

Analýza a specifikace požadavků

- Modely jazyka UML používané při specifikaci požadavků
 - Diagram případů užití (Use Case Diagram – UCD)
 - Diagram aktivit (Activity Diagram)
 - Stavový diagram (State Diagram)

- **Analýza a specifikace požadavků (8 %),**
 - **Cíl: Stanovení služeb, které zákazník požaduje od systému, a vymezení podmínek jeho vývoje a provozu.**
 - Transformace neformálních požadavků uživatele do strukturovaného popisu požadavků,
 - Zdůraznění požadavků uživatele, ne jak toho docílit (realizovat),
 - Provedení studie vhodnosti, identifikace a analýza rizik,
 - Získávání, analýza, definování a specifikace požadavků,
 - Plánování akceptačního testování.

- • architektonický a podrobný návrh (7 %),
- • implementace (12 %),
- • integrace a testování (6 %),
- • provoz a údržba (67 %).

- Zainteresované strany
 - Pojem stakeholder
 - (původně) dočasný držitel peněz či majetku
 - Člověk nebo skupiny lidí, bez jejichž podpory by organizace přestala existovat
 - (obecně chápán jako) zainteresované strany v projektu – zákazník, uživatel, analytik, návrhář, tester, manažer, . . .
 - Je důležité
 - Zapojit nejen zákazníka, ale všechny zainteresované strany (stakeholders)
 - Tyto strany na začátku identifikovat – pokud analytik v průběhu tvorby požadavků zjistí, že existuje ještě někdo, kdo by se měl k něčemu vyjádřit, zdržuje to průběh projektu; pokud se to nezjistí, může být v požadavcích chyba

- Typy požadavků
 - Obchodní požadavky (Business Requirements)
 - Proč zákazník potřebuje systém
 - ⇒ pochopení a definování cílů a smyslu projektu
 - Zaměřeno na obchodní cíle (úspora nákladů, času)
 - Úspora nákladů při práci s chemikáliemi
 - Uživatelské požadavky (User Requirements)
 - Úlohy, které uživatel se systémem provádí
 - ⇒ co je možné se systémem dělat
 - Zjistit dostupnost chemikálie na skladě nebo v laboratoři
 - Use cases, . . .
 - Funkční požadavky (Functional Requirements)

