



# **Vývoj požadavků ve fázi projektování**



KIV/ASWI 2017/2018

# Pro podrobnou specifikaci

## ► požadavků ...

---

... může být potřeba popsat cokoli z ...

- Očekávaná funkcionalita
- Mimofunkční požadavky
- Pravidla a procesy
- Uživatelské rozhraní
- Datový model
- Formální (dokazatelné) modely

# ► Detailní analýza požadavků

---

- Fáze projektování (rozpracování)
- Známe
  - zadání a cíl projektu
  - přehled důležitých funkcí / požadavků
  - možné zdroje problémů
- Co dál
  - doplnit detaily požadavků, kde je třeba
  - najít business mechanismy
  - vymyslet jejich realizaci

## **Podrobné požadavky – model užití**

# ► Podrobný popis případu užití



## Detailní rozbor komunikace aktér-systém

### Standardní průběh

- nejčastější sled akcí
- bez chyb a různých možností

### Vstupní a výstupní podmínky

- co potřebujeme pro standardní průběh

### Chybové stavy a alternativní průběhy

- určení míst výskytu, příčin, následků
- popis alternativních a chybových akcí

## Podrobnosti: UP Artifact: Use Case

Název a popis:

PU002 Půjčit exempláře

Umožňuje vlastníkovvi zaevidovat vypůjčení exemplářů

Standardní průběh:

```
# vlastník zvolí volbu "výpůjčka" v nabídce
# čtenář oznámí vlastníkovvi svoji identifikaci (jmé
# vlastník zadá nebo vyhledá čtenáře v seznamu zamě
<alt: čtenář nenalezen v evidenci>
# systém zobrazí všechny volné exempláře vlastníka
# pro všechny půjčované exempláře
## vlastník vyhledá vypůjčovaný exemplář ve svém fo
    podle PU004 Procházet katalog -- omezeno na fond
## systém ověří, že vybraný exemplář je k dispozici
    rezervovaný)
<alt: na exemplář je rezervace>
## systém zobrazí návrh výpůjčky s datem vrácení
## vlastník může data návrhu opravit, poté návrh od
## systém vytvoří záznam o výpůjčce exempláře čtená
    jeho data podle hodnot upravených vlastníkem
## systém informuje vlastníka o vytvoření výpůjčky
## vlastník předá exemplář čtenáři
# tento PU končí volbou "ukončit půjčování" zvoleno
```

Alternativní průběhy:

```
čtenář nenalezen v evidenci (krok 3)
- systém upozorní vhodným hlášením, tento PU končí

na exemplář je rezervace (krok 5)
- systém to oznámí vhodným hlášením
- tento PU pokračuje krokem 4 - další exemplář k pu
```

Vstupní podmínky:

(žádné)

Výstupní podmínky:

```
- exemplář je zapůjčen čtenáři
- je zaevidována výpůjčka
- pro exemplář je nastaven příznak "vypůjčen"
- systém je připraven pro libovolnou další operaci
```

# ► Určení detailů PU

---

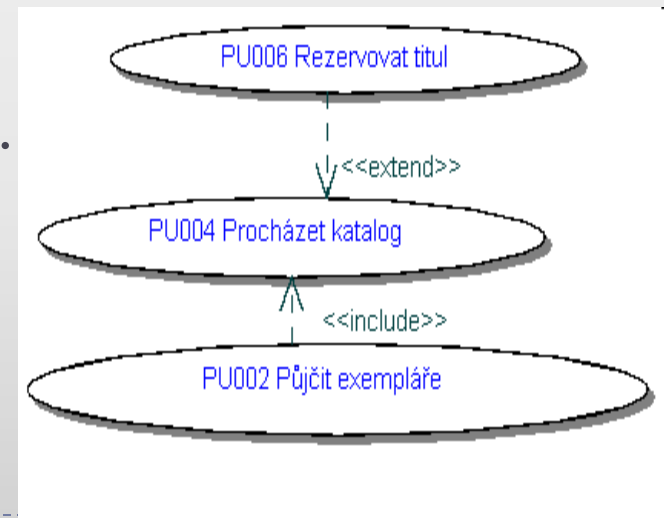
- Najít všechny scénáře
  - vyjasnění nejednoznačností v zadání
  - detaily normálního průběhu
  - všechny alternativy
- Určit výsledný stav systému
- Očekávat změny
  - primární, sekundární PU
  - společná funkčnost → vkládané, zobecněné PU
  - změny v aktérech, sekundární aktéři

