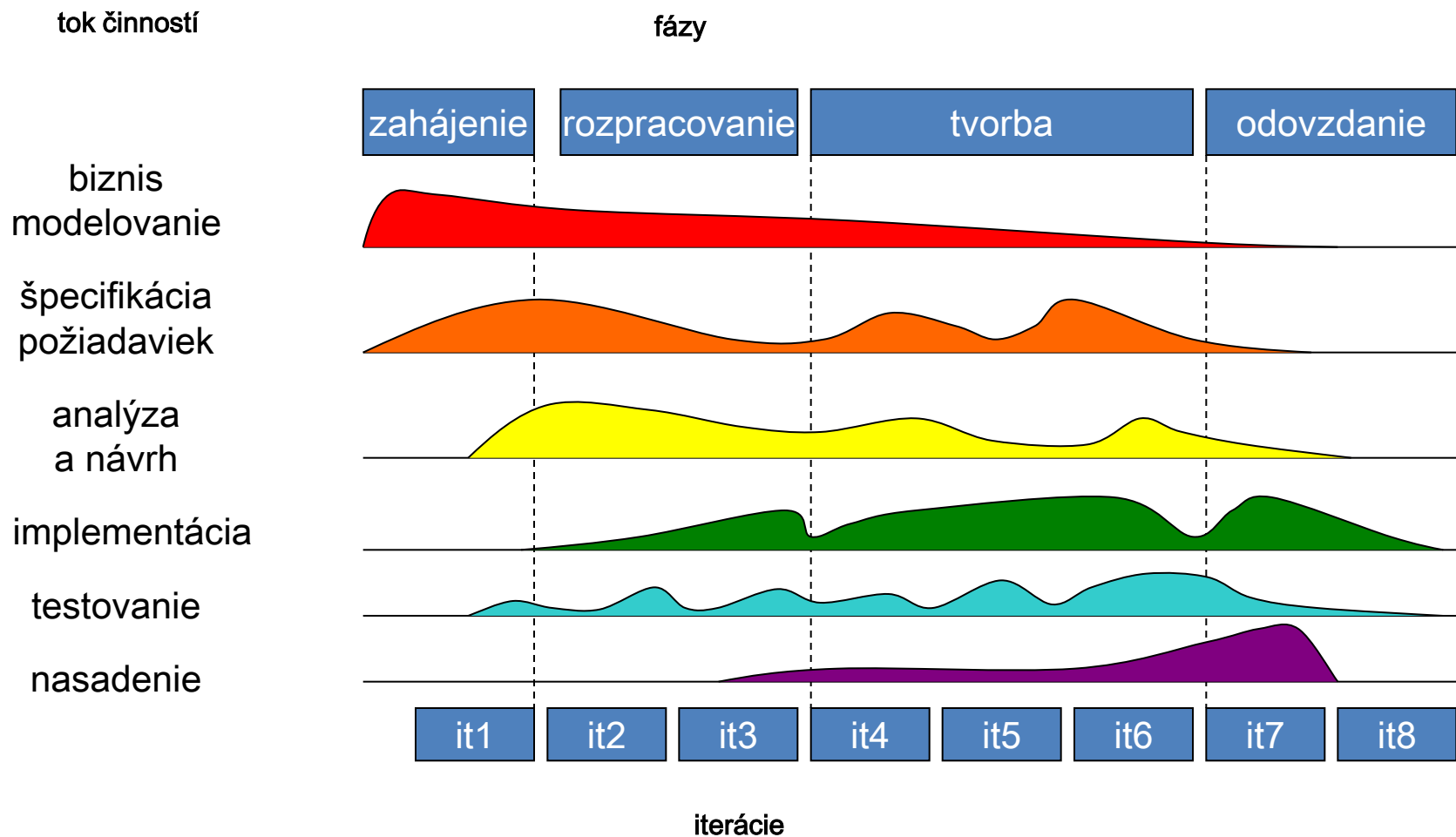
Model skutočného systému



Špecifikácia požiadaviek

Model softvéru (softvérového systému)

RUP – schéma (obsah x čas)



- určiť **ČO** má navrhovaný softvérový systém robiť (nie AKO)
- určiť funkcie systému
- iba terminológia z domény klienta
 - nepoužívať terminológiu IT v komunikácii s klientom
- vytvoriť zadanie projektu – základný dokument – spoločne odsúhlasený

- Funkčné
- Mimofunkčné
 - Často sú kritickejšie ako funkčné požiadavky
 - Niekedy sú dané vonkajšími faktormi (legislatívne požiadavky)
- Doménové
 - Môžu byť funkčné aj mimofunkčné

- Funkčné požiadavky
 - Užívatelia by mali mať možnosť prechádzať databázou objednávok
 - Systém by mal poskytovať užívateľom vhodné prehliadače na čítanie dokumentov v dátovom sklade
- Doménové požiadavky
 - Špecifikácia systému pre automatické spomalenie vlaku, ak prejde na červenú:
 - Spomalenie vlaku bude vypočítané ako $D = D_c + D_\Delta$, kde $D_\Delta = 9.81 \text{ ms}^2 \times (\text{gradient} / \alpha)$; hodnota α je známa pre rôzne typy vlakov

