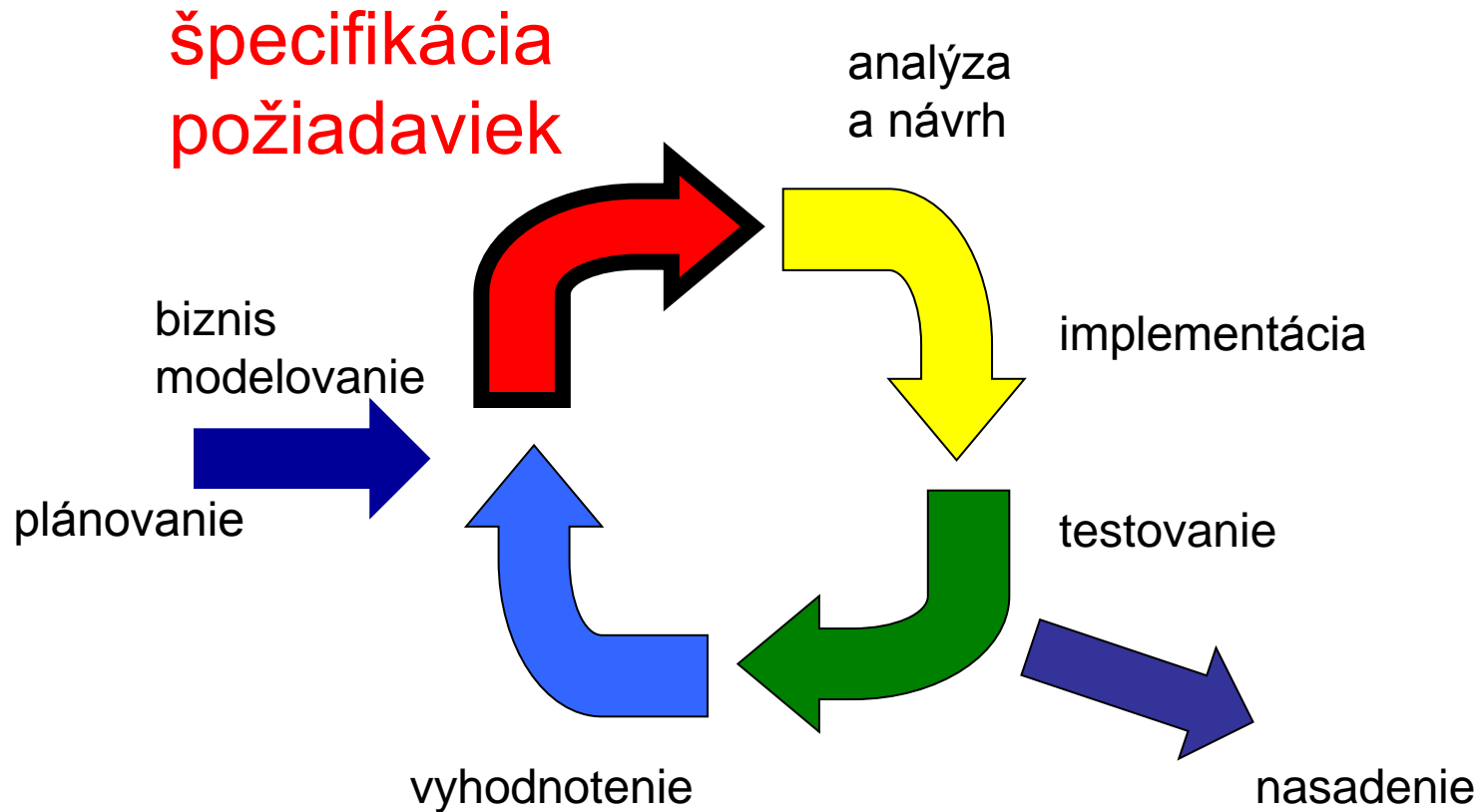


# 4

## Scenáře

# Špecifikácia požiadaviek



- 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