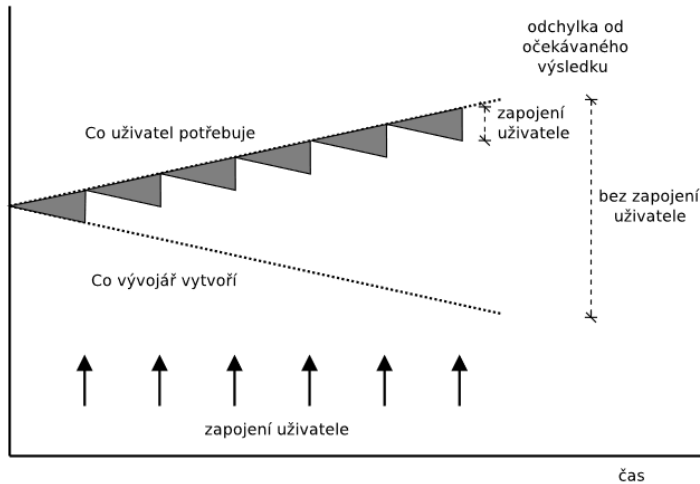
- Chování systému v různých podmínkách
 - ⇒ co musí být realizováno, aby mohly být vykonány úlohy (user requirements), a tím splněny obchodní požadavky (business requirements)
 - Co vše je potřeba pro zjištění dostupnosti chemikálie
 - Diagram aktivit, . . .
 - Nefunkční požadavky (Nonfunctional Requirements)
 - Vlastnosti a charakteristiky, které musí systém splňovat, a omezení, která musí respektovat
 - Požadavky na provoz systému
 - Statické – např. Počet uživatelů, . . .
 - Dynamické – např. Čas odezvy, počet transakcí na jednotku času, . . .
 - Požadavky na výsledný systém
 - Počítačové vybavení – např. HW náročnost (paměť, . . .)
 - Programové vybavení – např. Operační systém, programovací jazyky, . . .
 - Vyvíjený software – např. Efektivnost, spolehlivost, odolnost vůči chybám, přenositelnost, bezpečnost, . . .
 - Požadavky na vývojový proces
 - Dodržování norem
 - Odevzdání systému
 - Požadavky na rozhraní
 - Software → uživatel
 - Software → jiné součásti systému (HW, SW)
 - Externí požadavky
 - Legislativní požadavky (ochrana informací, . . .)
- Metody získávání informací
 - Interview (orientační, strukturované)
 - Základní běžná forma zjišťování potřeb zákazníka
 - Orientační – první setkání, získat základní přehled
 - Strukturované – připravené otázky, získat hlubší představu
 - Nejen naslouchat, ale navrhnout alternativy
 - Nevýhoda: Není možné uspokojit všechny stakeholdery (Interview nemůže být se všema)
 - Dotazníky
 - Lze obsáhnout velkou skupinu lidí
 - Na základě zkušeností připravené otázky s definovaným způsobem vyhodnocení
 - Pracovní setkání (workshop, elicitation meeting)
 - Skupina lidí (stakeholders) vyjednává o požadavcích a pracuje společně na specifikaci požadavků
 - Menší skupiny jsou efektivnější, různá setkání s různou skupinou lidí
 - Pozorování prací u zákazníka
 - Prostá specifikace nemusí být úplná, některé detaily nemusí být zachyceny, neboť jsou "zjevné" (nemusí pro každého)
 - Recept: "přidejte dvě nebo tři vejce"
 - Za jakých podmínek dvě vejce? Co skořápka?
 - Umožňuje lépe pochopit aktivity a procesy, ověřit získané informace, odhalit dosud neznámé informace
 - Časově náročné, není vhodné pro všechny projekty a uživatele
 - Další metody
 - Studium dokumentů
 - Přímá účast na pracích zákazníka

- Analýza existujícího softwarového systému
 - Kvalitní získávání informací o problémové oblasti a požadavcích snižuje riziko vytvoření systému, který nebude vyhovovat **potřebám** uživatele.
 - Důležitá je motivace ze strany zákazníka (uživatele).
 - Pro analytika jsou nutné komunikační schopnosti i zkušenosti.
- Problémy při specifikaci požadavků
- Přirozená neúplnost a nepřesnost
 - Nejasná a neúplná formulace požadavků zákazníkem
 - Neucelená představa uživatele o výsledném softwarovém systému
 - Problém rozhodování, jaké požadavky už nezačleňovat do specifikace
 - Pro komunikaci se používá přirozený jazyk, který je nejednoznačný
 - Nedostatek znalostí
 - Vývojář (analytik) se neorientuje v doménové problematice analyzované oblasti, nezná terminologii,
 - Specialista na doménovou oblast ve vývojovém týmu
 - Zákazník se neorientuje v problematice vývoje softwaru, nezná terminologii
 - Vyčleněný člověk od zákazníka (orientuje se ve vývoji, zaškolení, ...)
 - Nekonzistence požadavků
 - Různí uživatelé mají různé požadavky a priority
 - Různé požadavky uživatele a zákazníka (objednavatele)
 - Požadavky jsou mnohdy rozporné
 - Další problémy
 - Špatná predikovatelnost dopadu nového systému na organizaci, kde se nasadí
 - Otázka naplnění obchodních požadavků
 - Problémy s testováním a validací požadavků
 - Zapojení zákazníka
 - Prototypování, pravidelná setkání, . . .
 - Problémy plynou z použití přirozeného jazyka.
 - Vyřazení – Používají systém k výpůjčkám knih.
 - Kdo?
 - Deformace, zkreslení – čtenáři si nemohou půjčit další knihu, dokud nevrátí knihy s prošlou výpůjční lhůtou.
 - Když je zaplatí, tak mohou!
 - Zobecnění – Každý, kdo si chce vypůjčit knihu, musí mít průkazku.
 - A co výpůjčky mezi knihovnami?
 - Slovníček pojmů
 - Zachycuje obchodní jazyk (terminologii) pro daný projekt,
 - Řeší synonyma (slova téhož nebo podobného významu, např. účastník – aktér (v UML), směle – statečně, holka – dívka) výběrem nejčastějšího,

