

# Personalizované odporúčanie

## Tímový projekt

## Dokumentácia k riadeniu projektu

Tím číslo: 12  
Akad. rok: 2011/2012  
Predmet: Tímový projekt  
Kontakt: [tim12tp@googlegroups.com](mailto:tim12tp@googlegroups.com)  
Vedúci tímu: Ing. Michal Kompan  
Téma: Odporúčanie  
Názov tímu: 7 out of 10 Precision  
Verzia: 3.0

Bc. Michal Cádrik (SI)  
Bc. Martin Detko (SI)  
Bc. Igor Slotík (SI)  
Bc. Jakub Ševcech (SI)  
Bc. Ľudovít Mydla (IS)  
Bc. Matej Mihalik (SI)

# Obsah

1.Úvod.....	1 - 1
2.Ponuka.....	2 - 1
2.1.Predstavenie členov tímu.....	2 - 1
2.2.Motivácia.....	2 - 2
2.3.Koncepcia riešenia.....	2 - 3
3.Úlohy členov tímu.....	3 - 1
3.1.Zmeny po prvom semestri.....	3 - 1
4.Použité technológie.....	4 - 1
4.1.Jazyk implementácie:.....	4 - 1
4.2.Vývojové prostredia:.....	4 - 1
4.3.Nástroj na manažovanie projektu: .....	4 - 1
4.4.Správa verzií zdrojových kódov: .....	4 - 1
4.5.Databáza: .....	4 - 1
4.6.Testovanie: .....	4 - 1
4.7.Dokumentácia.....	4 - 1
5.Plán projektu.....	5 - 1
5.1.Dátumy začiatkov šprintov.....	5 - 1
5.2.Zoznam požadovaných funkcií (Product backlog).....	5 - 2
5.2.1.Ukončené a rozpracované funkcie v prvom semestri.....	5 - 2
5.2.2.Ďalšie naplánované funkcie na druhý semester.....	5 - 2
5.2.3.Ukončené funkcie v druhom semestri.....	5 - 3
5.3.Celkový plán projektu.....	5 - 3
5.4.Plán šprintu 01_Anofeles.....	5 - 4
5.5.Plán šprintu 02_Bubo_Bubo.....	5 - 6
5.6.Plán šprintu 03_Canis.....	5 - 8
5.7.Plán šprintu 04_Delphinidae_Delphis.....	5 - 10
5.8.Plán šprintu 05_Eptesicus_Fuscus.....	5 - 12
5.9.Plán šprintu 06_Felis Catus.....	5 - 14
5.10.Plán šprintu 07_Glaucomys_Sabrinus.....	5 - 16
5.11.Plán šprintu 08_Haliaeetus_Leucocephalus.....	5 - 17
5.12.Plán šprintu 09_Indicator_Exilis.....	5 - 19
5.13.Plán šprintu 10_Jynx_Torquilla.....	5 - 21
5.14.Plán šprintu 11_Kakamega_Poliothorax.....	5 - 23
5.15.Plán šprintu 12_Leporidae_Cuniculus.....	5 - 24
5.16.Vyhodnotenie plánov po prvom semestri.....	5 - 25
5.17.Vyhodnotenie plánov po druhom semestri.....	5 - 26
6.Zápisy zo stretnutí.....	6 - 1
6.1.Zápis z prvého stretnutia.....	6 - 1
6.2.Zápis z druhého stretnutia.....	6 - 4
6.3.Zápis z tretieho stretnutia.....	6 - 7
6.4.Zápis zo štvrtého stretnutia.....	6 - 9
6.5.Zápis z piateho stretnutia.....	6 - 13
6.6.Zápis zo šiesteho stretnutia.....	6 - 17
6.7.Zápis zo siedmeho stretnutia.....	6 - 21
6.8.Zápis z ôsmeho stretnutia.....	6 - 25
6.9.Zápis z deviateho stretnutia.....	6 - 30
6.10.Zápis z desiateho stretnutia.....	6 - 34
6.11.Zápis z jedenásteho stretnutia.....	6 - 36
6.12.Zápis z dvanásteho stretnutia.....	6 - 38
6.13.Zápis z trinásteho stretnutia.....	6 - 42
6.14.Zápis zo štrnásteho stretnutia.....	6 - 46

6.15.Zápis z pätnásteho stretnutia.....	6.- 50
6.16.Zápis zo šestnásteho stretnutia.....	6.- 54
6.17.Zápis zo sedemnásteho stretnutia.....	6.- 56
6.18.Zápis z osemnásteho stretnutia.....	6.- 58
7.Základná smernica.....	7.- 1
7.1.Konvencia názvov.....	7.- 1
7.2.Kontrola kódu.....	7.- 1
7.3.Povinnosti zapisovateľa stretnutia.....	7.- 2
7.4.Postup nahratia dokumentu na stránku tímu.....	7.- 2
8.Manažment kvality .....	8.- 1
8.1.Roly a zodpovednosti.....	8.- 1
8.2.Vytvorenie novej funkcionality v projekte.....	8.- 1
8.3.Vytvorenie JUnit testu.....	8.- 2
8.4.Použiteľné funkcie na overenie testovania.....	8.- 5
8.5.Použitie JMock testovania.....	8.- 7
8.6.Zmena tvorby JUnit testu.....	8.- 8
9.Manažment rizík .....	9.- 1
9.1.Roly a zodpovednosti.....	9.- 1
9.2.Spracovanie chyby.....	9.- 1
9.3.Vytvorenie novej chyby.....	9.- 3
9.4.Priradenie chyby.....	9.- 4
9.5.Oprava chyby.....	9.- 6
9.6.Vyriešenie chyby.....	9.- 6
9.7.Manažment rizík v druhom semestri.....	9.- 8
10.Manažment rozvrhu .....	10.- 1
10.1.Tvorba rozvrhu.....	10.- 1
10.2.Postup tvorby rozvrhu.....	10.- 1
10.3.Ukončenie šprintu – aktualizovanie stavu úloh v systéme Redmine.....	10.- 1
10.4.Riešenie úlohy.....	10.- 2
10.5.Zatvorenie úlohy.....	10.- 2
10.6.Odmietnutie úlohy.....	10.- 3
11.Manažment podpory vývoja .....	11.- 1
11.1.Systém na správu verzií GIT.....	11.- 1
12.Monitorovanie projektu.....	12.- 1
12.1.Nástroje na monitorovanie projektu.....	12.- 1
12.1.1.Redmine a jeho nástroje.....	12.- 1
12.2.Prehliadky modelov.....	12.- 9
12.2.1.Vypracovanie záznamu o vykonanej prehliadke.....	12.- 11
13.Manažment komunikácie .....	13.- 1
13.1.Nástroje na podporu komunikácie.....	13.- 1
13.2.Nástroje na podporu kolaborácie.....	13.- 1
13.3.Manažment požiadaviek na zmenu.....	13.- 3
13.3.1.Roly a ich zodpovednosti.....	13.- 3
13.3.2.Životný cyklus zmeny.....	13.- 3
13.3.3.Proces identifikácie a vytvorenia požiadavky na zmenu nástroji Redmine.....	13.- 4
13.3.4.Proces prijatia požiadavky na zmenu.....	13.- 5
13.3.5.Proces zamietnutia požiadavky na zmenu.....	13.- 7
13.3.6.Proces prehodnotenia zamietnutia požiadavky na zmenu.....	13.- 7
13.3.7.Proces uzatvorenia zamietnutej požiadavky na zmenu.....	13.- 8
14.Manažment tvorby dokumentácie .....	14.- 1
14.1.Nástroje na tvorbu dokumentácie.....	14.- 1
14.2.Používateľskú príručku.....	14.- 1

14.2.1. Formát dokumentu.....	14.- 3
14.2.2. Konvencia označovania v texte.....	14.- 5
14.2.3. Ukončenie tvorby dokumentu.....	14.- 5
14.2.4. Odovzdanie dokumentu a zdrojových súborov.....	14.- 6
14.3. Dokumentácia k riadeniu.....	14.- 6
14.3.1. Postup úpravy dokumentácie k riadeniu.....	14.- 6
14.4. Dokumentácia k inžinierskemu dielu.....	14.- 6
14.4.1. Postup tvorby dokumentácie k inžinierskemu dielu.....	14.- 7
14.5. Javadoc.....	14.- 7
14.6. Dokumentácia – všeobecné ustanovenia.....	14.- 7
14.6.1. Spôsob editovania dokumentov.....	14.- 7
14.6.2. Zmena úložiska dokumentov.....	14.- 7
15. Zmeny v dokumente.....	15.- 1
Príloha 1- Preberacie protokoly.....	Príloha 1- 1
Príloha 2- Preberacie protokoly.....	Príloha 2- 1
Príloha 3- Preberacie protokoly.....	Príloha 3- 1
Príloha 4- Prihláška na TP Cup.....	Príloha 4- 1
Téma: Personalizované odporúčanie.....	- 1
Predstavenie tímu.....	- 1
Motivácia.....	- 2
Konceptia riešenia.....	- 2
Príloha 5- Preberací protokol pre priebežné odovzdanie v 2. semestri.....	Príloha 5- 1

## Autori jednotlivých častí v dokumentácii riadenia

Autor	Vypracovaná časť
Michal Cádrik	Štruktúra dokumentu
Všetci	2. Ponuka, 3. Úlohy členov tímu, 4. Použité technológie
Matej Mihalik	5. Plán projektu
Ľudovít Mydla	6.1. Zápis z prvého stretnutia
Jakub Ševcech	6.2. Zápis z druhého stretnutia
Igor Slotík	6.3. Zápis z tretieho stretnutia
Martin Detko	6.4. Zápis zo štvrtého stretnutia
Matej Červeňák	6.5. Zápis z piateho stretnutia
Matej Mihalik	6.6. Zápis zo šiesteho stretnutia
Michal Cádrik	6.7. Zápis zo siedmeho stretnutia
Jakub Ševcech	6.8. Zápis z ôsmeho stretnutia
Igor Slotík	6.9. Zápis z deviateho stretnutia
Ľudovít Mydla	6.10. Zápis z desiateho stretnutia
Jakub Ševcech	6.11. Zápis z jedenásteho stretnutia
Igor Slotík	6.12. Zápis z dvanásteho stretnutia
Martin Detko	6.13. Zápis z trinásteho stretnutia
Matej Mihalik	6.14. Zápis zo štrnásteho stretnutia
Michal Cádrik	6.15. Zápis z pätnásteho stretnutia
Jakub Ševcech	6.16. Zápis zo šestnásteho stretnutia
Ľudovít Mydla	6.17. Zápis zo sedemnásteho stretnutia
Igor Slotík	6.18. Zápis z osemnásteho stretnutia
Jakub Ševcech, Michal Cádrik	7. Základná smernica
Martin Detko	8. Manažment kvality
Ľudovít Mydla	9. Manažment rizík
Matej Mihalik	Error: Reference source not found Error: Reference source not found
Igor Slotík	11. Manažment podpory vývoja
Matej Červeňák	12. Monitorovanie projektu
Jakub Ševcech	13. Manažment komunikácie
Michal Cádrik	14. Manažment tvorby dokumentácie
Michal Cádrik	15. Zmeny v dokumente
Jakub Ševcech	Príloha 1 - Preberacie protokoly
Jakub Ševcech	Príloha 2 - Preberacie protokoly
Michal Cádrik	Príloha 3 - Preberacie protokoly

Michal Cádrik, Jakub Ševcech	Príloha 4 - Prihláška na TP Cup
Michal Cádrik	Príloha 5 - Preberací protokol pre priebežné odovzdanie v 2. semestri

Tabuľka 1.I: Prehľad autorov jednotlivých častí dokumentu k riadeniu projektu

## Autori jednotlivých častí v dokumentácii inžinierskeho diela

Autor	Vypracovaná časť
Michal Cádrik	Štruktúra dokumentu a kostry jednotlivých šprintov
Michal Cádrik	Kolaboratívne odporúčanie, p. p. 02
Michal Cádrik	Prerobenie kolaboratívneho odporúčania na spôsob pluginov
Jakub Ševcech	Vytvorenie dátového úložiska, t. p. T01
Matej Mihalik	Analýza pridávania algoritmov, t. p. T02
Martin Detko	Upload dát, p. p. 07
Jakub Ševcech	Webová služba, p. p. 06
Matej Červeňák	Analýza pridávania algoritmov, t. p. T02
Eudovít Mydla	Odporúčanie založené na obsahu, p.p. 01
Jakub Ševcech	Spätná väzba, p.p. 09
Matej Mihalik	Výpočet podobnosti, p.p. 10
Jakub Ševcech, Matej Mihalik	Jednoduché pridávanie ďalších algoritmov, p.p. 05
Martin Detko	Vytvorenie dávkovača na spracovanie dát. p.p. 14
Michal Cádrik	Prerobenie kolaboratívneho odporúčania na spôsob pluginov, p.z. 01
Jakub Ševcech	Identifikácia používateľa pomocou cookies a nie IP adresy, p.z. 03
Igor Slotík	Vytvorenie metriky systémom pluginov, p.z. 04
Martin Detko	Príloha A, Príloha B, Príloha C
Jakub Ševcech	Príloha D – Prototyp > Architektúra systému, Konfigurácia serveru
Martin Detko	Príloha D – Prototyp > Dátový model
Jakub Ševcech	Príloha E - Používateľská príručka
Jakub Ševcech	Príloha F – Dátový model po prvom semestri
Michal Cádrik	2.6.1. Vytvorenie kombinovačov systémom pluginov, p. p. 15 > kaskádový kombinovač
Igor Slotík	2.6.1. Vytvorenie kombinovačov systémom pluginov, p. p. 15 > váhový kombinovač
Eudovít Mydla	2.6.1. Vytvorenie kombinovačov systémom pluginov, p. p. 15 > prepínací kombinovač
Jakub Ševcech	2.6.2. Nasadiť New Relic, t. p. T04.1
Jakub Ševcech	2.6.3. Capistrano, t. p. T04.2
Jakub Ševcech	2.6.4. Validácia XML, t. p. T04.3
Matej Mihalik	2.6.5. Prístup do DB v jave z ruby, t. p. T05.1
Matej Mihalik	2.6.6. Vkladanie testovacích dát do DB, t. p. T05.2

Michal Cádrik	2.7.1. Zmena kombinovačov a odporúčačov, t. z. TZ01
Michal Cádrik	2.8.1. Riešiče problému studeného štartu, p. p. 16
Ľudovít Mydla	2.8.2. Predspracovanie dokumentu, p. p. 18
Matej Mihalik	2.8.3. Vytvorenie základného pluginu na získavanie keywordov, p. p. 18.2
Igor Slotík	2.9.1. Precision a Recall metriky, p. p. 19
Jakub Ševcech	2.9.2. Spúšťanie metrik, t.p. TP06.1
Jakub Ševcech	2.9.3. Indexácia a migrácia DB, t.p. TP06.2
Jakub Ševcech	2.9.4. Spúšťanie podobnosti, t.p. TP06.3
Jakub Ševcech	2.9.5. Testovanie výkonu servera, t.p. TP06.4
Michal Cádrik	2.9.6. Zmena odporúčačov podľa novej špecifikácie, t. z. TZ02
Martin Detko	2.9.7. Implementácia keyword algoritmov do procesu importovania, t. z. TZ03
Ľudovít Mydla	2.10.1. Implementácia Porterovho algoritmu, p. p. 20
Martin Detko	2.10.2. Podpora AB testovania, p. p. 21
Michal Cádrik	2.10.3. Zmena kombinovačov a odporúčačov, t. z. TZ05.1
Jakub Ševcech	2.11.1. Pridanie tabuľky lemmas do migrácií, t. p. T07
Martin Detko	2.11.2. Zmena importu relácií - doplnenie argumentu, t. z. TZ06.1
Jakub Ševcech	2.11.3. Prerobiť API pre feedback, t.z. TZ06.2
Martin Detko, Jakub Ševcech	Príloha G – Dátový model po druhom semestri
Jakub Ševcech	Príloha H – Opis grafického používateľského prostredia
Jakub Ševcech	Príloha H – Opis API
Matej Mihalik	Príloha H - Používateľská príručka po druhom semestri > Získavanie kľúčových slov
Michal Cádrik	Príloha H - Používateľská príručka po druhom semestri > Opis algoritmov > Odporúčacie algoritmy > Opis kolaboratívneho odporúčania
Michal Cádrik	Príloha H - Používateľská príručka po druhom semestri > Opis algoritmov > Opis riešičov problému studeného štartu
Ľudovít Mydla	Príloha H - Používateľská príručka po druhom semestri > Opis algoritmov > Odporúčacie algoritmy > Odporúčanie založené na obsahu a podkapitola Spracovanie textu
Igor Slotík	Príloha H - Používateľská príručka po druhom semestri > Metriky
Jakub Ševcech	Príloha I, Príloha J, Príloha K
Michal Cádrik	Príloha L

Tabuľka 1: Autori jednotlivých častí v dokumentácii inžinierskeho diela



## Zoznam skratiek

Skratka	Plné znenie
AIS	Akademický informačný systém
API	Aplication interface
GUI	Graphic user interface
IDE	Integrated development enviroment
INFO	Informácie
MVC	Model view controller
TORCS	The open racing car simulator

Tabuľka 1.II: Zoznam skratiek použitých v dokumente

## Slovník pojmov

Pojem	Význam
Test-driven-development	Vývoj riadený testami

Tabuľka 1.III: Slovník pojmov použitých v dokumente

## **1. Úvod**

V tomto dokumente sa nachádza dokumentácia k riadeniu projektu „Personalizované odporúčanie“. Projekt vypracováva tím: Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihálik, Ľudovít Mydla, Igor Slotík, a Jakub Ševcech. Tím pracuje pod vedením Ing. Michala Kompana.

Dokument obsahuje deväť kapitol. Prvá kapitola obsahuje úvod a opisuje organizáciu dokumentu a obsah jednotlivých jeho častí. Druhá kapitola „Ponuka“ obsahuje ponuky vypracované pri uchádzaní sa o projekty. Tretie kapitola opisuje úlohy a zodpovednosti jednotlivých členov tímu. V nasledujúcej kapitole sú opísané použité technológie. Kapitola číslo päť obsahuje plán projektu. V šiestej kapitole sú vložené zápisy z pravidelných stretnutí. Metodiky použité pri vývoji projektu sú uvedené v kapitole číslo sedem. V predposlednej kapitole je uvedený zoznam zmien v dokumente. V poslednej kapitole sa nachádzajú preberacie protokoly, potvrdzujúce odovzdanie dokumentácií.

## 2. Ponuka

**Autor:** Všetci

### 2.1. Predstavenie členov tímu

#### **Bc. Michal Cádrik:**

Programuje v jazyku Java (IDE Eclipse), C/C++ (IDE Visual Studio, Qt). Bc. Michal Cádrik má skúsenosti s používaním operačných systémov Windows a Ubuntu. Javadocs používa ako jeden z nástrojov na vytváranie dokumentácie. Písanie dokumentácie v kancelárskych balíkoch MS Office a LibreOffice zvláda na dobrej úrovni. V bakalárskej práci sa venoval riešeniu hry Sudoku pomocou logických pravidiel. V poslednej dobe sa zaujíma o neurónové siete a fuzzy systémy.

#### **Bc. Martin Detko:**

Výborne ovláda programovacie jazyky C++ a C. Menšiu skúsenosť má s Javou. S týmito jazykmi robí v prostrediach Eclipse, Visual Studio, Qt, CodeBlocks, DevCpp. Bc. Martin Detko je používateľ operačného systému Microsoft Windows. Zadaním jeho bakalárskej práce bolo vytvoriť inteligentného agenta do prostredia TORCS, čo hovorí o jeho záujme o umelú inteligenciu, na ktorú zameriava aj svoje ďalšie štúdium na fakulte voľbou výberových predmetov ako neurónové siete a fuzzy systémy.

#### **Bc. Ľudovít Mydla:**

Na škole v rámci projektov používa najmä Javu + IDE Eclipse. Menšie (minimálne) znalosti má s používaním C, C++, Ruby. Mimo školy, v práci, používal Javu, JSP (v IDE IntelliJ), MSSQL, ObjectiveC (v IDE Xcode) na MacOS pri vývoji aplikácie pre iPhone, ale momentálne pracuje s PHP + MySQL + JS + CSS pri vývoji webových stránok s použitím CMS systému Wordpress alebo MVC Frameworku Codeigniter. Má skúsenosť s SVN a Git ako systémami na verziovanie, javadocom a systémom na sledovanie chýb Bugzilla. Momentálne sa zaujíma o vývoj pre mobilné platformy.

#### **Bc. Igor Slotík:**

Má veľkú skúsenosť s programovaním v Jave a C. Ako programovacie prostredia používa DevCpp, Visual Studio 2005, Netbeans, Eclipse. Používa hlavne Windows a má skúsenosť aj s Ubuntu. Na tvorbu a integráciu dokumentácie používa aj Javadoc. Vie robiť s HTML, CSS, Javascript. Bakalárska práca Bc. Igor Slotíka bola Automatická vizuálna detekcia inteligentnou kamerou. Získal hlboké skúsenosti z rôznych prístupov na detekciu ľudí, farebných modelov, ktoré sa používajú na detekciu ľudí a AdaBoost klasifikátora. Svoje riešenie implementoval pre inteligentnú kameru a pre kameru notebooku. Vie porovnávať výhody a nevýhody rôznych implementácií na detekciu ľudí.

#### **Bc. Jakub Ševcech:**

Má skúsenosť s programovacími jazykmi C/C++, Java, Ruby a JavaScript. Používa prostredia ako VisualStudio, CodeBlocks, Eclipse alebo RubyMine. V súčasnosti programuje v jazyku Ruby a na správu verzií používa systém Git. Má skúsenosti so systémom Redmine, ako systémom

na sledovanie chýb a systémom na spravovanie projektov. Má skúsenosti s test-driven-development a s vytváraním testov a špecifikácií prostredníctvom cucumber testov. Ako bakalársku prácu navrhol a realizoval metódu na pridávanie poznámok do webových stránok. Metódu implementoval ako webovú službu v jazyku Ruby a použitím frameworku Ruby on Rails, ako rozšírenie výučbového systému ALEF vyvíjaného na fakulte.

### **Bc. Matej Mihalik:**

Má dobré skúsenosti z prostredia tvorby webových stránok a aplikácií, kde používa technológie HTML + CSS + JavaScript (jQuery) + PHP + SQL (PostgreSQL) + formát SVG, s využitím frameworku CodeIgniter, podporujúceho MVC architektúru. Skúsenosti má s programovaním v jazyku Java (prostredia Eclipse a NetBeans) a menšie skúsenosti s jazykmi C/C++ (VisualStudio, CodeBlocks). Bakalársku prácu vypracoval na tému Podpora vyhodnocovania webových dotazníkov.