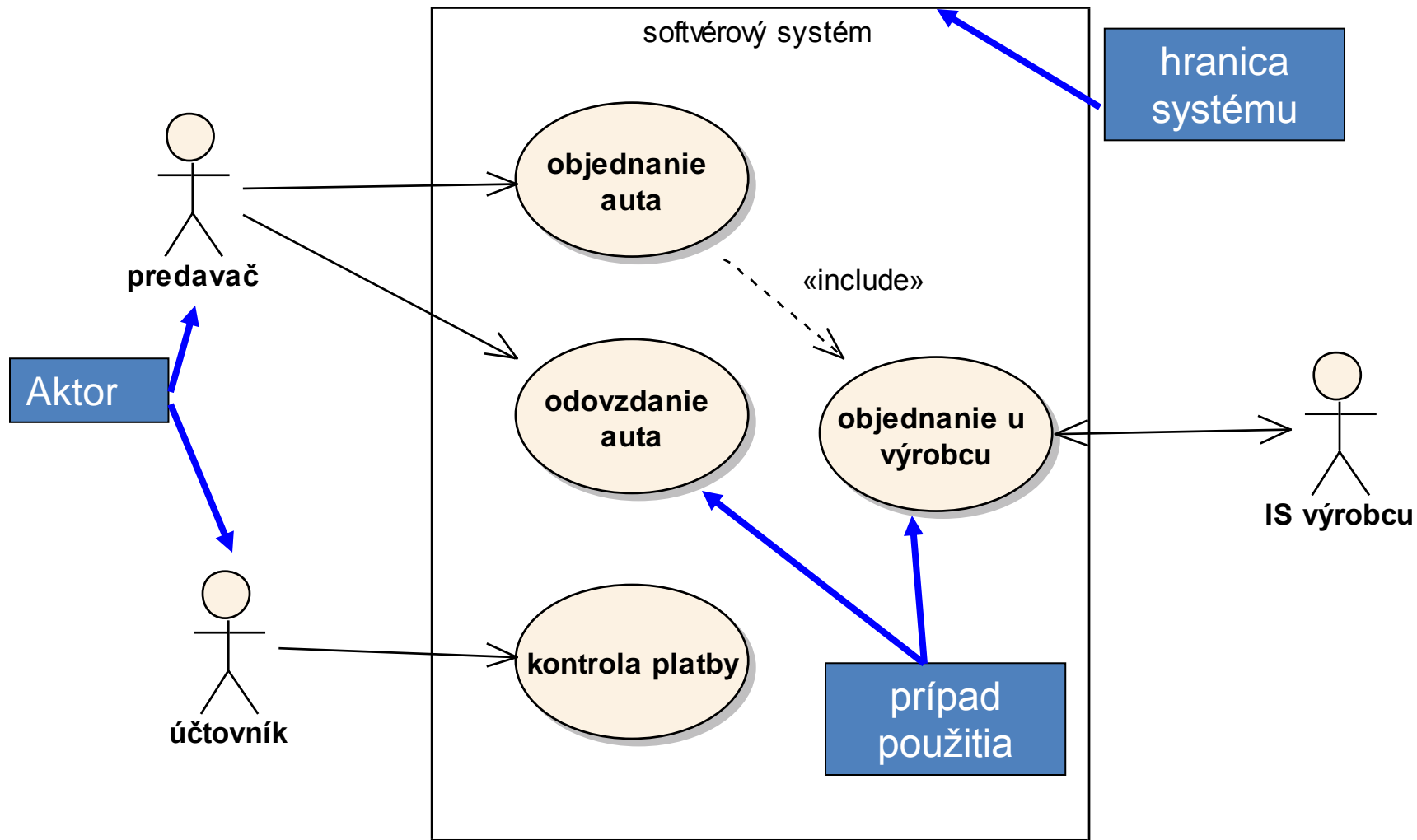
# Spôsob špecifikácie požiadaviek

- Prirodzený jazyk
- Formuláre
- Prípady použitia
- Pseudokódy a špecifikácie rozhraní
- Diagramy UML