Príklad - mimofunkčné požiadavky

- Ak je riadiaci systém lietadla nespoľahlivý, je nepoužiteľný
- Celá komunikácia medzi užívateľom a systémom by mala byť vyjadriteľná v znakovnej sade ISO 8859-2
- Proces vývoja systému a všetky dokumenty majú zodpovedať softvérovému procesu definovanému v štandarde XYZ
- Systém nemá operátorom systému poskytovať žiadne osobné informácie o zákazníkoch okrem mena a čísla zákazníka
- Systém by mal byť ľahko použiteľný s minimom chýb
- Systém by mal byť vytvorený v jazyku ADA
- Maximálna veľkosť systému má byť 4 MB

Delenie – podľa úrovne popisu

- Užívateľská špecifikácia požiadaviek
 - Vysokoúrovňový popis funkčných a mimofunkčných požiadaviek zákazníka
 - Musia byť zrozumiteľné pre užívateľa
- Systémová špecifikácia požiadaviek
 - Podrobnejšia špecifikácia užívateľských požiadaviek pre vývojárov
 - Musí byť presná
 - Úlohy
 - Základ pre kontrakt medzi zákazníkom a dodávateľom
 - Počiatočný bod pre dizajn systému

Príklad

Užívateľská špecifikácia požiadaviek

1. Systém musí poskytnúť spôsob prezentácie externých dokumentov a možnosť ich prehliadania

Vedeli by ste to už naprogramovať?



Užívateľská špecifikácia požiadaviek

1. Systém musí poskytnúť spôsob prezentácie externých dokumentov a možnosť ich prehliadania

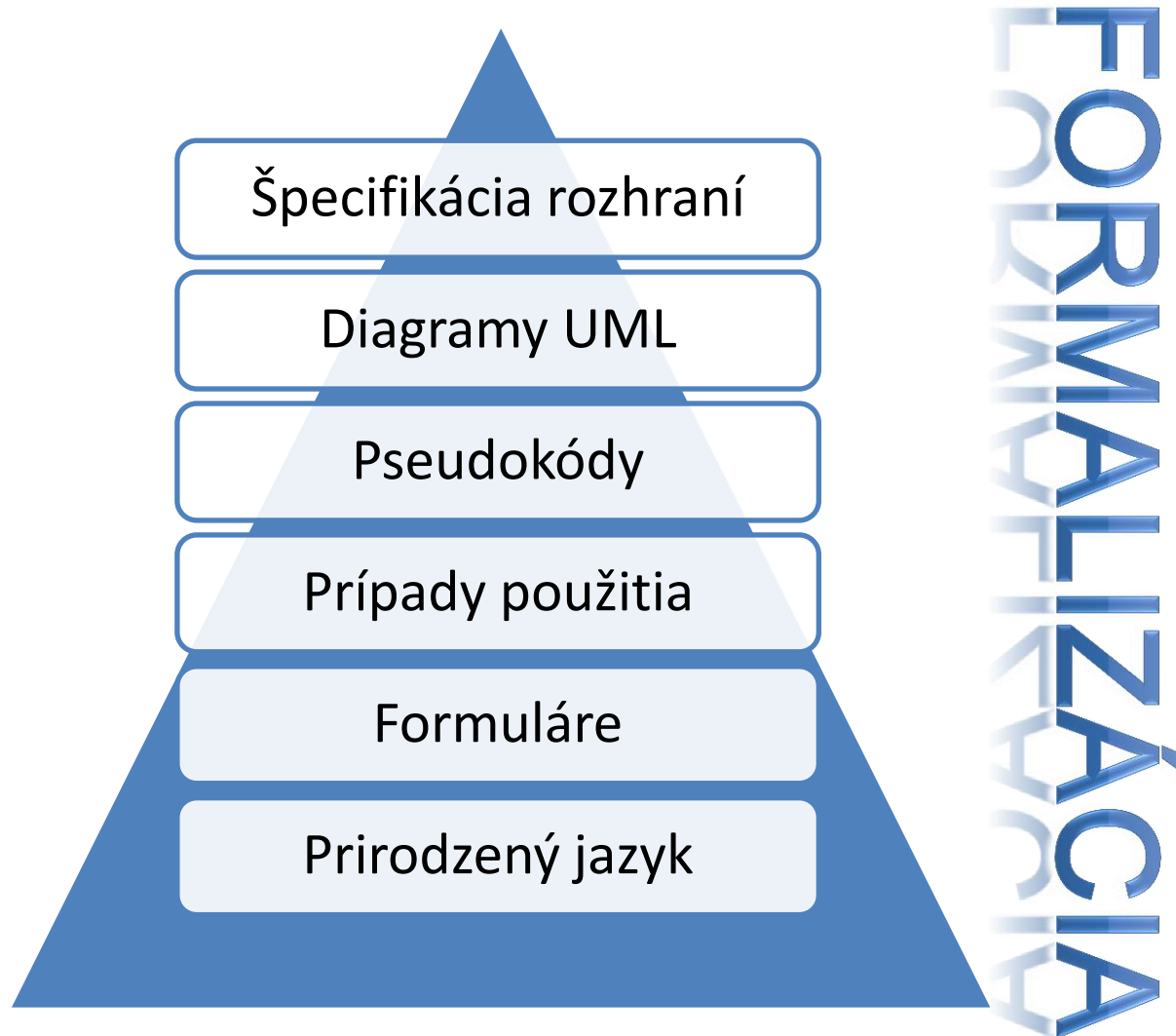
Systémová špecifikácia požiadaviek

- 1.1 Užívateľovi bude poskytnutá možnosť definovať typy dokumentov
- 1.2 Každý typ externého dokumentu bude na obrazovke reprezentovaný určitou ikonou
- 1.3 Užívateľovi bude poskytnutá možnosť definovať pre typ externého dokumentu vlastnú ikonu
- 1.4 Užívateľovi bude poskytnutá možnosť združiť typ externého dokumentu s prehliadačom
- 1.5 Pokiaľ si užívateľ vyberie ikonu reprezentujúcu externý dokument, výsledkom bude spustenie prehliadača združeného s typom externého dokumentu pre dokument reprezentovaný vybranou ikonou.