Podrobnosti: UP Task: Detail Use Case Scenarios

---

# ► Vztahy mezi případy užití

- Zdroje vztahů
  - z povahy věci + snaha o zvýšení obecnosti
- **Zahrnutí** jiného případu použití
  - rozšiřuje funkčnost, reuse vkládaného
  - kam vložit: uvést u vkládajícího
- Rozšíření případu použití o jiný
  - původní průběh nedotčen
  - zpracování nestandardních situací apod.
  - kam vložit: uvést u vkládaného
- **Generalizace** základního průběhu
  - specializované PU doplňují detaily



# ► Výsledný model užití

---

## ► Úplnost


- všichni aktéři
- popis: 20% případů na 99% ... 60% na 60% ... 20% na 1%
- pozor na „samozřejmosti“ a skryté požadavky
- včetně technických PU a re-use scénářů

## ► Přehlednost a srozumitelnost

- rozložení diagramu, struktura modelu
- hodně PU  $\Rightarrow$  package, více diagramů

## ► Textové popisy

- srozumitelné pro zákazníka
- forma dokumentu: schvalování, kontrakt



UML use case model neškáluje pro rozsáhlé systémy...



## ► **Příklad: UC model s detaily**

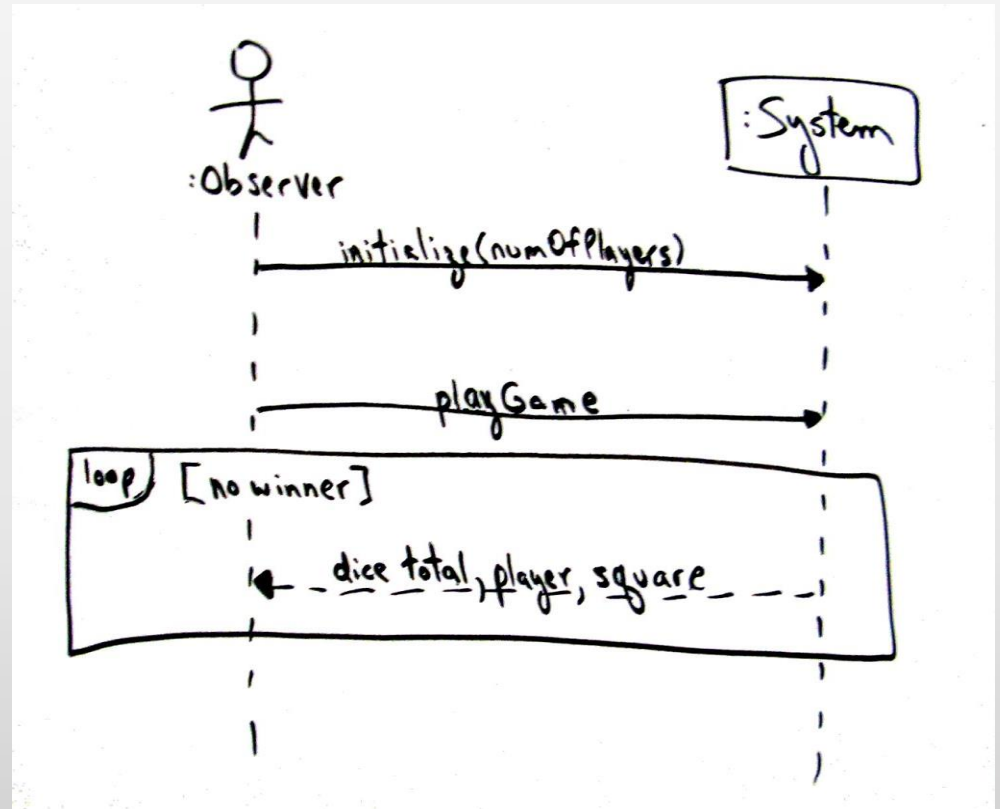
---

- Global Personal Marketplace System Requirements Specification (SRS)