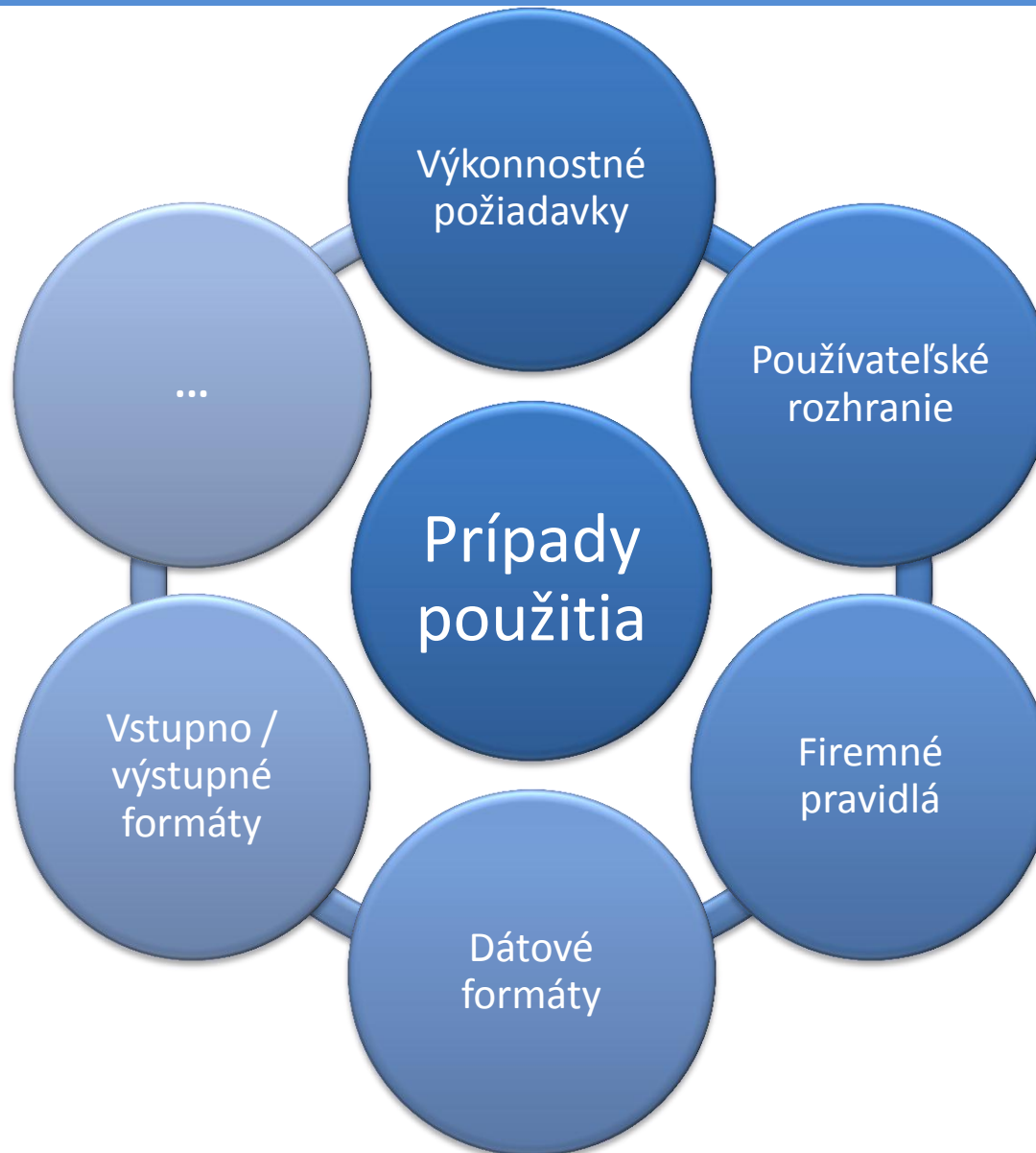
# Diagramy

- 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



# Prípady použitia - význam



# Vytvorenie popisu pre každý prípad použitia

- Krátka notácia názvu prípadu použitia
- Stručný popis prípadu použitia (1 – 3 vety)
- Zobrazte aj aktérov spojených s prípadom použitia
- V tejto etape sa môžu objaviť nové prípady použitia a staré môžu zaniknúť
- Popis toku udalostí pre každý prípad použitia
- Tvorba doplnkovej špecifikácie





Postupnosť činností v komunikácii aktora so systémom

- Forma scenára
  - štruktúrovaný text
  - diagram UML
    - sekvenčný diagram, diagram spolupráce, diagram aktivít, stavový diagram
- **Postupnosť činností v komunikácii aktora so systémom**

# Scenáre prípadov použitia

- každý UC má súbor scenárov
- ďalšie náležitosti scenára: aktori, podmienky uplatnenia UC
- jedna strana A4