### **Bc. Matej Červeňák:**

Má dobré skúsenosti z návrhu informačného systému, s prácou v jazyku UML a s nástrojom CASE studio, ktoré získal v predchádzajúcom štúdiu. Ďalej má znalosti jazykov Java (prostredie NetBeans) a MySQL. Používa operačný systém Windows a má menšie skúsenosti s prácou v prostredí operačného systému Fedora. Na výbornej úrovni ovláda grafické programy Photoshop, Illustrator a Flash, v ktorých pracuje s dizajnom webov, printovou grafikou a fotografiami. V bakalárskej práci sa inšpiroval práve touto tematikou - napodobňovanie fotografických filmových materiálov pomocou grafických editorov. Matej Červeňák sa zaujíma o analýzu a návrh systémov.

## **2.2. Motivácia**

Množstvo webových stránok a dokumentov robia web neprehľadným. Ľudia častokrát spotrebujú veľa času na nájdenie relevantných dokumentov, o ktoré majú záujem. Pomoc pri hľadaní zaujímavých článkov a šetrenie času je pre nás veľká výzva.

Viacerí členovia tímu sa zaujímajú o témy strojového učenia a veríme, že pri implementácii metód odporúčania využijeme ich záujem o podobné algoritmy. Takisto máme skúsenosť s prácou na skrytých Markovových modeloch, ktoré sú využívané pri niektorých metódach. V každom prípade máme základnú znalosť problémovej oblasti, na ktorej môžeme ďalej stavať. A najmä, ktorú si veľmi radi rozšírime.

Vieme, že v tejto oblasti sa spracováva veľké množstvo údajov. Používajú sa na to rôzne metódy ako regulárne výrazy, rôzne štatistické postupy a podobne. K tomuto sa viaže predmet Vyhľadávanie informácií, ktorý niektorí z nás majú tento semester zapísaný a vidíme to ako vhodnú príležitosť naučiť sa niečo, čo aj ďalej využijeme nie len v tomto predmete.

Motivuje nás hlavne:

1. Výzva, ako zladit' požiadavky od množstva ľudí na fakulte, ktorí pracujú s metódami na odporúčanie.
2. Naučenie sa niečoho, čo vieme využiť aj mimo projektu, ako napríklad regulárne výrazy, štatistické metódy spracovania dokumentov a rôzne postupy na spracovanie informácií v dokumentoch.
3. Chceme pracovať na zlepšení kvality vyhľadávania a odporúčania, a tak pomôcť pri riešení súčasného zahltenia informáciami.

4. Vytvoríme aplikáciu, ktorá sa bude reálne používať pri výskume na fakulte a môže tak pomôcť k zlepšeniu výsledkov študentov a fakulty samotnej.

### 2.3. Konceptia riešenia

Vytvoríme knižnicu najznámejších a najpoužívanejších existujúcich (state-of-the-art) metód na odporúčanie na základe obsahu a metód založených na kolaboratívnom prístupe. Táto knižnica (predpokladane napísaná v Jave) bude prístupná pre rôzne iné programovacie jazyky. Keďže na fakulte viacerí ľudia pracujú s programovacími jazykmi Ruby a C#, chceme používanie knižnice z týchto jazykov zabezpečiť napríklad vytvorením knižnice pre jazyk C# alebo vytvorením gemu pre jazyk Ruby na základe spomínanej knižnice. Pre ostatné jazyky a platformy chceme implementované metódy sprístupniť prostredníctvom webovej služby.

Knižnicu navrhujeme a implementujeme rozšíriteľným spôsobom tak, aby sme nechali používateľom možnosť na jednoduché začlenenie nových metód k už implementovaným metódam. Takto sa bude dať knižnica jednoducho rozširovať a bude možné držať krok s vývojom v oblasti odporúčania.

Okrem knižnice metód na odporúčanie vytvoríme aj jednotné rozhranie pre uchovávanie a prístup k dátovým vzorkám používaných na overovanie navrhnutých riešení. Budeme poskytovať rozhranie na použitie týchto dátových vzoriek na overovanie iných metód. Takto si môže vývojár novej metódy odporúčania porovnať svoju metódu s už existujúcimi metódami na rovnakých údajoch.

Vytvoríme jednotný prístup na jednotné vytváranie a uchovávanie rôznych typov modelov používateľov a na použitie takto vytvorených modelov pri overovaní rôznych metód.

### 3. Úlohy členov tímu

**Autor:** Všetci

#### **Manažér komunikácie**

Jakub Ševcech. Má na starosti delegovanie úloh na ďalších členov v tíme a vyhodnocovanie týchto úloh. Okrem toho sa snaží motivovať členov tímu, aby pracovali čo najefektívnejšie. Ako manažér komunikácie sprostredkováva komunikáciu medzi zákazníkom a členmi tímu.

#### **Manažér rizík**

Ľudovít Mydla. Snaží sa o efektívne využitie členov v tíme, teda snaží sa delegovať úlohy podľa schopností jednotlivých členov tímu. Snaží sa predvídať možné problémy počas vývoja predchádzať im.

#### **Manažér plánovania**

Matej Mihálik. Vytvára plán pre tím a pre jednotlivých členov tímu. Vyhodnocuje jeho plnenie a následne v ňom robí úpravy, poprípade ho spresňuje.

#### **Manažér kvality**

Martin Detko. Stará sa o kontrolu kvality procesov a vytváranie vzorových testov. Sleduje, či sú všetky komponenty dôkladne otestované.

#### **Manažér monitorovania**

Matej Červeňák. Monitoruje celý projekt a jeho aspekty, to jest sledovanie postupu pri riešení problémov, sledovanie správnosti odhadov, teda ich napĺňanie a mnoho iných. Okrem toho sleduje aj kvalitu výsledku a čas potrebný k realizácií.

#### **Manažér podpory vývoja**

Igor Slotík. Stará sa o verziovanie produktu. Zabezpečuje technologickú podporu integrovania výsledku a aj ostatných činností.

#### **Manažér dokumentácie**

Michal Cádrik. Riadi proces tvorby dokumentácie jednotlivých procesov. Vyberá nástroje pre dokumentáciu a vytvára šablóny pre jednotlivé časti. Navrhne kostru a štruktúru jednotlivých dokumentov. Pred odovzdaním kontroluje korektnosť dokumentov a taktiež ich pridávanie na stránku tímu.

#### **3.1. Zmeny po prvom semestri**

Na začiatku druhého semestra nastala zmena v zodpovednosti za úlohy pre dvoch členov tímu. Igor Slotík a Jakub Ševcech si medzi sebou vymenili svoje úlohy. Igor Slotík má teda v druhom semestri rolu manažéra komunikácie a Jakub Ševcech rolu manažéra podpory vývoja. Táto zmena vznikla z iniciatívy Jakuba Ševcecha, pretože si myslel že v novej role bude pre tím užitočnejší.

## 4. Použité technológie

**Autor:** Všetci

### 4.1. Jazyk implementácie:

Implementácia bude rozdelená na dve časti:

- implementáciu algoritmov na odporúčanie a ďalších pomocných algoritmov v jazyku Java
- webovú službu, ktorá tvorí rozhranie medzi používateľmi a implementovanými algoritmi v jazyku JRuby

### 4.2. Vývojové prostredia:

Keďže bude implementácia v dvoch jazykoch, použijeme aj dve vývojové prostredia:

- Eclipse pre vývoj v jazyku Java (<http://www.eclipse.org/>)
- RubyMine pre jazyk jRuby/Ruby (<http://www.jetbrains.com/ruby/>)

### 4.3. Nástroj na manažovanie projektu:

- Na manažovanie projektu používame nástroj Redmine. Vytvorili sme vlastnú inštanciu tohto nástroja dostupnú na adrese (<http://vm28.ucebne.fiit.stuba.sk/Redmine>).

### 4.4. Správa verzií zdrojových kódov:

- GIT.

### 4.5. Databáza:

- PostgreSQL

### 4.6. Testovanie:

- Pri testovaní využívame metodiku TDD (Test-driven-development).
- Pri testovaní komponentov v jazyku Java používame framework JUnit (<http://junit.org/>).
- Pri testovaní komponentov v jazyku jRuby používame frameworky RSpec (<http://rspec.info/>) a Capybara (<https://github.com/jnicklas/capybara>).

### 4.7. Dokumentácia

- Na tvorbu dokumentov používame nástroj LibreOffice.
- UML diagramy modelujeme v nástroji Software Ideas Modeler.
- Na dokumentáciu zdrojového kódu používame pre Javu Javadoc a pre Ruby Rdoc.



## **5. Plán projektu**

Pri vývoji produktu postupujeme metodikou scrum, ktorá sa zaraďuje medzi agilné metodiky vývoja softvéru. Pri tejto metodike nevytvárame definitívny plán, ale postupne plán dopĺňame a rozširujeme na začiatku každého šprintu. Preto táto časť dokumentu nie je finálna, ale sa bude priebežne dopĺňať o ďalšie časti.

### **5.1. Dátumy začiatkov šprintov**

- 13.10.2011 – Začiatok šprintu 01\_Anofeles
- 27.10.2011 – Začiatok šprintu 02\_Bubo\_Bubo
- 10.11.2011 – Začiatok šprintu 03\_Canis
- 24.11.2011 – Začiatok šprintu 04\_Delphinidae\_Delphis
- 08.12.2011 – Začiatok šprintu 05\_Eptesicus\_Fuscus
- 14.02.2012 – Začiatok šprintu 06\_Felis Catus
- 20.02.2012 – Začiatok šprintu 07\_Glaucomys\_Sabrinus
- 27.02.2012 – Začiatok šprintu 08\_Haliaeetus\_leucocephalus
- 05.03.2012 – Začiatok šprintu 09\_Indicator\_Exilis
- 12.03.2012 – Začiatok šprintu 10\_Jynx\_Torquilla
- 26.03.2012 – Začiatok šprintu 11\_Kakamega\_Poliothorax
- 02.04.2012 – Začiatok šprintu 12\_Leporidae\_Cuniculas

## 5.2. Zoznam požadovaných funkcií (Product backlog)

### 5.2.1. Ukončené a rozpracované funkcie v prvom semestri

ID	Funkcionalita	Hodnotenie
T01	Vytvorenie dátového úložiska	14
02	Kolaboratívne odporúčanie	16
T02	Analýza pridávania algoritmov	4
06	Webová služba	15
07	Upload dát	12
08	Získanie odporúčania	6
09	Spätná väzba	5
10	Výpočet podobnosti	24
01	Odporúčanie na základe obsahu	11
13	Vyhodnocovanie	15
05	Prerobenie na plugin systém jadra programu	30
14	Vytvorenie dávkovača na spracovanie dát	8

Tabuľka 5.1: Ukončené a rozpracované funkcie v prvom semestri

### 5.2.2. Ďalšie naplánované funkcie na druhý semester

ID	Funkcionalita	Odhadované hodnotenie
15	Kombinácia algoritmov (hybridné)	50
16	Cold-start problém	20
17	AB testovanie	30
18	Vytvoriť nástroj na integráciu odporúčania do ľub. stránky	15
19	Výpočet podobnosti sprístupniť vonku	6
20	Identifikácia používateľa v rôznych systémoch	6

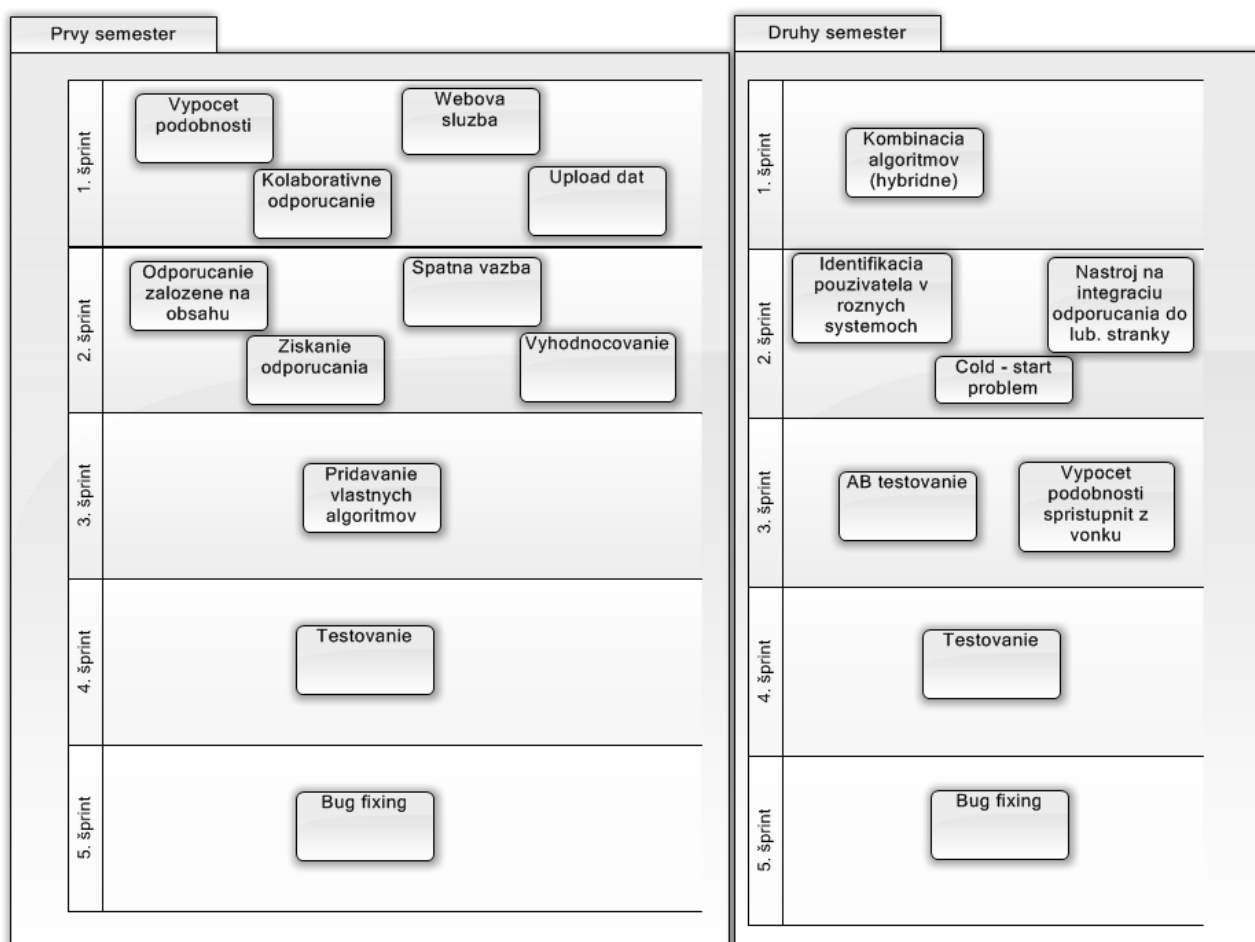
Tabuľka 5.2: Ďalšie naplánované funkcie na druhý semester

### 5.2.3. Ukončené funkcie v druhom semestri

ID	Funkcionalita	Hodnotenie
15	Kombinácia algoritmov (hybridné)	52
16	Cold-start problém	16
17	AB testovanie	6
21	Zabezpečenie bezpečnosti systému	20
22	Získavanie keywordov z dokumentov	31
T03	Rôzne zmeny, vylepšenia, optimalizácia a refaktoring	119.5
T04	Integrácia a Testovanie systému	56

Tabuľka 5.3: Ukončené funkcie v druhom semestri

### 5.3. Celkový plán projektu



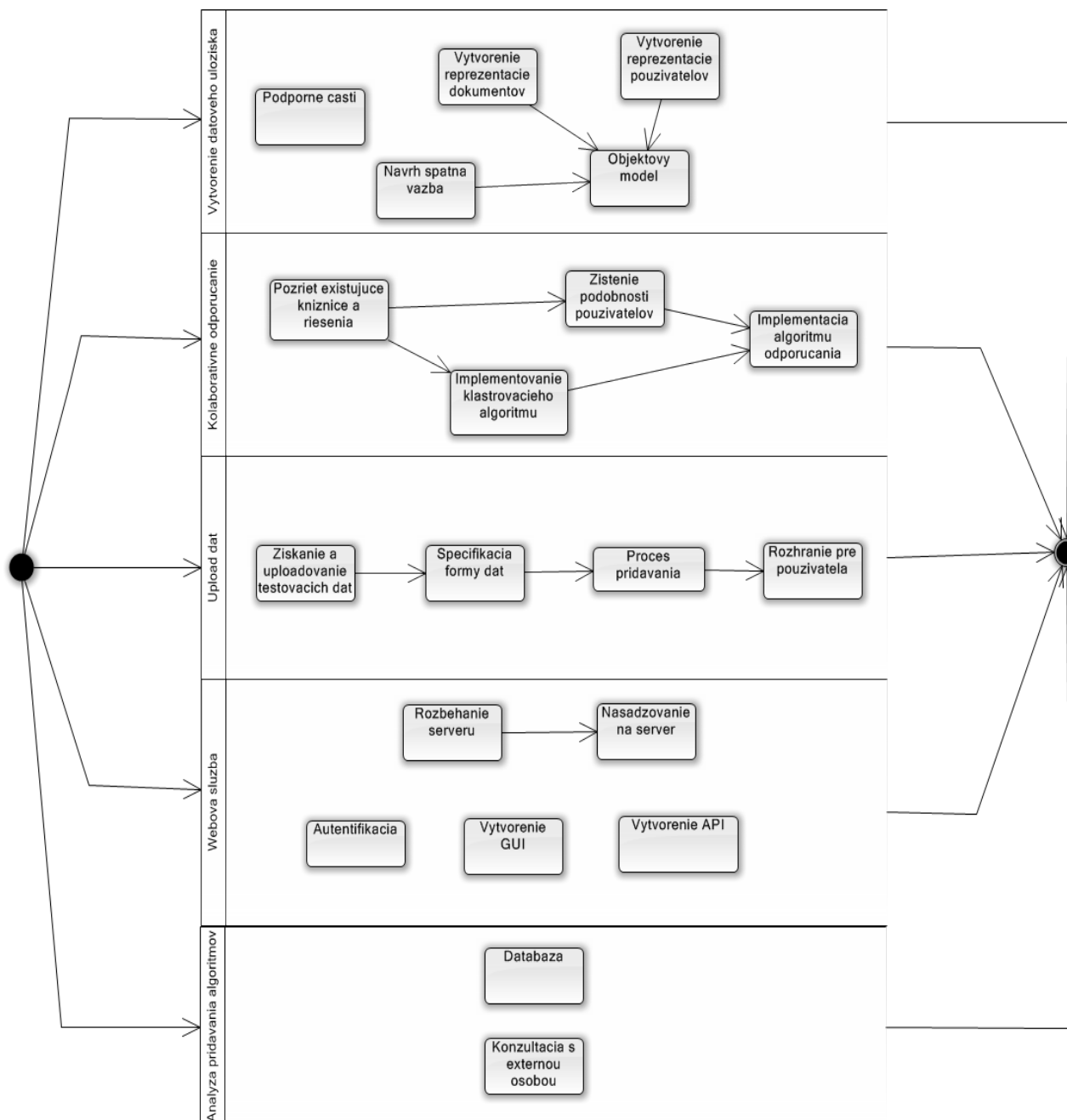
Obrázok 5.1: Celkový plán projektu

**5.4. Plán šprintu 01\_Anofeles**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>2.1</b>	Vytvorenie dátového úložiska	Jakub Ševcech	17.10.11	vysoká
2.1.1	Objektový model	Ludovit Mydla	17.10.11	vysoká
2.1.2	Návrh - spätná väzba	Jakub Ševcech	19.10.11	normálna
2.1.3	Podporné činnosti	Matej Červeňák	17.10.11	normálna
2,1,4	Vytvorenie reprezentácie dokumentov	Martin Detko	17.10.11	vysoká
2,1,5	Vytvorenie reprezentácie používateľov	Matej Červeňák	17.10.11	vysoká
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cadrik	19.10.11	normálna
2.2.1	Pozrieť existujúce knižnice a riešenia	Igor Slotík	17.10.11	vysoká
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	17.10.11	vysoká
2.2.3	Implementovanie klastrovacieho algoritmu	Igor Slotík	26.10.11	normálna
2.2.4	Implementácia algoritmu odporúčania	Michal Cadrik	19.10.11	normálna
<b>2.3</b>	Upload dát	Martin Detko	26.10.11	normálna
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	26.10.11	normálna
2.3.2	Proces pridávania	Martin Detko	26.10.11	normálna
2.3.3	Špecifikácia formy dát	Martin Detko	17.10.11	normálna
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	26.10.11	normálna
<b>2.4</b>	Webová služba	Jakub Ševcech	19.10.11	normálna
2.4.1	Rozbehovanie serveru	Ludovit Mydla	17.10.11	normálna
2.4.2	Autentifikácia	Jakub Ševcech	17.10.11	vysoká
2.4.3	Vytvorenie GUI	Jakub Ševcech	19.10.11	normálna
2.4.4	Nasadzovanie na server	Ludovit Mydla	17.10.11	normálna
2.4.5	Vytvorenie API	Jakub Ševcech	26.10.11	vysoká
<b>2.5</b>	Analýza pridávania algoritmov	Matej Červeňák	26.10.11	normálna
2.5.1	Databáza	Matej Mihalik	19.10.11	vysoká

2.5.2	Konzultácia s externou osobou	Matej Červeňák	26.10.11	normálna
-------	-------------------------------	----------------	----------	----------