# ► Systémové sekvenční diagramy

## ► Shrnutí komunikace aktér-systém

- oslí můstek pro návrh



## **Podrobné požadavky – agilní popis požadavků na funkčnost**

## ► Detailní požadavky agilně

- Přístup: „Customer collaboration over contract negotiation“ a „Responding to change over following a plan“
  - User Story je jen „poukázka na rozhovor“ (tím spíše Epic)
  - + podklad pro odhad a plánování (projekt / iterace)

royce1970managing--waterfall.pdf - Foxit Reader

checkout. Key management decisions are: when is the time and who is the person to do final checkout?

### STEP 5: INVOLVE THE CUSTOMER

For some reason what a software design is going to do is subject to wide interpretation even after previous agreement. It is important to involve the customer in a formal way so that he has committed himself at earlier points before final delivery. To give the contractor free rein between requirement definition and operation is inviting trouble. Figure 9 indicates three points following requirements definition where the insight, judgment, and commitment of the customer can bolster the development effort.

# ► User Stories: obsah



## ► Popis jedné funkčnosti z pohledu uživatele

- business value
- terminologie

## ► Hlavní vlastnosti

- stručnost
- ověřovací kritéria

**„INVEST“**

Mike Cohn (2004):  
*Writing Stories* (from  
User Stories Applied,  
Addison-Wesley)

• • •

„As a ... I want to ...  
[so that ...] .

ITB

As a student I want to purchase  
a parking pass so that I can  
drive to school

Tests:

- undergrad student: 1-term pass for \$100
- grad student: 1-term pass for \$150
- phd: 1-year pass for \$200
- cash payment
- card payment: Visa, MasterCard only
- receipt indicates type, duration, amount paid

# ► User Stories: forma

## ► Způsob uchování

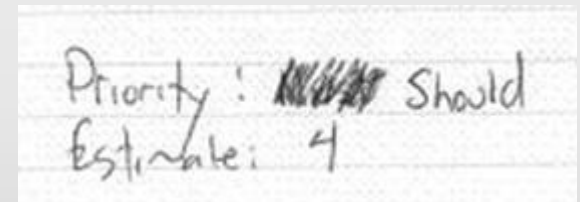
- papír
- excel
- bugzilla / xplanner

## ► Důležitá je flexibilita práce

## ► Doplnující položky

- Priorita (MoSCoW)
- Důležitost (1..100)
- Pracnost (**story points**, T-shirt sizing)