# Náležitosti (schéma) textového scenára

- ID a názov scenára
- Stručný popis scenára
- Aktor, ktorý inicializuje UC
- A priori podmienky pre UC
- Kroky v scenári – činnosti aktora a systému
- A posteriori podmienky UC
- Aktor, ktorý dostane výsledok UC

# Podmienky



**A priori podmienky** – vstupné podmienky, ktoré musia byť splnené pred začiatkom vykonávania UC

- Stav systému, pred zahájením UC



**A posteriori podmienky** – výstupné podmienky, ktoré musia byť splnené po skončení vykonávania UC

- Stav, ktorý systém dosiahne po skončení UC
- Jednoduché výrazy, ktoré je možné vyhodnotiť
- Výsledok vyhodnotenia má tvar pravda/nepravda

# Kroky scenára – tok udalostí

- <Číslo kroku><kto/čo><akcia>
- Dobrý príklad:
  1. UC začína, keď zákazník zvolí „vyplniť objednávku“
  2. Systém zobrazí formulár objednávky
  3. Zákazník vyplní meno a priezvisko

# Nejasnosti v krokoch scenára

- Zlý príklad:

1. Sú zadávané údaje o zákazníkovi

Čo je na tom zlé?



# Nejasnosti v krokoch scenára

- Zlý príklad:

## 1. Sú zadávané údaje o zákazníkovi

- ??? Kto zadáva údaje
  - ??? Aké údaje to sú
  - ??? Kam sú zadávané
- Nesmú zostať pochybnosti o vykonávaných krokoch
  - Nesmie sa vytvárať priestor pre rôzne interpretácie – zákazník vs. vývojár

# Vetvenie krokov scenára

- Niektoré kroky možno vykonať s podmienkou, ktorá sa dá vyhodnotiť ako pravda/nepravda
  1. UC začína, keď zákazník označí tovar v nákupnom košíku
  2. **AK** zákazník zadá „zmazať tovar“
    - 2.1 systém odstráni tovar z košíka
  3. **AK** zákazník určí nové množstvo tovaru
    - 3.1 systém zmení množstvo tovaru v košíku



# Hlavné a alternatívne kroky

- Niektoré kroky nemožno presne umiestniť – môžu sa urobiť v rôznych okamihoch
- Riešením je jeden hlavný tok udalostí a k nemu alternatívne kroky
- Alternatív môže byť viac
- Alternatívy sú za hlavným tokom
- Spoločné a priori podmienky
- Vlastné a posteriori podmienky

# Hlavné kroky – príklad

1. UC začína keď zákazník zvolí „zobraziť obsah košíka“
2. **AK** je košík prázdny
  - 2.1 Systém oznámi zákazníkovi, že košík neobsahuje žiadny tovar
  - 2.2 UC končí
3. Systém zobrazí zoznam všetkých tovarov v nákupnom košíku: ID tovaru, názov, množstvo a cenu

# Alternatívne kroky

- Alternatívne kroky 1:
  1. Zákazník môže kedykoľvek opustiť stránku nákupného košíka
- Výstupné podmienky:
- Alternatívne kroky 2:
  1. Zákazník môže kedykoľvek opustiť systém
- Výstupné podmienky:

# Opakovanie v krokoch scenára - PRE

1. UC začína, keď zákazník zvolí „nájst produkt“
2. Systém požiada zákazníka, aby vybral kritéria vyhľadávania
3. Zákazník urobí voľbu kritérií
4. Systém vyhľadá výrobky vyhovujúce kritériám
5. **AK** systém nájde vyhovujúce výrobky
  - 5.1 **PRE** každý nájdený výrobok systém zobrazí
    - 5.1.1 obrázok výrobku
    - 5.1.2 podrobnosti o výrobku
    - 5.1.3 cenu výrobku

## 6. KÝM zákazník prezerá zobrazené informácie

6.1 systém prehráva hudbu

6.2 systém zobrazuje reklamu v pruhu reklamy

# Príklad – tester

- popis
  - učiteľ vytvára, mení alebo ruší otázky v katalógu otázok
- predpoklady
  - učiteľ má systém na svojom PC
- aktor, ktorý inicializuje UC
  - učiteľ
- a priori podmienky pre UC
  - učiteľ je zaregistrovaný v systéme
  - učiteľ má prístupové práva do systému

- kroky
  1. učiteľ sa prihlási do systému
  2. učiteľ vyberie jednu z možností: vytvoriť novú otázku, zmeniť otázku alebo zrušiť otázku
  3. učiteľ pracuje s otázkami
  4. učiteľ ukončí prácu
  5. systém uloží novú verziu katalógu