- Výsledok – dokument špecifikácie požiadaviek (DŠP)
- Dve formy výstupu
 - Zákazník – vysokoúrovňový popis požiadaviek (user requirements)
 - Vývojár – podrobná špecifikácia systému (system requirements)

- Externé správanie sa systému
- Jednoducho štruktúrovaný
- Obmedzenia implementácie
- Charakterizovať prijateľné odpovede na nežiaduce udalosti
- Zaznamenať predstavu o životnom cykle systému

Spôsob špecifikácie požiadaviek



Prirodzený jazyk

- Zrozumiteľný pre vývojára aj užívateľa, používaný aj napriek nevýhodám
- Nevýhody
 - Nejednoznačnosť popisu
 - Zložité koncepcie (algoritmy) sú ťažko opísateľné
 - Veľmi flexibilný (jedná vec popísaná viacerými spôsobmi)
 - Neexistencia jednoduchej modularizácie (update zmien)
 - Automatizácia procesov
- Nutnosť sa vyhýbať
 - Dlhým súvetiam s vedľajšími vetami
 - Termínom s niekoľkými prijateľnými významami
 - Vyjadrenie niekoľkých požiadaviek jednou požiadavkou
 - Nekonzistencii termínov (používanie synonym)

- Popis špecifikovanej funkcie alebo entity
- Popis vstupov (odkiaľ prichádzajú)
- Popis výstupov (kam smerujú)
- Aké ďalšie entity špecifikovaná funkcia alebo entita používa
- Prípadné vstupné a výstupné podmienky
- Ak vznikajú postranné efekty, tak aj ich popis

Príklad

Funkce: Vlož prvek do diagramu.

Popis: Vloží prvek do existujícího diagramu. Uživatel určí typ prvku a jeho pozici.

Vstupy: Typ prvku, Pozice prvku, Identifikátor diagramu.

Zdroje: Typ prvku a Pozici prvku zadá uživatel, Identifikátor diagramu získáme z databáze diagramů.

Výstupy: Identifikátor diagramu.

Úložiště: Databáze diagramů. Při dokončení operace je proveden COMMIT.

Vyžaduje: Diagram odpovídající vstupnímu Identifikátoru diagramu.

Vstupní podmínka: Diagram je otevřen a zobrazen na obrazovce uživatele.

Výstupní podmínka: Diagram je nezměněn kromě přidání prvku určeného typu na určenou pozici.

Vedlejší efekty: Nejsou.

Pseudokódy

- Jazyk s abstraktnými konštrukciami
- Lepšie vyjadrenie vnorených podmienok a cyklov

Příklad (část popisu činnosti bankomatu):

```
-----  
Přečti kartu  
Vypiš výzvu: "Prosím zadejte PIN"  
Přečti zadané_PIN  
Opakuj nejvýše 3x  
    Přečti zadané_PIN  
    Jestliže zadané_PIN je PIN_karty pak opust' smyčku  
Jestliže zadané_PIN není PIN_karty pak ...  
-----
```

Špecifikácia rozhraní

- Ak má systém komunikovať s inými systémami musí byť špecifikované komunikačné a softvérové rozhranie
- Dva typy rozhraní, ktoré je nutné definovať
 - Procedurálne rozhranie
 - Popis predávaných dát
- Príklad špecifikácie procedurálneho rozhrania
 - Klasické jazyky – prototyp procedúry alebo funkcie, popis in/out parametrov, popis činnosti , a pod.
 - Objektové jazyky – všeobecný popis triedy, popis konštruktorov, popis metód , a pod.

**Ktorý UML diagram by ste použili na
popis funkčných požiadaviek?**



Diagramy UML

- Model prípadov použitia (Use Case Model)
- Používajú sa na popis kontextu systému a popis funkčných požiadaviek
- Základné prvky – aktori a prípady použitia



aktor (actor) – prvok okolia modelovaného systému (človek, hardvér, iný softvér. systém)

- body komunikácie systému a okolia



prípád použitia (use case) – základná funkcia systému z vonkajšieho pohľadu – z pohľadu klienta

Hľadanie aktorov a prípadov použitia

- Stretnutie ľudí (< 10) rôznych znalostí a skúseností (zákazníci a vývojári) – brainstorming
- **Identifikácia aktorov**
- **Identifikácia prípadov použitia**
- **Vytvorenie popisu pre každý prípad použitia**
- **Popis toku udalostí pre každý prípad použitia**
- *Štruktúrovanie prípadov použitia* → len v niektorých prípadoch
- *Identifikácia analytických tried* → len v niektorých prípadoch