- Řeší homonyma (slova s odlišným významem znějící stejně, např. Pila, zámek, diskrétní) definicí jejich významu

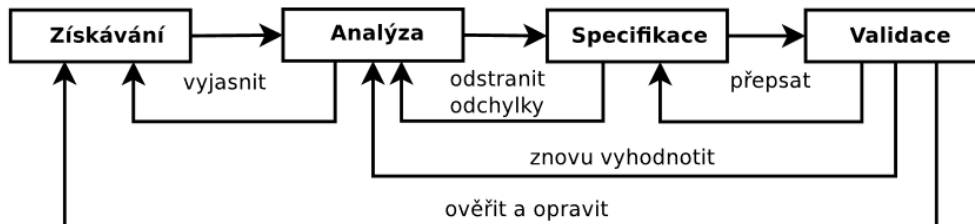
- Zapojení uživatelů

- Zapojení uživatele/zákazníka do procesu tvorby je důležité.



- Postup při specifikaci požadavků

- Tvorba požadavků je iterativní proces.

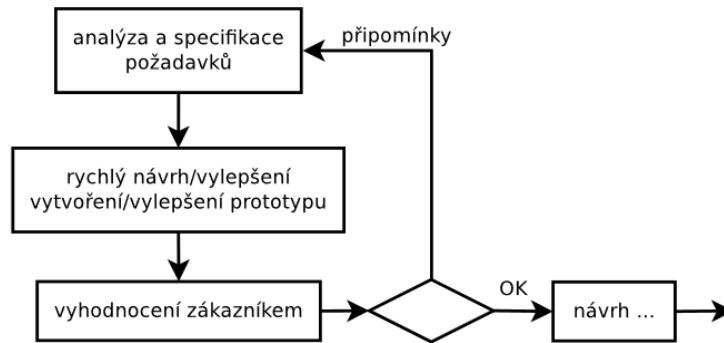


- **Získávání informací**
 - definice cílů projektu
 - identifikace uživatelských požadavků, . . .
 - interview, pozorování práce, . .
- **Analýza požadavků**
 - studie vhodnosti = odhad, zda je reálné vytvořit systém s danými vlastnostmi za daných podmínek; musí být provedena rychle a levně
 - zkoumání současného stavu
 - modelování, prototypování, . . .
- **Specifikace požadavků**
 - transformace informací z analýzy do dokumentu
 - specifikace nefunkčních požadavků, . . .
- **Validace požadavků**
 - vyhodnocení požadavků, simulování, prototypování, . . .
 - definování kritérií pro akceptování produktu

- Prototypování

- **Motivace**

- uživatelé mají problémy s přesnou specifikací svých požadavků, avšak dokáží lépe formulovat požadavky v reakci na práci s "reálným" produktem ⇒ prototypem
- prototypování dokáže ověřit správnost návrhu v počátečních fázích ⇒ validace požadavků (snižují se rizika projektu)
- může nastínit různé alternativy návrhu
- **Prototyp**
 - částečná implementace produktu
 - prototyp je většinou implementován rychle s cílem demonstrovat potenciální rozhraní či chování systému, není kladen důraz na kvalitu návrhu a programování



- **Problém dokončení prototypu**
 - není kladen důraz na kvalitu návrhu a implementace prototypu ⇒ prototyp je špatně udržovatelný
 - po vyjasnění specifikace a návrhu se prototyp dále nepoužívá
 - pokud se prototypy od začátku realizují plnohodnotně, lze je využít v implementaci (např. metodika RAD)

- **Dobrá specifikace požadavků**

- Měla by být:
 - **seřazená podle důležitosti**
 - poznačte si datum vytvoření požadavku
 - seskupte požadavky do tříd důležitosti
 - **sledovatelná**
 - původ (smysl) požadavku je jasný
 - na každý požadavek je možné se odkazovat v další dokumentaci
 - **modifikovatelná**
 - struktura a styl specifikace je konzistentní a bez redundancí
 - snadné úpravy a doplňování požadavků
 - **jednoznačná**
 - neumožňuje více interpretací
 - požadavky pište jasně a jednoznačně (přirozený jazyk je zdrojem nejednoznačnosti)
 - **úplná**
 - obsahuje všechny důležité požadavky a definice reakcí systému na všechny třídy vstupních údajů
 - specifikujte situace, ve kterých se porušuje akceptovatelné chování
 - Všechno důležité, ne všechno, co nás napadne
 - **konzistentní**