- a posteriori podmienky UC
  - nová verzia katalógu otázok je uložená
- aktor, ktorý dostane výsledok UC
  - učiteľ

- popis
  - učiteľ zadá svoje identifikačné údaje a systém ich overí
- predpoklady
  - učiteľ má systém na svojom PC
- aktor, ktorý inicializuje UC
  - učiteľ
- a priori podmienky pre UC
  - učiteľ je zaregistrovaný v systéme
  - učiteľ má prístupové práva do systému

- kroky
  1. systém predloží učiteľovi prihlasovací formulár
  2. učiteľ vyplní svoje identifikačné údaje (meno a heslo) a odošle ich do systému
  3. systém preverí správnosť zadaných údajov učiteľa
  4. ak sú údaje správne, systém zobrazí hlavné okno aplikácie
  5. ak nie sú údaje správne, systém žiada nové vyplnenie alebo zrušenie práce

- a posteriori podmienky UC
  - učiteľ má prístup do systému alebo ukončil prácu
- aktor, ktorý dostane výsledok UC
  - učiteľ

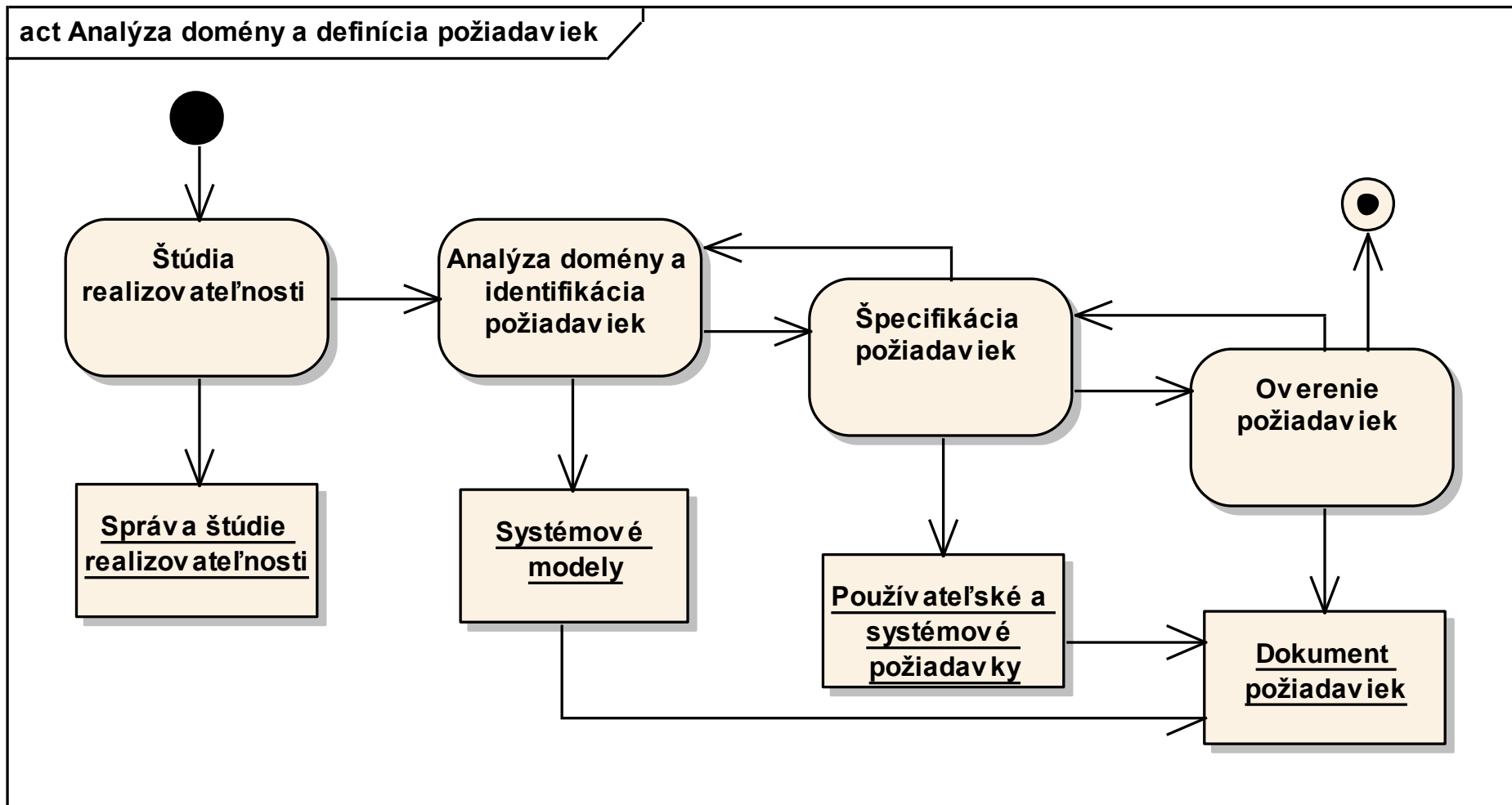
# Odporúčania

- Píšte zrozumiteľne
- Od menších podrobností k väčším
  - Názov aktora a jeho cieľ
  - Hlavný scenár
  - Alternatívne scenára
  - Kroky alternatívnych scenárov

# **Analýza domény a špecifikácia požiadaviek**



# Činnosti analýzy domény a ŠP



# Klasifikácia, detekcia riešenie konfliktov

- Neštruktúrovanú množinu požiadaviek sa logicky usporiada
- Požiadavky sa rozlišujú na
  - funkčné, mimofunkčné a doménové
  - užívateľské a systémové
  - je to potrebné pre tvorbu DŠP
- Detekcia a riešenie konfliktov medzi požiadavkami



# Detekcia a riešenie konfliktov

- Príklad konfliktu
  - Dvaja užívatelia vyžadujú nezlučiteľné vlastnosti
  - Rozpor medzi požadovanými schopnosťami a danými obmedzeniami
- Konflikt by nemali riešiť vývojári
- Rozhodnutie o konflikte by malo byť sledovateľné až ku konkrétnej osobe (zástupca zadávateľa) (vid' správa požiadaviek)

# Validácia požiadaviek

- Vstup – dokument špecifikácie požiadaviek
- Platnosť zmenených požiadaviek
- Konzistencia
- Úplnosť požiadaviek
- Kontrola realizovateľnosti
- Overiteľnosť
- Sledovateľnosť pôvodu požiadavky