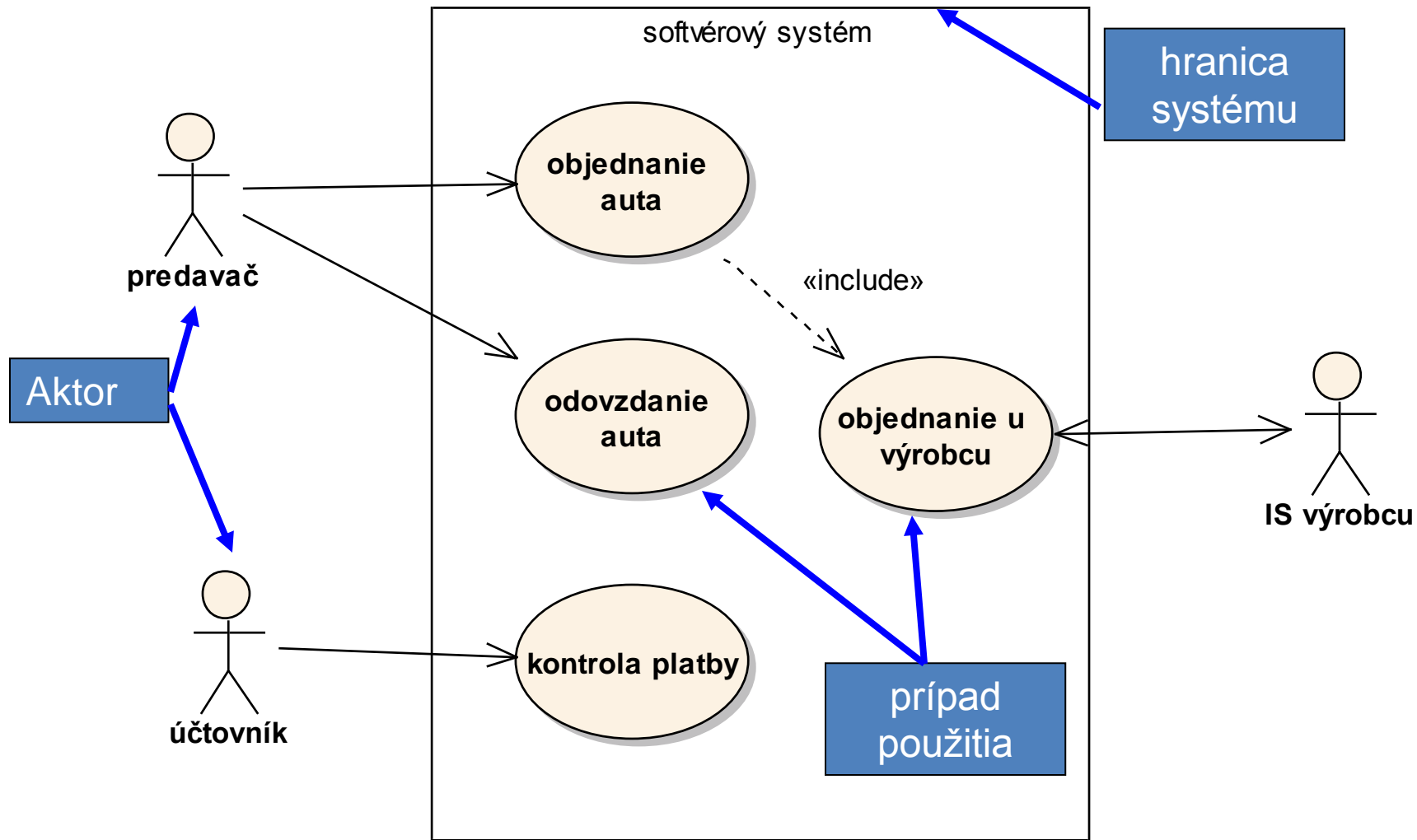
Postupnosť činností v komunikácii aktora so systémom

- Výsledok je niečo, čo predstavuje profit aktora
- Forma scenára
 - štruktúrovaný text
 - diagram UML
 - sekvenčný diagram, diagram spolupráce, diagram aktivít, stavový diagram

Diagramy

- diagram(y) prípadov použitia - UCD
- vizuálne prvky: aktori, prípady použitia a väzby (vzťahy) medzi nimi
- väzby alebo vzťahy medzi aktormi a prípadmi použitia

UCD - príklad



- **Aktor** – niečo mimo vlastný systém
 - používa systém
 - zadáva vstupy do systému
 - preberá výstupy zo systému
 - neriadi systém



- **Aktor** – kandidátom môže byť ten alebo to
 - priamo používa systém – používateľ
 - udržiava systém – administrátor
 - externý hardvér – snímač čipových kariet
 - iné (softvérové) spolupracujúce systémy



Ak softvérový systém v pravidelných intervaloch zálohuje systém, kto je aktor?



Identifikácia aktorov

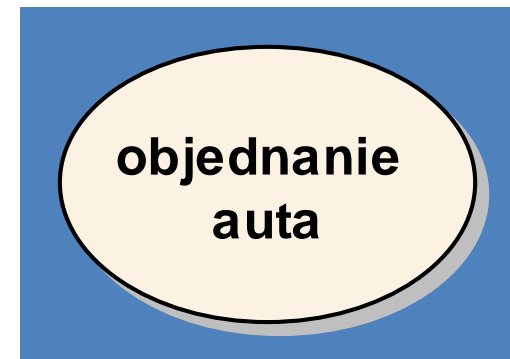
- Kto alebo čo bude systém používať?
- Začína sa s konkrétnymi ľuďmi → identifikácia úlohy, ktorú hrajú pri interakcii so systémom → mená aktorov
- Zaznamenávanie popisu – úlohu a zodpovednosť voči systému
- Aktormi sú aj externé systémy s ktorými komunikuje náš systém
- Neobmedzujeme a neštruktúrujeme aktora

Identifikácia aktorov

- Kto bude systém používať?
- Z akých systémov bude systém prijímať informácie?
- Do akých systémov bude systém informácie posielat'?
- Kto spúšťa systém?
- Kto udržiava informácie o užívateľoch? a pod.