- požadavek není v rozporu s jinými požadavky
 - buďte konzistentní v používání názvů
 - **verifikovatelná**
 - existuje proces kontroly, zda software splňuje požadavek
 - měřitelnost splnění požadavků
- **Během procesu specifikace požadavků:**
- **Udržujte specifikaci čitelnou pro zákazníka.**
 - zákazník se musí umět orientovat ve specifikaci
 - **Ve specifikaci nenavrhněte řešení.**
 - cílem specifikace je získat úplný a správný pohled na potřeby zákazníka
 - realizace požadavků je záležitostí dalších etap
 - **Validujte požadavky.**
 - prototyp snižuje riziko špatného pochopení požadavků
 - slabá specifikace ⇒ špatný odhad nákladů
 - **Zainteresujte uživatele**
 - uživatel se musí podílet na procesu formování a validace požadavků
 - nechte si výsledek zkontrolovat a potvrdit druhou stranou
- Přibližný odhad nákladů na opravu chyb ve specifikaci

Etapa	Náklady (člověko-hodiny)
Specifikace	2
Návrh	5
Implementace	15
Akceptační testování	50
Údržba	150

- Dobře identifikované požadavky snižují cenu vývoje softwaru!

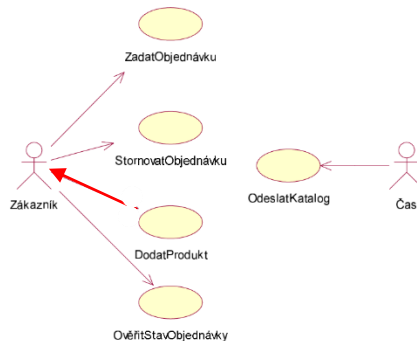
- Specifikace požadavků – dokumentace

- **Dokumentace spojená se specifikací požadavků**
 - různorodá, od textu až po formální specifikace
 - čím formálnější podoba, tím méně lidí je schopno tuto podobu akceptovat
 - praxe ukazuje, že pro mnoho softwarových projektů se používá kombinace strukturovaného jazyka, vizuálních modelů a dalších prezentačních technik (tabulky, matematické výrazy, . . .)
 - přiměřený rozsah dokumentace!
- **Formy dokumentace požadavků**
 - strukturovaný text
 - tabulky
 - vizuální modely
 - spustitelné modely
 - formální modely
 - . . .

- Specifikace požadavků – modelování

- **Modelování dat**
 - Entity Relationship Diagram (ERD) – strukturovaný model dat
 - Class Diagram – objektově orientovaný model dat a protokolu (rozhraní, zodpovědnosti)
- **Modelování funkčních požadavků**
 - Data Flow Diagram (DFD) – specifikace chování systému; strukturovaný přístup
 - Use Case Diagram (UCD) – diagram případů užití, specifikuje možnosti použití systému; jazyk UML (pro uživatelské požadavky, ale s kontextem i funkční)
 - UCD je doplněn dalšími modely z jazyka UML, např. Activity Diagram
- **Diagram případů užití**
 - **Use-case-driven approach (např. metodika RUP)**
 - Klíčovými aktivitami při specifikaci požadavků je nalezení účastníků a nalezení případů užití.
 - K tomu se využívají diagramy případů užití doplněné o detaily (specifikace) případů užití.
 - **Prvky diagramu užití**
 - hranice systému
 - účastník (aktor) – subjekt, který se systémem pracuje (může mít i speciální podobu, např. čas nebo jiný systém)
 - případ užití – funkce, kterou systém vykonává jménem jednotlivých účastníků nebo v jejich prospěch.
 - interakce – ukazuje účast aktora na provádění případu užití