# Metódy validácie



- Preskúmanie (reviews)
  - Manuálna tímová kontrola požiadaviek (od zákazníka kontraktora)
  - Formy preskúmania
    - Formálne preskúmanie DŠP – vývojový tím vysvetľuje zákazníkovi dôsledky každej požiadavky
    - Neformálne – diskusia o požiadavkách so zástupcami zákazníka
- Generovanie testovacích prípadov
  - Tvorba testov požiadaviek – časté odhaľovanie problémov
  - Ak je ťažké vytvoriť test – ťažká implementácia požiadavky



- Prototypovanie
  - Predvedenie spustiteľného modelu zákazníkovi – zistenie či zodpovedá jeho požiadavkám
  - Pomocou prototypu zákazník najlepšie pochopí správanie sa užívateľského rozhrania
- Automatická analýza konzistencie
  - Ak sú požiadavky vo forme modelu (formálna alebo štruktúrovaná notácia) – možná automatická kontrola konzistencie

# Správa požiadaviek



- Proces riadenia zmien systémových požiadaviek
- Požiadavky z hľadiska vývoja:
    - Trvalé
    - Nestále
  - Plánovanie správy požiadaviek stanovuje:
    - Spôsob identifikácie požiadaviek
    - Proces zmeny požiadaviek
    - Sledovateľnosť
    - Nástroje na uchovávanie informácií o požiadavkách

# Proces zmeny požiadaviek

- Analýza problému a špecifikácia zmeny
  - Identifikácia problému alebo návrh na zmenu požiadavky
  - Zisťovanie platnosti problému alebo zmeny
  - Výsledok – podrobnejší návrh zmeny
- Analýza zmeny a určenie jej ceny
  - Určenie, akú zmenu DŠP alebo dizajnu je potrebné realizovať
  - Odhad ceny zmeny alebo nového termínu dokončenia
  - Rozhodnutie o pokračovaní v procese zmeny
- Implementácia zmeny

# Sledovateľnosť požiadaviek (traceability)

- Definuje schopnosť sledovať požiadavky
- Nástroj – matica závislostí požiadaviek
  - Závislosť medzi požiadavkou v riadku od požiadavky v stĺpci
  - U (uses) – požiadavka v riadku používa možnosti dané požiadavkou v stĺpci
  - R (relates) – slabší vzťah, napr. obe požiadavky sú súčasťou rovnakého pod systému

Id pož.	1.1	1.2	1.3	1.4	1.5
1.1	.	U	R	.	.
1.2	.	.	U	.	.
1.3	R	.	.	.	.