Tabuľka 5.4: Plán 1. šprintu



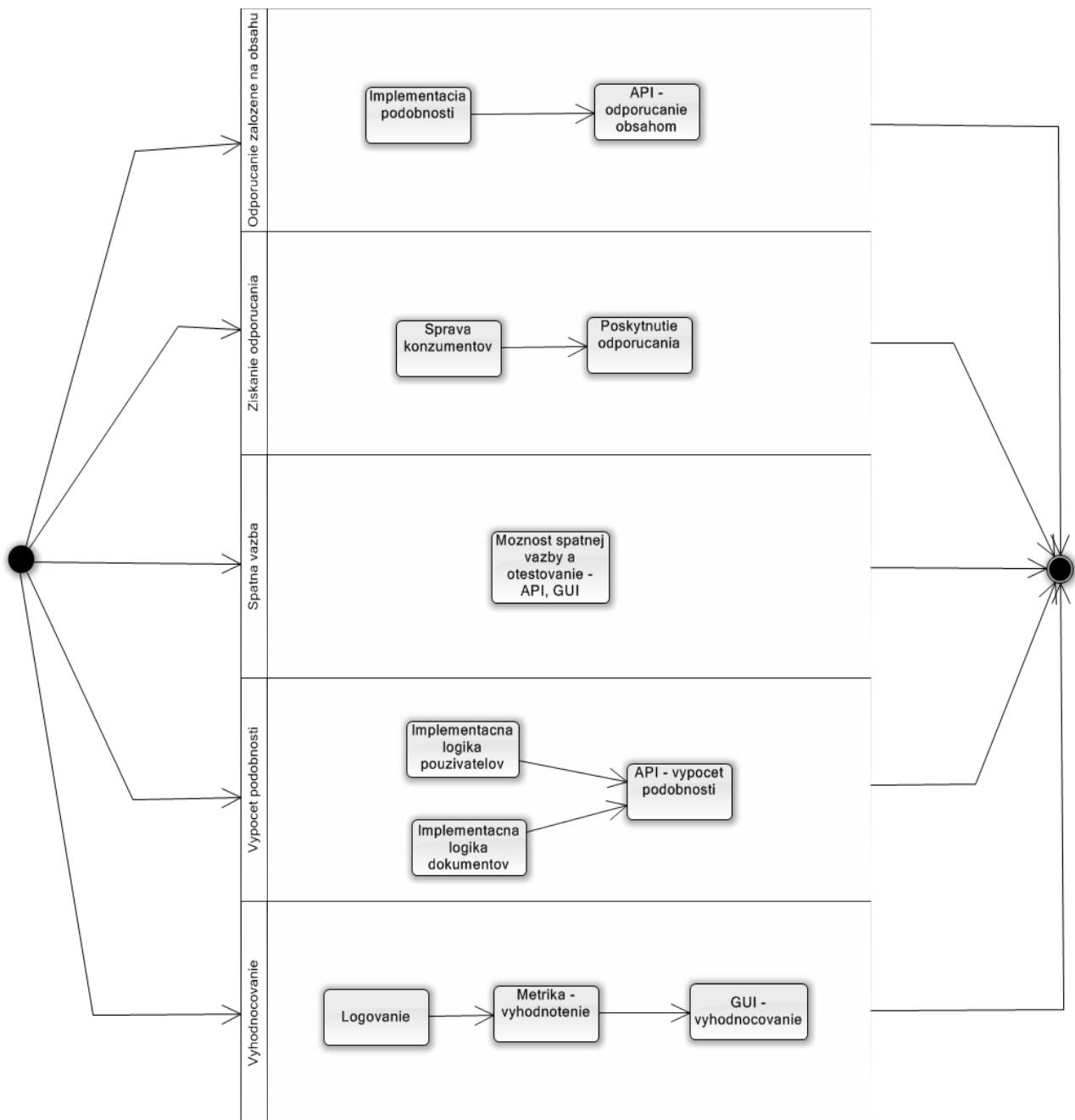
Obrázok 5.2: Plán 1. šprintu

**5.5. Plán šprintu 02\_Bubo\_Bubo**

ID	Popis	Zodpovední	Dátum splnenia	Priorita
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	11.06.11	normálna
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	11.02.11	vysoká
2.2.3	Implementovanie klastrovacieho algoritmu	Igor Slotík	11.06.11	normálna
2.2.4	Implementácia algoritmu odporúčania	Michal Cadrik	11.06.11	normálna
<b>2.3</b>	Upload dát	Martin Detko	11.06.11	normálna
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	11.06.11	normálna
2.3.2	Proces pridávania	Martin Detko	11.02.11	vysoká
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	11.06.11	normálna
<b>2.4</b>	Webová služba	Jakub Ševcech	11.06.11	normálna
2.4.1	Rozbehanie serveru	Ludovít Mydla	11.02.11	vysoká
2.4.4	Nasadzovanie na server	Ludovít Mydla	11.02.11	vysoká
2.4.5	Vytvorenie API	Jakub Ševcech	11.02.11	vysoká
<b>3.1</b>	Odporúčanie založené na obsahu	Ludovít Mydla	11.10.11	normálna
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	11.10.11	normálna
3.1.2	Implementácia podobnosti	Ludovít Mydla	11.10.11	normálna
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	11.10.11	normálna
3.2.1	Správa konzumentov	Jakub Ševcech	11.10.11	normálna
3.2.2	Poskytnutie odporúčania	Michal Cádrik	11.10.11	normálna
<b>3.3</b>	Spätná väzba	Jakub Ševcech	11.10.11	normálna
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	11.10.11	normálna
<b>3.4</b>	Výpočet podobnosti	Jakub Ševcech	11.10.11	normálna
3.4.1	Implementačná logika používateľov	Matej Mihalik	11.10.11	normálna
3.4.2	Implementačná logika dokumentov	Matej Mihalik	11.10.11	normálna
3.4.3	API – Výpočet podobnosti	Jakub Ševcech	11.10.11	normálna
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	11/10.11	normálna

3.5.1	Logovanie	Martin Detko	11.07.11	vysoká
3.5.2	Metrika-vyhodnotenie	Igor Slotík	11.10.11	normálna
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	11.10.11	normálna

Tabuľka 5.5: Plán 2. šprintu



Obrázok 5.3: Plán 2. šprintu

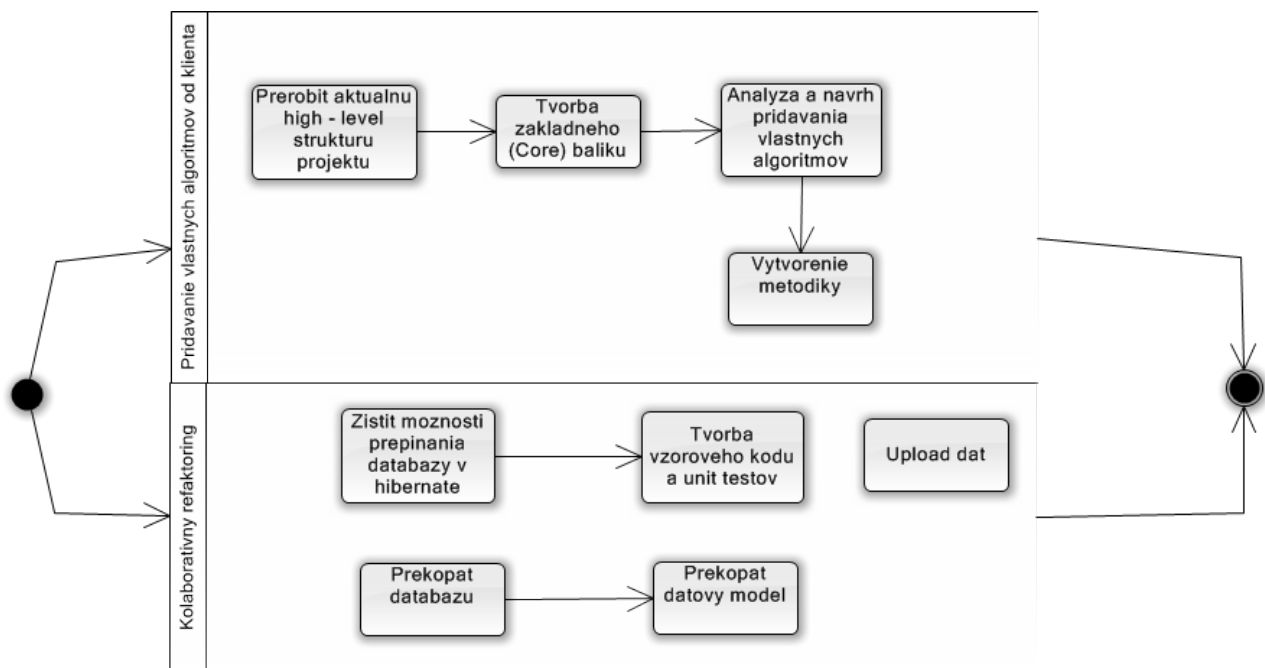
**5.6. Plán šprintu 03\_Canis**

ID	Popis	Zodpovední	Dátum splnenia	Priorita
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna
2.2.4	Implementácia algoritmu odporúčania	Michal Cadrik	06.11.11	normálna
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	normálna
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna
3.1.2	Implementácia podobnosti	Ľudovít Mydla	10.11.11	normálna
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna
<b>3.5</b>	Vyhodnocovanie		10.11.11	normálna
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta		24.10.11	normálna
4.1.1	Tvorba základného (Core) balíku	Ľudovít Mydla	24.10.11	normálna
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.10.11	normálna
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.10.11	normálna
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.10.11	normálna
<b>4.2</b>	Kolaboratívny refaktoring		24.10.11	normálna
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.10.11	normálna



4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.10.11	normálna
4.2.3	Upload dát	Martin Detko	24.10.11	normálna
4.2.4	Prekopať databázu	Jakub Ševcech	24.10.11	normálna
4.2.5	Prekopať dátový model	Matej Červeňák	24.10.11	Normálna

Tabuľka 5.6: Plán 3. šprintu

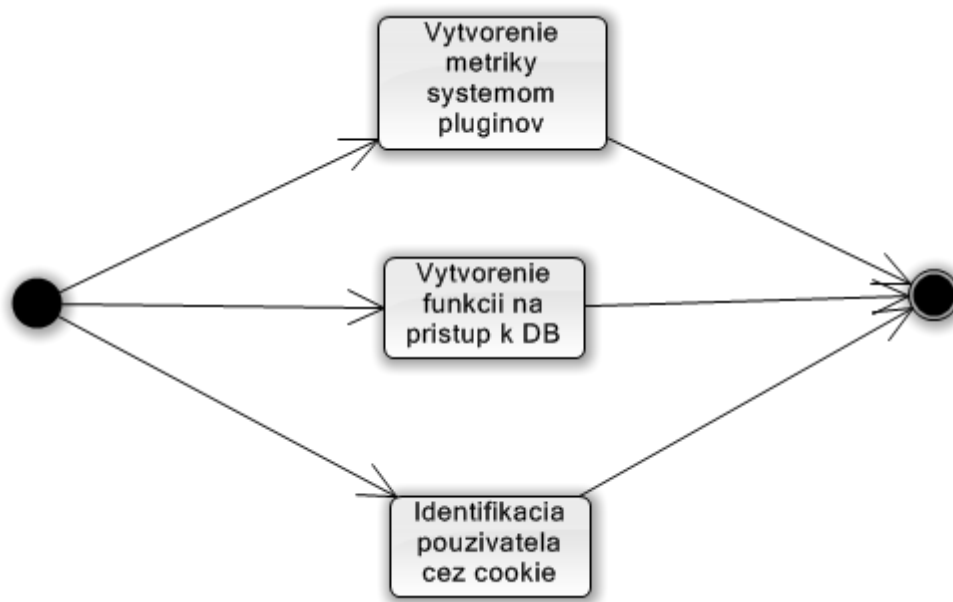


Obrázok 5.4: Plán 3. šprintu

**5.7. Plán šprintu 04\_Delphinidae\_Delphis**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	normálna
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna
<b>3.5</b>	Vyhodnocovanie		10.11.11	normálna
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta		24.11.11	normálna
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	normálna
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	normálna
<b>4.2</b>	Kolaboratívny refaktoring		24.11.11	normálna
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.11.11	normálna
4.2.3	Upload dát	Martin Detko	24.11.11	normálna
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	normálna
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.2011	Normálne
5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihálik	1.12.2011	Normálna
5.1.3	Identifikácia používateľa cez cookie	Jakub Ševcech	1.12.2011	Vysoká

Tabuľka 5.7: plán 4. šprintu

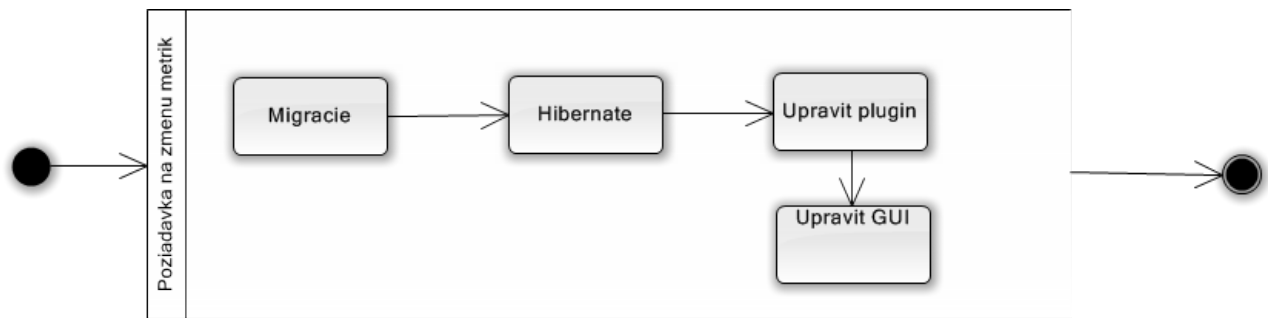


Obrázok 5.5: plán 4. šprintu

**5.8. Plán šprintu 05\_Eptesicus\_Fuscus**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	Normálna
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	Normálna
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	Normálna
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	Normálna
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	Normálna
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	Normálna
3.5.1	Logovanie	Martin Detko	07.11.11	Vysoká
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	Normálna
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	Normálna
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	Normálna
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	Normálna
<b>4.2</b>	Kolaboratívny refaktoring	Ľudovít Mydla	24.11.11	Normálna
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.11.11	Normálna
4.2.3	Upload dát	Martin Detko	24.11.11	Normálna
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	Normálna
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.11	Normálne
5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihálik	1.12.11	Normálna
5.1.3	Identifikácia používateľa cez cookie	Jakub Ševcech	1.12.11	Vysoká
<b>6.1</b>	Požiadavka na zmenu metrík	Igor Slotík	4.12. 2011	Vysoká
6.1.1	Upraviť GUI	Jakub Ševcech	1.12. 2011	Vysoká
6.1.2	Upraviť plugin	Igor Slotík	4.12. 2011	Vysoká
6.1.3	Migrácie	Jakub Ševcech	1.12. 2011	Vysoká
6.1.4	Hibernate	Matej Mihálik	1.12. 2011	Vysoká

Tabuľka 5.8: plán 5. šprintu

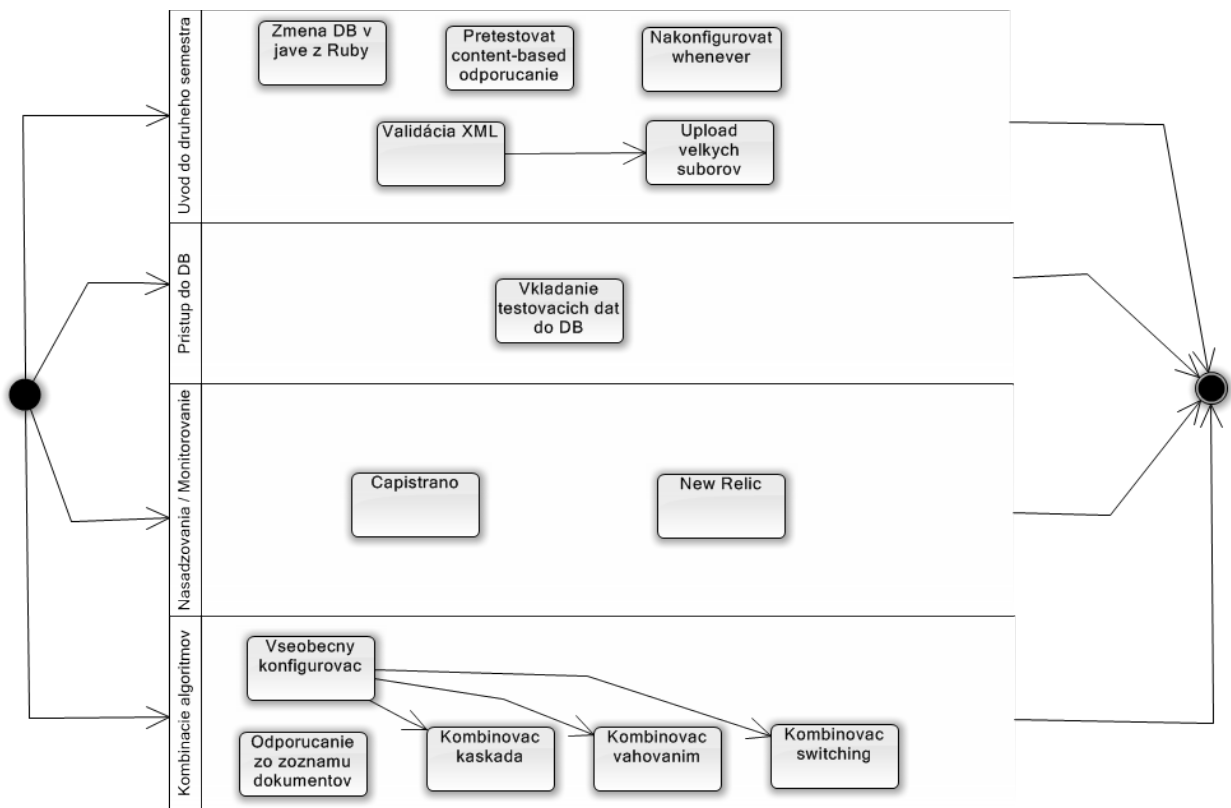


Obrázok 5.6: Plán 5. šprintu

**5.9. Plán šprintu 06\_Felis Catus**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
7.1	Úvod do 2. semestra	Jakub Ševcech	19.2.2012	normálna
7.1.1	Zmena DB v jave z Ruby	Matej Mihálik	19.2.2012	normálna
7.1.2	Nakonfigurovať whenever	Jakub Ševcech	19.2.2012	normálna
7.1.3	Validácia XML	Jakub Ševcech, Martin Detko	19.2.2012	normálna
7.1.4	Upload veľkých súborov	Jakub Ševcech	19.2.2012	normálna
7.1.5	Pretestovať content-based odporúčanie	Eudovít Mydla	19.2.2012	normálna
7.2	Prístup do DB	Matej Mihálik	19.2.2012	normálna
7.2.1	Vkladanie testovacích dát do DB	Matej Mihálik	19.2.2012	normálna
7.3	Nasadzovanie / monitorovanie	Jakub Ševcech	19.2.2012	normálna
7.3.1	Capistrano	Jakub Ševcech	19.2.2012	normálna
7.3.2	New Relic	Jakub Ševcech	19.2.2012	normálna
7.4	Kombinácia algoritmov	Martin Detko	19.2.2012	normálna
7.4.1	Všeobecný konfigurovač	MD, LM, IS, MM	19.2.2012	normálna
7.4.2	Kombinovač kaskáda	Michal Cádrik	19.2.2012	normálna
7.4.3	Kombinovač váhovaním	Igor Slotík	19.2.2012	normálna
7.4.4	Kombinovač switching	Martin Detko	19.2.2012	normálna
7.4.5	Odporúčanie zo zoznamu dokumentov	Michal Cádrik, Eudovít Mydla	19.2.2012	normálna

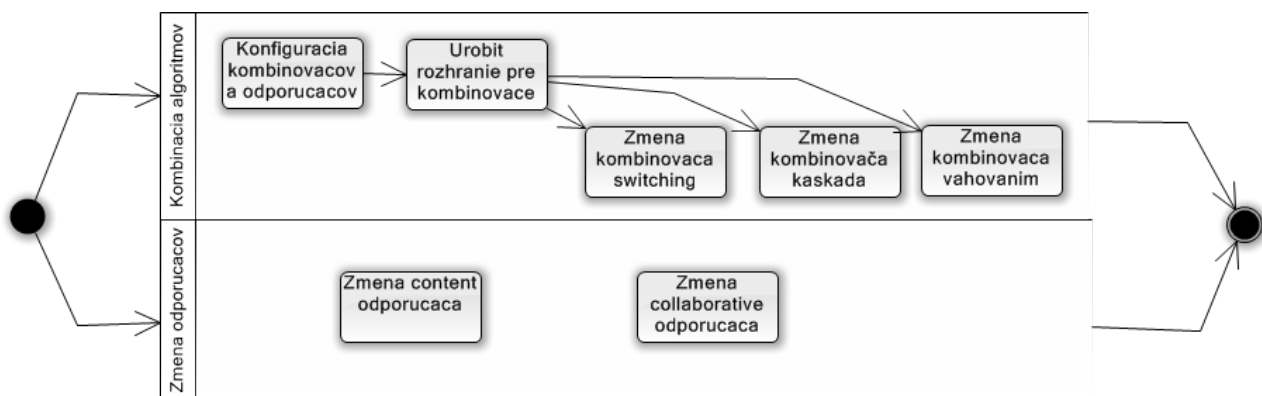
Tabuľka 5.9: plán 6. šprintu



### 5.10. Plán šprintu 07\_Glaucomys\_Sabrinus

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
7.4	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna
7.4.5	Zmena kombinovača switching	Martin Detko	26.2.2012	normálna
7.4.6	Zmena kombinovača kaskáda	Michal Cádrik	26.2.2012	normálna
7.4.7	Zmena kombinovača váhovaním	Igor Slotík	26.2.2012	normálna
7.4.8	Urobiť rozhranie pre kombinovače	Jakub Ševcech	26.2.2012	normálna
7.4.9	Konfigurácia kombinovačov a odporúčačov	Martin Detko	26.2.2012	normálna
8.1	Zmena odporúčačov	Michal Cádrik	26.2.2012	normálna
8.1.1	Zmena content odporúčača	Ľudovít Mydla	26.2.2012	normálna
8.1.1	Zmena collaborative odporúčača	Michal Cádrik	26.2.2012	normálna