Diagram případů užití



- **Detail (Specifikace) případu užití**
 - Konkretizace (specifikace) případu užití.
 - Neexistuje standard, většinou se však využívá tabulka.
 - Případ užití má svůj
 - název,
 - jedinečný identifikátor a
 - specifikaci.
 - Specifikace případu užití má:
 - vstupní podmínky,
 - tok událostí a

- následné podmínky.

název	Případ užití: Platit daň z přidané hodnoty
identifikátor	ID: UC1
účastníci	Účastníci: Čas finanční úřad
stav před	Vstupní podmínky: 1. Je konec fiskálního čtvrtletí?
kroky	Tok událostí: 1. Případ užití začíná na konci fiskálního čtvrtletí. 2. Systém určuje výši daně z přidané hodnoty, kterou je třeba odvést státu. 3. Systém odesílá elektronickou platbu finančnímu úřadu.
stav po	Následné podmínky: 1. Finanční úřad přijímá daň z přidané hodnoty.

- Alternativní toky:

Případ užití: Zobrazit košík	
ID: UC11	
Účastníci:	Zákazník
Vstupní podmínky:	1. Zákazník je přihlášen do systému.
Tok událostí:	1. Případ užití začíná volbou „zobrazit obsah košíku“. 2. KDYŽ je košík prázdný: 2.1 Systém oznámí Zákazníkovi, že košík neobsahuje žádné položky. 2.2 Případ užití končí. 3. Systém zobrazí seznam všech položek v nákupním košíku zákazníka včetně ID, názvu, množství a ceny každé položky.
Následné podmínky:	...
	...
Alternativní tok 1:	1. Zákazník může kdykoliv opustit obrazovku košíku.
Následné podmínky:	
Alternativní tok 2:	1. Zákazník může kdykoliv opustit systém.
Následné podmínky:	

- **Pokročilé techniky modelování případů užití**

- Pokročilé techniky:
 - zobecnění účastníka
 - zobecnění případu užití

- relace <<include>>
- relace <<extend>>
- Pokročilé techniky používejte pouze pro zjednodušení modelu.
- Příklad užití je způsob zápisu požadavků.
 - Musí být tedy čitelný i pro uživatele.
 - Je proto žádoucí, aby byl co nejjednodušší.
- Uživatelé těžko chápou zobecnění účastníka.
- Význam relace <<extend>> často nechápu ani analytici či návrháři.
- **Závěr:** Pokročilé techniky používejte co nejméně!

Diagram bez zobecnění účastníka

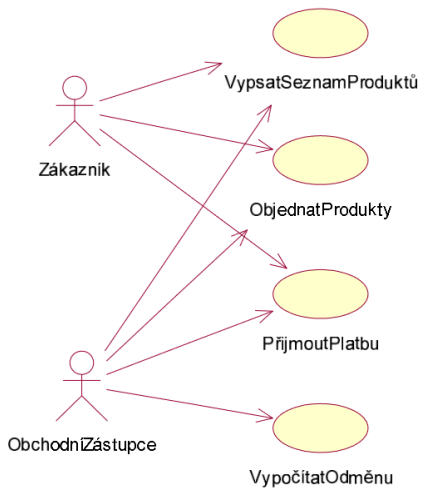
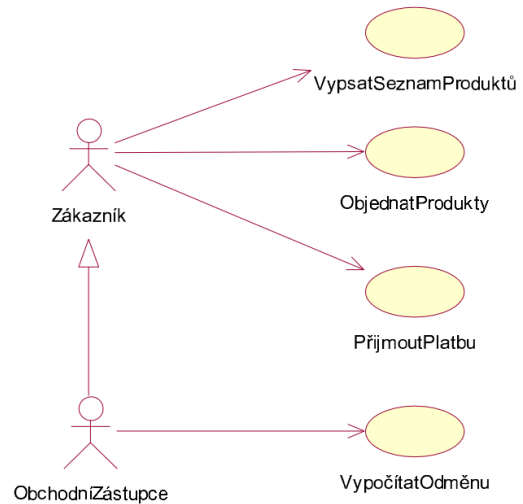
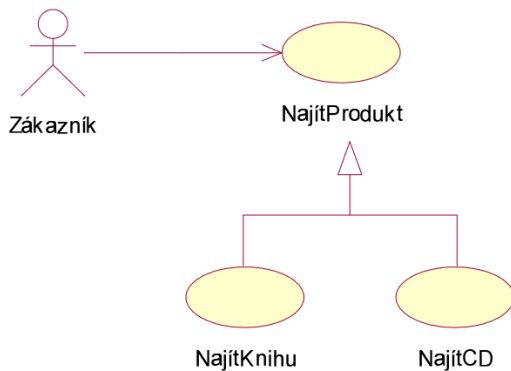