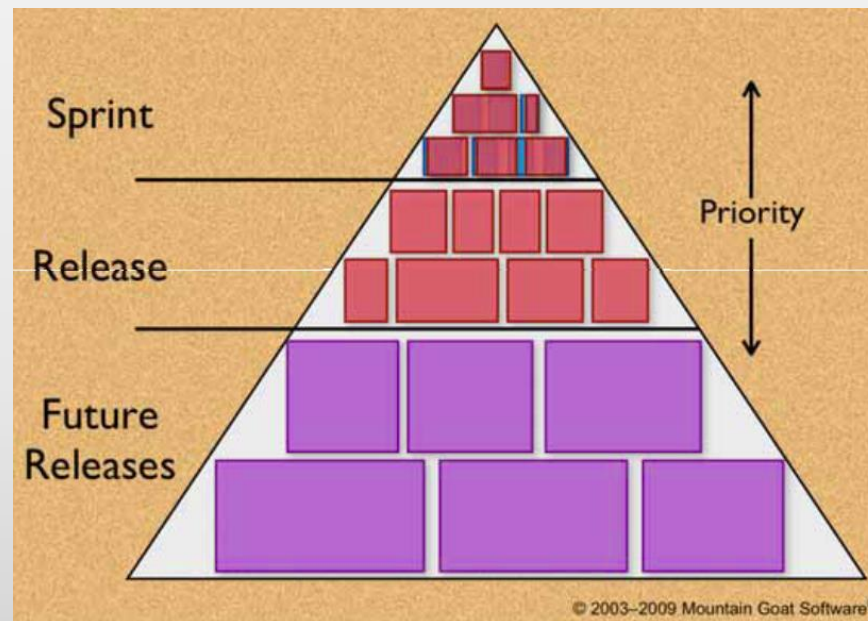
card / conversation  
/ confirmation



# ► Způsob dokumentace





---

**Stále stejný:** Product backlog



# ► Příklady Product Backlogu

## EPICS

- As a patron, I want seamless integration with Illiad
-  As a patron, I want a queue of holds that can be used to automatically generate holds for me so I stay at an reasonable level of items out.
-  As a patron, I want improved recommendations for other books that are similar to a title I am looking at or that I may want to read based on my reading history.
- As a user, I would like to access my library information via Facebook, recommend titles to my friends, add reviews, etc.
- As a collection development, I would like to add functionality for patron driven acquisitions.
-  As a patron, I want to easily access related data for a title (FRBR)
-  As a librarian, I would like a staff only interface that can be accessed by iPad which would allow increased functionality.



## MiM-4.Iterace

2012-05-09

33 Úkoly (33 uzavřených — 0 otevřených)

### Související úkoly

- Enhancement #970: ~~Návrh a implementace prvku - tlačítko.~~
- Enhancement #971: ~~Návrh a implementace prvku - posuvník.~~
- Enhancement #973: ~~Návrh a implementace prvku - seznam.~~
- Enhancement #975: Implementace hlasového vyvolání akcí.
- Enhancement #976: Rozpoznání gest.
- Enhancement #977: Autentifikace uživatele zvednutím ruky.
- Enhancement #1048: Uložení sekvencí gest do souboru





# **Mimofunkční požadavky**

## **Popis chování**

# ► Vlastnosti systému

---

Je třeba vědět nejen *co*, ale také *jak dobře*.

- Co to je: mimo-funkční požadavky
  - doby odezvy, objemy, spolehlivost, ...
  - nutný doplněk požadavků na funkce
  - dopad na architekturu, implementaci (někdy významný)

# ► **FURPS+**

---

## ► Model třídění vlastností

- Functionality, Usability, Reliability, Performance, Supportability + constraints
- původně Hewlett-Packard

## ► Omezující podmínky

- normy, zákony
- obchodní pravidla
- implementační omezení (technologie, rozhraní)
- fyzické charakteristiky



# ► Reprezentace vlastností

- Účel: možnost ověřit splnění v implementaci
- **Měřitelný** způsob (numericky)
  - obvyklá hodnota, povolené odchylky / četnost / přírůstky
  - zvážení realizovatelnosti funkčních požadavků
  - vhodná reprezentace pro neměřitelné
- Popis vlastností
  - u případů užití
  - u tříd problémové oblasti
  - vázané na celý systém