Tabuľka 5.10: plán 7. šprintu

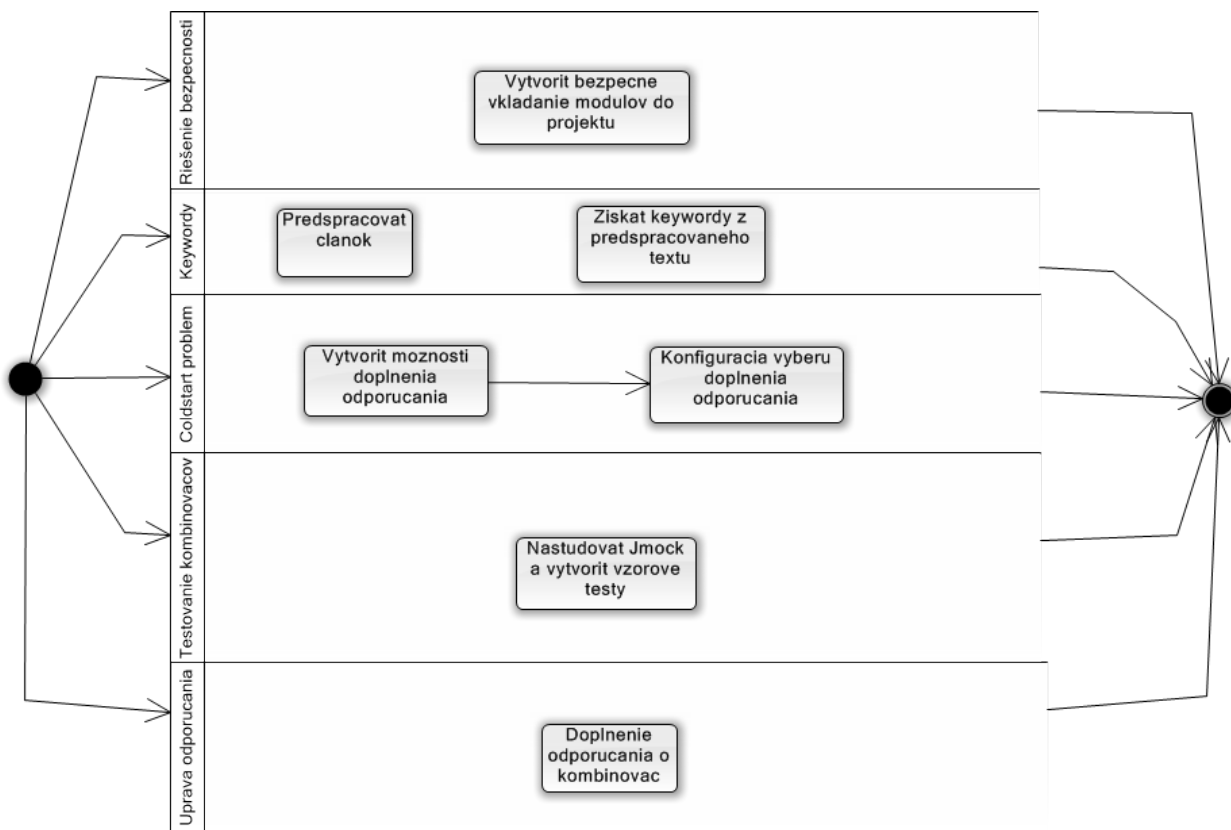




**5.11. Plán šprintu 08\_Haliaeetus\_leucocephalusa**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna
<b>9.2</b>	Získať keywordy z článku	Matej Mihálik	5. 3. 2012	normálna
9.2.1	Predspracovať článok	Ľudovít Mydla	5. 3. 2012	normálna
9.2.2	Získať keywordy z predspracovaného textu	Matej Mihálik	5. 3. 2012	normálna
<b>9.3</b>	Coldstart problem	Michal Cadrik	5. 3. 2012	normálna
9.3.1	Vytvoriť možnosti doplnenia odporúčania	Michal Cádrik	5. 3. 2012	normálna
9.3.2	Konfigurácia výberu doplnenia odporúčania	Martin Detko	5. 3. 2012	normálna
<b>9.4</b>	Testovanie kombinovačov	Igor Slotík	5. 3. 2012	normálna
9.4.1	Naštudovať Jmock a vytvoriť vzorové testy	Igor Slotík	5. 3. 2012	normálna
<b>9.5</b>	Úprava poskytnutia odporúčania	Martin Detko	5. 3. 2012	normálna
9.5.1	Doplnenie odporúčania o kombinovač	Martin Detko	5. 3. 2012	normálna

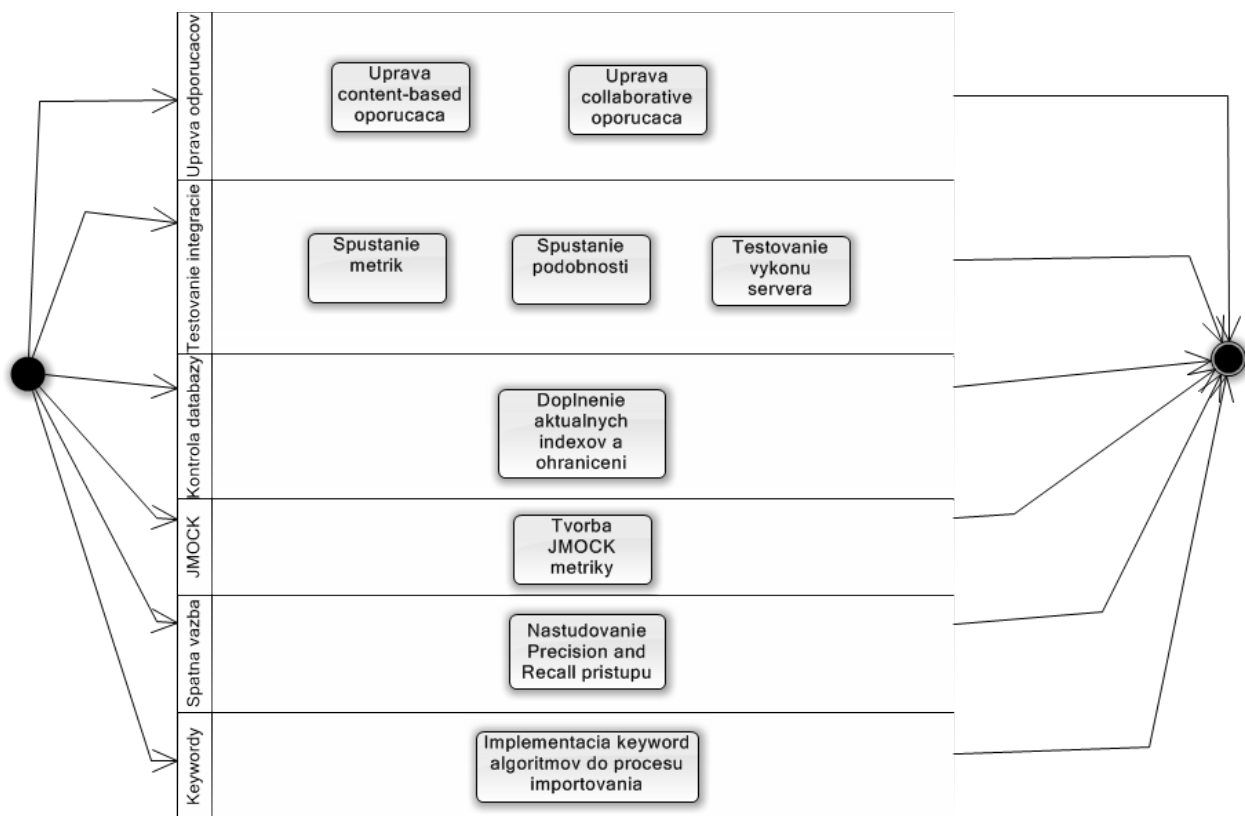
Tabuľka 5.11: plán 8. šprintu



**5.12. Plán šprintu 09\_Indicator\_Exilis**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>10.1</b>	Úprava odporúčačov podľa nového rozhrania	Ludovít Mydla	12. 3. 2012	normálna
10.1.1	Úprava content-based oporúčača	Ludovít Mydla	12. 3. 2012	Normálna
10.1.2	Úprava collaborative oporúčača	Michal Cádrik	12. 3. 2012	normálna
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna
10.2.1	Spúšťanie metrik	Martin Detko	12. 3. 2012	normálna
10.2.2	Spúšťanie podobnosti	Jakub Ševcech	12. 3. 2012	normálna
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna
<b>10.3</b>	Kontrola databázy	Jakub Ševcech	12. 3. 2012	normálna
10.3.1	Doplnenie aktuálnych indexov a ohraňení	Jakub Ševcech	12. 3. 2012	normálna
<b>10.4</b>	JMOCK	Igor Slotík	512 3. 2012	normálna
10.4.1	Tvorba JMOCK metriky	Igor Slotík	12. 3. 2012	normálna
<b>10.5</b>	Spätná väzba	Igor Slotík	12. 3. 2012	normálna
10.5.1	Naštudovanie Precision and Recall prístupu	Igor Slotík	12. 3. 2012	normálna
<b>10.6</b>	Integrácia vlastných algoritmov na extrakciu keywordov	Matej Mihalik	12. 3. 2012	normálna
10.6.1	Implementácia keyword algoritmov do procesu importovania	Martin Detko	12. 3. 2012	normálna

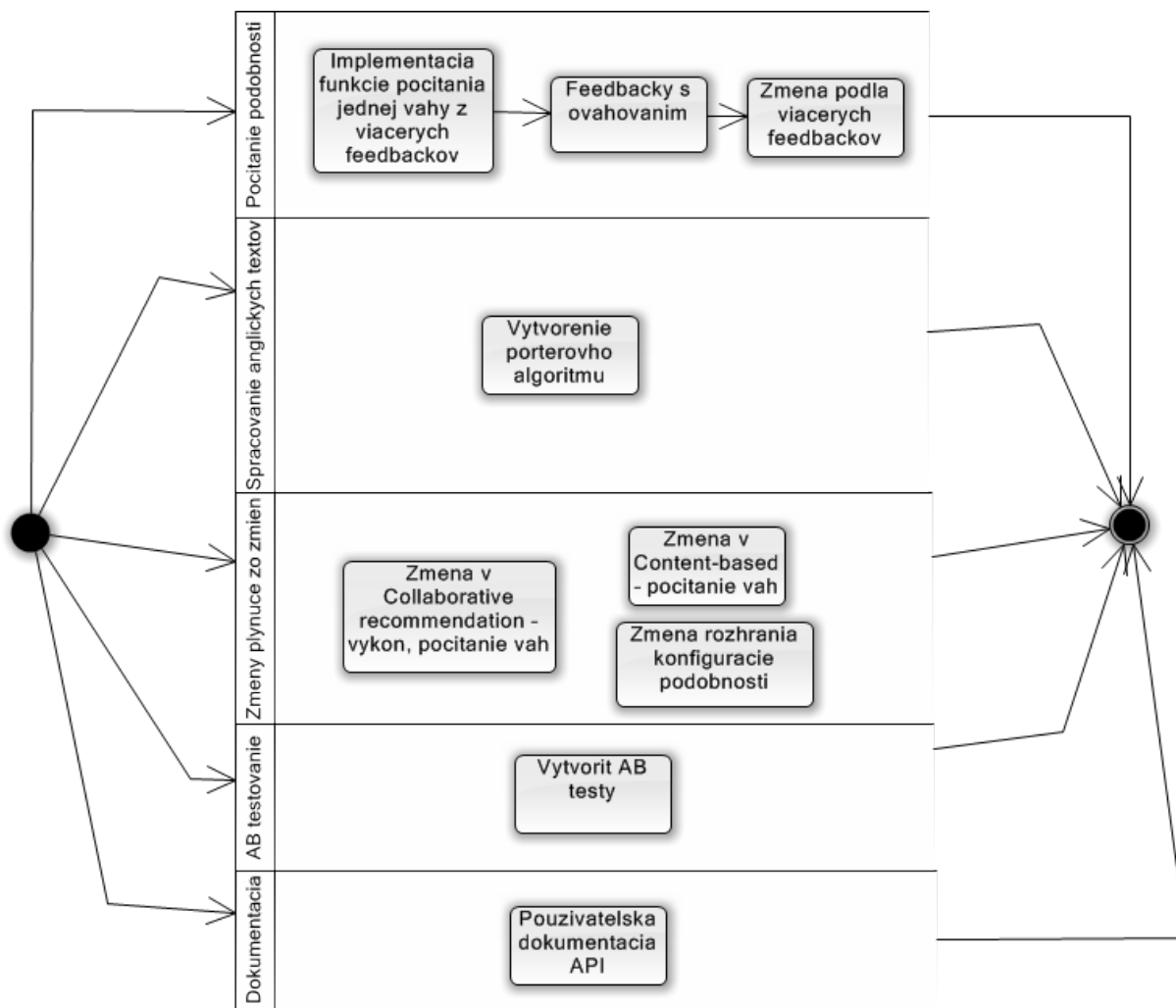
Tabuľka 5.12: plán 9. šprintu



**5.13. Plán šprintu 10\_Jynx\_Torquilla**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>11.1</b>	Zmena počítania podobností	Matej Mihalik	19. 3. 2012	normálna
11.1.1	Zmena podľa viacerých feedbackov	Matej Mihalik	19. 3. 2012	normálna
11.1.2	Feedbacky s ováhováním	Matej Mihalik	19. 3. 2012	normálna
11.1.3	Implementácia funkcie počítania jednej váhy z viacerých feedbackov	Matej Mihalik	19. 3. 2012	normálna
<b>11.3</b>	Spracovanie anglických textov	Eudovít Mydla	19. 3. 2012	normálna
10.3.1	Vytvorenie porterovho algoritmu	Eudovít Mydla	19. 3. 2012	normálna
<b>11.4</b>	Zmeny plynúce zo zmien	Igor Slotík	19. 3. 2012	normálna
11.4.1	Zmena v Collaborative recommendation – výkon, počítanie váh	Michal Cádrik	19. 3. 2012	normálna
11.4.2	Zmena v Content-based – počítanie váh	Eudovít Mydla	19. 3. 2012	normálna
11.4.3	Zmena rozhrania konfigurácie podobnosti	Jakub Ševcech	19. 3. 2012	normálna
11.4.3	Vyladenie metrik	Igor Slotík	19. 3. 2012	normálna
11.4.4	Zmena v metrikách	Jakub Ševcech	19. 3. 2012	normálna
<b>11.5</b>	AB testovanie	Martin Detko	19. 3. 2012	normálna
11.5.1	Vytvoriť AB testy	Martin Detko	19. 3. 2012	normálna
<b>11.6</b>	Dokumentácia	Michal Cádrik	19. 3. 2012	normálna
11.6.1	Používateľská dokumentácia API	Michal Cádrik	19. 3. 2012	normálna

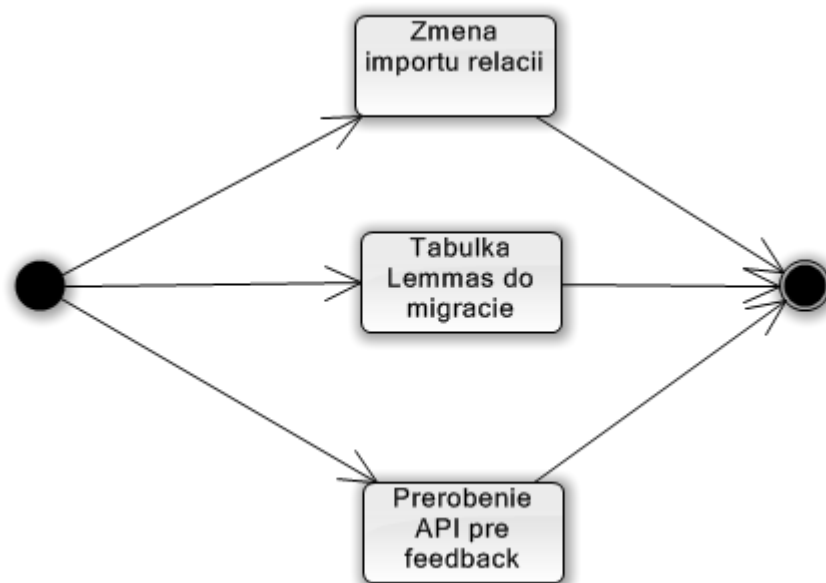
Tabuľka 5.13: plán 10. šprintu



**5.14. Plán šprintu 11\_Kakamega\_Poliothorax**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
12.1	Zmena importu relácií	Martin Detko	2. 4. 2012	normálna
12.2	Tabuľka Lemmas do migrácie	Jakub Ševcech	2. 4. 2012	normálna
12.3	Prerobenie API pre feedback	Jakub Ševcech	2. 4. 2012	normálna

Tabuľka 5.14: plán 11. šprintu

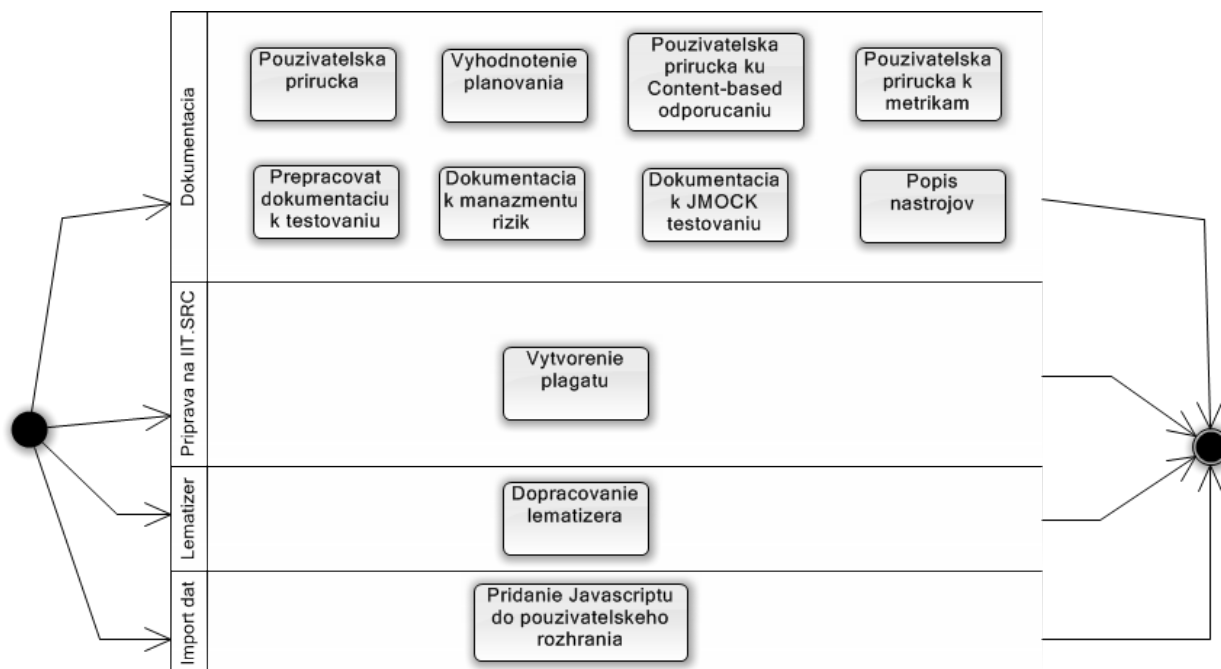


**5.15. Plán šprintu 12\_Leporidae\_Cuniculas**

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita
<b>13.1</b>	Dokumentácia	Michal Cádrik	7. 10. 2012	normálna
13.1.1	Používateľská príručka	Michal Cádrik	7. 10. 2012	normálna
13.1.2	Prepracovať dokumentáciu k testovaniu	Martin Detko	7. 10. 2012	normálna
13.1.3	Vyhodnotenie plánovania	Matej Mihalik	7. 10. 2012	normálna
13.1.4	Dokumentácia k manažmentu rizík	Eudovít Mydla	7. 10. 2012	normálna
13.1.5	Používateľská príručka ku Content-based odporúčaniam	Eudovít Mydla	7. 10. 2012	normálna
13.1.6	Dokumentácia k JMOCK testovaniu	Igor Slotík	7. 10. 2012	normálna
13.1.7	Používateľská príručka k metrikám	Igor Slotík	7. 10. 2012	normálna
13.1.8	Popis nástrojov	Jakub Ševcech	7. 10. 2012	normálna
<b>13.2</b>	Príprava na IIT.SRC	Eudovít Mydla	12.10. 2012	normálna
13.2.1	Vytvorenie plagátu	Eudovít Mydla	12.10. 2012	normálna
<b>13.3</b>	Lematizér	Eudovít Mydla	12.10. 2012	normálna
13.3.1	Dopracovanie lematizéra	Eudovít Mydla	12.10. 2012	normálna
<b>13.4</b>	Import dát	Jakub Ševcech	12.10. 2012	normálna
13.4.1	Pridanie Javascriptu do používateľského rozhrania	Jakub Ševcech	12.10. 2012	normálna

Tabuľka 5.15: plán 12. šprintu





### 5.16. Vyhodnotenie plánov po prvom semestri

Ako sme na začiatku tejto kapitoly uviedli, náš tím postupuje pri vývoji metodikou scrum. Keďže všetci členovia tímu doteraz nepracovali touto metodikou, dopustili sme sa v prvých šprintoch niekoľkých základných chýb.

Zle sme si odhadli čas potrebný na splnenie úloh a teda sme si ich na prvé šprinty naplánovali príliš veľa. To malo za následok nedokončenie úloh včas a prenesenie týchto úloh do ďalšieho šprintu. Tam sa tento problém ešte zväčšil.

Okrem toho sme sa stretli s viacerými problémami spojenými s použitou technológiou, ktoré nám umožňovali len veľmi pomalé napredovanie v plnení úloh. Rôzni členovia tímu pracujú na rôznych platformách a väčšina z nich nemala doterajšie skúsenosti s niektorými použitými technológiami (JRuby). Preto mali problémy nainštalovať potrebné nástroje na svojich počítačoch. Rovnako sme sa stretli s problémom nasadiť tieto technológie na pridelený server.

Skúsili sme zmeniť použité technológie. Upustili sme od používania jazyka JRuby a začali sme implementovať v jazyku Java s použitím frameworku Spring. Po stretnutí s kontrolórom kvality sme sa však opäť vrátili k pôvodným technológiám (JRuby a Java). Touto zmenou technológií sme stratili veľmi veľa času, ktorý sa prejavil na nedokončení úloh na konci prvých dvoch šprintov. Ďalší čas sme stratili už pri prvých problémoch s inštaláciou technológií.

V posledných šprintoch sme sa sústredili najmä na dokončovanie stále nekompletných úloh z minulých šprintov, a nové úlohy ktoré nám pribudli sa týkali najmä zlepšovania a úpravy funkcií, ktoré nám už boli zadané predtým. Mali sme plno tímových stretnutí kde sme navrhovali novú a lepšiu štruktúru projektu, databázy a aj nové postupy a metodiky ktoré dodržiavame. Ďalej sme prispôbovali už napísaný kód týmto novým zmenám v štruktúre projektu. Zároveň sme sa už naučili lepšie odhadovať čas potrebný na splnenie jednotlivých úloh, čo prináša realistickejšie a už aj splniteľnejšie plány pre jednotlivé šprinty.

### **5.17. Vyhodnotenie plánov po druhom semestri**

Narozdiel od prvého semestra sme s plnením plánu na druhý semester celkom spokojní. Podarilo sa nám splniť všetky väčšie úlohy, ktoré sme si stanovili a to dokonca aj viac – menej v stanovenom čase. Zároveň nám ostala kopa času, ktorú sme mohli venovať rôznym úpravám a vylepšeniam pôvodných návrhov a riešení algoritmov, ako aj dôkladnému integračnému a unit testovaniu.

Tento úspech pripisujeme najmä lekciám, ktoré sme sa naučili v prvom semestri. Jednak sme sa naučili lepšie odhadovať čas potrebný na splnenie jednotlivých úloh, a taktiež sme získali lepší prehľad v problémovej oblasti a rovnako aj v našom vlastnom systéme, teda sme jednoduchšie a dôkladnejšie chápali úlohy, ktoré nás čakajú.