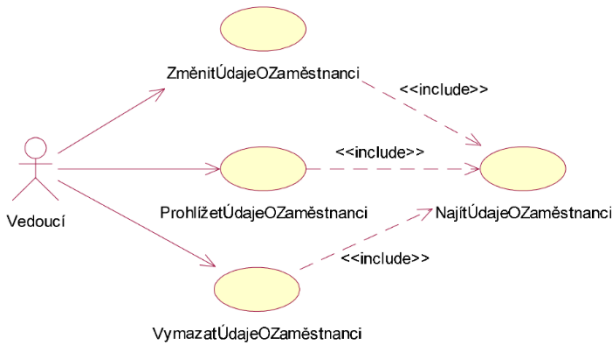
Diagram se zobecněním účastníka



Zobecnění případu užití

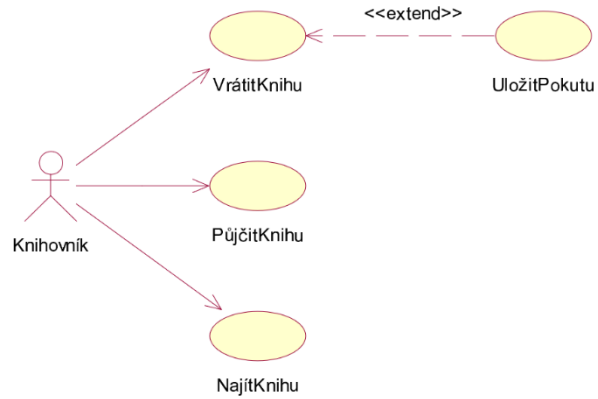


Relace <<include>>



Zevšeobecnění procesů, který se opakuje

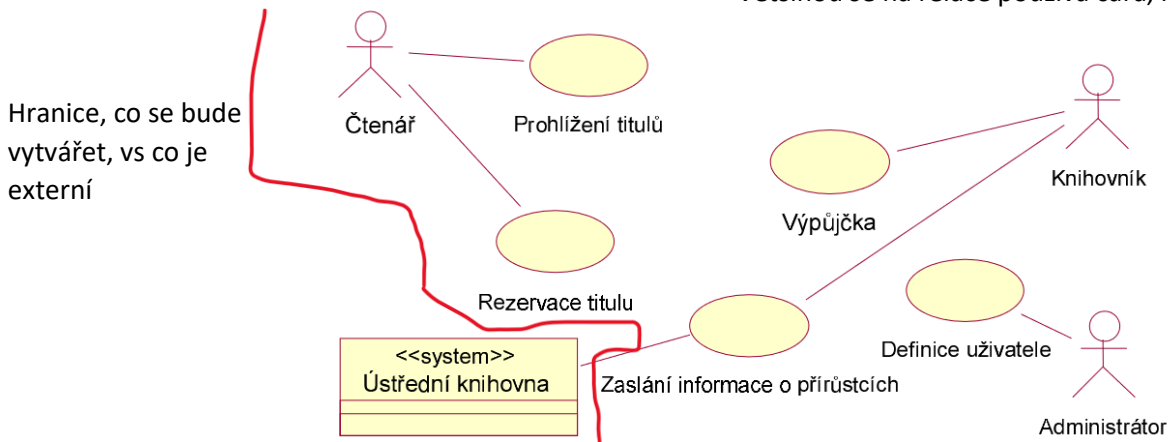
Relace <<extend>>



Rozšiřuje (extende) proces, nemusí být (extender) vyvolán

Diagram případů užití – alternativní notace

Většinou se na relace používá čára, ne šipka



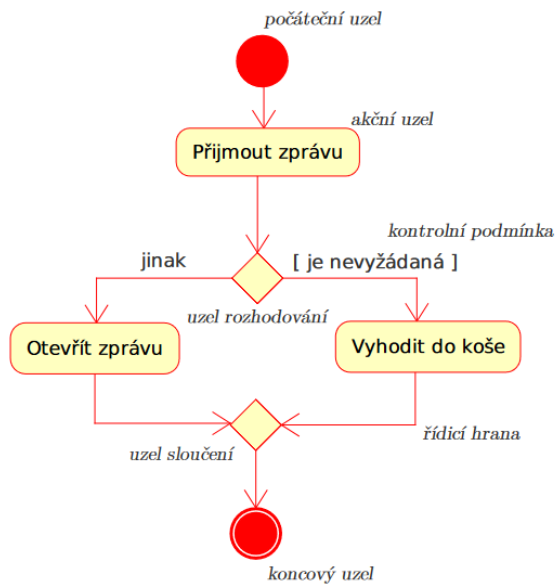
Hranice, co se bude vytvářet, vs co je externí

Tímhle obdélníkem zdůrazňujeme že se nejedná o člověka

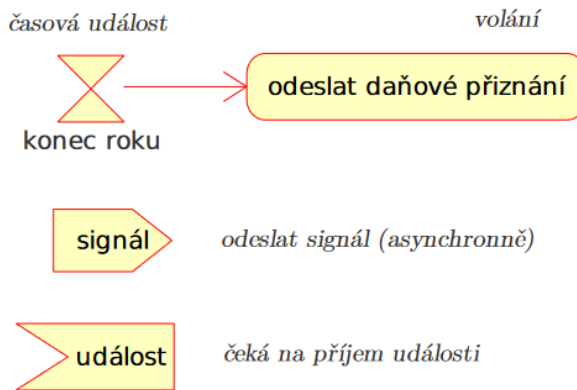
- Diagramy aktivit (Activity Diagrams)

- reprezentují objektově orientované vývojové diagramy
- modelování procesů, kterých se typicky účastní více objektů