# Matica závislostí - EA

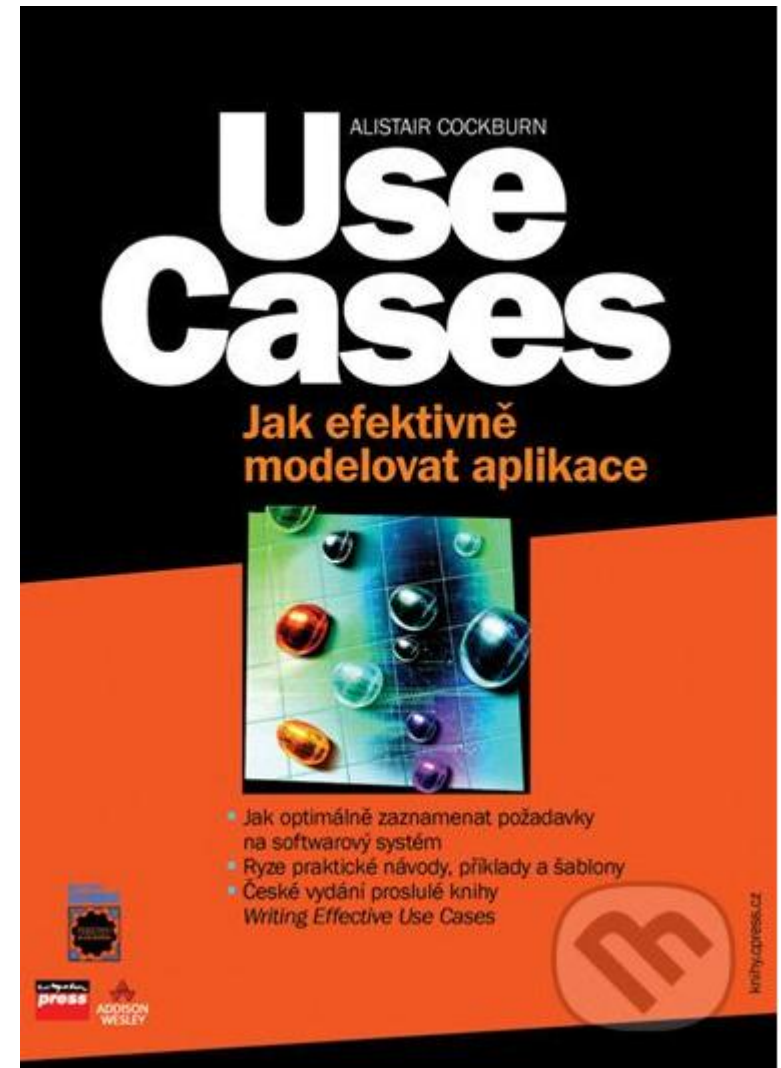
	Monitorovanie systému::001: Lokálne monitorovanie	Monitorovanie systému::002: Centrálné monitorovanie	Monitorovanie systému::003: Automatický reštart GTN Servera	Správa dát::201: Archivácia logov	Správa dát::202: Import dát	Správa dát::203: Zmeniť systém zálohovania	Správa dát::204: Čistenie databáz	Správa systému::101: Automatizácia nasadzovania systému	Správa systému::102: Konfigurácia častí systému
Užívateľské požiadavky::Bezpečný a spoľahlivý		X							
Užívateľské požiadavky::Hromadná realizácia				X					
Užívateľské požiadavky::Jednoduchá lokalizácia						X			
Užívateľské požiadavky::Možnosti rozširovania									



- Čo je to scenár a z akých častí sa skladá?
- Aké sú metódy validácie požiadaviek?
- Čo všetko zahŕňa správa požiadaviek?

- **Use Cases**

- Jak efektivně modelovat aplikace
- Alistair Cockburn



# Ďakujem za pozornosť

Vaše otázky...