Čas, ktorý nám ešte ostáva do konca semestra plánujeme stráviť ďalším testovaním, prípadným odchyťovaním chýb a rozširovaním už existujúcej funkcionality.

## 6. Zápisy zo stretnutí

### 6.1. Zápis z prvého stretnutia

**Autor:** Ľudovít Mydla

---

Dátum:	06.10.2011
Trvanie:	12:00 – 14:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	-
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Ľudovít Mydla

---

#### Téma stretnutia:

Dohodnutie ohľadom servera a tímového plagátu.

#### Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
1.1	spraviť jeden účet na serveri	Ľudovít Mydla	3.10.2011	normálna	splnená
1.2	nakonfigurovať Redmine	Jakub Ševcech	3.10.2011	normálna	splnená
1.3	Nakonfigurovať FTP	Jakub Ševcech	3.10.2011	normálna	pozastavená
1.4	Rozbehať Wordpress	Matej Mihalik, Ľudovít Mydla	3.10.2011	normálna	splnená
1.5	Spraviť plagátik návrh	Matej Červeňák	29.10.2011	normálna	splnená
1.6	Konečná verzia plagátu	Matej Červeňák	31.10.2011	normálna	splnená
1.7	Pozrieť podrobnosti ohľadom TP Cup a sledovať termíny	Michal Cádrik	31.11.2011	normálna	rozpracovaná
1.8	Nájsť informácie ohľadom spôsobu testovania v C a C++	Igor Slotík, Matej Červeňák	31.12.2011	normálna	pozastavená

## Opis stretnutia:

- Skontrolovanie stavu úloh z minulého stretnutia
- Ustanovenie SCRUM, ako metodiky vývoja. Pripravujeme službu
- Hlasovanie za jazyk. Vyhrala Java + Ruby (minimálne 5 zo 7)
- Určenie IDE a nástrojov = Eclipse, RubyMine, Github
- Rozhodnutie naštudovania si tém a spravenie 5-minútových prezentácií.
- Rozhodnutie o príprave backlogu = v pondelok večer po prezentáciách
- Načrtnutie dátového modelu + diskusia
- Model nemá byť úplne generický, ale aby pokryl väčšinu prípadov a metód
- Rozdelenie tém prezentácií:
  - Kolaboratívne odp. = Michal Cádrik
  - Content-based = Ľudovít Mydla
  - Skupinové odp. = Martin Detko
  - Hybridné odp. = Michal Cádrik
  - User Model = Jakub Ševcech
  - Info o testovaní Javy a TDD = Igor Slotík

## Stanovené úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
2.1	Doplniť na stránku O nás, Technológie atď.	Matej Červeňák	10/10/11	normálna	nová
2.2	Postarať sa o vizuál stránky. Nový template napr.	Matej Červeňák	10/10/11	normálna	nová
2.3	Upraviť práva na upload pre wordpress	Jakub Ševcech	10/10/11	vysoká	nová
2.4	Registrovať sa na <a href="http://gitbus.fiit.stuba.sk/">http://gitbus.fiit.stuba.sk/</a>	všetci	10/10/11	normálna	nová
2.5	Nájsť info. o testovaní v Jave, najmä test-drive development	Igor Slotík	10/10/11	normálna	nová
2.6	Spísať používané tooly a dať do AISu	Jakub Ševcech	13/10/11	normálna	nová
2.7	Nahádzať zápis zo stretnutia na stránku a	Ľudovít Mydla	07/10/11	vysoká	nová

	do redminu				
2.8	Inštalácia PostgreSQL	Matej Mihálik	10/10/11	normálna	nová
2.9	Nakreslenie dátového modelu	Martin Detko	10/10/11	normálna	nová

Poznámky:

-

**6.2. Zápis z druhého stretnutia****Autor:** Jakub Ševcech

---

Dátum:	13.10.2011
Trvanie:	12:00 – 14:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech, Ľudovít Mydla
Neprítomní:	-
Zapisovateľ:	Jakub Ševcech

---

*Téma stretnutia:*

Vytvorenie product-backlogu.  
Spustenie prvého šprintu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
2.1	Doplniť na stránku O nás, Technológie atď.	Matej Červeňák	10/10/11	normálna	splnená
2.2	Postarať sa o vzhľad stránky. Nový template napr.	Matej Červeňák	10/10/11	normálna	splnená
2.3	Upraviť práva na upload pre wordpress	Jakub Ševcech	10/10/11	vysoká	splnená
2.4	Registrovať sa na <a href="http://gitbus.fiit.stuba.sk/">http://gitbus.fiit.stuba.sk/</a>	všetci	10/10/11	normálna	splnená
2.5	Nájsť info o testovaní v Jave, najmä test-drive development	Igor Slotík	10/10/11	normálna	splnená
2.6	Spísať používané tooly a dať do AISu	Jakub Ševcech	13/10/11	normálna	splnená
2.7	Nahádzať zápis zo stretnutia na stránku a do redminu	Ľudovít Mydla	07/10/11	vysoká	splnená
2.8	Inštalácia PostgreSQL	Matej Mihalik	10/10/11	normálna	splnená

2.9	Nakreslenie dátového modelu	Martin Detko	10/10/11	normálna	splnená
-----	-----------------------------	--------------	----------	----------	---------

*Opis stretnutia:*

- Stretnutie viedol Ľudo Mydla
- Skontrolovali sme stav úloh z minulého stretnutia
- Vedúci projektu (Ing. Michal Kompan) v úlohe majiteľa produktu (product owner) predstavil vlastnosti, ktoré by ako zákazník od projektu očakával (user story)
- Všetci členovia tímu sa dohodli na ohodnotení jednotlivých úloh podľa ich časovej náročnosti
- Rozdelili sme jednotlivé na vlastnosti na podúlohy a určili sme členov tímu zodpovedných za jednotlivé vlastnosti ako aj za podúlohy. Takto sme vlastne vytvorili product backlog.
- Vedúci si vybral úlohy, ktoré by chcel mať hotové v prvom šprinte.
- Spustili sme prvý šprint

*Stanovené úlohy:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.1</b>	Vytvorenie dátového úložiska	Jakub Ševcech	17.10.11	vysoká	nová
2.1.1	Objektový model	Ľudovít Mydla	17.10.11	vysoká	nová
2.1.2	Návrh - spätná väzba	Jakub Ševcech	19.10.11	normálna	nová
2.1.3	Podporné činnosti	Matej Červeňák	17.10.11	normálna	nová
2.1.4	Vytvorenie reprezentácie dokumentov	Martin Detko	17.10.11	vysoká	nová
2.1.5	Vytvorenie reprezentácie používateľov	Matej Červeňák	17.10.11	vysoká	nová
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	19.10.11	normálna	nová
2.2.1	Pozrieť existujúce knižnice a riešenia	Igor Slotík	17.10.11	vysoká	nová
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	17.10.11	vysoká	nová
2.2.3	Implementovanie klastrovacieho	Igor Slotík	26.10.11	normálna	nová

	algoritmu				
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	19.10.11	normálna	nová
<b>2.3</b>	Upload dát	Martin Detko	26.10.11	normálna	nová
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	26.10.11	normálna	nová
2.3.2	Proces pridávania	Martin Detko	26.10.11	normálna	nová
2.3.3	Špecifikácia formy dát	Martin Detko	17.10.11	normálna	nová
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	26.10.11	normálna	nová
<b>2.4</b>	Webová služba	Jakub Ševcech	19.10.11	normálna	nová
2.4.1	Rozbehovanie serveru	Ľudovít Mydla	17.10.11	normálna	nová
2.4.2	Autentifikácia	Jakub Ševcech	17.10.11	vysoká	nová
2.4.3	Vytvorenie GUI	Jakub Ševcech	19.10.11	normálna	nová
2.4.4	Nasadzovanie na server	Ľudovít Mydla	17.10.11	normálna	nová
2.4.5	Vytvorenie API	Jakub Ševcech	26.10.11	vysoká	nová
<b>2.5</b>	Analýza pridávania algoritmov	Matej Červeňák	26.10.11	normálna	nová
2.5.1	Databáza	Matej Mihalik	19.10.11	vysoká	nová
2.5.2	Konzultácia s externou osobou	Matej Červeňák	26.10.11	normálna	nová

Poznámky:

-



**6.3. Zápis z tretieho stretnutia****Autor:** Igor Slotík

---

Dátum:	20.10.2011
Trvanie:	12:00 – 14:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech, Ľudovít Mydla
Neprítomní:	-
Zapisovateľ:	Igor Slotík

---

*Téma stretnutia:*

Medzi-kontrola prvého šprintu

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.1</b>	Vytvorenie dátového úložiska	Jakub Ševcech	27.10.11	vysoká	rozpracovaná
2.1.1	Objektový model	Ľudovít Mydla	10/19/11	vysoká	splnená
2.1.2	Návrh - spätná väzba	Jakub Ševcech	27.10.11	normálna	rozpracovaná
2.1.3	Podporné činnosti	Matej Červeňák	27.10.11	normálna	rozpracovaná
2.1.4	Vytvorenie reprezentácie dokumentov	Martin Detko	27.10.11	vysoká	rozpracovaná
2.1.5	Vytvorenie reprezentácie používateľov	Matej Červeňák	27.10.11	vysoká	rozpracovaná
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	27.10.11	normálna	rozpracovaná
2.2.1	Pozrieť existujúce knižnice a riešenia	Igor Slotík	27.10.11	vysoká	rozpracovaná
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	27.10.11	vysoká	rozpracovaná
2.2.3	Implementovanie	Igor Slotík	27.10.11	normálna	rozpracovaná

	klastrovacieho algoritmu				
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	27.10.11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	27.10.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	27.10.11	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	27.10.11	normálna	rozpracovaná
2.3.3	Špecifikácia formy dát	Martin Detko	27.10.11	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	27.10.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	27.10.11	normálna	rozpracovaná
2.4.1	Rozbeh serveru	Eudovít Mydla	27.10.11	normálna	rozpracovaná
2.4.2	Autentifikácia	Jakub Ševcech	27.10.11	vysoká	rozpracovaná
2.4.3	Vytvorenie GUI	Jakub Ševcech	27.10.11	normálna	rozpracovaná
2.4.4	Nasadzovanie na server	Eudovít Mydla	27.10.11	normálna	rozpracovaná
2.4.5	Vytvorenie API	Jakub Ševcech	27.10.11	vysoká	rozpracovaná
<b>2.5</b>	Analýza pridávania algoritmov	Matej Červeňák	20.10.11	normálna	splnená
2.5.1	Databáza	Matej Mihalik	20.10.11	vysoká	splnená
2.5.2	Konzultácia s externou osobou	Matej Červeňák	20.10.11	normálna	splnená

*Opis stretnutia:*

- Stretnutie viedol Jakub Ševcech
- Medzi-kontrola stavu úloh, ktoré boli určené na začiatku šprintu
- Riešenie problémov s momentálne nedokončenými úlohami
- Keďže bol problém s inštaláciou *JRuby* na server, dohodli sme sa, že budeme používať len *Javu*
- Na tomto stretnutí sme sa dohodli na ďalšej technológii: *Spring*

*Poznámky:* -

**6.4. Zápis zo štvrtého stretnutia****Autor:** Martin Detko

---

Dátum:	27.10.2011
Trvanie:	12:00 – 14:30
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Martin Detko

---

*Téma stretnutia:*

Vyhodnotenie prvého šprintu a naplánovanie ďalšieho.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.1</b>	Vytvorenie dátového úložiska	Jakub Ševcech	17.10.11	vysoká	ukončená
2.1.1	Objektový model	Eudovít Mydla	17.10.11	vysoká	ukončená
2.1.2	Návrh - spätná väzba	Martin Detko	19.10.11	normálna	ukončená
2.1.3	Podporné činnosti	Matej Červeňák	17.10.11	normálna	ukončená
2.1.4	Vytvorenie reprezentácie dokumentov	Martin Detko	17.10.11	vysoká	ukončená
2.1.5	Vytvorenie reprezentácie používateľov	Matej Červeňák	17.10.11	vysoká	ukončená
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	nesplnená	normálna	rozpracovaná
2.2.1	Pozrieť existujúce knižnice a riešenia	Igor Slotík	10/19/11	vysoká	ukončená
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	nesplnená	vysoká	rozpracovaná
2.2.3	Implementovanie klastrovacieho	Igor Slotík	nesplnená	normálna	rozpracovaná

	algoritmu				
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	nesplnená	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	nesplnená	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	nesplnená	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	10/25/11	normálna	ukončená
2.3.3	Špecifikácia formy dát	Martin Detko	nesplnená	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	nesplnená	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	nesplnená	normálna	rozpracovaná
2.4.1	Rozbehovanie serveru	Ľudovít Mydla	nesplnená	normálna	rozpracovaná
2.4.2	Autentifikácia	Jakub Ševcech	10/26/11	vysoká	ukončená
2.4.3	Vytvorenie GUI	Jakub Ševcech	10/26/11	normálna	ukončená
2.4.4	Nasadzovanie na server	Ľudovít Mydla	nesplnená	normálna	rozpracovaná
2.4.5	Vytvorenie API	Jakub Ševcech	nesplnená	vysoká	rozpracovaná
<b>2.5</b>	Analýza pridávania algoritmov	Matej Červeňák	10/20/11	normálna	zrušená
2.5.1	Databáza	Matej Mihalik	10/20/11	vysoká	zrušená
2.5.2	Konzultácia s externou osobou	Matej Červeňák	10/20/11	normálna	zrušená

*Opis stretnutia:*

- Vyhodnotenie stavu úloh
- Riešenie stavu veľa nedokončených úloh a zle odhadnutý čas
  - Presunutie nevyriešených úloh do ďalšieho šprintu
- Časové ohodnotenie úloh, pridelenie zodpovednosti a rozdelenie na podúlohy:
  - Odporúčanie založené na obsahu
  - Výpočet podobnosti
  - Vyhodnocovanie
- Zahŕnutie nových úloh do šprintu
  - Odporúčanie založené na obsahu
  - Výpočet podobnosti
  - Vyhodnocovanie

- Spätná väzba
- Získanie odporúčania

## Stanovené úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	11/06/11	normálna	rozpracovaná
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	11/02/11	vysoká	rozpracovaná
2.2.3	Implementovanie klastrovacieho algoritmu	Igor Slotík	11/06/11	normálna	rozpracovaná
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	11/06/11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	11/06/11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	11/06/11	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	11/02/11	vysoká	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	11/06/11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	11/06/11	normálna	rozpracovaná
2.4.1	Rozbehovanie serveru	Eudovít Mydla	11/02/11	vysoká	rozpracovaná
2.4.4	Nasadzovanie na server	Eudovít Mydla	11/02/11	vysoká	rozpracovaná
2.4.5	Vytvorenie API	Jakub Ševcech	11/02/11	vysoká	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	11/10/11	normálna	nová
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	11/10/11	normálna	nová
3.1.2	Implementácia podobnosti	Eudovít Mydla	11/10/11	normálna	nová
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	11/10/11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	11/10/11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	11/10/11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	11/10/11	normálna	nová

3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	11/10/11	normálna	nová
<b>3.4</b>	Výpočet podobnosti	Jakub Ševcech	11/10/11	normálna	nová
3.4.1	Implementačná logika používateľov	Matej Mihálik	11/10/11	normálna	nová
3.4.2	Implementačná logika dokumentov	Matej Mihálik	11/10/11	normálna	nová
3.4.3	API – Výpočet podobnosti	Jakub Ševcech	11/10/11	normálna	nová
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	11/10/11	normálna	nová
3.5.1	Logovanie	Martin Detko	11/07/11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	11/10/11	normálna	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	11/10/11	normálna	nová

Poznámky: -

**6.5. Zápis z piateho stretnutia****Autor:** Matej Červeňák

---

Dátum:	3.11.2011
Trvanie:	12:00 – 13:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Matej Červeňák

---

*Téma stretnutia:*

Pribežné vyhodnotenie úloh šprintu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Pribežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna	rozpracovaná
2.2.2	Zistenie podobnosti používateľov	Matej Mihalik	02.11.11	vysoká	rozpracovaná
2.2.3	Implementovanie klastrovacieho algoritmu	Igor Slotík	06.11.11	normálna	rozpracovaná
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	06.11.11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	02.11.11	vysoká	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
2.4.1	Rozbehovanie serveru	Ludovít Mydla	02.11.11	vysoká	rozpracovaná
2.4.4	Nasadzovanie na server	Ludovít Mydla	02.11.11	vysoká	rozpracovaná
2.4.5	Vytvorenie API	Jakub Ševcech	02.11.11	vysoká	ukončená

<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	normálna	nová
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Ľudovít Mydla	10.11.11	normálna	nová
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.4</b>	Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	nová
3.4.1	Implementačná logika používateľov	Matej Mihálik	10.11.11	normálna	nová
3.4.2	Implementačná logika dokumentov	Matej Mihálik	10.11.11	normálna	nová
3.4.3	API – Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	nová
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová

*Opis stretnutia:*

- Vyhodnotenie stavu úloh
- Diskusia k rozpracovaným úlohám

*Stanovené úlohy:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna	rozpracovaná
2.2.2	Zistenie podobnosti používateľov	Matej Mihálik	02.11.11	vysoká	rozpracovaná
2.2.3	Implementovanie	Igor Slotík	06.11.11	normálna	rozpracovaná



	klastrovacieho algoritmu				
2.2.4	Implementácia algoritmu odporúčania	Michal Cadrik	06.11.11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	02.11.11	vysoká	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
2.4.1	Rozbehanie serveru	Ludovít Mydla	02.11.11	vysoká	rozpracovaná
2.4.4	Nasadzovanie na server	Ludovít Mydla	02.11.11	vysoká	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Ludovít Mydla	10.11.11	normálna	nová
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Ludovít Mydla	10.11.11	normálna	nová
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.4</b>	Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	nová
3.4.1	Implementačná logika používateľov	Matej Mihálik	10.11.11	normálna	nová
3.4.2	Implementačná logika dokumentov	Matej Mihálik	10.11.11	normálna	nová
3.4.3	API – Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	nová
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna	nová
3.5.3	GUI - Vyhodnocovanie	Matej	10.11.11	normálna	nová

		Červeňák			
--	--	----------	--	--	--

*Poznámky: -*

**6.6. Zápis zo šiesteho stretnutia****Autor:** Matej Mihálik

---

Dátum:	10.11.2011
Trvanie:	12:00 – 13:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihálik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Matej Mihálik

---

*Téma stretnutia:*

Vyhodnotenie úloh druhého šprintu a naplánovanie nasledujúceho (tretieho).

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna	rozpracovaná
2.2.2	Zistenie podobnosti používateľov	Matej Mihálik	02.11.11	vysoká	ukončená
2.2.3	Implementovanie klastrovacieho algoritmu	Igor Slotík	06.11.11	normálna	ukončená
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	06.11.11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna	rozpracovaná
2.3.2	Proces pridávania	Martin Detko	02.11.11	vysoká	ukončená
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
2.4.1	Rozbehnutie serveru	Ludovít Mydla	02.11.11	vysoká	ukončená
2.4.4	Nasadzovanie na server	Ludovít Mydla	02.11.11	vysoká	ukončená

<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Ľudovít Mydla	10.11.11	normálna	rozpracovaná
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.4</b>	Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	ukončená
3.4.1	Implementačná logika používateľov	Matej Mihálik	10.11.11	normálna	ukončená
3.4.2	Implementačná logika dokumentov	Matej Mihálik	10.11.11	normálna	ukončená
3.4.3	API – Výpočet podobnosti	Jakub Ševcech	10.11.11	normálna	zamietnutá
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová

*Opis stretnutia:*

- Vyhodnotenie stavu úloh
- Diskusia k rozpracovaným úlohám
- Potreba dokončiť rozpracované úlohy
- Pridelenie nových úloh do nového šprintu
- Zmena technológií z Java na JRuby / Java.
- Každý člen tímu nainštalovať a spojiť virtuálny stroj – Ubuntu
- Podľa návodu / inštrukcií Jakuba Ševcecha
- Revízia projektovej dokumentácie a dokumentácie k riadeniu
- Písať celé v slovenčine, vyhnúť sa anglickým termínom (in progress, atd'...)