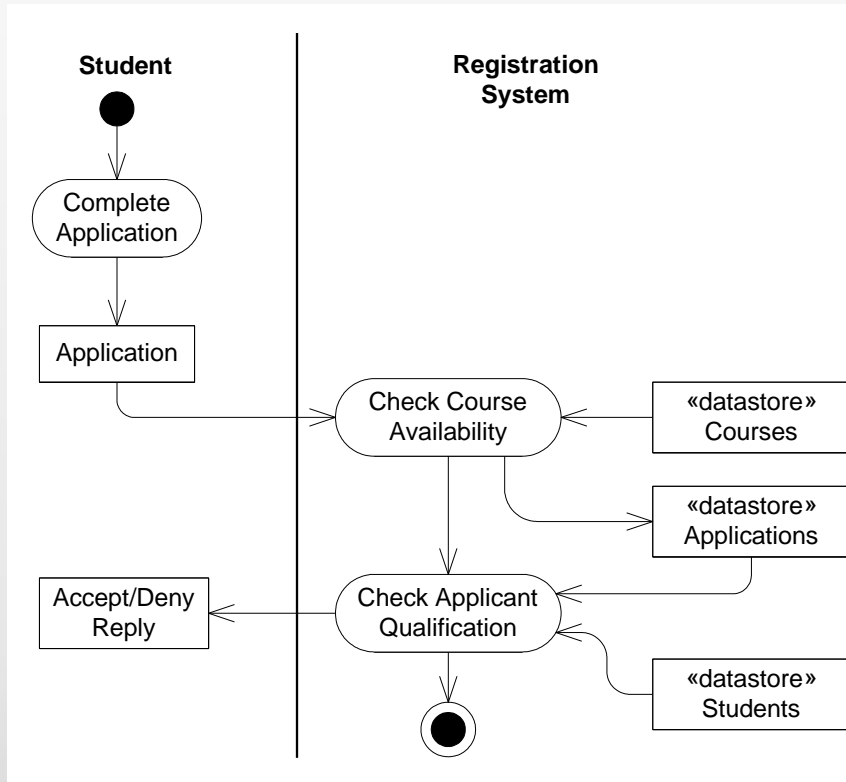
Příklad: CoCoME

## ► Business rules

---

- Popisují fakta o fungování „business logiky“ (know-how)
  - volně přiřazená k datům, procesu
  - často platná globálně v celém systému
- Strukturální
  - „jak se co z čeho vytvoří“
  - pravidla integrity systému
- Chování
  - triggerery
  - pre/post-conditions

# ► Procesní modely



- Popis scénáře funkčnosti při složitém rozhodování
  - když text nepřehledný
- Nadřazený proces dělený do funkcí (PU)
  - business process model
- Notace
  - UML diagram aktivit
  - DFD

# ► Chování aplikace

---

## ► Typický chování UI

- správa dat, průvodci, přihlašování, ...
- náhrada modelu uživatelského rozhraní

## ► Prototyp uživatelského rozhraní

- papír → PPT → demo

## ► Stavový model

- stav=obrazovka / krok průvodce / ...

# ► Prototyp uživatelského rozhraní

---

- **Prototyp** = reprezentace vnějšího rozhraní produktu nebo jeho části, v abstraktní (např. papírové) či konkrétní (spustitelná aplikace) podobě.
  - podklad pro určování funkcí a ovládání aplikace
  - diskuse nad prototypem  $\Rightarrow$  korekce neshod v porozumění
- Diskuse k prototypování
  - jednoduchá testovací data
  - abstraktní podoba uživatelského rozhraní
  - řízení konečným automatem
- Kam s ním?
  - vyhodit  $\times$  uchovat
  - hraniční třídy pro objektový návrh



# ► Formální metody

## ► Matematický model chování aplikace/systému

## ► Dokazatelná

- správnost, bezrozpornost
- úplnost

## ► Specializované notace

- Z,VDM
- B-method

## ► ... a nástroje

*Tariff*

$\Delta FS$

$op? : Op$

$\Phi FileId?$

$\Phi FileId!$

$data! : Data$

$idset! : \mathbb{F} FileId$

$cost! : Money$

$report! = SuccessReport \Rightarrow$

$op? = NewFileOp \Rightarrow cost! = NewFileCost$

$op? = WriteFileOp \Rightarrow cost! = WriteFileCost$   
 $+ StoreByteCost * (expires' - updated') * \#contents'$

$op? = ReadFileOp \Rightarrow cost! = ReadFileCost + ReadByteCost * \#data!$

$op? = DestroyFileOp \Rightarrow cost! = DestroyFileCost$   
 $- StoreByteCost * (expires' - updated') * \#contents'$

$op? = FileStatusOp \Rightarrow cost! = FileStatusCost$

$op? = SetFileExpiryOp \Rightarrow cost! = SetFileExpiryCost$   
 $+ StoreByteCost * (expires' - expires) * \#contents'$

$op? = SetFileLengthOp \Rightarrow cost! = SetFileLengthCost$   
 $+ StoreByteCost * (expires' - updated')$   
 $* (\#contents' - \#contents)$

$report! \in \{NoSuchFileReport, NoSpaceReport, NotOwnerReport,$   
 $NotKnownUserReport\} \Rightarrow cost! = FSErrorCost$



# **Modelování datové části**



# ► Použití doménového modelu

---

## ► Komunikace

- dorozumění s klientem

## ► Objektová analýza a návrh

- s těmito třídami můžeme najisto počítat
- analýza přidá zodpovědnosti, detaily vlastností a chování
- návrh je transformuje a přidá implementační třídy