- lze je připojit k libovolnému modelovanému elementu
 - případ užití
 - třída
 - komponenty
 - ...
- Využití diagramu aktivit
 - modelování scénářů případů užití
 - modelování detailů operace nebo algoritmů
 - ...
- Prvky diagramu
 - uzly
 - akční uzly: modelují aktivitu
 - řídicí uzly: modelují rozhodování; počáteční uzel; koncový uzel; . . .
 - objektové uzly: modelují objekty podílející se na aktivitách

- hrany
 - řídicí hrany: modelují přechody mezi uzly
 - objektové hrany: modelují cesty objektů mezi uzly
- Možnosti modelování
 - tok událostí (včetně toku dat)
 - rozhodování
 - větvení a spojení
 - iterace
 - paralelní toky
- Příklad:

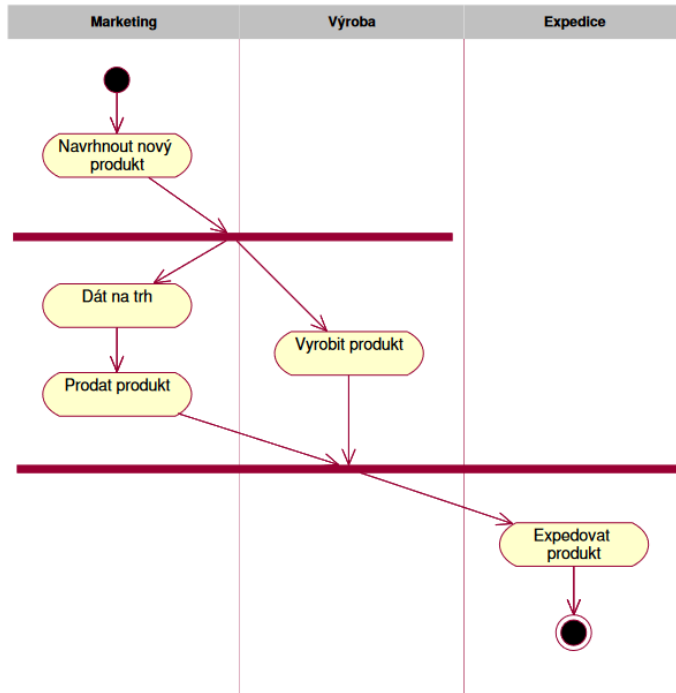


- Speciální akční uzly:



