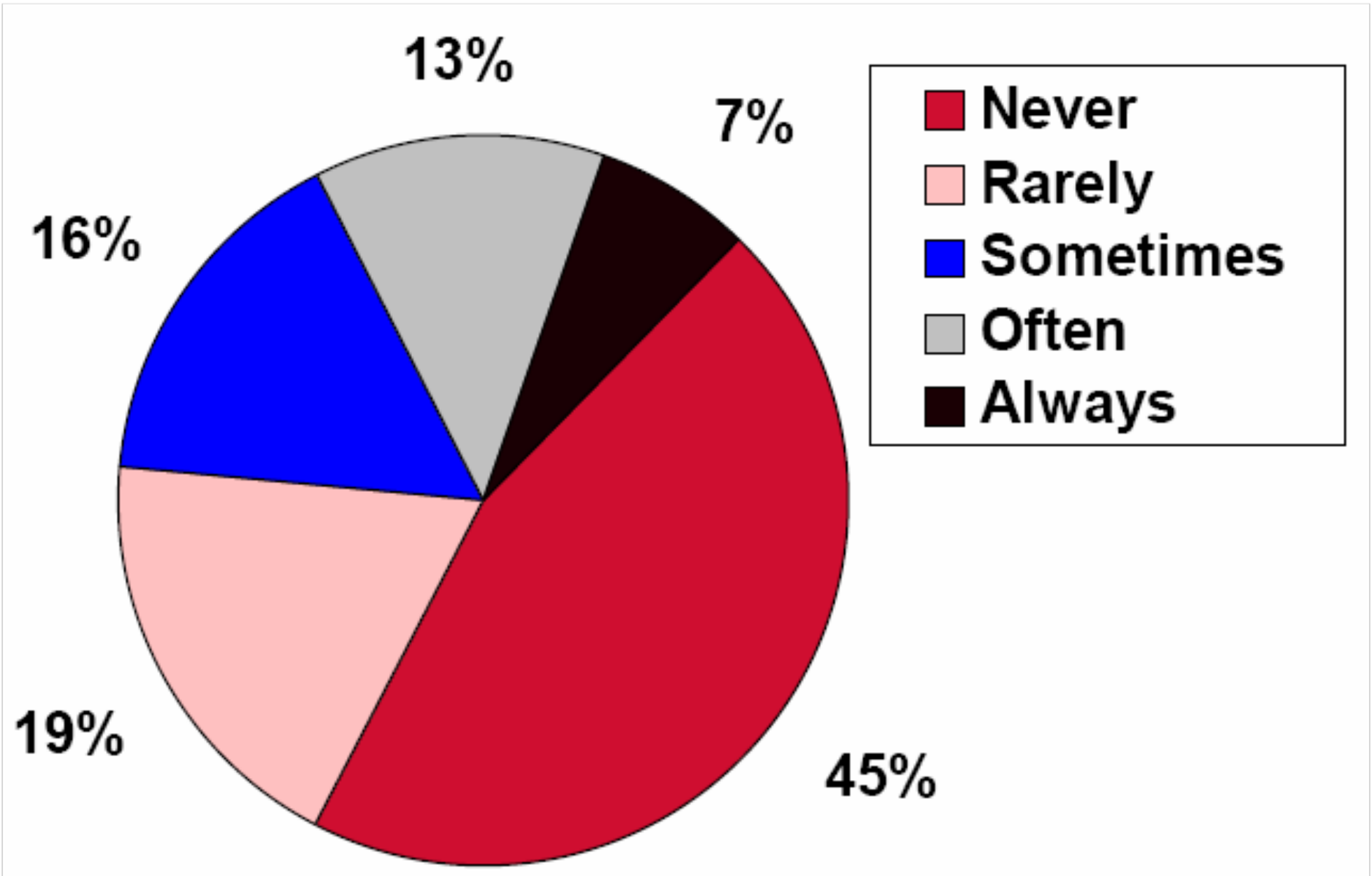
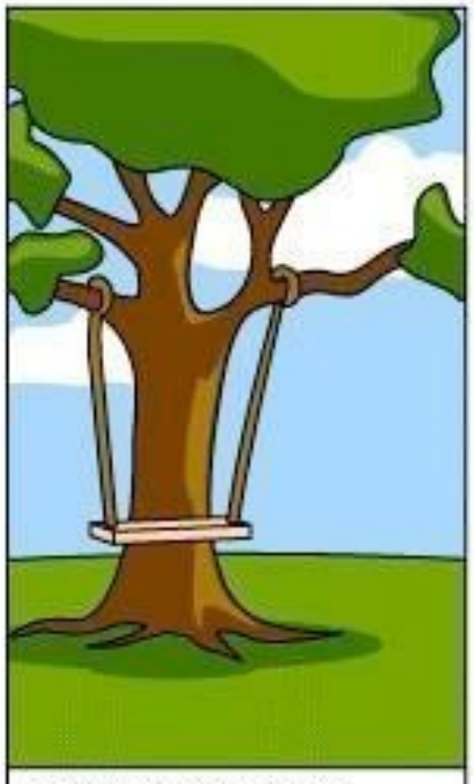
SPECIFIKACE POŽADAVKŮ NA SOFTWARE