- Nepoužívať značenie verzie v dokumente
- Ganttové grafy zbytočné, bez výpovednej hodnoty

## Stanovené úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna	rozpracovaná
2.2.4	Implementácia algoritmu odporúčania	Michal Cadrik	06.11.11	normálna	rozpracovaná
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Eudovít Mydla	10.11.11	normálna	rozpracovaná
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.5</b>	Vyhodnocovanie		10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna	nová
3.5.3	GUI -	Matej	10.11.11	normálna	nová

	Vyhodnocovanie	Červeňák			
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta		24.10.11	normálna	nová
4.1.1	Tvorba základného (Core) balíku	Ľudovít Mydla	24.10.11	normálna	nová
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.10.11	normálna	nová
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.10.11	normálna	nová
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.10.11	normálna	nová
<b>4.2</b>	Kolaboratívny refaktoring		24.10.11	normálna	nová
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.10.11	normálna	nová
4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.10.11	normálna	nová
4.2.3	Upload dát	Martin Detko	24.10.11	normálna	nová
4.2.4	Prekopať databázu	Jakub Ševcech	24.10.11	normálna	nová
4.2.5	Prekopať dátový model	Matej Červeňák	24.10.11	normálna	nová

Poznámky:

-

**6.7. Zápis zo siedmeho stretnutia****Autor:** Michal Cádrik

---

Dátum:	15.11.2011
Trvanie:	11:00 – 13:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Michal Cádrik

---

*Téma stretnutia:*

Priebežné vyhodnotenie úloh šprintu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.2</b>	Kolaboratívne odporúčanie	Michal Cádrik	06.11.11	normálna	ukončená
2.2.4	Implementácia algoritmu odporúčania	Michal Cádrik	06.11.11	normálna	ukončená
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Matej Červeňák	06.11.11	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Eudovít Mydla	10.11.11	normálna	rozpracovaná
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová

<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	normálna	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	normálna	nová
4.1.1	Tvorba základného (Core) balíku	Ľudovít Mydla	24.11.11	normálna	nová
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	normálna	nová
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	normálna	nová
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.11.11	normálna	nová
<b>4.2</b>	Kolaboratívny refaktoring	Ľudovít Mydla	24.11.11	normálna	nová
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.11.11	normálna	nová
4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.11.11	normálna	nová

*Opis stretnutia:*

- stretnutie viedol Matej Mihalik
- Jakub pokročil so skriptom na importovanie dumpov
- navrhovali sme pluginový systém, nenavrhl sme ho
- napísali sme metodiku
- prerobený dátový model
- stretnutie štvrtok od 08:00 do 12:00 v D208 cez A blok (odovzdať kartu na vrátnici a zapísať sa)
- dohodnutie sa na napísanie prihlášky do TP cup
- úložisko je recommender (kedysi pôvodné) – treba naklonovať !!!
- odporúča sa aj počas spracovávania podobností používateľov



## Stanovené úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Jakub Ševcech	06.11.11	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Eudovít Mydla	10.11.11	normálna	rozpracovaná
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	nová
<b>3.5</b>	Vyhodnocovanie		10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	vysoká	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta		24.11.11	normálna	nová
4.1.1	Tvorba základného (Core) balíku	Eudovít Mydla	24.11.11	normálna	nová
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	normálna	nová
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	normálna	rozpracovaná
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.11.11	normálna	rozpracovaná
<b>4.2</b>	Kolaboratívny		24.11.11	normálna	nová

	refactoring				
4.2.1	Tvorba vzorového kódu a unit testov	Eudovít Mydla	24.11.11	normálna	nová
4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.11.11	normálna	nová
4.2.3	Upload dát	Martin Detko	24.11.11	normálna	nová
4.2.4	Prekopat' databázu	Jakub Ševcech	24.11.11	normálna	nová
4.2.5	Prekopat' dátový model	Matej Červeňák	24.11.11	normálna	ukončená
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	normálna	nová

Poznámky:

-

**6.8. Zápis z ôsmeho stretnutia****Autor:** Jakub Ševcech

---

Dátum:	22.11.2011
Trvanie:	11:00 – 12:30
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Jakub Ševcech

---

*Téma stretnutia:*

Ukončenie tretieho šprintu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	rozpracovaná
2.3.1	Získanie a uploadovanie testovacích dát	Jakub Ševcech	06.11.11	normálna	rozpracovaná
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	rozpracovaná
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Eudovít Mydla	10.11.11	normálna	rozpracovaná
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	nová
3.3.1	Možnosť spätnej	Martin Detko	10.11.11	normálna	nová

	väzby a otestovanie – GUI, API				
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	normálna	nová
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	vysoká	nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	normálna	nová
4.1.1	Tvorba základného (Core) balíku	Ľudovít Mydla	24.11.11	normálna	nová
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	normálna	nová
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	normálna	rozpracovaná
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.11.11	normálna	rozpracovaná
<b>4.2</b>	Kolaboratívny refaktoring	Ľudovít Mydla	24.11.11	normálna	nová
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.11.11	normálna	nová
4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.11.11	normálna	nová
4.2.3	Upload dát	Martin Detko	24.11.11	normálna	nová
4.2.4	Prekopať databázu	Jakub Ševcech	24.11.11	normálna	nová
4.2.5	Prekopať dátový model	Matej Červeňák	24.11.11	normálna	ukončená
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	normálna	nová

*Opis stretnutia:*

- Stretnutie viedol Michal Cádrik.
- Do zajtra je potrebné odovzdať prihlášku do TP CUPu. Treba ju najskôr poslať Michalovi Kompanovi na kontrolu.
- Chýbajú spísané metodiky na dodržiavanie štábnej kultúry a pravidiel vývoja. Je potrebné

sa na týchto činnostiach dohodnúť, spísať metodiky a zabezpečiť ich dodržiavanie.

- Vznikla požiadavka na zmenu skriptu na vytvorenie dát na upload tak, aby bol používateľ identifikovaný pomocou cookie a nie IP adresou. Túto zmenu zapracuje Jakub Ševcech v nasledujúcom šprinte.
- Vznikla požiadavka na zmenu, kde treba prerobiť metriky na vypočítavanie úspešnosti odporúčania do podoby pluginov. Túto zmenu zapracuje Igor Slotík v nasledujúcom šprinte.
- Nasledujúci šprint je naplánovaný len na jeden týždeň. Toto je z dôvodu zvýšenia efektivity práce počas týždňa, ako aj z dôvodu organizácie konca semestra.
- V nasledujúcom šprinte je potrebné ukončiť úlohy, ktoré sa prenášajú z minulých šprintov a dokončiť tiež novo vzniknuté úlohy.

#### Stanovené úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>2.3</b>	Upload dát	Martin Detko	06.11.11	normálna	ukončená
2.3.1	Získanie a uploadovanie testovacích dát	Jakub Ševcech	06.11.11	normálna	ukončená
2.3.4	Rozhranie pre používateľa	Jakub Ševcech	06.11.11	normálna	ukončená
<b>2.4</b>	Webová služba	Jakub Ševcech	06.11.11	normálna	ukončená
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	normálna	rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	normálna	nová
3.1.2	Implementácia podobnosti	Ľudovít Mydla	10.11.11	normálna	ukončená
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	normálna	nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	normálna	nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	normálna	nová
<b>3.3</b>	Spätná väzba	Jakub Ševcech	10.11.11	normálna	ukončená
3.3.1	Možnosť spätnej väzby a otestovanie – GUI, API	Martin Detko	10.11.11	normálna	ukončená
<b>3.5</b>	Vyhodnocovanie		10.11.11	normálna	rozpracovaná
3.5.1	Logovanie	Martin Detko	07.11.11	vysoká	nová
3.5.2	Metrika-vyhodnotenie	Igor Slotík	10.11.11	vysoká	ukončená
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	normálna	nová

4.1	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	normálna	ukončená
4.1.1	Tvorba základného (Core) balíku	Ľudovít Mydla	24.11.11	normálna	ukončená
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	normálna	rozpracovaná
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	normálna	rozpracovaná
4.1.4	Prerobiť aktuálnu high - level štruktúru projektu	Jakub Ševcech	24.11.11	normálna	ukončená
4.2	Kolaboratívny refaktoring	Ľudovít Mydla	24.11.11	normálna	rozpracovaná
4.2.1	Tvorba vzorového kódu a unit testov	Ľudovít Mydla	24.11.11	normálna	rozpracovaná
4.2.2	Zistiť možnosti prepínania databázy v hibernate	Matej Mihalik	24.11.11	normálna	ukončená
4.2.3	Upload dát	Martin Detko	24.11.11	normálna	nová
4.2.4	Prekopať databázu	Jakub Ševcech	24.11.11	normálna	ukončená
4.2.5	Prekopať dátový model	Matej Červeňák	24.11.11	normálna	ukončená
4.3	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	normálna	rozpracovaná
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.2011	Normálne	nová
5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihalik	1.12.2011	Normálna	nová
5.1.3	Identifikácia používateľa cez cookie	Jakub Ševcech	1.12.2011	Vysoká	nová

Poznámky:

**6.9. Zápis z deviateho stretnutia****Autor:** Igor Slotík

---

Dátum:	1.12.2011
Trvanie:	12:00 – 14:00
Miestnosť:	blok D – softvérové štúdio
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Igor Slotík

---

*Téma stretnutia:*

Ukončenie štvrtého šprintu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	Normálna	Rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	Normálna	Nová
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	Normálna	Nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	Normálna	Nová
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	Normálna	Nová
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	Normálna	Rozpracovaná
3.5.1	Logovanie	Martin Detko	07.11.11	Vysoká	Nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	Normálna	Nová
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	Normálna	Rozpracovaná
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	Normálna	Rozpracovaná
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	Normálna	Rozpracovaná
<b>4.2</b>	Kolaboratívny refaktoring	Eudovít Mydla	24.11.11	Normálna	Rozpracovaná
4.2.1	Tvorba vzorového kódu a unit testov	Eudovít Mydla	24.11.11	Normálna	Rozpracovaná

4.2.3	Upload dát	Martin Detko	24.11.11	Normálna	nová
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	Normálna	Rozpracovaná
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.2011	Normálne	Nová
5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihálik	1.12.2011	Normálna	Nová
5.1.3	Identifikácia používateľa cez cookie	Jakub Ševcech	1.12.2011	Vysoká	Nová

*Opis stretnutia:*

- Stretnutie viedol Jakub Ševcech
- Od posledného stretnutia pribudol v Redmine nový plugin, ktorý slúži na podporu SCRUM-u a funguje princípom drag-and-drop.
- Dohodli sme sa na určení závislostí úloh na každom stretnutí a hraničných termínov pre každú novú úlohu, aby sa skrátilo čakanie na vykonanie závislých úloh.
- Vznikla požiadavka na zmenu metrik, ktorá zahŕňa upraviť GUI, upraviť plugin, vytvoriť migrácie a hibernate prispôbiť zmenám.

*Stanovené úlohy:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>3.1</b>	Odporúčanie založené na obsahu	Eudovít Mydla	10.11.11	Normálna	Rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	Normálna	Nová
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	Normálna	Nová
3.2.1	Správa konzumentov	Jakub Ševcech	10.11.11	Normálna	Ukončená
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	Normálna	Rozpracovaná
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	Normálna	Rozpracovaná
3.5.1	Logovanie	Martin Detko	07.11.11	Vysoká	Nová
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák	10.11.11	Normálna	Nová
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	Normálna	Rozpracovaná
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	Normálna	Rozpracovaná
4.1.3	Vytvorenie metodiky	Michal Cádrik	24.11.11	Normálna	Ukončená
<b>4.2</b>	Kolaboratívny	Eudovít Mydla	24.11.11	Normálna	Ukončená



	refactoring				
4.2.1	Tvorba vzorového kódu a unit testov	Eudovít Mydla	24.11.11	Normálna	Ukončená
4.2.3	Upload dát	Martin Detko	24.11.11	Normálna	Ukončená
<b>4.3</b>	Napísať prihlášku do TP Cup	Michal Cádrik	22.11.11	Normálna	Ukončená
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.11	Normálne	Rozpracovaná
5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihálik	1.12.11	Normálna	Rozpracovaná
5.1.3	Identifikácia používateľa cez cookie	Jakub Ševcech	1.12.11	Vysoká	Ukončená
<b>6.1</b>	Požiadavka na zmenu metrik	Igor Slotík	4.12. 2011	Vysoká	Nová
6.1.1	Upraviť GUI	Jakub Ševcech	1.12. 2011	Vysoká	Nová
6.1.2	Upraviť plugin	Igor Slotík	4.12. 2011	Vysoká	Nová
6.1.3	Migrácie	Jakub Ševcech	1.12. 2011	Vysoká	Nová
6.1.4	Hibernate	Matej Mihálik	1.12. 2011	Vysoká	Nová

Poznámky:

-

**6.10. Zápis z desiateho stretnutia****Autor:** Ľudovít Mydla

Dátum: 8.12.2011  
 Trvanie: 12:00 – 12:45  
 Miestnosť: blok D – softvérové štúdio  
 Prítomní  
     Pedagóg: Ing. Michal Kompan  
     Členovia tímu: Michal Cádrik, Matej Červeňák, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech  
 Neprítomní: -  
 Zapisovateľ: Ľudovít Mydla

*Téma stretnutia:*  
 Záverečné pokyny.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	10.11.11	Normálna	Rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	10.11.11	Normálna	Čaká na spätnú väzbu
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	10.11.11	Normálna	Čaká na spätnú väzbu
3.2.2	Poskytnutie odporúčania	Michal Cádrik	10.11.11	Normálna	Čaká na spätnú väzbu
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	10.11.11	Vysoká	Rozpracovaná
3.5.1	Logovanie	Martin Detko	07.11.11	Vysoká	Ukončená
3.5.3	GUI - Vyhodnocovanie	Matej Červeňák, spravil Jakub Ševcech	10.11.11	Normálna	Ukončená
<b>4.1</b>	Pridávanie vlastných algoritmov od klienta	Jakub Ševcech	24.11.11	Normálna	Ukončená
4.1.2	Analýza a návrh pridávania vlastných algoritmov	Igor Slotík	24.11.11	Normálna	Ukončená
5.1.1	Vytvorenie metriky systémom pluginov	Igor Slotík	1.12.11	Normálne	Ukončená

5.1.2	Vytvorenie funkcie na prístup k DB	Matej Mihálik	1.12.11	Normálna	Ukončená
<b>6.1</b>	Požiadavka na zmenu metrik	Igor Slotík	4.12. 2011	Vysoká	Ukončená
6.1.1	Upraviť GUI	Jakub Ševcech	1.12. 2011	Vysoká	Ukončená
6.1.2	Upraviť plugin	Igor Slotík	4.12. 2011	Vysoká	Ukončená
6.1.3	Migrácie	Jakub Ševcech	1.12. 2011	Vysoká	Ukončená
6.1.4	Hibernate	Matej Mihálik	1.12. 2011	Vysoká	Ukončená

*Opis stretnutia:*

- Stretnutie viedol Igor Slotík
- Skontroloval sa stav úloh. Nové úlohy sú ukončené. Stále zostávajú úlohy z predchádzajúcich šprintov
- Matej Mihálik nás oboznámil s problémom insertovania a selectovania cez Hibernate pomocou objektov
- V najbližší štvrtok, 15.12.2011 okolo 1 hodiny poobede budeme prezentovať projekt
- Dokumentáciu projektu odovzdávame vo forme zakladača, aby sa dalo pridávať do budúcnosti

*Úlohy, ktoré je potrebné dokončiť:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>3.1</b>	Odporúčanie založené na obsahu	Ľudovít Mydla	12/13/11	Normálna	Rozpracovaná
3.1.1	API – Odporúčanie obsahom	Michal Cádrik	12/15/11	Normálna	Čaká na spätnú väzbu
<b>3.2</b>	Získanie odporúčania	Michal Cádrik	12/15/11	Normálna	Čaká na spätnú väzbu
3.2.2	Poskytnutie odporúčania	Michal Cádrik	12/15/11	Normálna	Čaká na spätnú väzbu
<b>3.5</b>	Vyhodnocovanie	Igor Slotík	12/15/11	Vysoká	Rozpracovaná

*Poznámky:*

-

**6.11. Zápis z jedenásteho stretnutia****Autor:** Jakub Ševcech

---

Dátum:	14.2.2012
Trvanie:	9:00 – 11:00
Miestnosť:	D - 220
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Jakub Ševcech

---

*Téma stretnutia:*

Prvé stretnutie v letnom semestri

*Výhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

Všetky úlohy z predchádzajúceho semestra boli dokončené.

*Opis stretnutia:*

- Stretnutie viedol Ľudovít Mydla.
- Dohodli sme sa na novom rozdelení úloh v tíme
  - Martin Detko – Manažment kvality
  - Michal Cádrik – Manažment dokumentácie
  - Matej Mihalik – Manažment plánovania
  - Igor Slotík – Manažment komunikácie
  - Jakub Ševcech – Manažment podpory vývoja
  - Ľudovít Mydla – Manažment rizík
- Určili sme úlohu Jakubovi Ševcechovi vytvoriť/prerobiť metodiku pre prácu s nástrojom na verziovanie zdrojových kódov GIT

*Stanovené úlohy:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
7.1	Úvod do 2. semestra	Jakub Ševcech	19.2.2012	normálna	Nová
7.1.1	Zmena DB v jave z Ruby	Matej Mihalik	19.2.2012	normálna	Nová
7.1.2	Nakonfigurovať	Jakub Ševcech	19.2.2012	normálna	Nová

	whenever				
7.1.3	Validácia XML	Jakub Ševcech, Martin Detko	19.2.2012	normálna	Nová
7.1.4	Upload veľkých súborov	Jakub Ševcech	19.2.2012	normálna	Nová
7.1.5	Pretestovať content-based odporúčanie	Ľudovít Mydla	19.2.2012	normálna	Nová
<b>7.2</b>	Prístup do DB	Matej Mihalik	19.2.2012	normálna	Nová
7.2.1	Vkladanie testovacích dát do DB	Matej Mihalik	19.2.2012	normálna	Nová
<b>7.3</b>	Nasadzovanie / monitorovanie	Jakub Ševcech	19.2.2012	normálna	Nová
7.3.1	Capistrano	Jakub Ševcech	19.2.2012	normálna	Nová
7.3.2	New Relic	Jakub Ševcech	19.2.2012	normálna	Nová
<b>7.4</b>	Kombinácia algoritmov	Martin Detko	19.2.2012	normálna	Nová
7.4.1	Všeobecný konfigurovač	MD, LM, IS, MM	19.2.2012	normálna	Nová
7.4.2	Kombinovač kaskáda	Michal Cádrik	19.2.2012	normálna	Nová
7.4.3	Kombinovač váhovaním	Igor Slotík	19.2.2012	normálna	Nová
7.4.4	Kombinovač switching	Martin Detko	19.2.2012	normálna	Nová
7.4.5	Odporúčanie zo zoznamu dokumentov	Michal Cádrik, Ľudovít Mydla	19.2.2012	normálna	Nová

Poznámky:

-

**6.12. Zápis z dvanásteho stretnutia****Autor:** Igor Slotík

---

Dátum:	20.2.2012
Trvanie:	11:00 – 12:00
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Igor Slotík

---

*Téma stretnutia:*

Vyhodnotenie 6. šprintu.

Diskusia o možných zlepšeniach odporúčača a rozdelenie nových úloh na 7. šprint.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>7.1</b>	Úvod do 2. semestra	Jakub Ševcech	26.2.2012	normálna	Nová
7.1.1	Zmena DB v jave z Ruby	Matej Mihalik	26.2.2012	normálna	Nová
7.1.2	Nakonfigurovať whenever	Jakub Ševcech	26.2.2012	normálna	Nová
7.1.3	Validácia XML	Jakub Ševcech, Martin Detko	18.2.2012	normálna	Uzatvorená
7.1.4	Upload veľkých súborov	Jakub Ševcech	26.2.2012	normálna	Nová
7.1.5	Pretestovať content-based odporúčanie	Ľudovít Mydla	17.2.2012	normálna	Spätná väzba
<b>7.2</b>	Prístup do DB	Matej Mihalik	26.2.2012	normálna	Nová
7.2.1	Vkladanie testovacích dát do DB	Matej Mihalik	26.2.2012	normálna	Nová
<b>7.3</b>	Nasadzovanie / monitorovanie	Jakub Ševcech	26.2.2012	normálna	Nová
7.3.1	Capistrano	Jakub Ševcech	19.2.2012	normálna	Uzatvorená
7.3.2	New Relic	Jakub Ševcech	26.2.2012	normálna	Nová
<b>7.4</b>	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna	Nová
7.4.1	Všeobecný	Ľudovít Mydla	19.2.2012	normálna	Ukončená

	konfigurovač				
7.4.2	Kombinovač kaskáda	Michal Cádrik	26.2.2012	normálna	Rozpracovaná
7.4.3	Kombinovač váhovaním	Igor Slotík	26.2.2012	normálna	Rozpracovaná
7.4.4	Kombinovač switching	Martin Detko	26.2.2012	normálna	Nová
7.4.5	Odporúčanie zo zoznamu dokumentov	Michal Cádrik, Ľudovít Mydla	19.2.2012	normálna	Ukončená

*Opis stretnutia:*

- Stretnutie viedol Jakub Ševcech.
- Ľudovít Mydla vytvoril metodiku pre prácu s Gitom.
- Martin Detko bol vybraný spolupracovať s Jakubom Ševcechom pri práci s JRuby.
- Dohodli sme sa na zmene odporúčačov a kombinovačov.
- V dátovom modeli sa zmení tabuľka *implementations*, kde pribudnú dva parametre. Jeden bude identifikovať typ odporúčača a druhý druh podobnosti.

## Úlohy, ktoré je potrebné dokončiť do nasledujúceho šprintu:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
7.1	Úvod do 2. semestra	Jakub Ševcech	26.2.2012	normálna	Nová
7.1.1	Zmena DB v jave z Ruby	Matej Mihálik	26.2.2012	normálna	Nová
7.1.2	Nakonfigurovať whenever	Jakub Ševcech	26.2.2012	normálna	Nová
7.1.4	Upload veľkých súborov	Jakub Ševcech	26.2.2012	normálna	Nová
7.2	Prístup do DB	Matej Mihálik	26.2.2012	normálna	Nová
7.2.1	Vkladanie testovacích dát do DB	Matej Mihálik	26.2.2012	normálna	Nová
7.3	Nasadzovanie / monitorovanie	Jakub Ševcech	26.2.2012	normálna	Nová
7.3.2	New Relic	Jakub Ševcech	26.2.2012	normálna	Nová
7.4	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna	Nová
7.4.2	Kombinovač kaskáda	Michal Cádrik	26.2.2012	normálna	Rozpracovaná
7.4.3	Kombinovač váhovaním	Igor Slotík	26.2.2012	normálna	Rozpracovaná
7.4.4	Kombinovač switching	Martin Detko	26.2.2012	normálna	Nová
7.4.5	Zmena kombinovača switching	Martin Detko	26.2.2012	normálna	Nová

7.4.6	Zmena kombinovača kaskáda	Michal Cádrik	26.2.2012	normálna	Nová
7.4.7	Zmena kombinovača váhovaním	Igor Slotík	26.2.2012	normálna	Nová
7.4.8	Urobiť rozhranie pre kombinovače	Jakub Ševcech	26.2.2012	normálna	Nová
7.4.9	Konfigurácia kombinovačov a odporúčačov	Martin Detko	26.2.2012	normálna	Nová
<b>8.1</b>	Zmena odporúčačov	Michal Cádrik	26.2.2012	normálna	Nová
8.1.1	Zmena content odporúčača	Eudovít Mydla	26.2.2012	normálna	Nová
8.1.1	Zmena collaborative odporúčača	Michal Cádrik	26.2.2012	normálna	Nová

Poznámky:

-



**6.13. Zápis z trinásteho stretnutia****Autor:** Martin Detko

---

Dátum:	27.2.2012
Trvanie:	11:00 – 12:00
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla
Neprítomní:	Jakub Ševcech
Zapisovateľ:	Martin Detko

---

*Téma stretnutia:*

Vyhodnotenie 7. šprintu.  
 Ukážka práce s testovacou databázou v jave  
 Rozdelenie nových úloh na 8. šprint.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>7.1</b>	Úvod do 2. semestra	Jakub Ševcech	26.2.2012	normálna	Ukončená
7.1.1	Zmena DB v jave z Ruby	Matej Mihalik	26.2.2012	normálna	Ukončená
7.1.2	Nakonfigurovať whenever	Jakub Ševcech	26.2.2012	normálna	Ukončená
7.1.4	Upload veľkých súborov	Jakub Ševcech	26.2.2012	normálna	Ukončená
<b>7.2</b>	Prístup do DB	Matej Mihalik	26.2.2012	normálna	Ukončená
7.2.1	Vkladanie testovacích dát do DB	Matej Mihalik	26.2.2012	normálna	Ukončená
<b>7.3</b>	Nasadzovanie / monitorovanie	Jakub Ševcech	26.2.2012	normálna	Ukončená
7.3.2	New Relic	Jakub Ševcech	26.2.2012	normálna	Ukončená
<b>7.4</b>	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna	Rozpracovaná
7.4.2	Kombinovač kaskáda	Michal Cádrik	26.2.2012	normálna	Ukončená
7.4.3	Kombinovač váhovaním	Igor Slotík	26.2.2012	normálna	Ukončená
7.4.4	Kombinovač switching	Ľudovít Mydla	26.2.2012	normálna	Rozpracovaná
7.4.5	Zmena kombinovača switching	Ľudovít Mydla	26.2.2012	normálna	Rozpracovaná

7.4.6	Zmena kombinovača kaskáda	Michal Cádrik	26.2.2012	normálna	Ukončená
7.4.7	Zmena kombinovača váhovaním	Igor Slotík	26.2.2012	normálna	Ukončená
7.4.8	Urobiť rozhranie pre kombinovače	Jakub Ševcech	26.2.2012	normálna	Rozpracovaná
7.4.9	Konfigurácia kombinovačov a odporúčačov	Martin Detko	26.2.2012	normálna	Zrušená

*Opis stretnutia:*

- Stretnutie viedol Igor Slotík.
- Matej Mihalik predviedol ako vytvárať testovanie v Jave pomocou nových funkcií
- Rozhodli sme sa stanoviť si pravidlo pri určovaní podobností, že najväčšie číslo znamená najväčšiu podobnosť.
- Dohodli sme sa, že ani tvorca implementácií ich nemôže ani mazať a ani meniť.