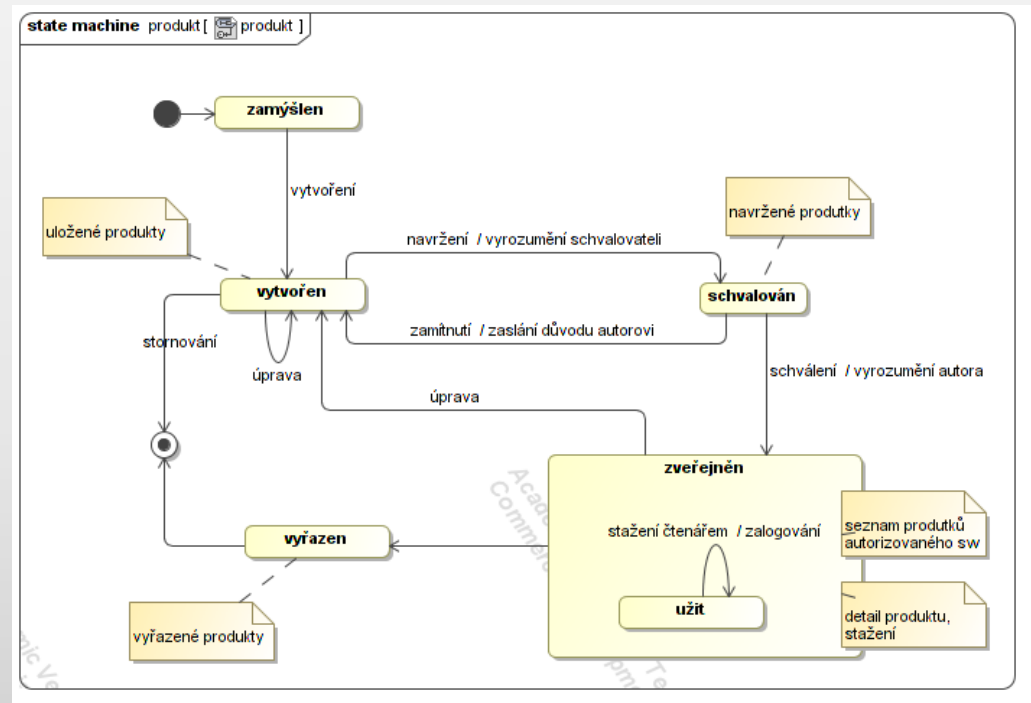
## ► Datové modelování

- doménový model → logický datový model



# ► Vývoj datových entit

- (Datová) entita typicky prochází vývojem
  - životní cyklus
- Model = stavový diagram
  - vazba na doménový / datový model
  - vazba na pravidla
  - vazba na funkčnosti (+ procesy)



## ► CRUD(L) matice

---

- Cíl: vědět kdo/co manipuluje s jakými údaji
- Create-Read-Update-Delete(-List)
- Úroveň detailu
  - analytická – uživatel, případ užití × informace
  - návrhová – třída, funkce, proces × tabulka

	<b>Contact History</b>	<b>Maps</b>	<b>Points Of Interest</b>	<b>Preferences</b>
Plan Route	Read & Create	Read	Read	
Request Hotel		Read	Read	
Visit Web Site			Read & Update	Read & Update



# **Formy a vlastnosti specifikace detailů požadavků**



# ► Formy specifikace požadavků

- Textový popis
  - shopping list
  - backlog, story cards
  - strukturovaný text
- Grafické notace, modely
  - případy užití
  - procesní atd. modely
- Strukturovaný dokument
  - normy (IEEE 830-1998)
  - RUP šablona
- Implementace
  - prototyp
  - uživatelská příručka