Jaroslav Žáček
jaroslav.zacek@osu.cz
http://www1.osu.cz/~zacek/

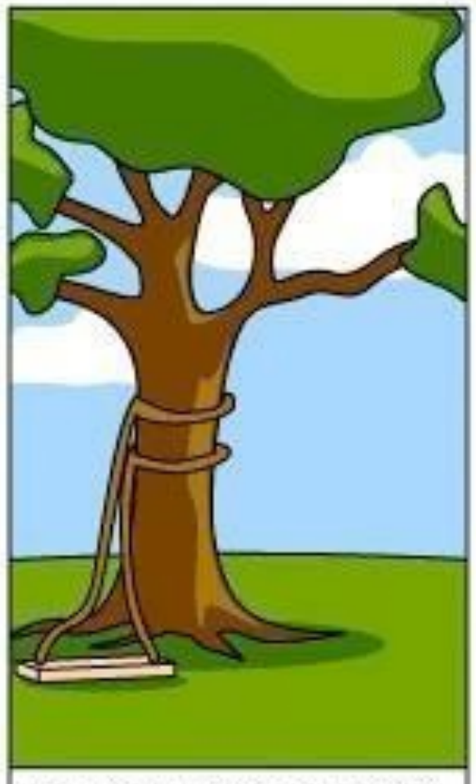

13%
7%
16%
19%
45%
Never
Rarely
Sometimes
Often
Always

How the customer explained it



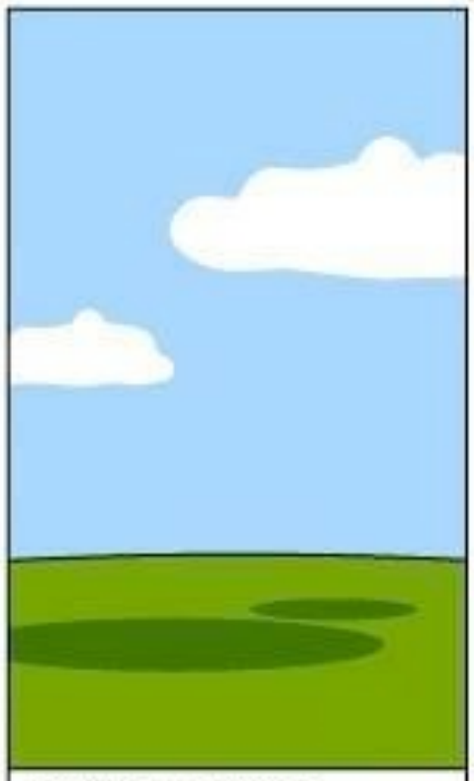
How the Project Leader understood it

How the Analyst designed it

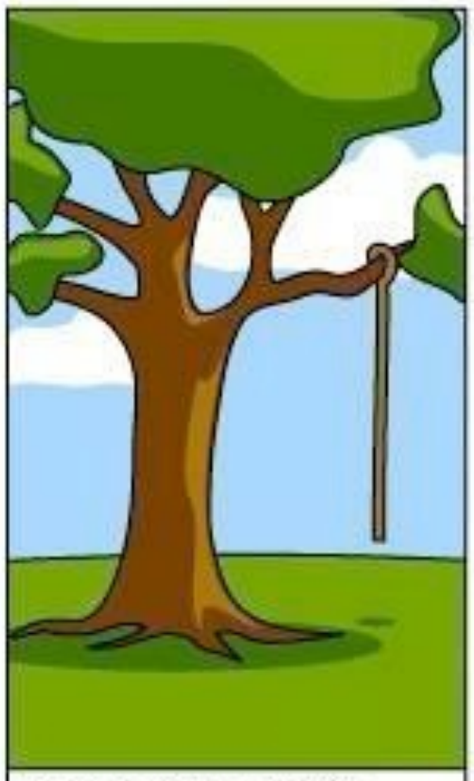


How the Programmer wrote it

How the Business Consultant described it

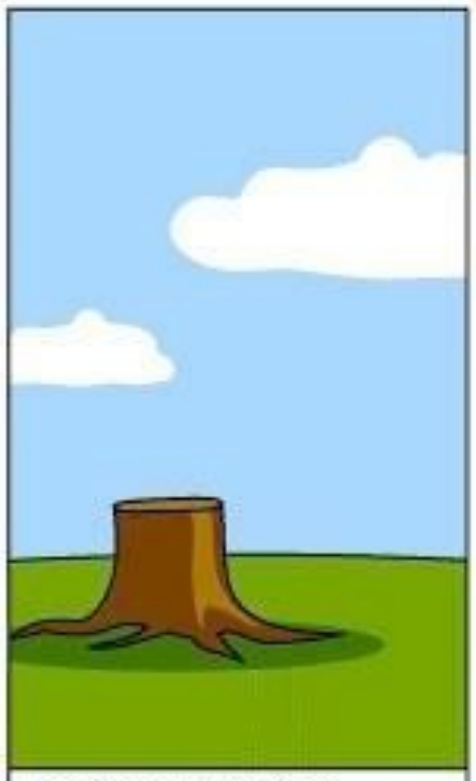


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed

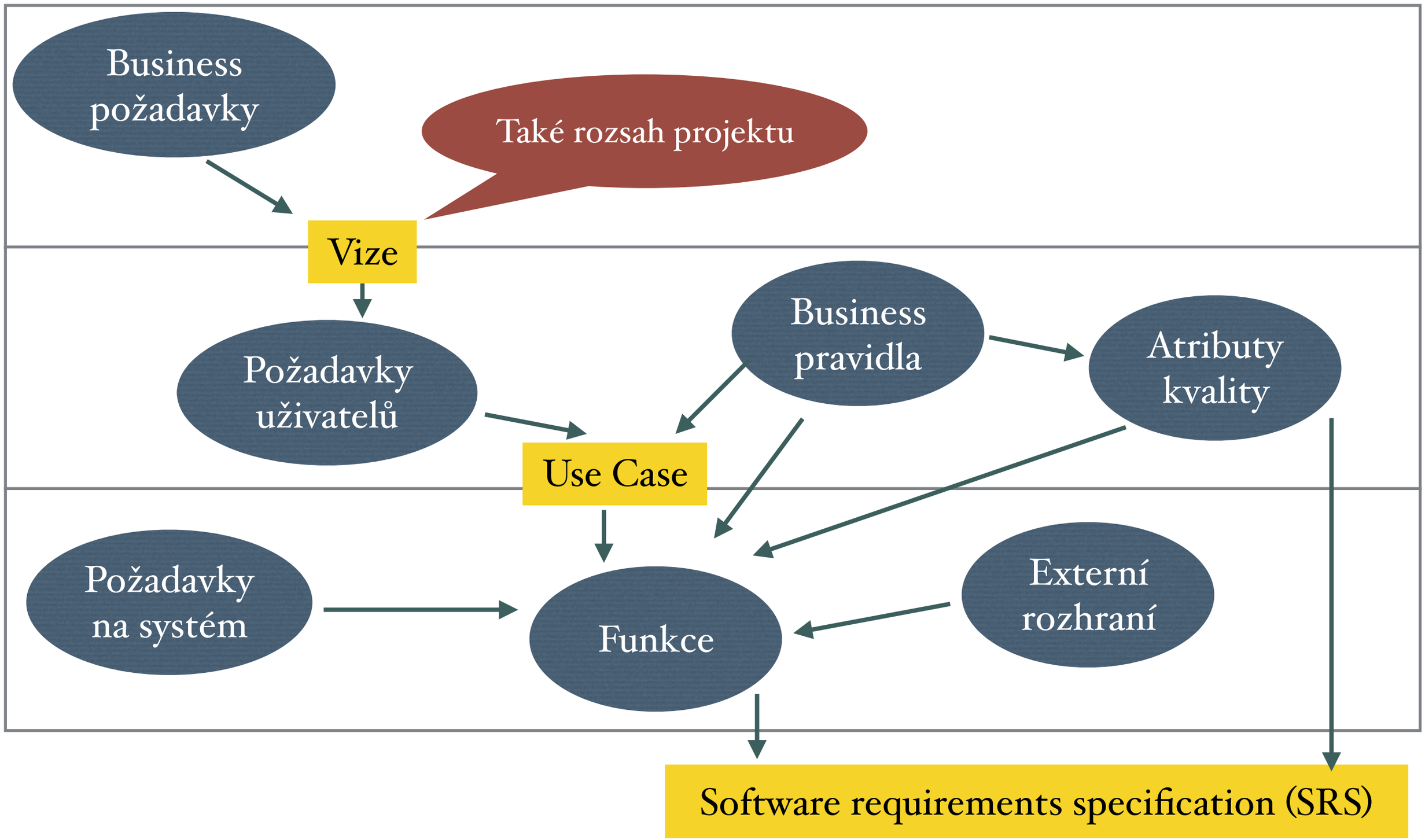
DŮVODY PRO FORMALIZACI SRS

- Podle Chaos Report organizace Standish Group jsou požadavky jedním z přispěvatelů k problémům softwarových projektů. V roce 1997 na prvním místě, 2006 na 4. místě.
- Výzkumy i zkušenosti hodnotí měnící se požadavky jako jednu z příčin neúspěšnosti projektů.
- Pouze 20% požadavků na software je využito, zhruba 45% se nikdy nepoužije.

ZÁKLADNÍ DĚLENÍ POŽADAVKŮ

- **Funkční** - popisuje co systém dělá, formalizace např. pomocí UC
- **Nefunkční** - popisují další vlastnosti, které uživatel přímo nevidí při práci s aplikací (doba odezvy, architektura - přístup k aplikaci z více míst)

VE SKUTEČNOSTI JE TO TROCHU SLOŽITĚJŠÍ



OBECNÝ POSTUP PŘI SPECIFIKACI POŽADAVKŮ

- Pochopení problému a jeho analýza
- Identifikace lidí se zájmem na projektu (tzv. stakeholders)
- Definice systému (scope) a jeho hranic (boundaries)
- Identifikace omezení, které musí systém mít (celá firma používá jako OS Linux)

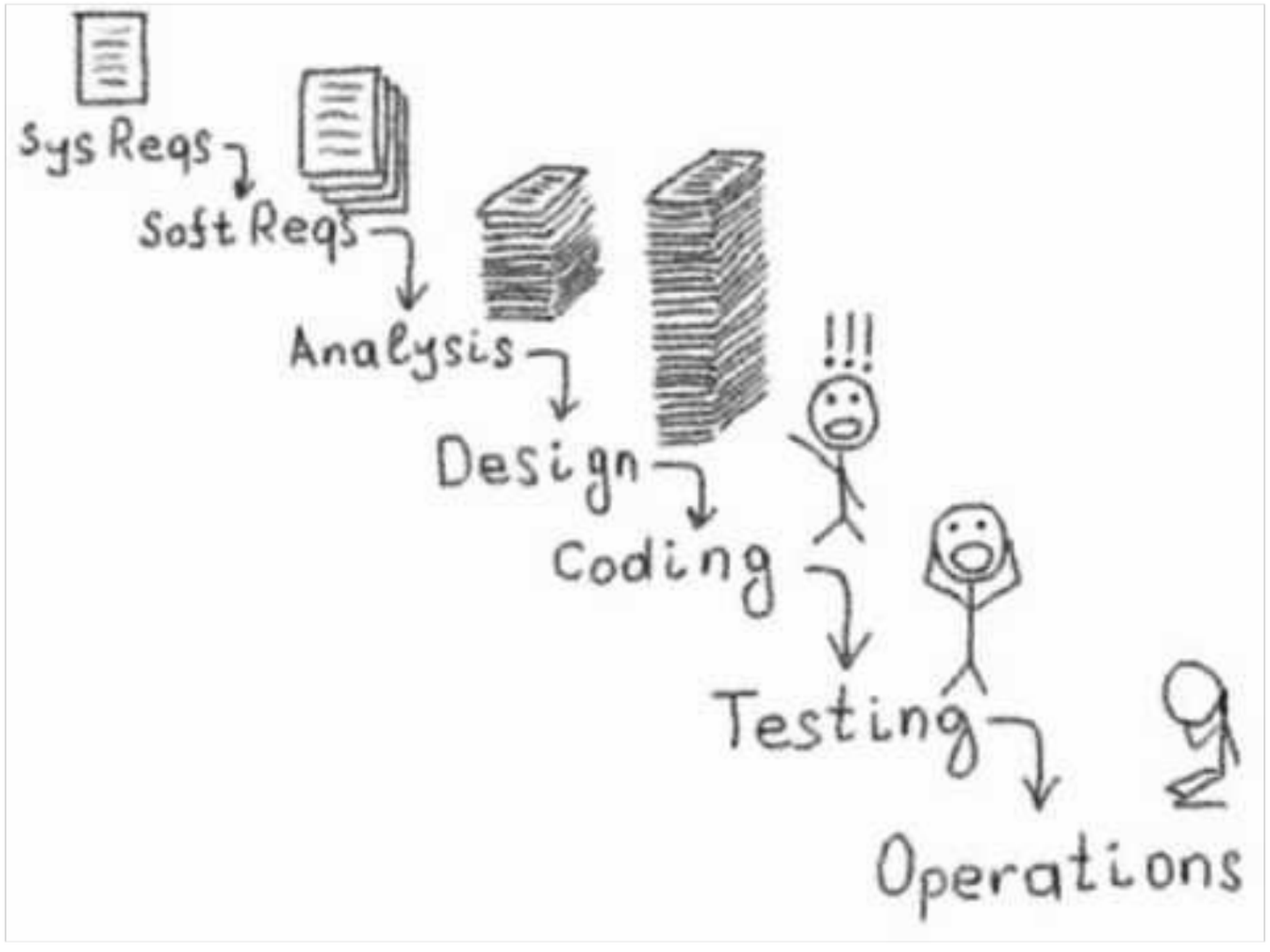
ZPŮSOBY IDENTIFIKACE POŽADAVKŮ

- Rozhovory s vybranými uživateli
- Requirements workshop
- Prototyp (HTML, obecné GUI)
- User Stories, Post-it lístečky

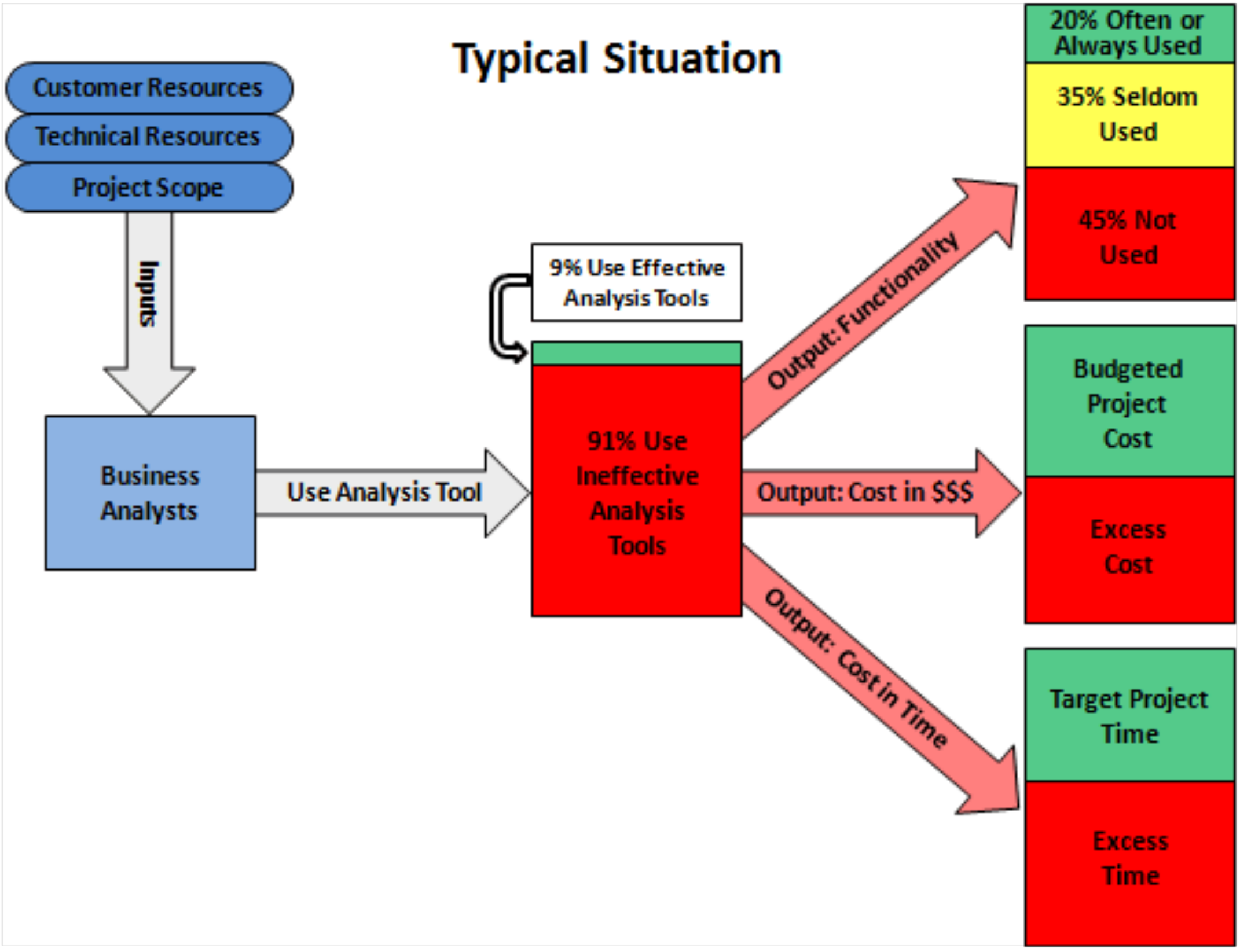
TRADIČNÍ ZPŮSOBY SOUPISU POŽADAVKŮ

- Sesbírat veškeré požadavky na systém detailně.
- Na základě požadavků navrhnout architekturu aplikace a detailně vypracovat důkladnou analýzu.
- Navrhnout databázi, pak teprve funkce nad ní.
- Forma zápisu převážně dokumentová.

ŽIVOTNÍ CYKLUS VÝVOJE SOFTWARE



Typical Situation



MOŽNÉ NEVÝHODY

- Ztráta celkového přehledu.
- Navzájem se vylučující požadavky.
- Funkce jsou zafixovány po celou dobu vývoje (již proběhla analýza).
- Funkce jsou většinou zaznamenány z pohledu systému, ne uživatele.

JAK TO ZLEPŠIT?

- Jakým způsobem byste chtěli zapisovat požadavky u zákazníka?
- Jak to zařídit, aby tomu zápisu rozuměli vaši kolegové?