## Nové úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Nová
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna	Nová
<b>9.2</b>	Získať keywordy z článku	Matej Mihalik	5. 3. 2012	normálna	Nová
9.2.1	Predspracovať článok	Ľudovít Mydla	5. 3. 2012	normálna	Nová
9.2.2	Získať keywordy z predspracovaného textu	Matej Mihalik	5. 3. 2012	normálna	Nová
<b>9.3</b>	Coldstart problém	Michal Cádrik	5. 3. 2012	normálna	Nová
9.3.1	Vytvoriť možnosti doplnenia odporúčania	Michal Cádrik	5. 3. 2012	normálna	Nová
9.3.2	Konfigurácia výberu doplnenia odporúčania	Martin Detko	5. 3. 2012	normálna	Nová
<b>9.4</b>	Testovanie kombinovačov	Igor Slotík	5. 3. 2012	normálna	Nová
9.4.1	Naštudovať Jmock a vytvoriť vzorové testy	Igor Slotík	5. 3. 2012	normálna	Nová
<b>9.5</b>	Úprava poskytnutia odporúčania	Martin Detko	5. 3. 2012	normálna	Nová
9.5.1	Doplnenie odporúčania o kombinovač	Martin Detko	5. 3. 2012	normálna	Nová

*Poznámky: -*

**6.14. Zápis zo štrnásteho stretnutia****Autor:** Matej Mihalik

---

Dátum:	05.3.2012
Trvanie:	11:00 – 13:30
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla, Jakub Ševcech
Nepřítomní:	-
Zapisovateľ:	Matej Mihalik

---

*Téma stretnutia:*

- Vyhodnotenie 8. šprintu.
- Prezentácia JMOCK knižnice
- Diskusia ohľadne rôznych menších zmien v aplikácii
- Rozdelenie nových úloh na 9. šprint.

*Vyhodnotenie úloh z predchádzajúceho stretnutia/ Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
7.4	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna	Rozpracovaná
7.4.2	Kombinovač kaskáda	Michal Cádrik	26.2.2012	normálna	Ukončená
7.4.3	Kombinovač váhovaním	Igor Slotík	26.2.2012	normálna	Ukončená
7.4.4	Kombinovač switching	Ľudovít Mydla	26.2.2012	normálna	Rozpracovaná
7.4.5	Zmena kombinovača switching	Ľudovít Mydla	26.2.2012	normálna	Rozpracovaná
7.4.6	Zmena kombinovača kaskáda	Michal Cádrik	26.2.2012	normálna	Ukončená
7.4.7	Zmena kombinovača váhovaním	Igor Slotík	26.2.2012	normálna	Ukončená
7.4.8	Urobiť rozhranie pre kombinovače	Jakub Ševcech	26.2.2012	normálna	Rozpracovaná
7.4.9	Konfigurácia kombinovačov a odporúčačov	Martin Detko	26.2.2012	normálna	Zrušená
9.1	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
9.1.1	Vytvoriť bezpečné vkladanie modulov do	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná

	projektu				
<b>9.2</b>	Získať keywordy z článku	Matej Mihalik	5. 3. 2012	normálna	Rozpracovaná
9.2.1	Predspracovať článok	Ľudovít Mydla	5. 3. 2012	normálna	Rozpracovaná
9.2.2	Získať keywordy z predspracovaného textu	Matej Mihalik	5. 3. 2012	normálna	Rozpracovaná
<b>9.3</b>	Coldstart problem	Michal Cádrik	5. 3. 2012	normálna	Rozpracovaná
9.3.1	Vytvoriť možnosti doplnenia odporúčania	Michal Cádrik	5. 3. 2012	normálna	Rozpracovaná
9.3.2	Konfigurácia výberu doplnenia odporúčania	Martin Detko	5. 3. 2012	normálna	Rozpracovaná
<b>9.4</b>	Testovanie kombinovačov	Igor Slotík	5. 3. 2012	normálna	Ukončená
9.4.1	Naštudovať Jmock a vytvoriť vzorové testy	Igor Slotík	5. 3. 2012	normálna	Ukončená
<b>9.5</b>	Úprava poskytnutia odporúčania	Martin Detko	5. 3. 2012	normálna	Rozpracovaná
9.5.1	Doplnenie odporúčania o kombinovač	Martin Detko	5. 3. 2012	normálna	Rozpracovaná

*Opis stretnutia:*

- Stretnutie viedol Martin Detko.
- Historický míľnik - oslávili sme prvé lokálne spustené odporúčenie.
- Prezentácia testovania s využitím JMOCK knižnice by Igor Slotík.
- Rozprávali sme a dohodli sme sa na zmenách DB modelu vyplývajúcich zo spracovávania spätnej väzby.
- Upravovali sme interface pre odporúčače.
- Martin Detko navrhol nový spôsob ako pristúpiť k implementácii kaskádového kombinovača, nakoľko sa nám však tento spôsob zdal výpočtovo príliš náročný, nakoniec sme sa ho vzdali.
- Pripomenuli sme si potrebu dohliadnuť na vytvorenie dostatočne robustného setu funkcií nad našou databázou, pre všeobecné verejné využitie. Dohodli sme sa nakoniec však, že máme aj urgentnejšie úlohy, ktoré treba najskôr vyriešiť, a tak sme túto činnosť odložili na prediskutovanie na budúcom stretnutí.

Nové úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>10.1</b>	Úprava odporúčačov podľa nového rozhrania	Ľudovít Mydla	12. 3. 2012	normálna	Nová
10.1.1	Úprava content-based odporúčača	Ľudovít Mydla	12. 3. 2012	Normálna	Nová
10.1.2	Úprava collaborative odporúčača	Michal Cádrik	12. 3. 2012	normálna	Nová
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna	Nová
10.2.1	Spúšťanie metrík	Martin Detko	12. 3. 2012	normálna	Nová
10.2.2	Spúšťanie podobnosti	Jakub Ševcech	12. 3. 2012	normálna	Nová
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna	Nová
<b>10.3</b>	Kontrola databázy	Jakub Ševcech	12. 3. 2012	normálna	Nová
10.3.1	Doplnenie aktuálnych indexov a ohraňení	Jakub Ševcech	12. 3. 2012	normálna	Nová
<b>10.4</b>	JMOCK	Igor Slotík	12 3. 2012	normálna	Nová
10.4.1	Tvorba JMOCK metriky	Igor Slotík	12. 3. 2012	normálna	Nová
<b>10.5</b>	Spätná väzba	Igor Slotík	12. 3. 2012	normálna	Nová
10.5.1	Naštudovanie Precision and Recall prístupu	Igor Slotík	12. 3. 2012	normálna	Nová
<b>10.6</b>	Integrácia vlastných algoritmov na extrakciu keywordov	Matej Mihalik	12. 3. 2012	normálna	Nová
10.6.1	Implementácia keyword algoritmov do procesu importovania	Martin Detko	12. 3. 2012	normálna	Nová

Poznámky:

-

**6.15. Zápis z pätnásteho stretnutia****Autor:** Michal Cádrik

---

Dátum:	12.3.2012
Trvanie:	11:00 – 12:50
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Michal Cádrik

---

*Téma stretnutia:*

- Vyhodnotenie 9. šprintu.
- Zefektívnenie niektorých algoritmov.
- Debata o počítaní váh podobností a odporúčaní.
- Rozdelenie nových úloh na 10. šprint.

*Vyhodnotenie úloh z predchádzajúceho stretnutia / Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>7.4</b>	Kombinácia algoritmov	Martin Detko	26.2.2012	normálna	Ukončená
7.4.4	Kombinovač switching	Ľudovít Mydla	26.2.2012	normálna	Ukončená
7.4.5	Zmena kombinovača switching	Ľudovít Mydla	26.2.2012	normálna	Ukončená
7.4.8	Urobiť rozhranie pre kombinovače	Jakub Ševcech	26.2.2012	normálna	Ukončená
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
<b>9.2</b>	Získať keywordy z článku	Matej Mihalik	5. 3. 2012	normálna	Ukončená
9.2.1	Predspracovať článok	Ľudovít Mydla	5. 3. 2012	normálna	Ukončená
9.2.2	Získať keywordy z predspracovaného textu	Matej Mihalik	5. 3. 2012	normálna	Ukončená
<b>9.3</b>	Coldstart problem	Michal Cádrik	5. 3. 2012	normálna	Rozpracovaná
9.3.1	Vytvoriť možnosti doplnenia odporúčania	Michal Cádrik	5. 3. 2012	normálna	Rozpracovaná
9.3.2	Konfigurácia výberu	Martin Detko	5. 3. 2012	normálna	Ukončená

	doplňenia odporúčania				
<b>9.5</b>	Úprava poskytnutia odporúčania	Martin Detko	5. 3. 2012	normálna	Ukončená
9.5.1	Doplňenie odporúčania o kombinovač	Martin Detko	5. 3. 2012	normálna	Ukončená
<b>10.1</b>	Úprava odporúčačov podľa nového rozhrania	Ľudovít Mydla	12. 3. 2012	normálna	Ukončená
10.1.1	Úprava content-based odporúčača	Ľudovít Mydla	12. 3. 2012	normálna	Ukončená
10.1.2	Úprava collaborative odporúčača	Michal Cádrik	12. 3. 2012	normálna	Ukončená
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
10.2.1	Spúšťanie metrík	Jakub Ševcech	12. 3. 2012	normálna	Ukončená
10.2.2	Spúšťanie podobnosti	Jakub Ševcech	12. 3. 2012	normálna	Ukončená
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
<b>10.3</b>	Kontrola databázy	Jakub Ševcech	12. 3. 2012	normálna	Ukončená
10.3.1	Doplňenie aktuálnych indexov a ohraničení	Jakub Ševcech	12. 3. 2012	normálna	Ukončená
<b>10.4</b>	JMOCK	Igor Slotík	512 3. 2012	normálna	Ukončená
10.4.1	Tvorba JMOCK metriky	Igor Slotík	12. 3. 2012	normálna	Ukončená
<b>10.5</b>	Spätná väzba	Igor Slotík	12. 3. 2012	normálna	Ukončená
10.5.1	Naštudovanie Precision and Recall prístupu	Igor Slotík	12. 3. 2012	normálna	Ukončená
<b>10.6</b>	Integrácia vlastných algoritmov na extrakciu keywordov	Matej Mihalik	12. 3. 2012	normálna	Ukončená
10.6.1	Implementácia keyword algoritmov do procesu importovania	Martin Detko	12. 3. 2012	normálna	Ukončená

*Opis stretnutia:*

- Stretnutie viedol Matej Mihalik.
- Rozprava o možnostiach zefektívnenia algoritmov.
- Rozprava o zmenu počítania podobnosti, nemožnosť zistenia typu spätnej väzby (kombinovať spätnú väzbu alebo pridať flag do DB).
- Výsledok - je lepšie mať iba jednu spätnú väzbu, ktorá je kombinovaná (zoznam typov a váh).



- Dôležitosť feedback type v odporúčačoch.
- Výsledok kombinovania viacerých feedbackov je  $\log(\text{suma}/\text{počet})$  v jave implementovaná ako helper funkcia.
- Debata ohľadom lematizérov pre rôzne jazyky (pridanie atribútu pre dokument – jazyk)

Nové úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>11.1</b>	Zmena počítania podobností	Matej Mihalik	19. 3. 2012	normálna	Nová
11.1.1	Zmena podľa viacerých feedbackov	Matej Mihalik	19. 3. 2012	normálna	Nová
11.1.2	Feedbacky s ováňovaním	Matej Mihalik	19. 3. 2012	normálna	Nová
11.1.3	Implementácia funkcie počítania jednej váhy z viacerých feedbackov	Matej Mihalik	19. 3. 2012	normálna	Nová
<b>11.3</b>	Spracovanie anglických textov	Ľudovít Mydla	19. 3. 2012	normálna	Nová
10.3.1	Vytvorenie Porterovho algoritmu	Ľudovít Mydla	19. 3. 2012	normálna	Nová
<b>11.4</b>	Zmeny plynúce zo zmien	Igor Slotík	19. 3. 2012	normálna	Nová
11.4.1	Zmena v Collaborative recommendation – výkon, počítanie váh	Michal Cádrik	19. 3. 2012	normálna	Nová
11.4.2	Zmena v Content-based – počítanie váh	Ľudovít Mydla	19. 3. 2012	normálna	Nová
11.4.3	Zmena rozhrania konfigurácie podobnosti	Jakub Ševcech	19. 3. 2012	normálna	Nová
11.4.3	Vyladenie metrík	Igor Slotík	19. 3. 2012	normálna	Nová
11.4.4	Zmena v metrikách	Jakub Ševcech	19. 3. 2012	normálna	Nová
<b>11.5</b>	AB testovanie	Martin Detko	19. 3. 2012	normálna	Nová
11.5.1	Vytvoriť AB testy	Martin Detko	19. 3. 2012	normálna	Nová
<b>11.6</b>	Dokumentácia	Michal Cádrik	19. 3. 2012	normálna	Nová
11.6.1	Používateľská dokumentácia API	Michal Cádrik	19. 3. 2012	normálna	Nová

Poznámky:

-

**6.16. Zápis zo šestnásteho stretnutia****Autor:** Jakub Ševcech

---

Dátum:	19.3.2012
Trvanie:	11:00 – 12:50
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Jakub Ševcech

---

*Téma stretnutia:*

- Vyhodnotenie 10. šprintu.
- Rozdelenie úloh na nasledujúci týždeň.
- Naplánovanie integrácie projektu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia / Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
<b>9.3</b>	Coldstart problem	Michal Cádrik	5. 3. 2012	normálna	Ukončená
9.3.1	Vytvoriť možnosti doplnenia odporúčania	Michal Cádrik	5. 3. 2012	normálna	Ukončená
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
<b>11.1</b>	Zmena počítania podobností	Matej Mihalik	19. 3. 2012	normálna	Rozpracovaná
<b>11.3</b>	Spracovanie anglických textov	Ľudovít Mydla	19. 3. 2012	normálna	Ukončená
10.3.1	Vytvorenie Porterovho algoritmu	Ľudovít Mydla	19. 3. 2012	normálna	Ukončená
<b>11.4</b>	Zmeny plynúce zo zmien	Igor Slotík	19. 3. 2012	normálna	Ukončená
11.4.1	Zmena v Collaborative recommendation –	Michal Cádrik	19. 3. 2012	normálna	Rozpracovaná

	výkon, počítanie váh				
11.4.2	Zmena v Content-based – počítanie váh	Ľudovít Mydla	19. 3. 2012	normálna	Rozpracovaná
11.4.3	Zmena rozhrania konfigurácie podobnosti	Jakub Ševcech	19. 3. 2012	normálna	Ukončená
11.4.3	Vyladenie metrik	Igor Slotík	19. 3. 2012	normálna	Rozpracovaná
11.4.4	Zmena v metrikách	Jakub Ševcech	19. 3. 2012	normálna	Ukončená
<b>11.5</b>	AB testovanie	Martin Detko	19. 3. 2012	normálna	Ukončená
11.5.1	Vytvoriť AB testy	Martin Detko	19. 3. 2012	normálna	Ukončená
<b>11.6</b>	Dokumentácia	Michal Cádrik	19. 3. 2012	normálna	Rozpracovaná
11.6.1	Používateľská dokumentácia API	Michal Cádrik	19. 3. 2012	normálna	Rozpracovaná

*Opis stretnutia:*

- Stretnutie viedol Michal Cádrik.
- Zhodnotili sme stav súčasných úloh a dohodli sme sa že nebudeme začínať ďalší šprint, ale predĺžime ten súčasný. Dohodli sme sa na tom, že všetky rozpracované úlohy musia byť dokončené najneskôr vo štvrtok tak, aby sme mohli nasadiť aplikáciu na server a integrovať ju so systémom ALEF.
- Dohodli sme sa na tom, že po dokončení rozpracovaných úloh sa sústredíme na vytvorenie používateľskej dokumentácie.

## Nové úlohy:

Na nasledujúci týždeň sme nenaplánovali žiadne ďalšie úlohy. Určili sme si, že do štvrtka 22.3.2012 musia byť všetky rozpracované úlohy dokončené tak, aby sme mohli integrovať a nasadiť aplikáciu na server.

*Poznámky:*

-

**6.17. Zápis zo sedemnásteho stretnutia****Autor:** Ľudovít Mydla

---

Dátum:	26.3.2012
Trvanie:	11:00 – 12:50
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla, Jakub Ševcech
Nepřítomní:	-
Zapisovateľ:	Ľudovít Mydla

---

*Téma stretnutia:*

- Vyhodnotenie 11. šprintu.
- Rozdelenie úloh na nasledujúci týždeň.
- Naplánovanie testovania a uzatvorenia projektu.

*Vyhodnotenie úloh z predchádzajúceho stretnutia / Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna	Rozpracovaná
<b>11.1</b>	Zmena počítania podobností	Matej Mihalik	19. 3. 2012	normálna	Ukončená
11.4.1	Zmena v Collaborative recommendation – výkon, počítanie váh	Michal Cádrik	19. 3. 2012	normálna	Ukončená
11.4.2	Zmena v Content-based – počítanie váh	Ľudovít Mydla	19. 3. 2012	normálna	Zrušená
11.4.3	Vyladenie metrik	Igor Slotík	19. 3. 2012	normálna	Rozpracovaná
<b>11.6</b>	Dokumentácia	Michal Cádrik	19. 3. 2012	normálna	Ukončená
11.6.1	Používateľská dokumentácia API	Michal Cádrik	19. 3. 2012	normálna	Ukončená

*Opis stretnutia:*

- Stretnutie viedol Jakub Ševcech.
- Zhodnotili sme úlohy posledného týždňa. Dohodli sme sa že v pondelok, uzavrieme v Redmine všetky úlohy, okrem tých, ktoré sa naozaj nestihli spraviť za daný týždeň. Takisto sme sa dohodli, že začneme už testovať, ladiť a optimalizovať náš systém a takisto ho aj dokumentovať v zdrojovom kóde a v tlačenej dokumentácii

Nové úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>12.1</b>	Zmena importu relácií	Martin Detko	2. 4. 2012	normálna	Nová
12.2	Tabuľka Lemmas do migrácie	Jakub Ševcech	2. 4. 2012	normálna	Nová
<b>12.3</b>	Prerobenie API pre feedback	Jakub Ševcech	2. 4. 2012	normálna	Nová

Poznámky:

-

**6.18. Zápis z osemnásteho stretnutia****Autor:** Igor Slotík

---

Dátum:	2.4.2012
Trvanie:	11:00 – 12:30
Miestnosť:	Softvérové laboratórium
Prítomní	
Pedagóg:	Ing. Michal Kompan
Členovia tímu:	Michal Cádrik, Martin Detko, Matej Mihalik, Igor Slotík, Ľudovít Mydla, Jakub Ševcech
Neprítomní:	-
Zapisovateľ:	Igor Slotík

---

*Téma stretnutia:*

Vyhodnotenie 12. šprintu.  
 Dokumentácia k inžinierskemu dielu a dokumentácia k riadeniu.  
 JUNIT testovanie.  
 Nasadzovanie.  
 Testovanie nasadených častí.  
 Rozdelenie úloh na ďalší šprint.

*Vyhodnotenie úloh z predchádzajúceho stretnutia / Priebežný stav úloh šprintu:*

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>9.1</b>	Riešenie bezpečnosti	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
9.1.1	Vytvoriť bezpečné vkladanie modulov do projektu	Jakub Ševcech	5. 3. 2012	normálna	Rozpracovaná
<b>10.2</b>	Testovanie integrácie	Jakub Ševcech	12. 3. 2012	normálna	Uzavretá
10.2.3	Testovanie výkonu servera	Jakub Ševcech	12. 3. 2012	normálna	Uzavretá
<b>11.1</b>	Zmena počítania podobností	Matej Mihalik	19. 3. 2012	normálna	Uzavretá
11.4.1	Zmena v Collaborative recommendation – výkon, počítanie váh	Michal Cádrik	19. 3. 2012	normálna	Uzavretá
11.4.3	Vyladenie metrík	Igor Slotík	19. 3. 2012	normálna	Uzavretá
<b>11.6</b>	Dokumentácia	Michal Cádrik	19. 3. 2012	normálna	Uzavretá
11.6.1	Používateľská dokumentácia API	Michal Cádrik	19. 3. 2012	normálna	Uzavretá
<b>12.1</b>	Zmena importu relácií	Martin Detko	2. 4. 2012	normálna	Uzavretá

12.2	Tabuľka Lemmas do migrácie	Jakub Ševcech	2. 4. 2012	normálna	Uzavretá
<b>12.3</b>	Prerobenie API pre feedback	Jakub Ševcech	2. 4. 2012	normálna	Uzavretá

*Opis stretnutia:*

- Stretnutie viedol Ľudovít Mydla.
- Zhodnotili sme úlohy posledného týždňa.
- Dohodli sme sa na čo najrýchlejšej tvorbe dokumentácie k inžinierskemu dielu a k riadeniu (podrobnejšie rozdelenie úloh je v tabuľke).
- Ladenie, optimalizácia a testovanie.
- Každý je zodpovedný za nasadzovanie svojej časti projektu.
- Jakub Ševcech zmenil logo nášho tímu.
- Ľudovít Mydla vytvorí plagát na IT.SRC.
- Ľudovít Mydla dopracuje lematizér, aby poradie spracovania dokumentu bolo v nasledovnom poradí:
  1. Odstránenie HTML
  2. Úprava Upper-case na lower-case
  3. Odstránenie znamienok
  4. Samotná lematizácia

## Nové úlohy:

ID	Popis	Zodpovednosť	Dátum splnenia	Priorita	Stav
<b>13.1</b>	Dokumentácia	Michal Cádrik	7. 10. 2012	normálna	Nová
13.1.1	Používateľská príručka	Michal Cádrik	7. 10. 2012	normálna	Nová
13.1.2	Prepracovať dokumentáciu k testovaniu	Martin Detko	7. 10. 2012	normálna	Nová
13.1.3	Vyhodnotenie plánovania	Matej Mihalik	7. 10. 2012	normálna	Nová
13.1.4	Dokumentácia k manažmentu rizík	Ľudovít Mydla	7. 10. 2012	normálna	Nová
13.1.5	Používateľská príručka ku Content-based odporúčaniam	Ľudovít Mydla	7. 10. 2012	normálna	Nová
13.1.6	Dokumentácia k JMOCK testovaniu	Igor Slotík	7. 10. 2012	normálna	Nová
13.1.7	Používateľská príručka	Igor Slotík	7. 10. 2012	normálna	Nová

	k metrikám				
13.1.8	Popis nástrojov	Jakub Ševcech	7. 10. 2012	normálna	Nová
<b>13.2</b>	Príprava na IIT.SRC	Eudovít Mydla	12.10. 2012	normálna	Nová
13.2.1	Vytvorenie plagátu	Eudovít Mydla	12.10. 2012	normálna	Nová
<b>13.3</b>	Lematizér	Eudovít Mydla	12.10. 2012	normálna	Nová
13.3.1	Dopracovanie lematizéra	Eudovít Mydla	12.10. 2012	normálna	Nová
<b>13.4</b>	Import dát	Jakub Ševcech	12.10. 2012	normálna	Nová
13.4.1	Pridanie Javascriptu do používateľského rozhrania	Jakub Ševcech	12.10. 2012	normálna	Nová

*Poznámky:*

Dôležité: Nasledujúce stretnutie sa uskutoční vo štvrtok 12.4. 2012 o 8:00 v softvérovom štúdiu.