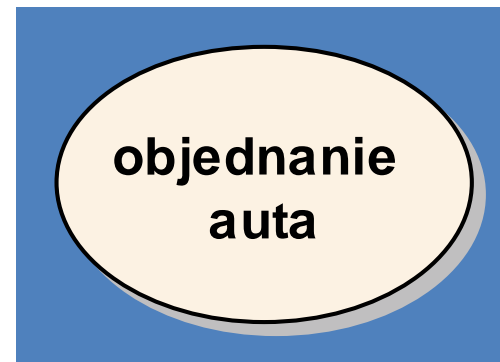
UCD – prípad použitia ⁽¹⁾

- **Prípad použitia** – popisuje určitý spôsob použitia systému z pohľadu aktora
 - vyjadruje chovanie systému
 - popisuje postupnosť správ medzi aktorom a systémom
 - poskytuje aktorovi určitý výsledok (hodnotu)

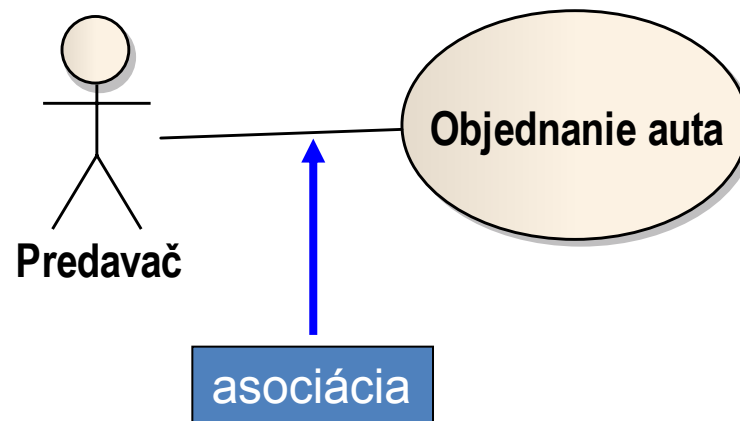


UCD – prípad použitia ⁽²⁾

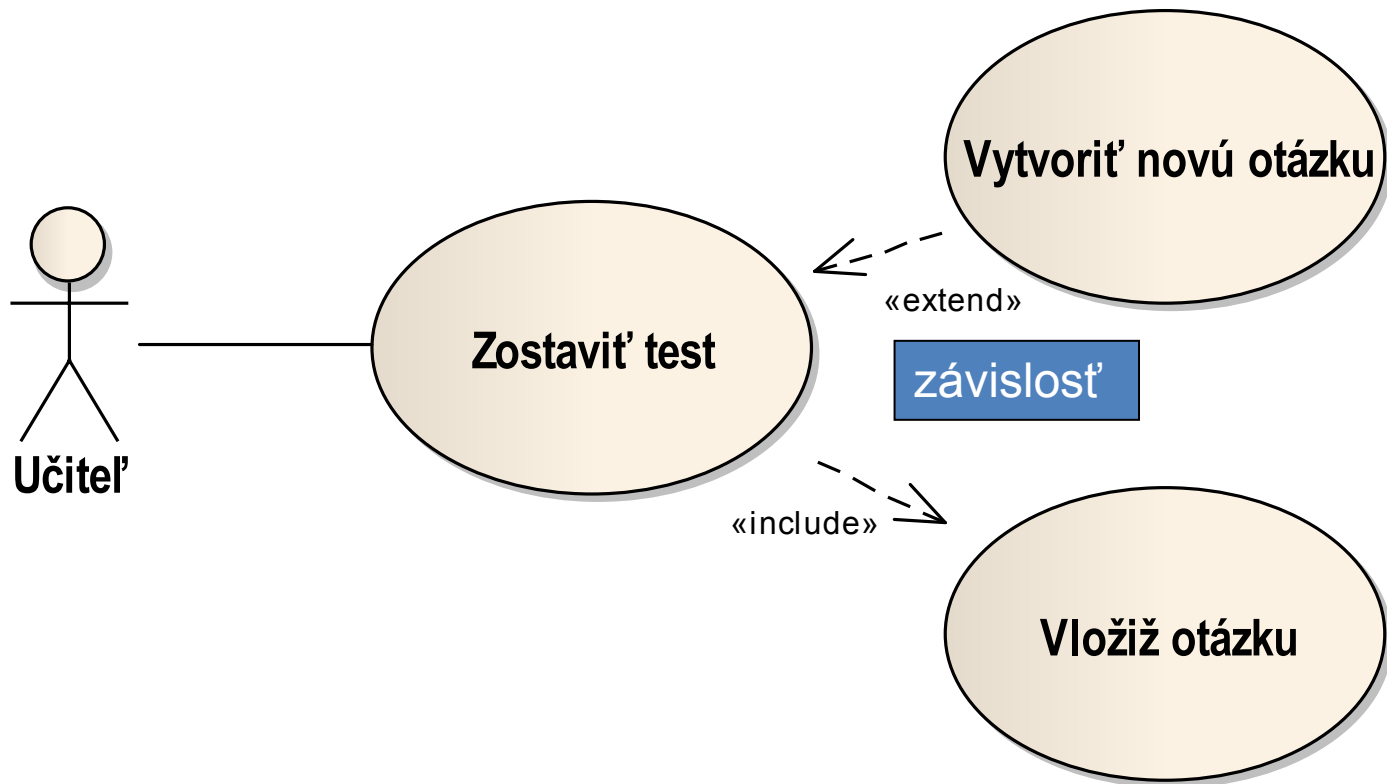
- **Prípad použitia** – je prostriedok na
 - vyjadrenie požiadaviek na systém
 - uľahčenie komunikácie s klientom
 - komunikáciu medzi vývojármi
 - testovanie systému
 - určovanie ceny
 - štruktúrovanie systému