UML

- Obecný grafický jazyk pro vizualizaci, dokumentaci a popis chování a funkcí v oblasti softwarového inženýrství.
- Od roku 1997 standardizován organizací OMG.
- Nejedná se o programovací jazyk.

DOSTUPNÉ MODELY

UML 1.x:

- Use Case diagram
- Class diagram
- Object diagram
- Sequence diagram
- Statechart
- Activity diagram
- Component diagram
- Deployment diagram

UML 2.x:

- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Activity diagram
- State Machine diagram
- Use Case diagram
- Communication diagram
- Interaction overview diagram
- Sequence diagram
- Timing Diagram

DOSTUPNÉ MODELY

UML 1.x:

- Use Case diagram
- Class diagram
- Object diagram
- Sequence diagram
- Statechart
- Activity diagram
- Component diagram
- Deployment diagram

UML 2.x:

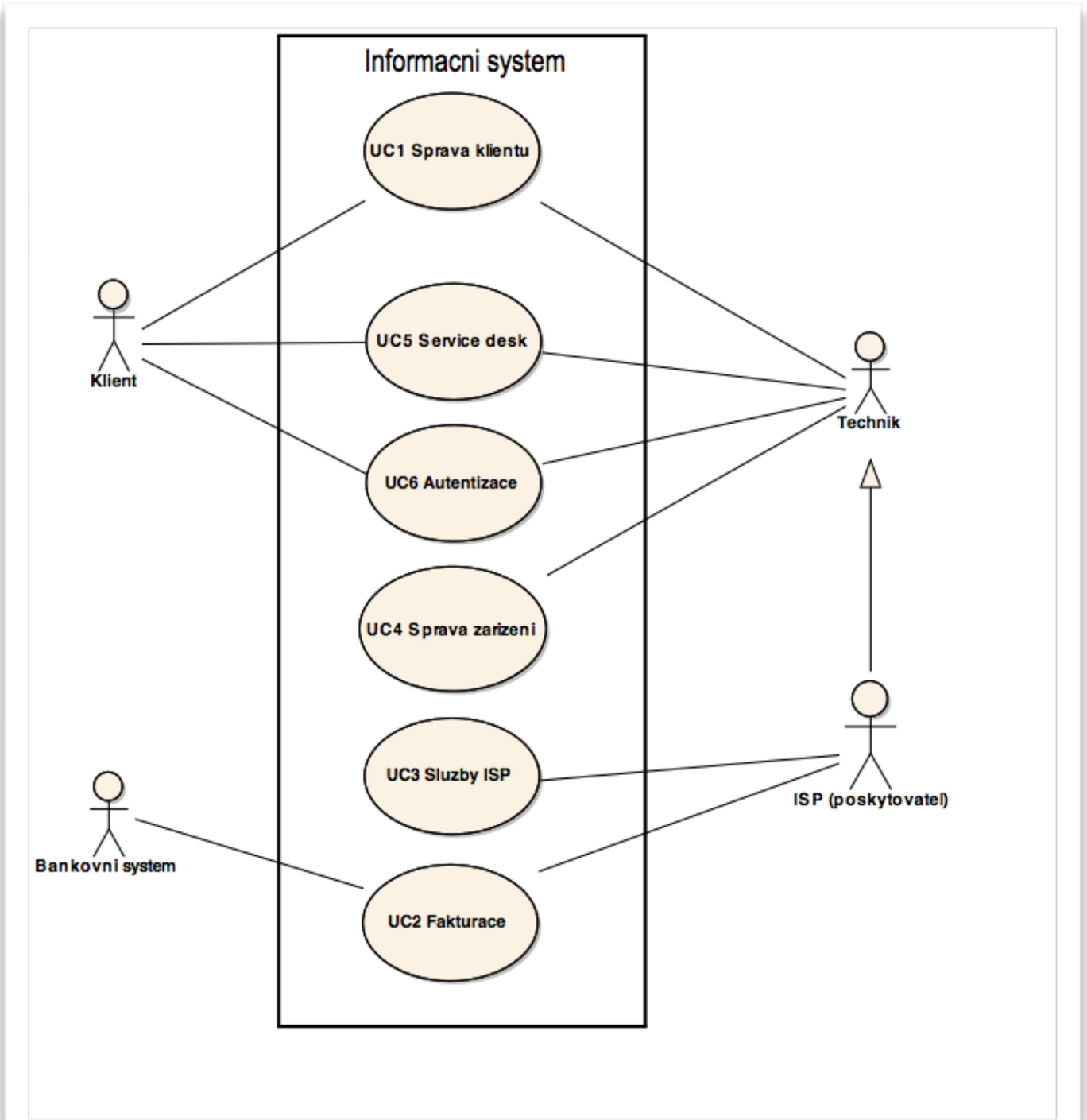
- **Class diagram**
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Activity diagram
- State Machine diagram
- **Use Case diagram**
- Communication diagram
- Interaction overview diagram
- **Sequence diagram**
- Timing Diagram

USE CASE

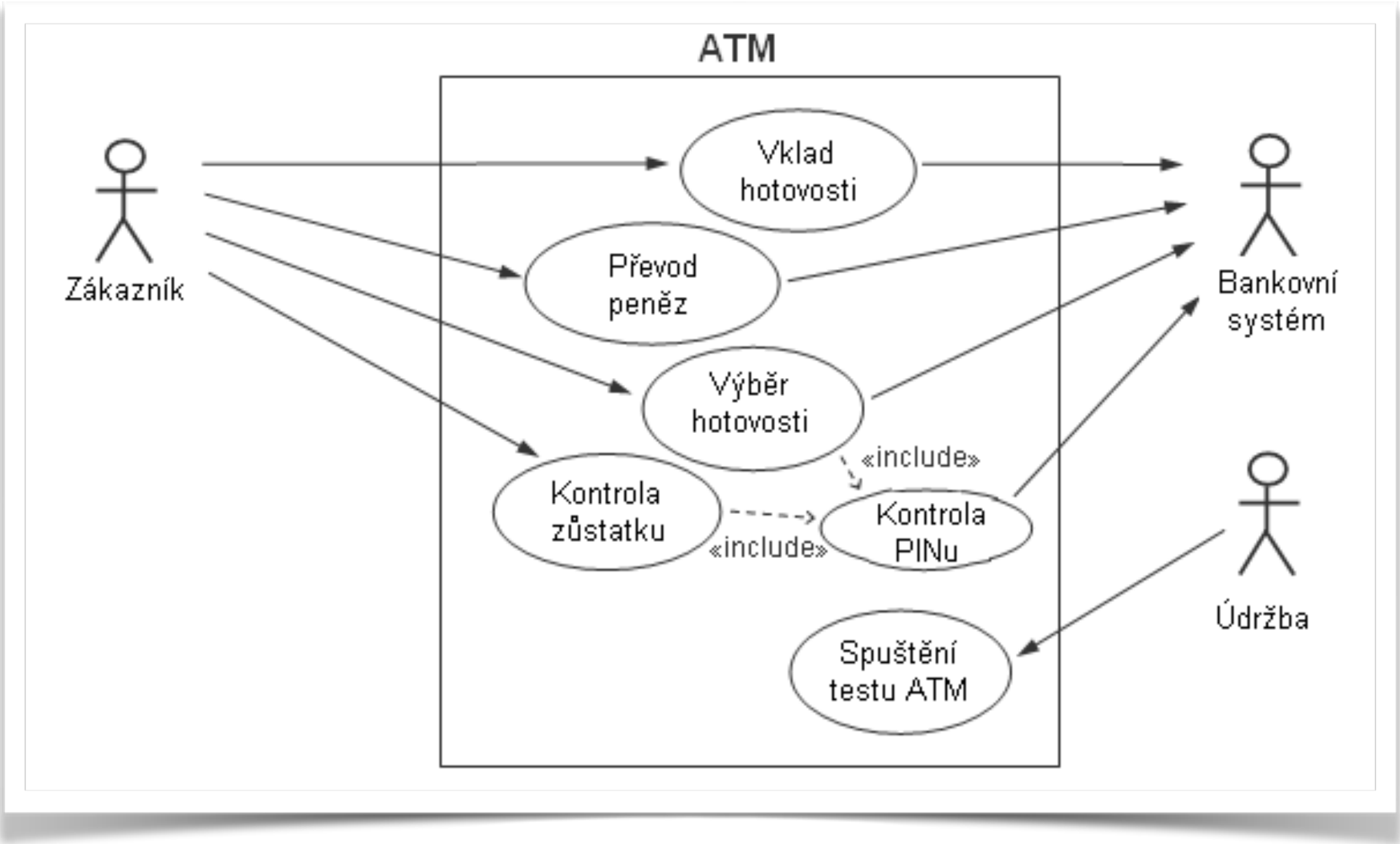
- **Actor** - vyjadřuje jednotku, která má nějaké interakce se systémem (člověk, jiný software, organizace).
- **Scenario** - specifická sekvence akcí a interakcí mezi aktorem a systémem který budujeme. Občas označován jako instance případu užití.
- **Use Case** (případ užití) - kolekce úspěšných a neúspěšných scénářů, které mají “něco společného” a podporují konkrétní cíl.

USE CASE

- Hranice systému
- Aktoři
- Use Cases - případy užití
- Relace, vazby

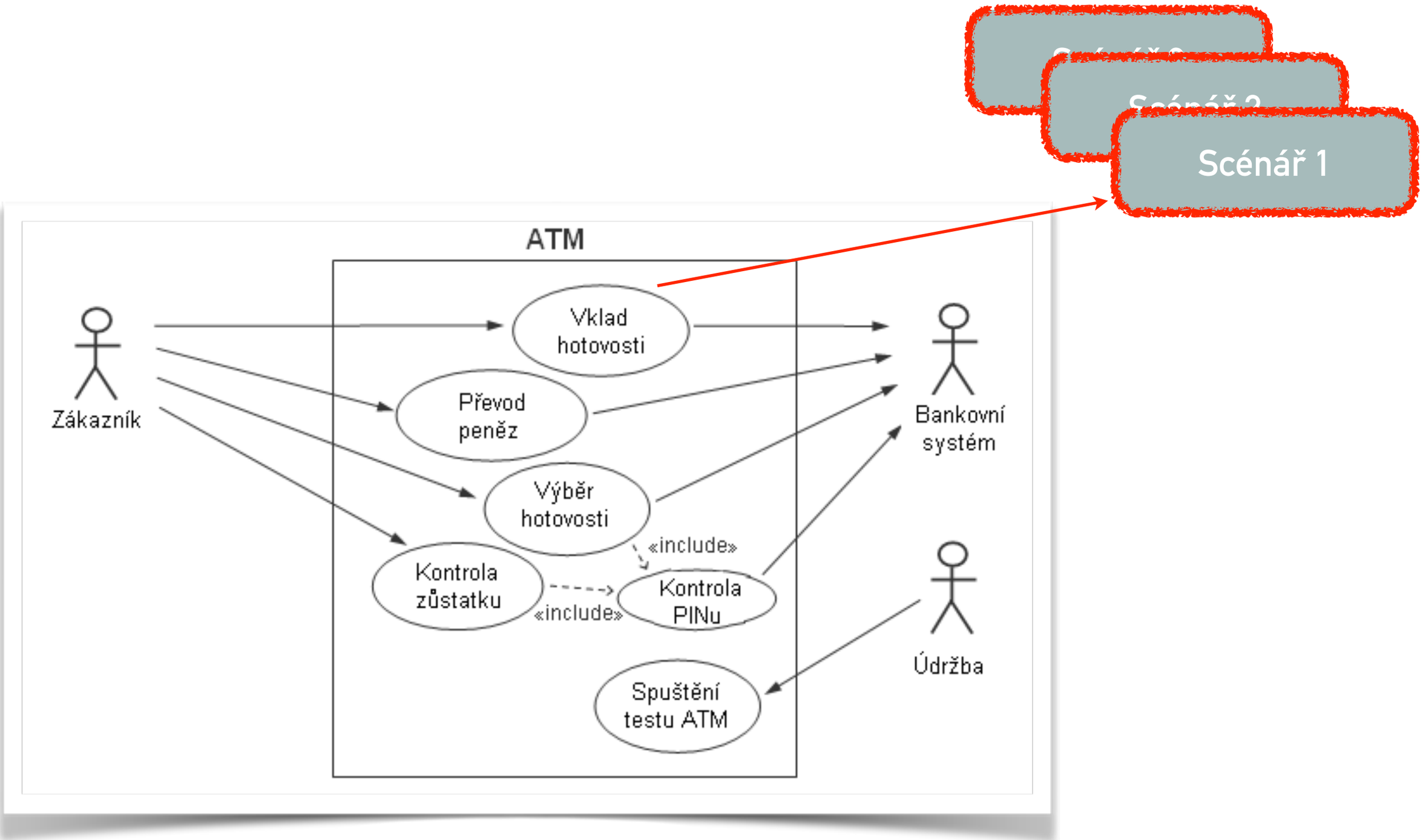


PŘÍKLAD POUŽITÍ – BANKOMAT



…stačí vám tento zápis k vytvoření aplikace?