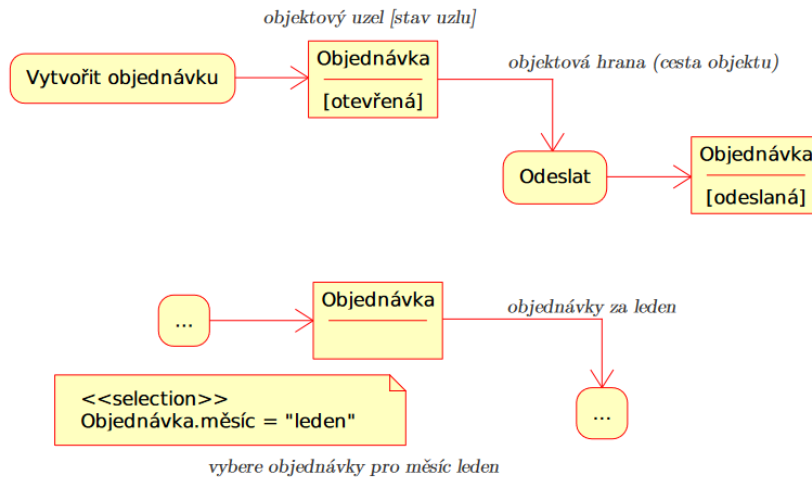
- Možnost modelování paralelních aktivit

- rozvětvení, spojení
- oddíly aktivit



-

- objektové uzly a hrany
- stav objektu, výběr objektů



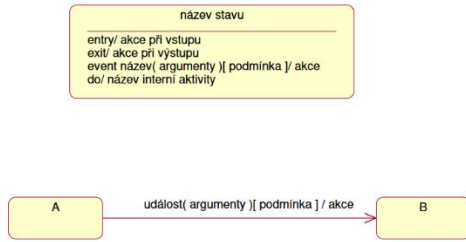
-

- Stavové diagramy

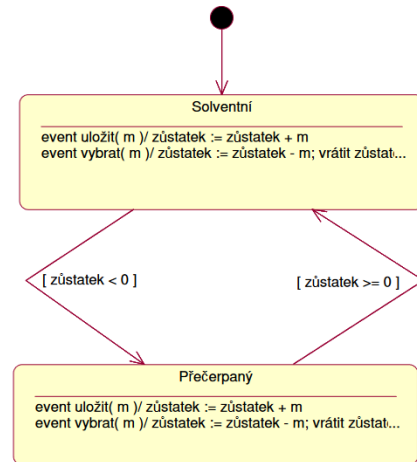
- modelování životního cyklu jednoho reaktivního objektu
- vycházejí ze stavového automatu (Harel)
- mohou modelovat dynamické chování těchto reaktivních objektů
 - třídy, resp. instancí tříd (nejčastější)
 - případy užití
 - podsystemy
 - systemy
- Reaktivní objekt
 - reaguje na vnější události

- životní cyklus je modelován jako řada stavů, přechodů a událostí
- chování je důsledkem předchozího chování (následný stav závisí na aktuálním stavu)

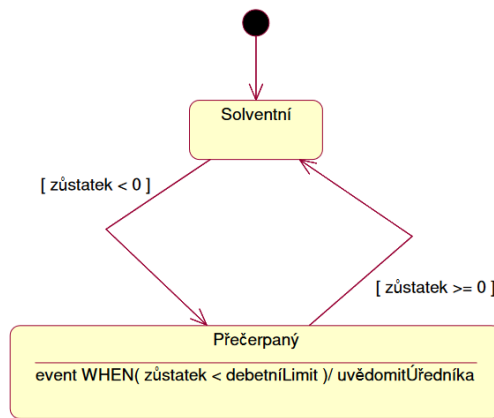
- stav
- přechod mezi stavy



- událost volání



- událost změny (WHEN)



- sekvenční složený stav
- událost AFTER