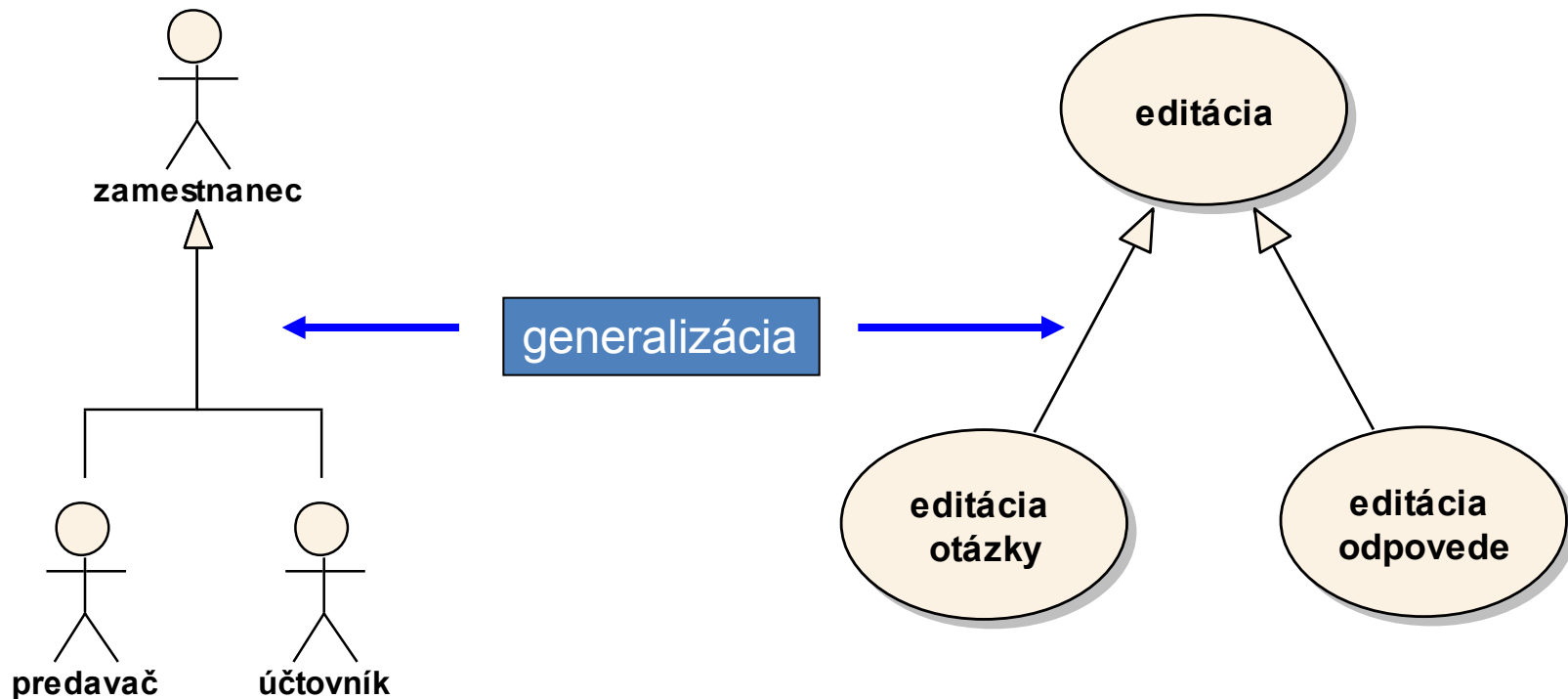
- **vzťahy** – medzi prvkami modelu prípadov použitia