USE CASE A SCÉNÁŘE



FORMÁLNOST UC

- **Brief** - nejvyšší abstrakce, většinou jeden hlavní scénář, např. Proces prodeje
- **Casual** - pokrývá více scénářů, např. Zpracování objednávek
- **Fully dressed** - detailně zpracovány všechny kroky, obsahují sekce Preconditions a Guaranties

STYL BLACK-BOX

- Nejčastější způsob zapsání UC.
- Nepopisují vnitřní práci systému, ale odpovědnosti systému za určité činnosti.

Styl Black-box	Styl White-box
Systém zaznamenává objednávky.	Systém zapisuje objednávky do databáze…(nebo hůře)…systém generuje SQL INSERT pro objednávku

ŠABLONA

- Primary Actor
- Stakeholders and Interests
- Preconditions
- Success Guarantee

- Main Success Scenario (také označován jako Basic Flow)
- Extensions (také označovány jako Alternative Flow)
- Special Requirements
- Technology and Data Variations List
- Frequency or Occurence
- Open Issues

PŘÍKLAD SCÉNÁŘE – NÁKUP V SAMOOBSLUZE

Handle Returns

Main Success Scenario: A customer arrives at a checkout with items to return. The cashier uses the POS system to record each returned item ...

Alternate Scenarios:

If the credit authorization is reject, inform the customer and ask for an alternate payment method.

If the item identifier is not found in the system, notify the Cashier and suggest manual entry of the identifier code (perhaps it is corrupted).

If the system detects failure to communicate with the external tax calculator system, ...

PŘÍKLAD

Use Case UC1: Process Sale

Primary Actor: Cashier
Stakeholders and Interests:
- Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer short ages are deducted from his/her salary.
- Salesperson: Wants sales commissions updated.
- Customer: Wants purchase and fast service with minimal effort. Wants proof of pur chase to support returns.
- Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server compo nents (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.
- Government Tax Agencies: Want to collect tax from every sale. May be multiple agen cies, such as national, state, and county.
- Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.

Preconditions: Cashier is identified and authenticated.
Success Guarantee (Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

BASIC FLOW

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
   Price calculated from a set of price rules.

   Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale and payment information to the
   external Accounting system (for accounting and commissions) and
   Inventory system (to update inventory).
9. System presents receipt.
10. Customer leaves with receipt and goods (if any).

ALTERNATIVE FLOW

Extensions (or Alternative Flows):

*a. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

1. System signals error to the Cashier, records the error, and enters a clean state.
2. Cashier starts a new sale.

3a. Invalid identifier:

1. System signals error and rejects entry. 3b. There are multiple of same item category and tracking unique item identity not

important (e.g., 5 packages of veggie-burgers):

1. Cashier can enter item category identifier and the quantity.

3-6a: Customer asks Cashier to remove an item from the purchase:

1. Cashier enters item identifier for removal from sale.
2. System displays updated running total.

3-6b. Customer tells Cashier to cancel sale:

1. Cashier cancels sale on System.

3-6c. Cashier suspends the sale:

1. System records sale so that it is available for retrieval on any POS terminal. 4a. The system generated item price is not wanted (e.g., Customer complained about something and is offered a lower price):

1. Cashier enters override price.
2. System presents new price.

SPECIAL REQUIREMENTS/TECHNOLOGY

Special Requirements:

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.
- Credit authorization response within 30 seconds 90% of the time.
- Somehow, we want robust recovery when access to remote services such the inven tory system is failing.
- Language internationalization on the text displayed.
- Pluggable business rules to be insertable at steps 3 and 7.

Technology and Data Variations List:

3a. Item identifier entered by bar code laser scanner (if bar code is present) or key-board.
3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.
7a. Credit account information entered by card reader or keyboard.
7b. Credit payment signature captured on paper receipt. But within two years, we pre-dict many customers will want digital signature capture.

JAK ČASTO SE OPAKUJE, NA CO JE POTŘEBA SI DÁT POZOR

Frequency of Occurrence: Could be nearly continuous.

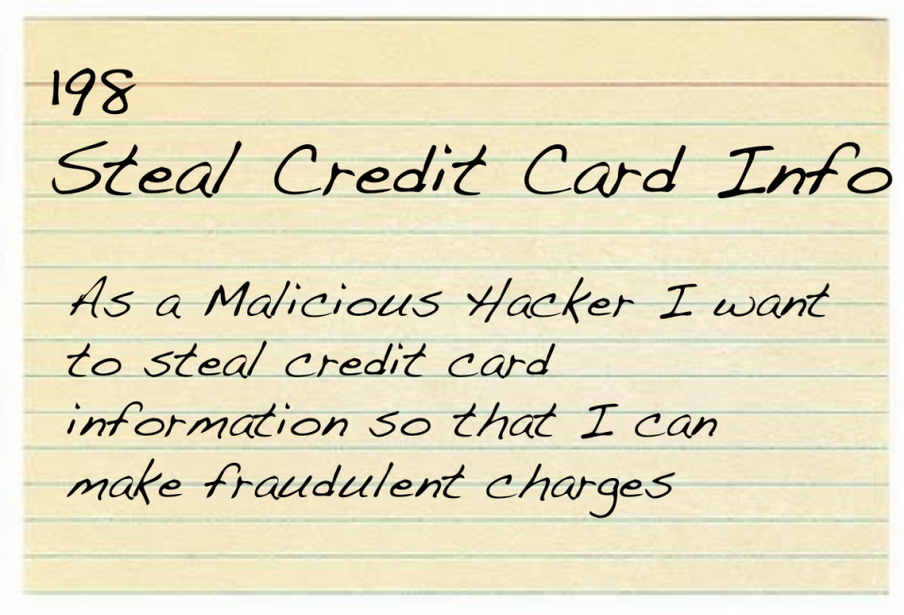
Open Issues:

- What are the tax law variations?
- Explore the remote service recovery issue.
- What customization is needed for different businesses?
- Must a cashier take their cash drawer when they log out?
- Can the customer directly use the card reader, or does the cashier have to do it?

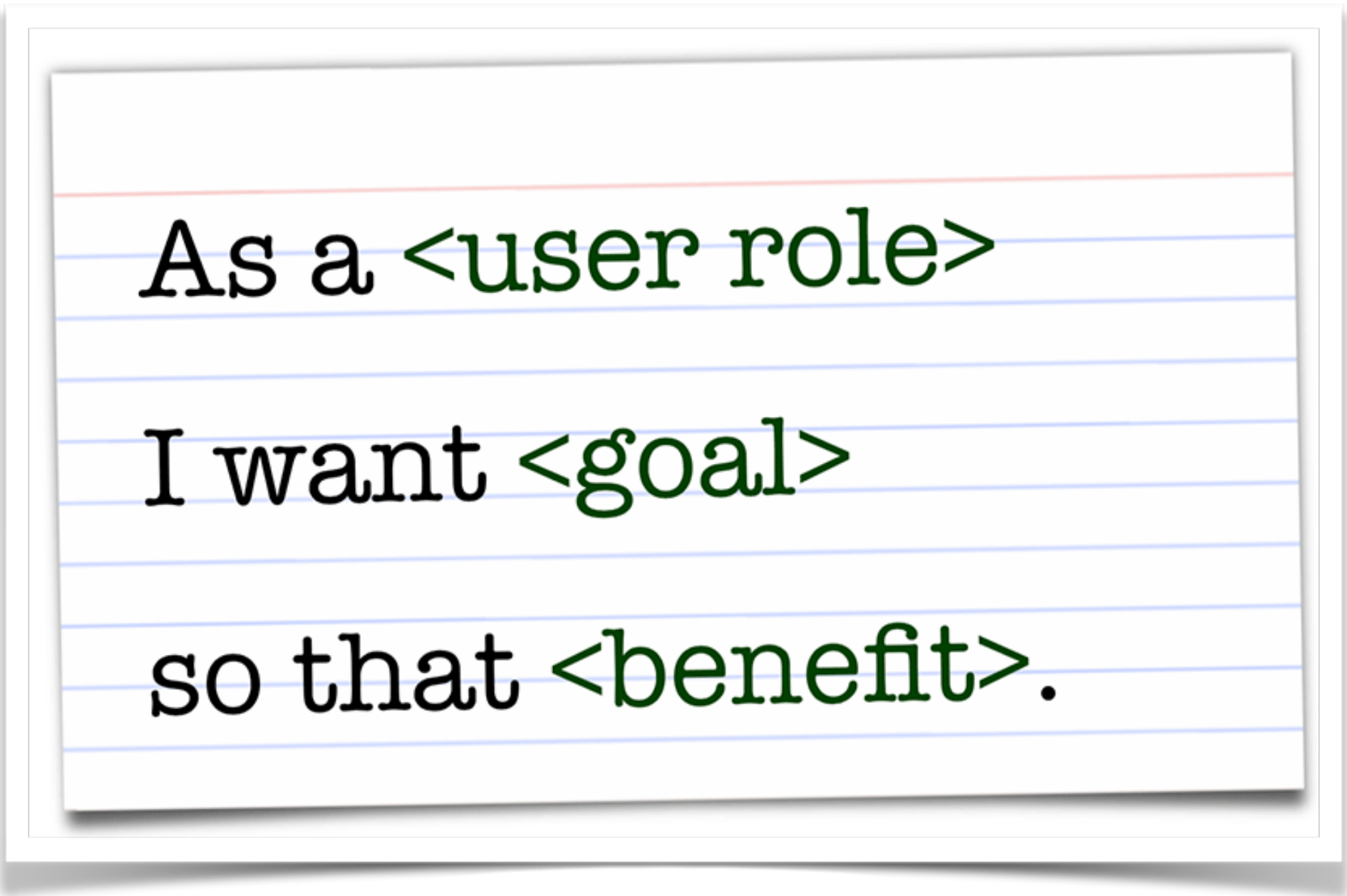
USER STORIES

USER STORY

- *Termín z XP, Scrum*
- *Textový popis požadavku na budoucí software v řeči, kterou denně používá uživatel*
- *Psáno na malém papírku, kartičce (aby nenarostl do velkých rozměrů)*
- *User Stories jsou psány zákazníkem*



USER STORY



USER STORY

- As Max, I want to invite my friends, so we can enjoy this service together.
- As Sascha, I want to organize my work, so I can feel more in control.
- As a manager, I want to be able to understand my colleagues progress, so I can better report our sucess and failures.