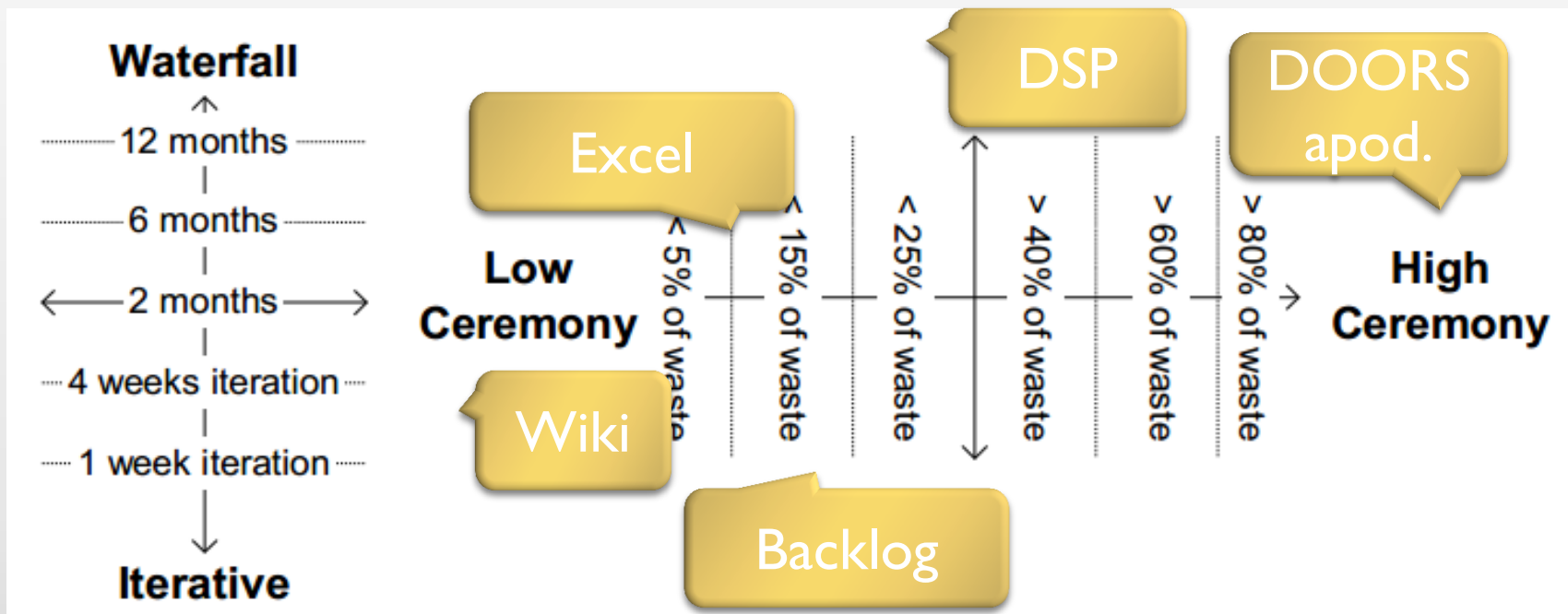
*Klíčové vlastnosti*  
- úplnost  
- bezrozpornost  
- trasovatelnost

Specializované metodiky, postupy a nástroje – DOORS, RequisitePro

Detaily:  
Wiegiers „COS SRS“, RUP Concepts: Traceability,  
UP Checkpoints: System-Wide Requirements