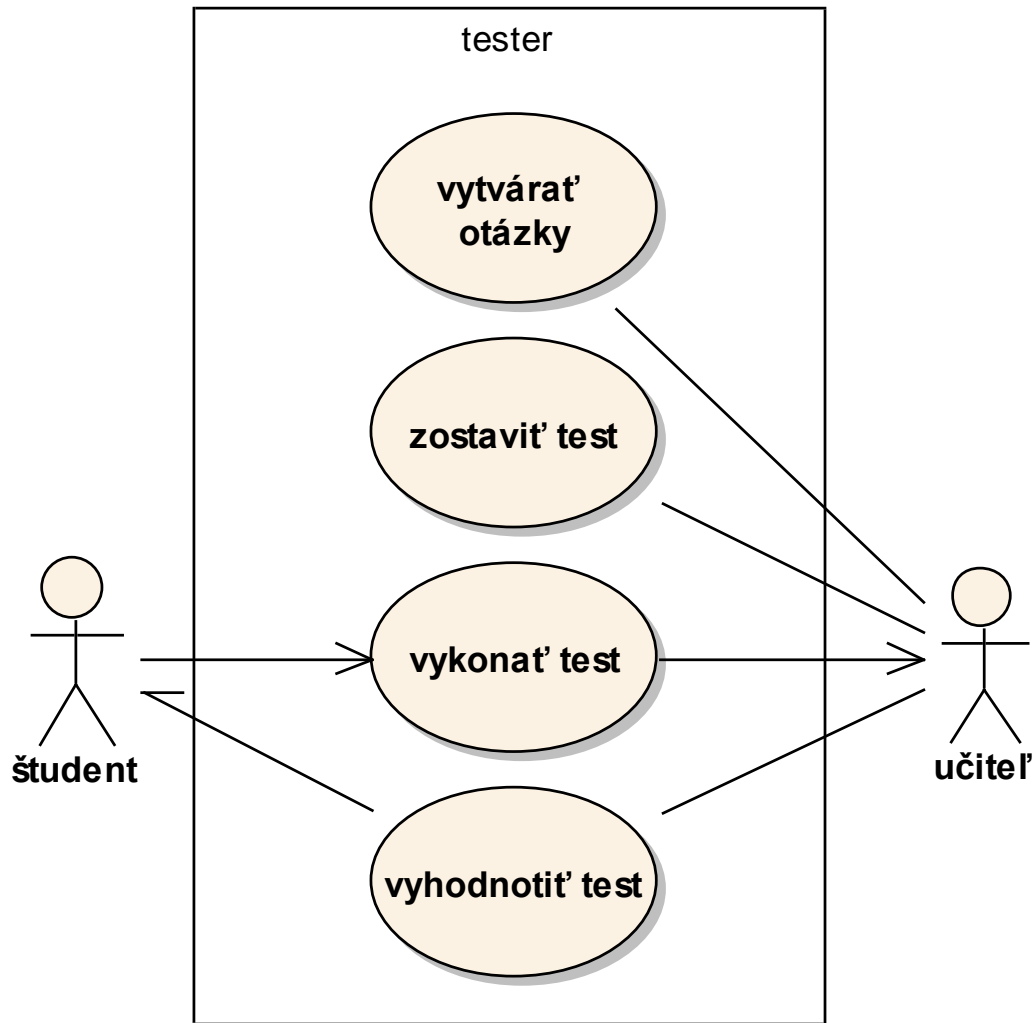
- **vzťahy** – medzi prvkami modelu prípadov použitia



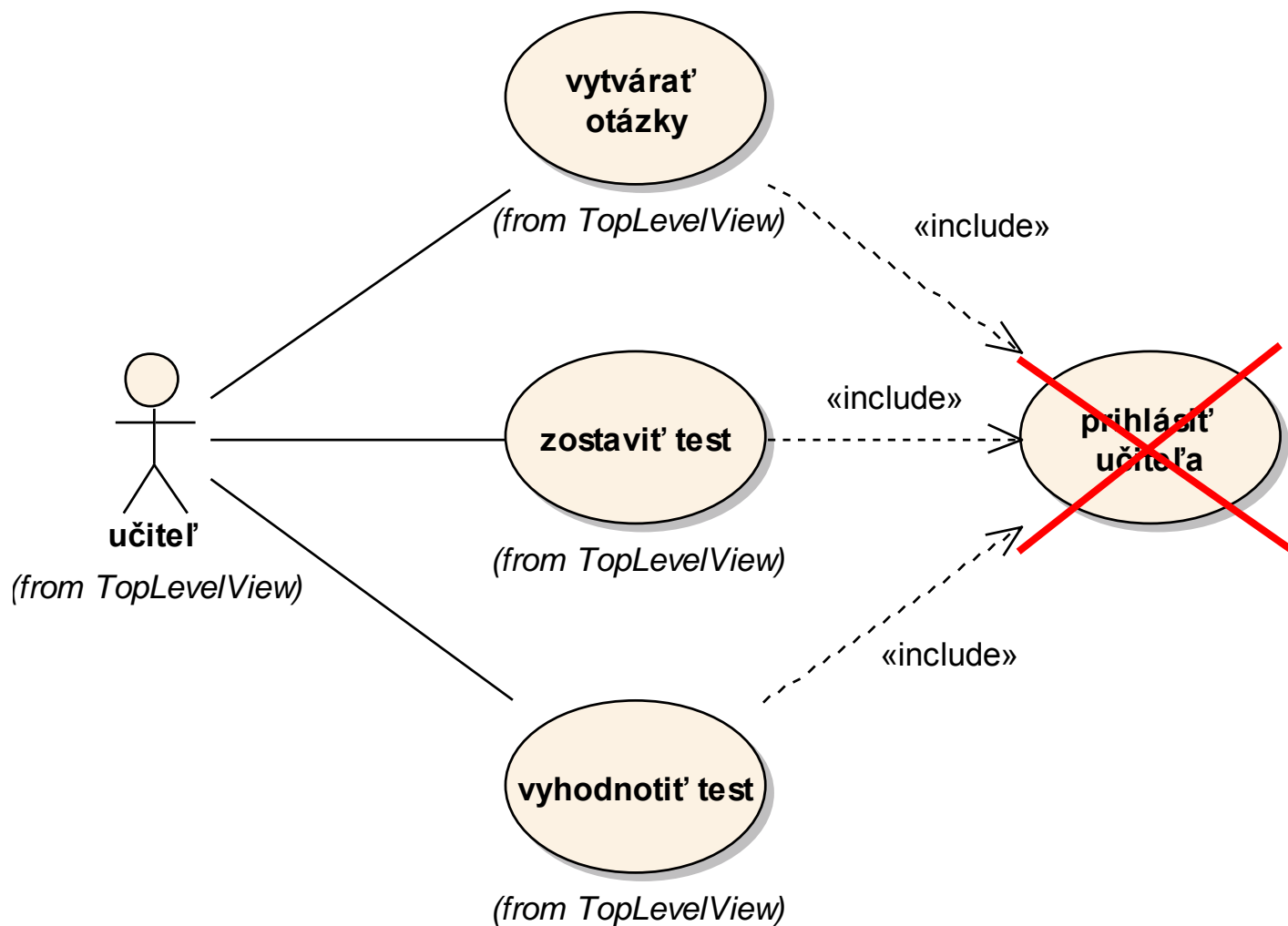
- **vzťahy** – medzi prvkami modelu prípadov použitia