USER STORY

sg1 / SG-45

Hosting a založení MySQL serveru

Edit | Comment | Assign | More | To Do | In Progress | Workflow | Admin

Details

Type:	Story	Status:	READY FOR ACCEP...
Priority:	Medium		(View Workflow)
		Resolution:	Unresolved
Labels:	None		
Epic Link:	Architektura aplikace		
Sprint:	SG Sprint 6, SG Sprint 7, SG Sprint 8, SG Sprint 9		

Description

DoR

- Byla nalezena (či domluvena) možnost pro vytvoření hostingu pro MySQL server

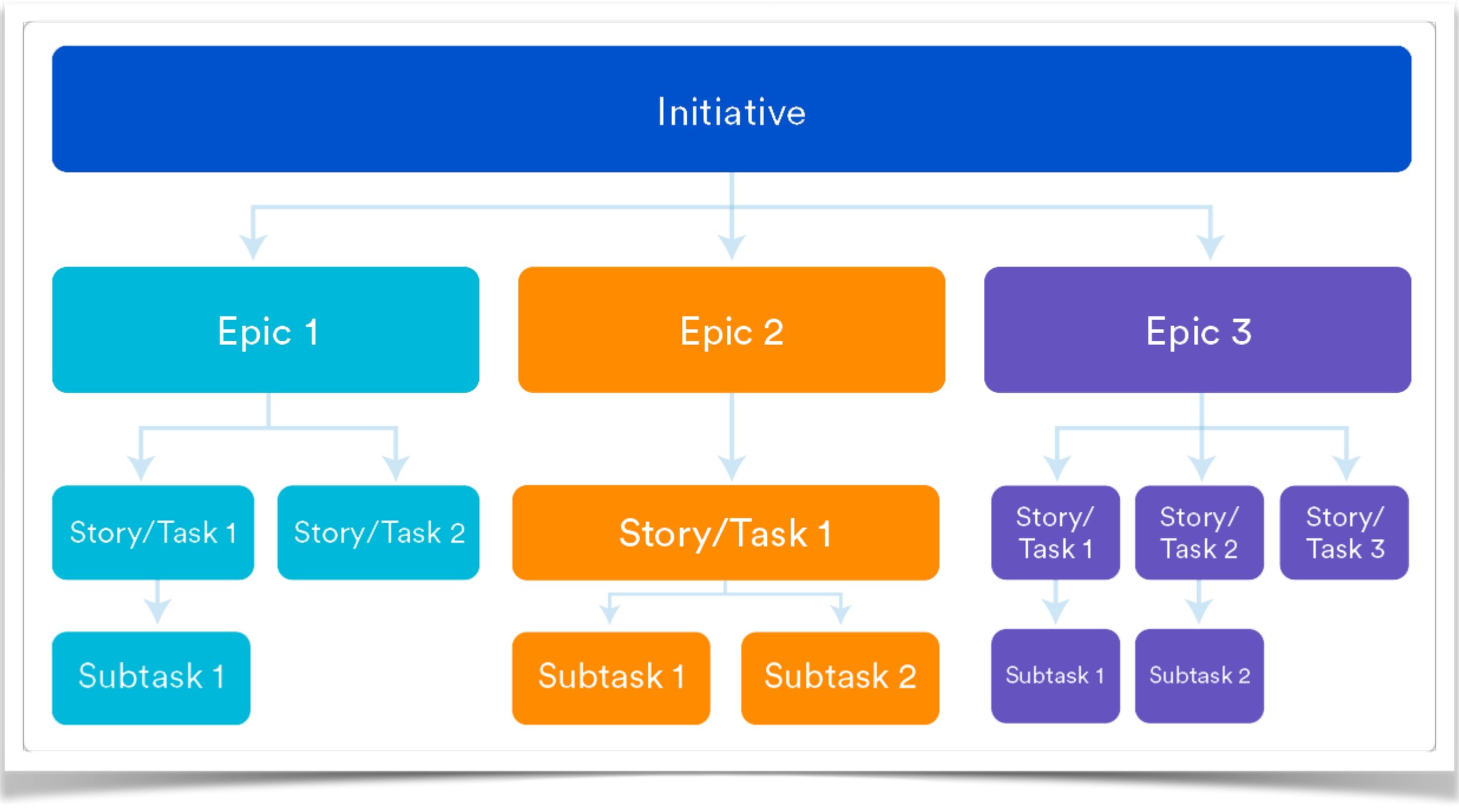
DoD

- Byl vytvořen hosting MySQL serveru
- Je k dispozici pro přijímaní z EspHubServeru a posílání dat do GUI-CORE
- Informace o přístupu do MySQL serveru jsou uloženy na wiki

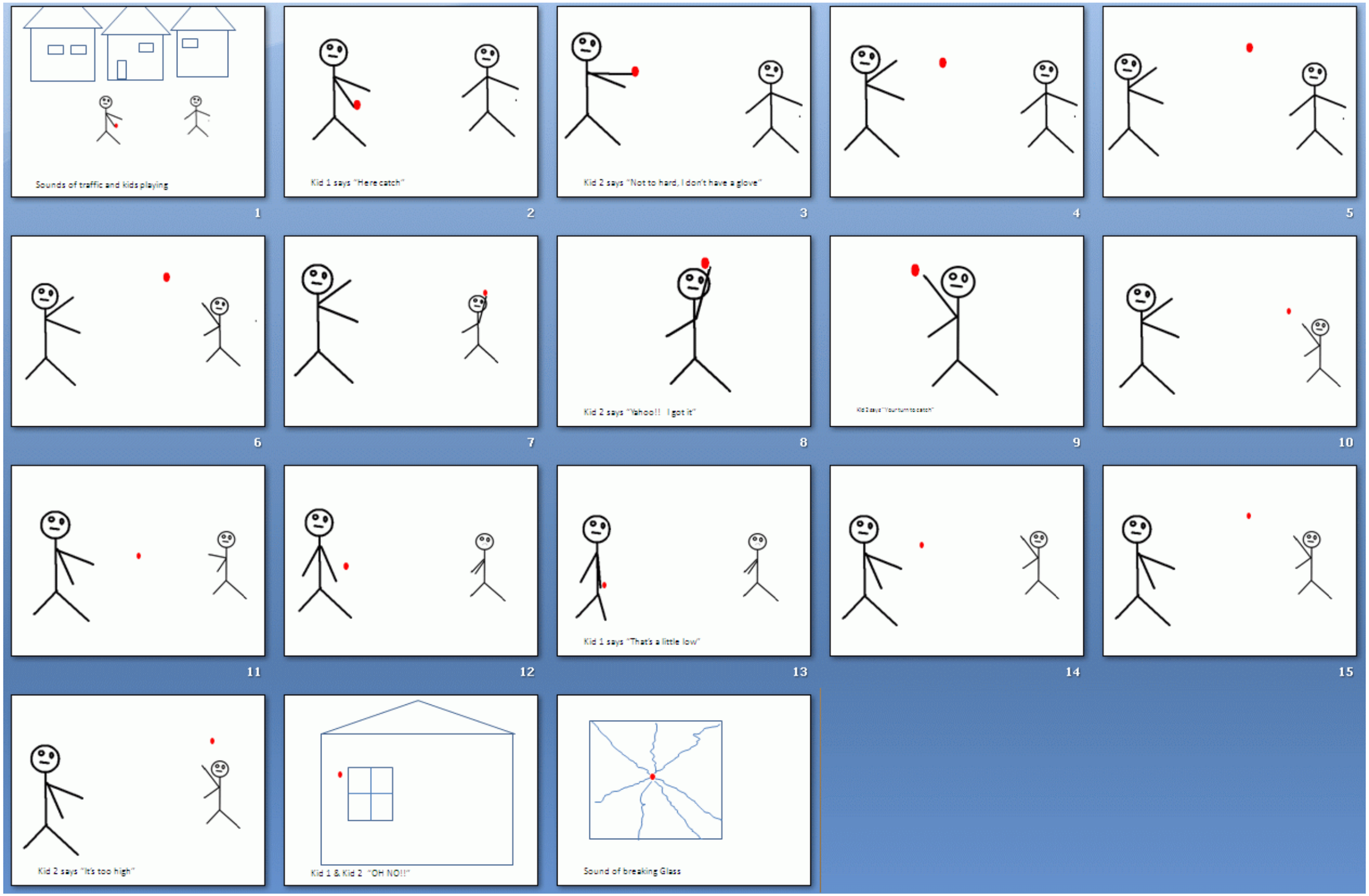
USER STORY

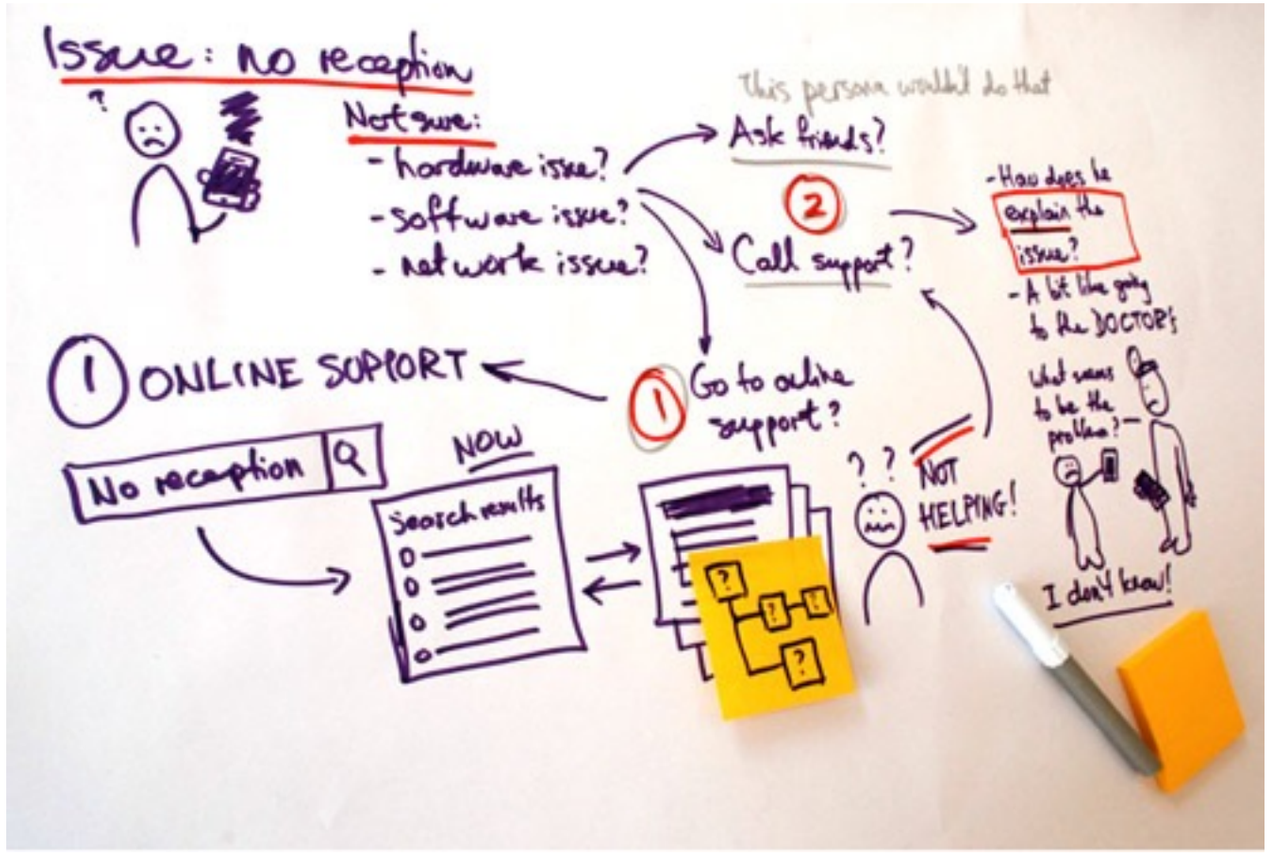
- **Definitions of Ready (DoR)**
  - US je jasná, je připravená pro vývoj
  - US je testovatelná
  - US je doručitelná v nejbližším časovém období
- **Definitions of Done (DoD)**
  - Jakou kvalitu očekáváme?
  - Jak je zajištěna z pohledu bezpečnosti?
  - Jak je zajištěn výkon (bude 10 uživatelů fungovat podobně jako 100 000 uživatelů)?
  - Jak je zajištěna dokumentace (JavaDoc, dokument pro uživatele)?
- **Acceptance Criteria (AC)**
  - Za jakých podmínek UC akceptuje zákazník.