# ► Varianty dle velikosti projektu



# ▶ Úplnost specifikace požadavků

---

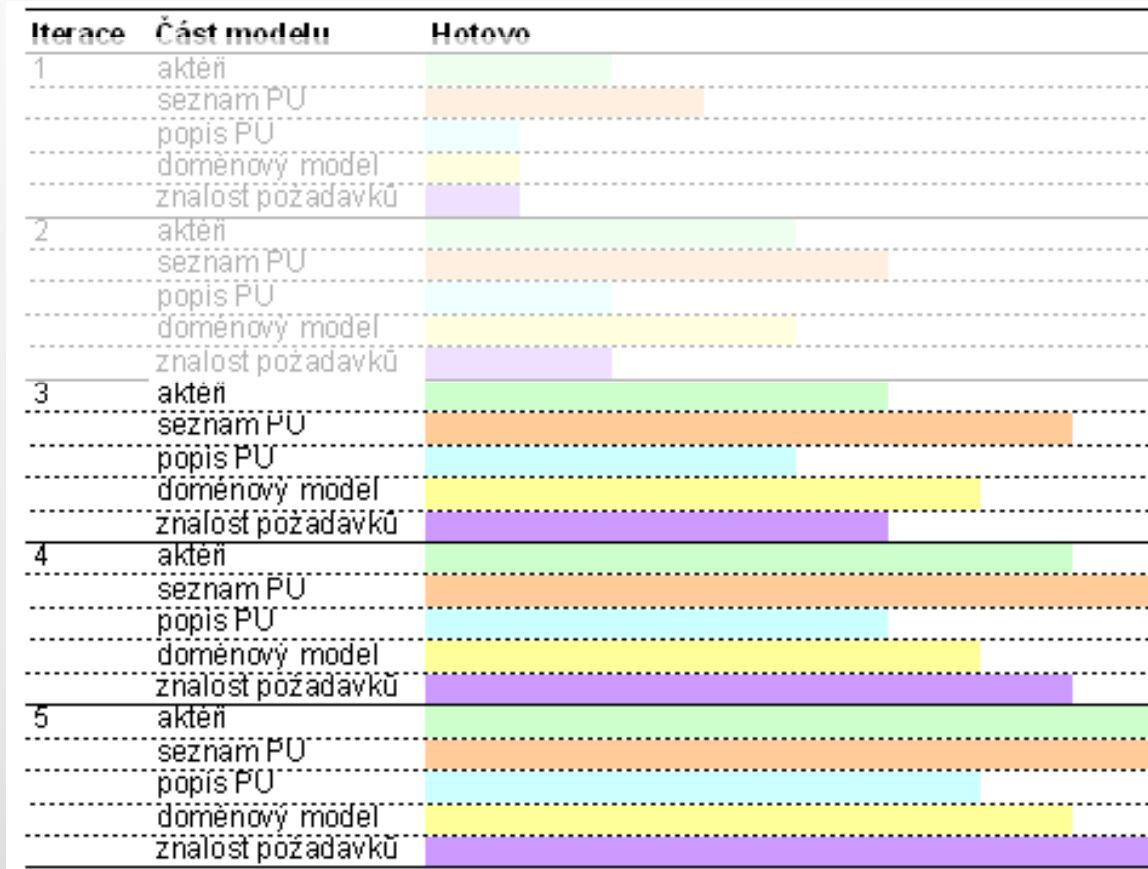
## ▶ Fáze zahájení projektu

- ▶ přesně cíl/vize projektu
- ▶ seznam klíčových aktérů, jejich cíle
- ▶ seznam/diagram podstatných případů užití (dle cíle)
- ▶ stručný popis klíčových PU, znalost klíčových vlastností

## ▶ Fáze projektování

- ▶ kompletní seznam aktérů, popis důležitých
- ▶ kompletní specifikace 80-100% funkčnosti (přesné popisy důležitých funkčností, stručné povědomí u všech)
- ▶ přesný popis mimofunkčních vlastností
- ▶ kompletní znalost datového modelu

## ► Vývoj znalosti požadavků

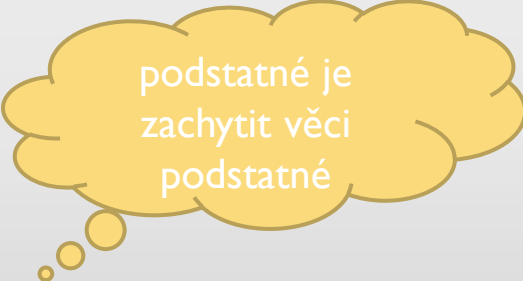


- V závislosti na iteraci, fázi, release

## ► Shrnutí

---

- Různé prostředky pro zachycení požadavků
  - kontext – vize, aktéři
  - funkčnost – případy užití, user stories, procesní model, ...
  - struktury – doménový model, stavový
  - vlastnosti – u PU, funkcí, doménových tříd, samostatně
- **Práce se specifikacemi požadavků**
  - viz iterativní proces: zpřesňování, design, plánování, realizace
- **Poznávání požadavků je nekončící proces**
  - zákazník a jeho představy se mění
  - poznání analytiků se zpřesňuje analýzou i implementací



podstatné je  
zachytit věci  
podstatné