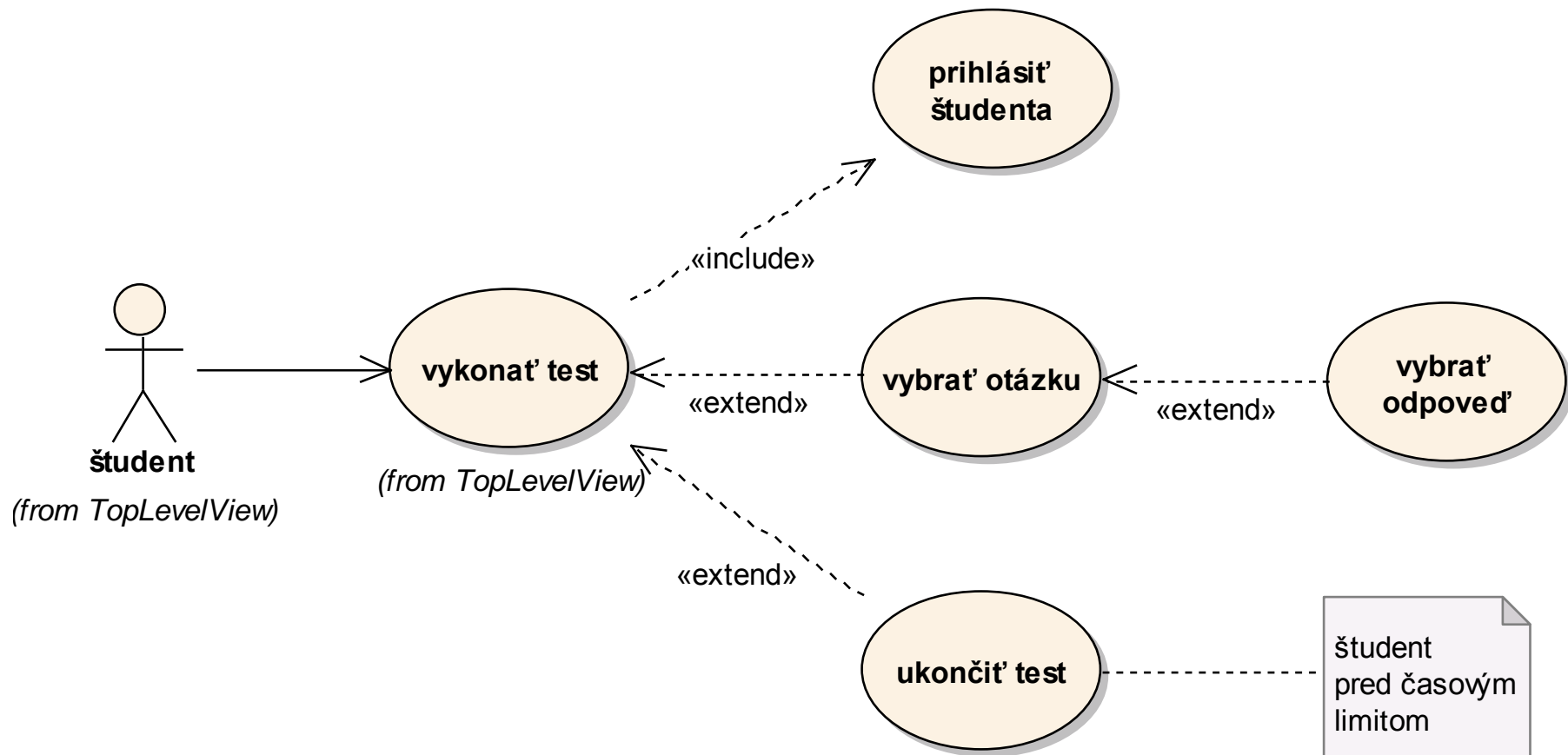
Príklad - tester



Tester - učiteľ



Tester - študent



- Čo je to špecifikácia požiadaviek a rozdiel oproti biznis modelovaniu?
- Typy požiadaviek
- Ako ich špecifikovať?

Ďakujem za pozornosť

Vaše otázky...