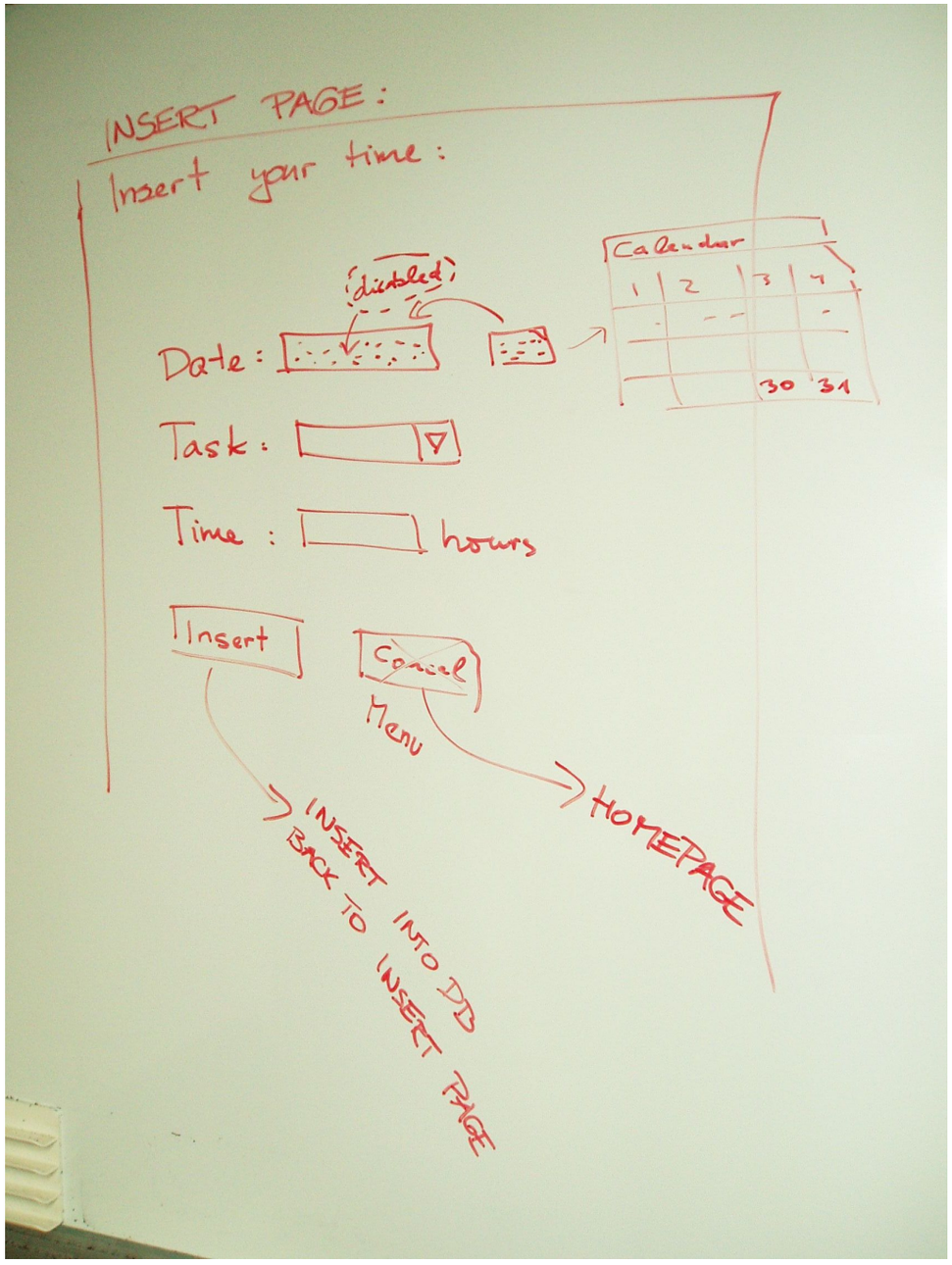
USER STORY

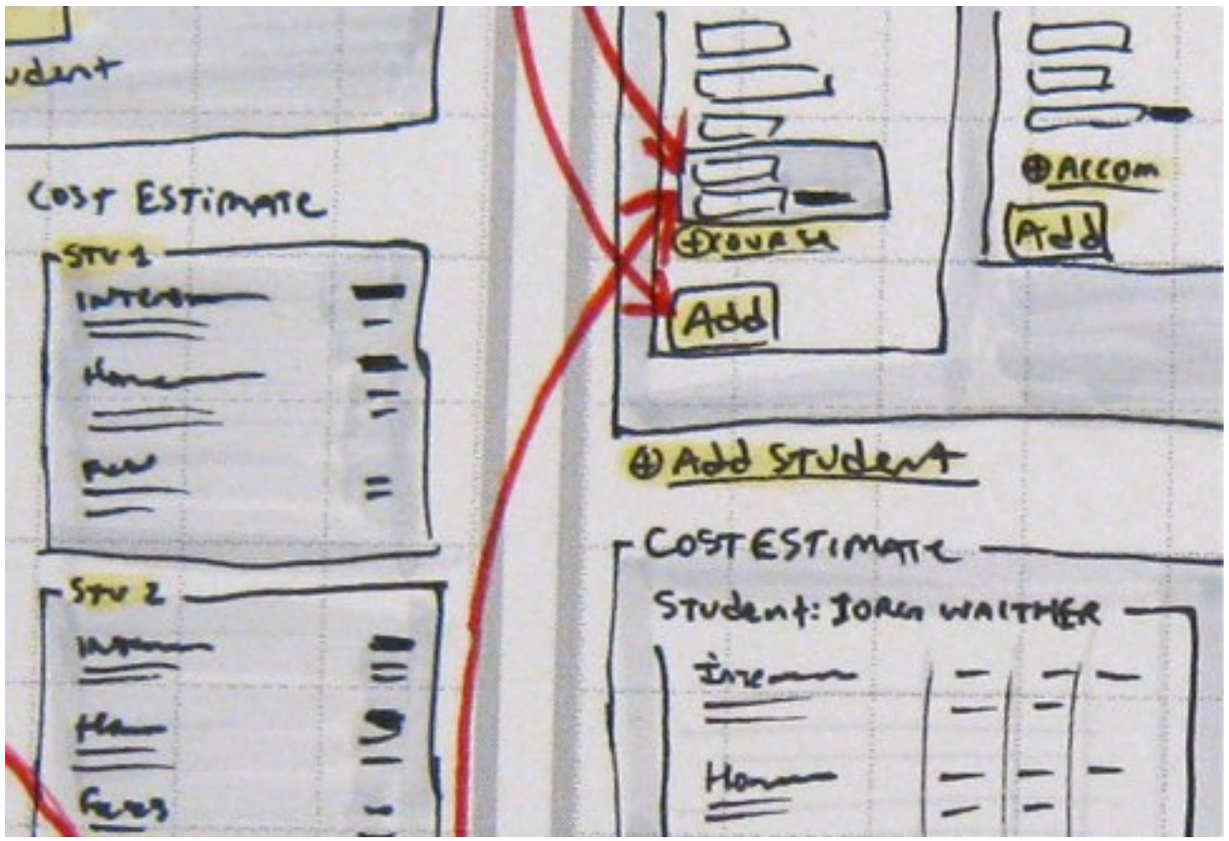

Initiative
Epic 1
Epic 2
Epic 3
Story/Task 1
Story/Task 2
Subtask 1
Story/Task 1
Subtask 1
Subtask 2
Story/ Task 1
Story/ Task 2
Story/ Task 3
Subtask 1
Subtask 2

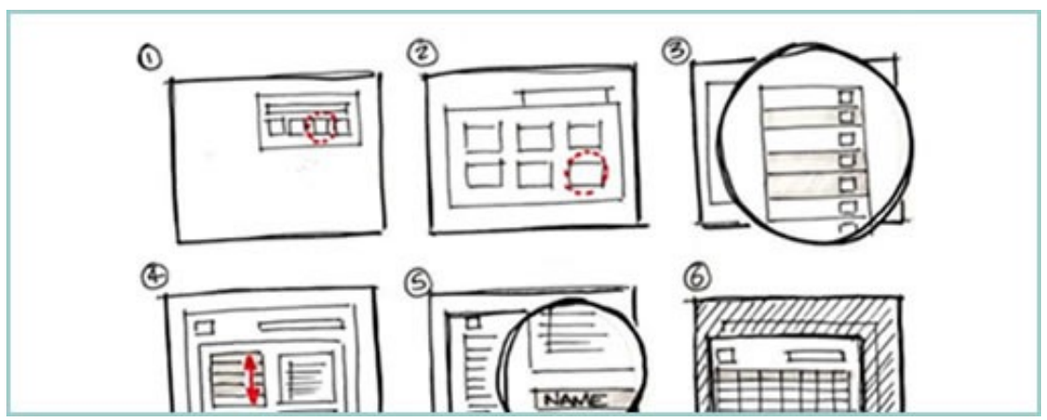
STORYBOARD


Sounds of traffic and kids playing
1
Kid 1 says "Here catch"
2
Kid 2 says "Not to hard, I don't have a glove"
3
4
5
6
7
Kid 2 says "Yahoo!! I got it"
8
9
10
11
12
Kid 1 says "That's a little low"
13
14
15
Kid 2 says "It's too high"
Kid 1 & Kid 2 "OH NO!!"
Sound of breaking Glass


Issue: no reception
Not sure:
- hardware issue?
- software issue?
- network issue?
This person wouldn't do that
Ask friends?
2
Call support?
-How does he explain the issue?
-A bit like going to the DOCTOR's
What seems to be the problem?
I don't know!
1 ONLINE SUPPORT
1 Go to online support?
No reception
NOW
Search results
??
NOT HELPING!


INSERT PAGE:
Insert your time:
Calendar
(dictated)
Date:
Task:
Time:
hours
Insert
Cancel
Menu
HOMEPAGE
INSERT INTO DB
BACK TO INSERT PAGE


COST ESTIMATE
STU 1
STU 2
Add
Add STUDENT
COST ESTIMATE
STUDENT: JÖRG WALTHER

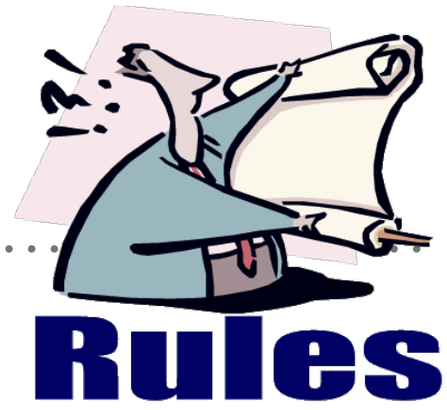

NAME

IEEE 830

IEEE 830

- IEEE Recommended Practice for Software Requirements Specification
- Doporučený přístup ke specifikaci SW požadavků
- Definuje zaměření (scope)
  - Popis praktik SRS
  - SRS za účelem vývoje SW
  - Částečně za účelem výběru SW
- Odkazy na relevantní standardy IEEE
  - IEEE 610.12 – Standard Glossary of Software Engineering Terminology
  - IEEE 1042 – Guide to SW Configurations Management
- Definice
  - Kontrakt, zákazník, dodavatel, uživatel.

NÁPLŇ IEEE 830



- Čím se zabývat:
  - Funkcionalita
  - Externí rozhraní
  - Výkon
  - Atributy
  - Návrhová omezení

- ***SRS by měla být:***
  - *Correct (přesná)*
  - *Unambiguous (jednoznačná)*
  - *Complete*
  - *Consistent*
  - *Ranked for importance (ohodnocená podle důležitosti)*
  - *Verifiable (ověřitelná)*
  - *Modifiable (přizpůsobitelná)*
  - *Traceable (sledovatelná)*

OBSAH SRS PODLE IEEE 830

Table of Contents

PŘÍKLAD – ŠABLONA DLE IEEE 830

Table of Contents

FURPS+

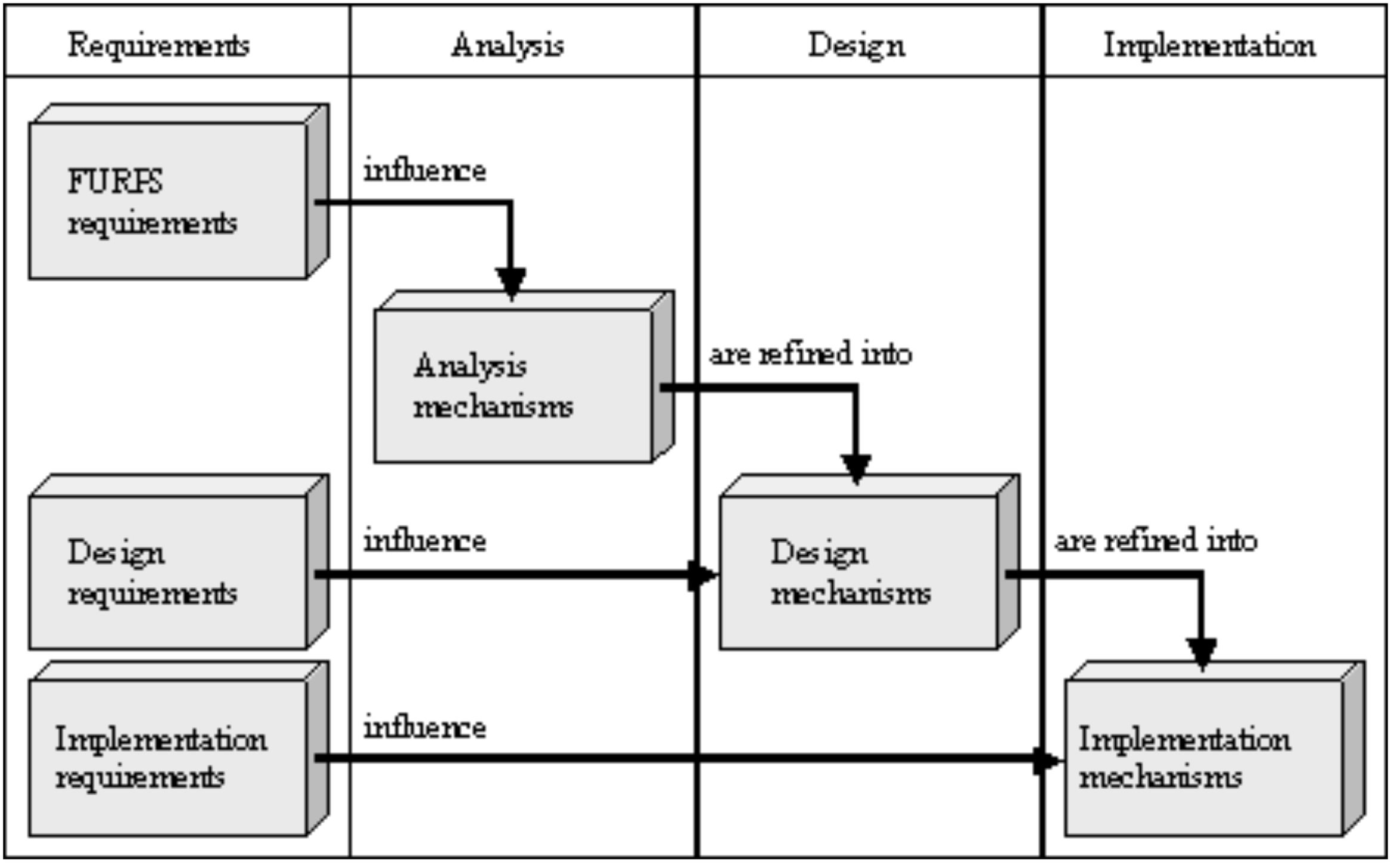
FURPS+

- Systém klasifikace požadavků z pohledu architektury navrhovaného systému
- Autorem Robert Grady (HP), alternativně lze použít standard ISO 9126
- Oblasti:
  - Functionality - funkce, bezpečnost
  - Usability - uživatelské rozhraní, dokumentace, nápověda
  - Reliability - spolehlivost, schopnost zotavení z chyby
  - Performance - odezva, přesnost, propustnost, využitelnost zdrojů
  - Supportability - podporovatelnost, udržitelnost, lokalizace
- \+ znamená, že bychom neměli zapomenout ani na:
  - Návrhové, implementační, fyzické požadavky

FUNKČNÍ POŽADAVKY, KTERÉ OVLIVŇUJÍ ARCHITEKTURU

Function	Description
Auditing	Provide audit trails of system execution.
Licensing	Provide services for tracking, acquiring, installing, and monitoring license usage.
Localization	Provide facilities for supporting multiple human languages.
Mail	Provide services that allow applications to send and receive mail.
Online help	Provide online help capability.
Printing	Provide facilities for printing.
Reporting	Provide reporting facilities.
Security	Provide services to protect access to certain resources or information.
System management	Provide services that facilitate management of applications in a distributed environment.
Workflow	Provide support for moving documents and other work items, including review and approval cycles.

TRANSFORMACE POŽADAVKŮ FURPS DO VÝSLEDNÉHO PRODUKTU

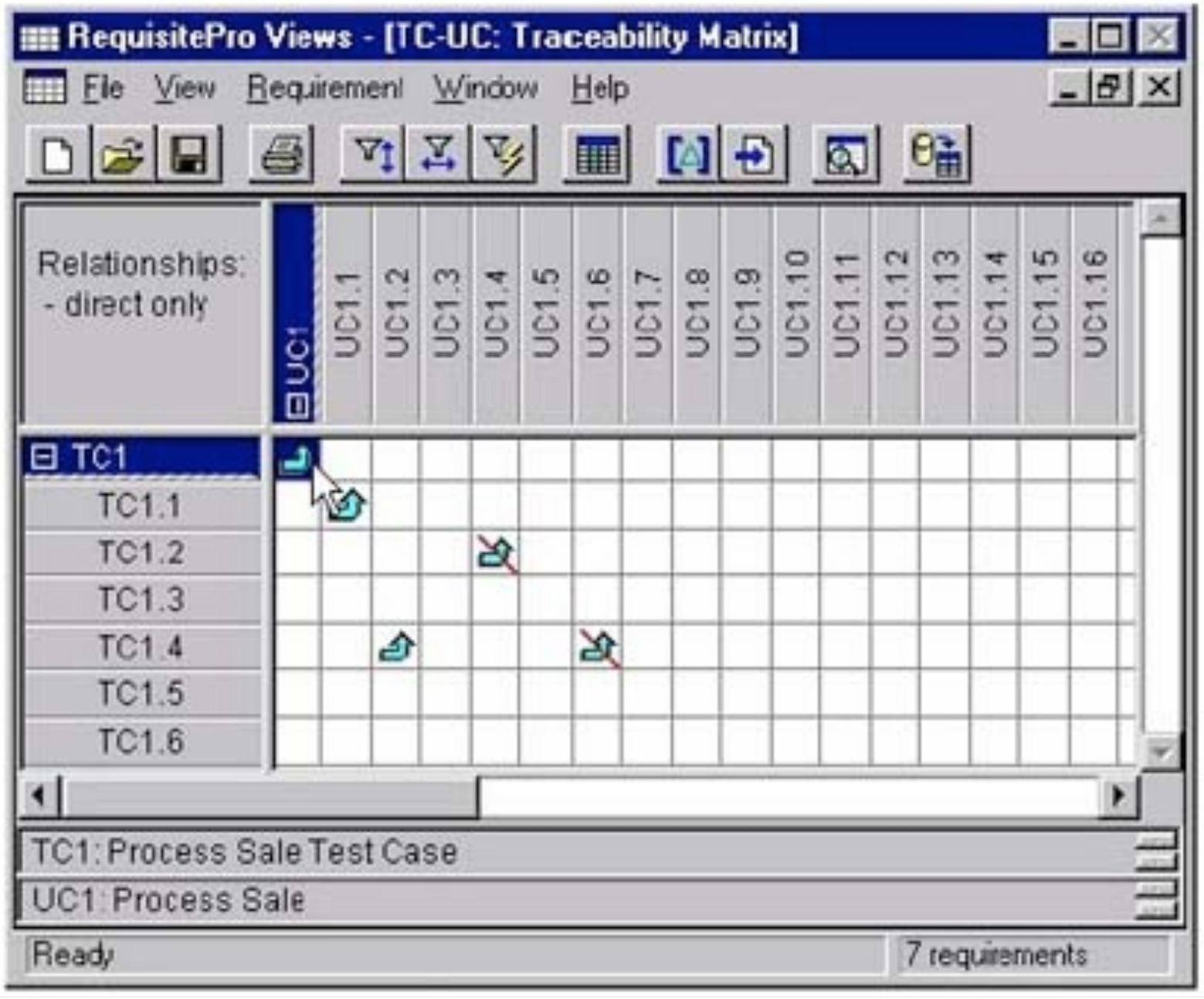


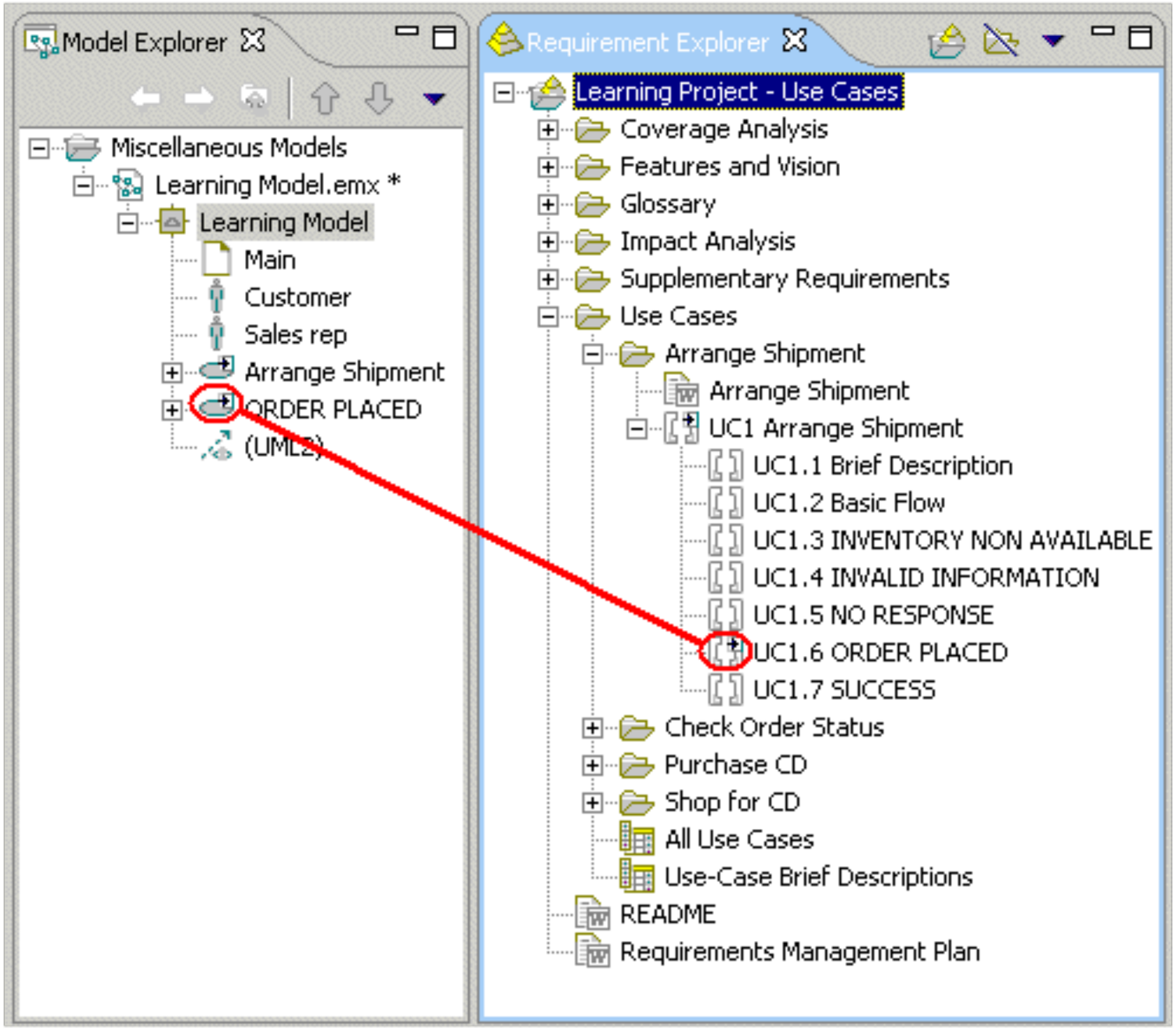
PODPORA NÁSTROJI

APPLICATION LIFECYCLE MANAGEMENT

POŽADAVKY NA NÁSTROJ

- Automatizace a usnadnění činností.
- Kontrola konzistence modelů.
- Traceability.
- Znovupoužití.
- Možnost generování GUI.
- Sledovat a řídit proces vývoje SW.


RequisitePro Views - [TC-UC: Traceability Matrix]
File View Requirement Window Help
Relationships:
- direct only
UC1
UC1.1
UC1.2
UC1.3
UC1.4
UC1.5
UC1.6
UC1.7
UC1.8
UC1.9
UC1.10
UC1.11
UC1.12
UC1.13
UC1.14
UC1.15
UC1.16
TC1
TC1.1
TC1.2
TC1.3
TC1.4
TC1.5
TC1.6
TC1: Process Sale Test Case
UC1: Process Sale
Ready
7 requirements


Model Explorer
Miscellaneous Models
Learning Model.emx *
Learning Model
Main
Customer
Sales rep
Arrange Shipment
ORDER PLACED
(UML2)
Requirement Explorer
Learning Project - Use Cases
Coverage Analysis
Features and Vision
Glossary
Impact Analysis
Supplementary Requirements
Use Cases
Arrange Shipment
Arrange Shipment
UC1 Arrange Shipment
UC1.1 Brief Description
UC1.2 Basic Flow
UC1.3 INVENTORY NON AVAILABLE
UC1.4 INVALID INFORMATION
UC1.5 NO RESPONSE
UC1.6 ORDER PLACED
UC1.7 SUCCESS
Check Order Status
Purchase CD
Shop for CD
All Use Cases
Use-Case Brief Descriptions
README
Requirements Management Plan

[RaQuestSample.rqe] - RaQuest

File Edit View Search Project Tree List Matrix Requirement Tools Help

- On-line bookstore
  - Bussiness Requirement
    - Higher volume - fast
    - Passing on of roles
    - View of customer m
    - Reduce wasted tim
  - Stakeholder Requireme
    - Relation between o
    - Create a secure on-
    - High Volume Throu
    - Efficient stock contr
  - Solution Requirements
    - Functional Requirer
      - Manage User A
        - Add Users
        - Remove Us
        - Report on U
        - Secure Acc
        - Store User I
        - Validate Us
      - Provide Online
        - ShoppingBa
        - Process Cre
      - Manage Invento
        - Receive Bo
        - List Stock L
        - Order Book
        - Store and M
        - Add Books
      - Process Order
        - Package O

Project | Members

ID	Type	Summary	Priority	Status	Risk	Difficulty	Stability
001	Functio...	Higher volume - faster client accessibility.	Medium	Approv...	Medium	Medium	Medium
002	Functio...	Passing on of roles leading to inefficiency ...	High	Proposed	Medium	Medium	High
003	Functio...	View of customer messages directly relate...	High	Proposed	Medium	Medium	High
004	Functio...	Reduce wasted time sending messages t...	High	Proposed	Medium	Medium	High
005	Functio...	Relation between orders and email inquir...	Medium	Proposed	Medium	Medium	Medium
006	Functio...	Create a secure on-line ordering system.	Medium	Validated	Medium	Medium	Medium
007	Functio...	High Volume Through put	Medium	Validated	Medium	Medium	Medium

All Requirements

001 Higher volume - faster client | Update | New

Summary | Detail | User-defined Attributes | UML Items | Files | Members | Test

Summary

Higher volume - faster client accessibility.

ID 001 Revision 3

Version 1.0 Phase 1.0

Due Date 1/20/2011 Type Functional

Created 10/5/2010 5:33:24 PM Status Approved

Created by raquest

Updated 12/10/2010 4:05:05 PM

Updated by Ichiro Approved by Ichiro

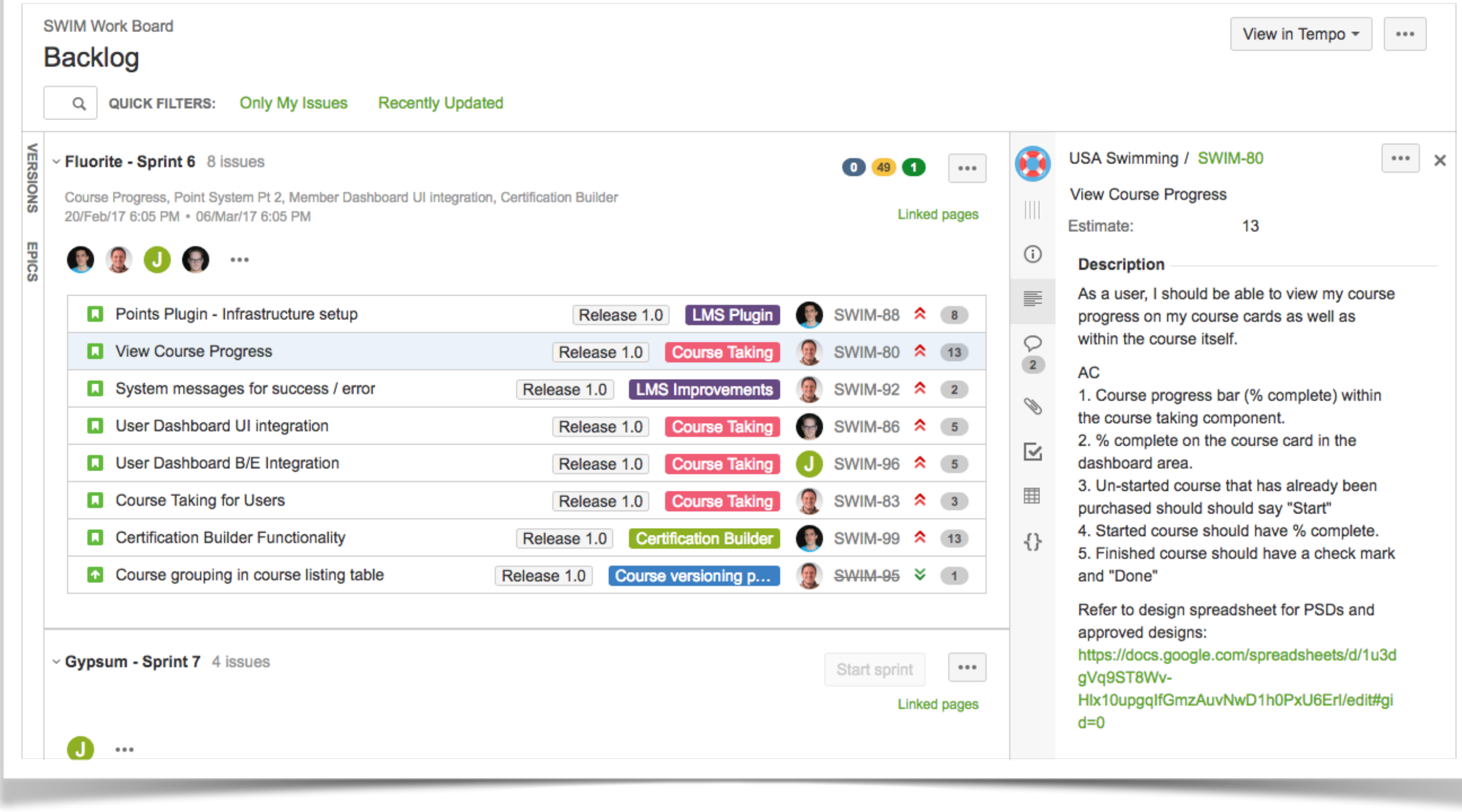
Lock Requirement

Review Required

Approved

Current Baseline: Nothing

JIRA


SWIM Work Board
Backlog
View in Tempo
QUICK FILTERS:
Only My Issues
Recently Updated
VERSIONS
EPICS
Fluorite - Sprint 6 8 issues
0 49 1
Course Progress, Point System Pt 2, Member Dashboard UI integration, Certification Builder
20/Feb/17 6:05 PM • 06/Mar/17 6:05 PM
Linked pages
Points Plugin - Infrastructure setup Release 1.0 LMS Plugin SWIM-88 8
View Course Progress Release 1.0 Course Taking SWIM-80 13
System messages for success / error Release 1.0 LMS Improvements SWIM-92 2
User Dashboard UI integration Release 1.0 Course Taking SWIM-86 5
User Dashboard B/E Integration Release 1.0 Course Taking SWIM-96 5
Course Taking for Users Release 1.0 Course Taking SWIM-83 3
Certification Builder Functionality Release 1.0 Certification Builder SWIM-99 13
Course grouping in course listing table Release 1.0 Course versioning p... SWIM-95 1
Gypsum - Sprint 7 4 issues
Start sprint
Linked pages
USA Swimming / SWIM-80
View Course Progress
Estimate: 13
Description
As a user, I should be able to view my course progress on my course cards as well as within the course itself.
AC
1. Course progress bar (% complete) within the course taking component.
2. % complete on the course card in the dashboard area.
3. Un-started course that has already been purchased should should say "Start"
4. Started course should have % complete.
5. Finished course should have a check mark and "Done"
Refer to design spreadsheet for PSDs and approved designs:
https://docs.google.com/spreadsheets/d/1u3dgVq9ST8Wv-HIx10upgqIfGmzAuvNwD1h0PxU6ErI/edit#gid=0

Application Lifecycle Management

Domain: NEVA, Project: ALM_DEMO, User: neva | Logout

< Back | Forward > | Tools | Help

Dashbo... | Manage... | Releases | Libraries | Require... | Testing | Defects

Releases | Edit | View

- Releases
  - Mercury Tours Applicatio
    - Release 10.5
      - Launch
      - Web Portal Integr
      - Code Freeze
      - Release to Produ
      - Cycle 1 - New Fe
      - Cycle 2 - New Fe
      - Cycle 3 a - Sanity
      - Cycle 3 b - Sanity
      - Cycle 4 a - Full
      - Cycle 4 b - Full

Details | Release Scope * | Master Plan | Scorecard | Status | Attachments

Generate | Scorecard Layout

Release 10.5 Scorecard

	Code Freeze					Release to Production			
	Defects Fixed per Day	Passed Requirements	Rejected Defects	Severe Defects	Tests Executed	Defects Fixed per Day	Passed Requirements	Severe Defects	Tests Executed
Booking System - New Booking Options	2	95 %	10 %	1	100 %	0	100 %	1	100 %
Online Recurring Booking Service	2	95 %	6 %	0	100 %	0	100 %	0	93 %
Reservation Extension Service	2	95 %	5 %	0	100 %	0	100 %	0	100 %
Self Service Profile Management	2	95 %	7 %	0	100 %	0	100 %		

Measured value: 93 of 100
OK value >= 100 %
Critical value<= 90%

Tests Executed

Milestone: Release to Production **Milestone Scope Item:** Online Recurring Booking Service **Last KPI Date:** 7/12/2010

Breakdown Over Time

Close | Help

...ons end when a milestone reaches its due date.

Server Time: 7/29/2010 10:47 AM

Internet | 100%

HP Application Lifecycle Management 11.00