## 7. Základná smernica

**Autor:** Jakub Ševcech, Michal Cádrik

### 7.1. Konvencia názvov

Typ	Názov
Trieda	Camel Case, začiatkové písmeno veľké
Metóda	Camel Case, začiatkové písmeno malé
Balík	Začiatočné písmeno malé

Tabuľka 7.1: Tabuľka konvencie názvov

#### Konvencia balíkov

- *Ruby projekt*
  - *controller (prepojenie logiky so zobrazovaním)*
  - *model (model databázy a logika aplikácie)*
  - *view (prezentácia HTML / XML)*
  - *adresár \lib\jrecommender (obsahuje Java projekt)*
    - *src*
      - *core (jadro odporúčača)*
        - *model (triedy objektového modelu databázy)*
        - *interfaces (rozhrania na implementovanie jednotlivých častí odporúčača)*
        - *worker*
      - *balíky ďalších funkcií*
    - *lib (.jar súbory závislostí)*

### 7.2. Kontrola kódu

Kontrola kódu po druhom vývojárovi (code-review) je zabezpečovaná nasledovne:

Vývojár zodpovedný za používateľský príbeh určí dvojicu ľudí, ktorí si budú navzájom kontrolovať kód, vrátane testov a dokumentácie. Ak napríklad Ľudovít Mydla je zodpovedný za používateľský príbeh a v rámci neho sú úlohy pridelené Jakubovi Ševcechovi, Michalovi Cádrikovi a Martinovi Detkovi, tak Ľudovít Mydla môže určiť Michala Cádrika aby kontroloval kód Jakubovi Ševcechovi, Martina Detka aby kontroloval kód Michalovi Cádrikovi, atď.

Kontrolujú sa nasledovné aspekty kódu:

- Správne formátovanie a odsadzovanie kódu.
- Správne používanie zátvoriek.

- Komentáre v kóde musia byť v angličtine.
- Pre implementáciu v jazyku Java musí byť použitý javadoc v angličtine, ktorý popisuje všetky vstupné parametre a výstupnú hodnotu. Pre implementáciu komponentov v jazyku JRuby musí byť podobne použitý nástroj RDoc.
- Musí byť uvedený autor danej triedy alebo metódy.
- Prehľadnosť kódu.
- Efektívnosť implementácie.
- Pokrytie testami.
- Dokumentácia pripojená k implementácií.

### 7.3. Povinnosti zapisovateľa stretnutia

1. Zapísať priebeh stretnutia s pridelenými úlohami podľa šablóny.
  - poznámka1: Na oddelovanie číslovania používajte bodky a nie čiarky !!!
  - poznámka2: Korektne si píšme mená navzájom.
2. Na začiatku každého šprintu vytvoriť v nástroji Redmine záznam o novom šprinte. Názov šprintu je tvorený menami zvierat v latinčine podľa abecedy vo formáte *XX\_zviera*, kde *XX* je poradové číslo šprintu.
3. Vytvoriť úlohy v Redmine najneskôr na ďalší deň po stretnutí.
4. Odovzdať *.odt* súbor do Redmine a *.pdf* súbor nahrať na stránku tímu.
5. Pridať zápis do dokumentácie k riadeniu.
6. Viest' nasledujúce stretnutie.

Názvová konvencia zápisov zo stretnutia:

Názov	Umiestnenie	Príklad
Zápis zo stretnutia číslo X.pdf	<a href="http://vm28.ucebne.fiit.stuba.sk/webpage/?page_id=37">http://vm28.ucebne.fiit.stuba.sk/webpage/?page_id=37</a>	Zápis zo stretnutia číslo 1.pdf
Zapis_z/zo_XXXXX_stretnutia.odt	<a href="http://vm28.ucebne.fiit.stuba.sk/Redmine/projects/recommend/files">http://vm28.ucebne.fiit.stuba.sk/Redmine/projects/recommend/files</a>	Zapis_z_prveho_stretnutia.odt

Tabuľka 7.2: Konvencia názvov pre zápisy zo stretnutí

### 7.4. Postup nahratia dokumentu na stránku tímu

1. Otvorenie stránky [vm28.ucebne.fiit.stuba.sk/webpage/wp-admin](http://vm28.ucebne.fiit.stuba.sk/webpage/wp-admin).
2. Meno je *admin* a heslo každý pozná (ak nie kontaktujte iných členov tímu).
3. Vľavo kliknúť na *Pages* potom na *Dokumenty* a záložka *Visual*.
4. Kliknúť na znak hviezdičky vedľa *Upload/Insert*.
5. Zobrazí sa okno. Kliknúť na *Select files*.
6. Vybrať súbor pre nahranie.

7. Zmeniť názov v Title podľa tabuľky 7.2
8. Kliknúť na *Insert into post*.
9. Premiestniť a naformátovať post podľa predošlých a zaradiť ho na správne miesto.
10. Na pravej strane kliknúť *Upload*.
11. Skontrolujte výsledok na stránke.

## 8. Manažment kvality

**Autor:** Martin Detko

### 8.1. Roly a zodpovednosti

Roly	Zodpovednosti
Vývojár	• vytvorenie samotnej funkcionality
	• vytvorenie jednotkových testov
	• spúšťanie jednotkových testov
	• zdokumentovanie výsledkov
Manažér kvality	• kontrola pokrytia testovacích prípadov testami
	• vyhodnotenie výsledkov testov
Tester	• vytvorenie dávkového spúšťania testov
	• hlásenie nefunkčných testov
Manažér podpory vývoja	• zabezpečenie nástrojov pre testovanie

Tabuľka 8.1: Roly a zodpovednosti pri testovaní

### 8.2. Vytvorenie novej funkcionality v projekte

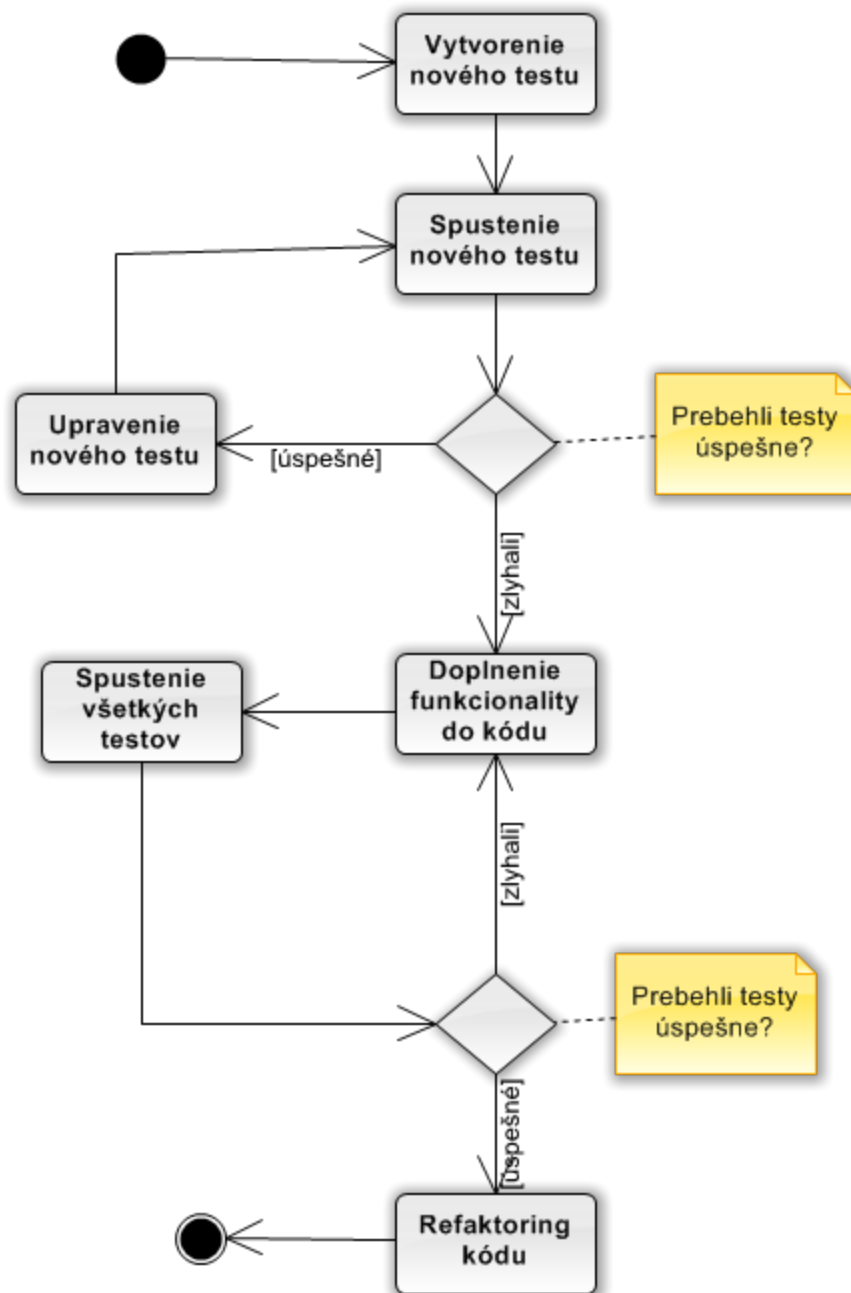
Pri vyvíjaní sa používa vývoj riadený testami. Pri pridávaní novej funkcionality sa postupuje nasledovne:

Vstup	Rozpracovaný fungujúci projekt bez požadovanej funkcionality.
Výstup	Rozpracovaný fungujúci projekt s požadovanou funkcionalitou.
Zodpovedný	Vývojár

1.	Vytvorenie nového jednotkového testu.
2.	Spustenie nového testu. Ak test zlyhal, krok 3 sa preskočí.
3.	Upravenie nového testu. A pokračuje sa krokom 2.
4.	Doplnenie funkcionality do kódu.
5.	Spustenie všetkých testov. Ak test zlyhal pokračuje sa krokom 4.
6.	Úprava vzhľadu kódu.

Postup je zachytený aj na obrázku diagramu aktivít č.1. Po pridání jednej funkcionality môže vývoj

pokračovať ďalším pridávaním funkcionalít alebo sa zastaviť.



Obrázok 8.1: Diagram aktivít, vývoj riadený testami

### 8.3. Vytvorenie JUnit testu

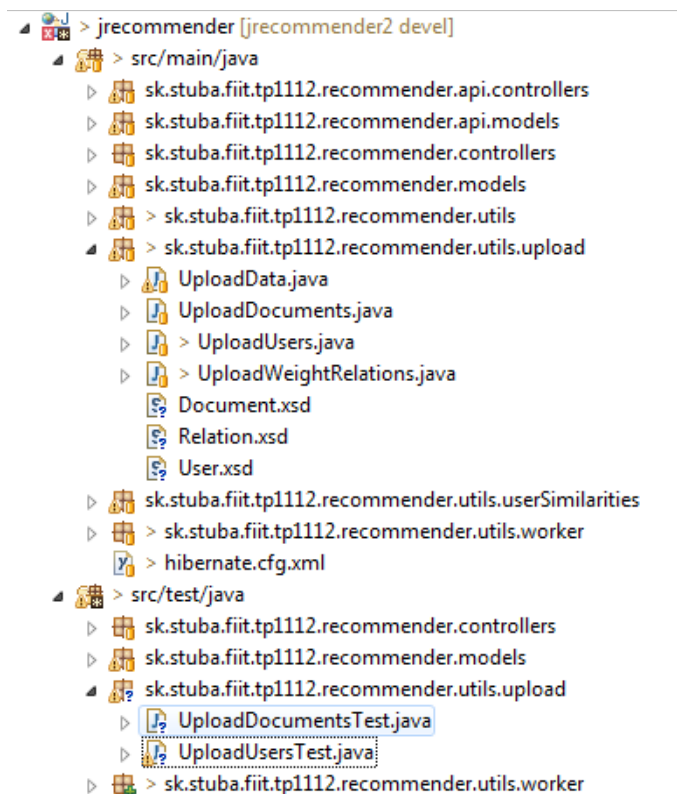
Pre testovanie časti produktu vyvíjaného v Jave sa využíva framework JUnit 4.0. V ďalšej časti je opísaný postup vytvárania nových jednotkových testov v JUnit.

#### Vytvorenie testovacej triedy

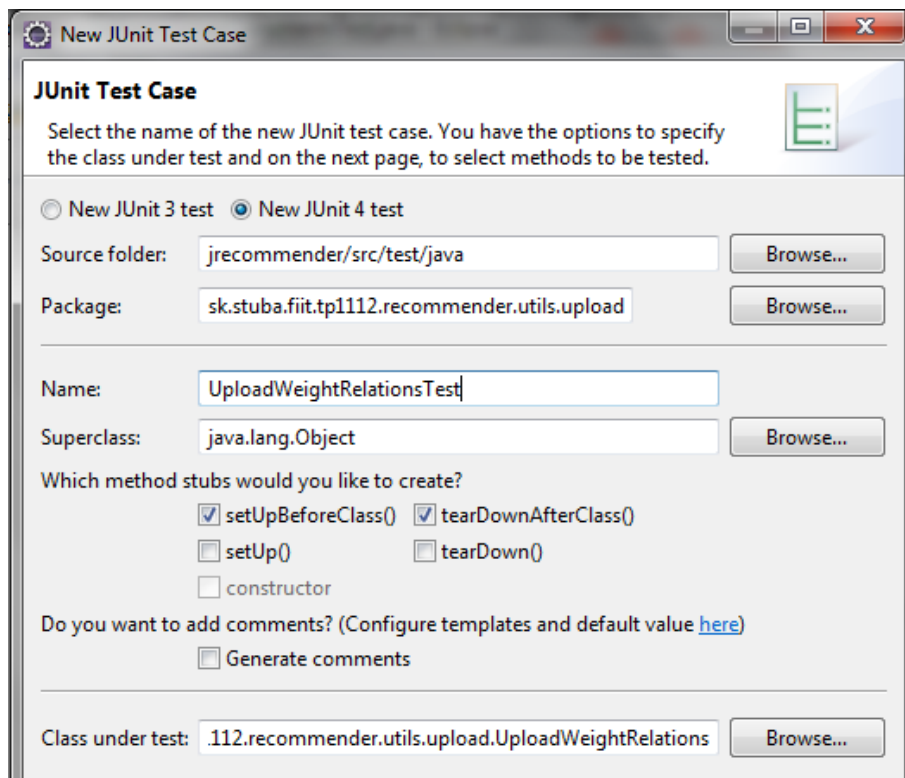
Pred vytvorením testovacích metód je potrebné vytvoriť testovacie triedy. Tieto triedy sa budú starať o inicializáciu testovacích dát. Okrem toho tieto triedy vytvárajú systém, v ktorom sa ľahšie orientuje. Postup vytvorenia novej testovacej triedy je nasledujúci.

Vstup	Vytvorená prázdna trieda, pre ktorú ideme vytvárať jednotkové testy. Spustené prostredie Eclipse.
Výstup	Vytvorená testovacia trieda.
Zodpovedný	Vývojár

1.	Ak balíček pre novú testovaciu triedu ešte neexistuje, vytvorí sa v adresári src/test/java. Názov balíčka bude rovnaký ako názov balíčka, v ktorom sa nachádza trieda, ktorá bude testovaná. Presnejšiu štruktúru vidieť na obrázku č. 8.2
2.	Pravým tlačítkom sa vyberie balíček, do ktorého bude vložená testovacia trieda. Vyberie sa možnosť File > JUnit Test Case.
3.	Následne v dialógu, obrázok č. 8.3, sa vyberie JUnit 4 test, potom sa vyplní trieda, ktorú chcem otestovať. Názov testovanej triedy sa zadá ako [názov balíčka].[názov testovanej triedy]. Potom sa vyplní názov testovacej triedy ako [názov testovacej triedy]Test.
4.	V prípade potreby inicializácie premenných alebo načítania testovacej databázy sa vyberie možnosť SetUpBeforeClass(). Ak sa v testoch bude robiť s databázou, je potrebné vyčistiť tabuľky databázy. V tomto prípade sa vyberie možnosť TearDownAfterClass().
5.	Ak bude nutné pred každou testovacou funkciou inicializovať premenné na rovnakú hodnotu alebo inicializovať niečo iné pred každou funkciou, vyberie sa možnosť setUp(). Ak bude každá testovacia funkcia meniť určité inicializované nastavenia, vyberie sa funkcia tearDown(), aby po skončení testovacej metódy, sa inicializované premenné uviedli do pôvodného stavu.
6.	Vyberieme tlačítko Finish.
7.	Na začiatku triedy sa vytvorí komentár, ktorý obsahuje meno autora.
8.	Ak bola vytvorená aspoň jedna z funkcií SetUpBeforeClass(), TearDownAfterClass(), setUp() alebo tearDown(), je nutné doimplementovať ich funkčnosť.



Obrázok 8.2: Štruktúra a umiestnenie testovacích tried voči triedam, ktoré sú testované



Obrázok 8.3: Tvorba novej testovacej triedy

### Vytvorenie testovacej metódy

Pred vytvorením samotnej funkcionality systému je potrebné spraviť najprv jednotkové testy. Tak ako vývoj riadený testami vytvára najprv testy funkcionality, tak aj pre jednotlivé metódy je potrebné najprv vytvoriť metódy, ktoré otestujú ich funkčnosť.

Vstup	Vytvorená a otvorená testovacia trieda na editovanie v Eclipse. Nová metóda funkcionality ešte neexistuje.
Výstup	Vytvorená testovacia metóda.
Zodpovedný	Vývojár

1.	Zadeklaruje sa nová metóda v triede, pre ktorú sa idú vytvárať testy.
2.	V testovacej triede sa vytvorí nová testovacia metóda s názvom test[metóda na otestovanie], kde [metóda na otestovanie] začína tento krát veľkým písmenom. V prípade potreby viacerých testovacích metód na pokrytie jednej metódy sa názov metódy doplní slovami, ktoré opisujú oblasť, ktorej sa test týka. Pred metódou sa musí nachádzať reťazec „@Test“.
3.	V prípade testovania výnimiek, sa musí pridať @Test(expected=[výnimka].class), kde [výnimka] je chyba, ktorá sa očakáva pri spustení metódy, že ju vráti.
4.	Ak bude treba testovanie časovo obmedziť, treba namiesto @Test pridať @Test(timeout=[počet milisekúnd]), kde [počet milisekúnd] je počet milisekúnd, koľko môže test najdlhšie trvať.
5.	Ak je potrebné na testovanie súbor, tento súbor sa vloží do adresára testFiles.
6.	Do novej testovacej metódy sa doplní kód, ktorý pokryje danú testovanú oblasť.
7.	Následne pravým tlačítkom sa vyberie vytváraná testovacia trieda a zvolí sa možnosť Run as > JUnit Test.
8.	V prípade, že vytváraná testovacia metóda skončila úspešne je nutné vrátiť sa k bodu 6.
9.	Nakoniec sa pred novú testovaciu metódu doplní komentár s obsahom, kto ju vytvoril a čo v nej testuje.

#### 8.4. Použiteľné funkcie na overenie testovania

Pri vytváraní metód pre testovanie sa používajú metódy triedy *Assert* a *Assume*. Pre splnenie testov musia byť splnené len metódy triedy *Assert*. V prípade metód triedy *Assume* sa test zastaví a bude ignorovaný.

Najdôležitejšie metódy triedy *Assert* a ich použitie je nasledovné (Tabuľka č. 8.2):



Metóda	Opis
assertArrayEquals	Porovná, či obsah dvoch polí bol rovnaký. Ak je test nepravdivý, jednotkový test skončí neúspechom.
assertEquals	Porovná, či sa hodnoty dvoch objektov, reťazcov alebo čísel rovnajú. Ak nie, jednotkový test skončí neúspechom.
assertFalse	Skontroluje, či je boolean výraz nepravdivý. Ak je výraz pravdivý, jednotkový test skončí neúspechom.
assertNotNull	Skontroluje, či objekt nemá hodnotu null. Ak jeho hodnota je null, tak test skončí neúspechom.
assertNotSame	Skontroluje, či referencie neukazujú na ten istý objekt. Ak sa jedná o ten istý objekt, test skončí neúspechom.
assertNull	Skontroluje, či objekt má hodnotu null. Ak jeho hodnota nie je null, tak test skončí neúspechom.
assertSame	Overí, či referencie odkazujú na ten istý objekt. V prípade, že nie, jednotkový test skončí neúspechom.
assertThat	Skontroluje, či argument vyhovuje podmienke. Ak nie, tak test skončí neúspechom.
assertTrue	Skontroluje, či je boolean výraz pravdivý. Ak výraz nie je pravdivý, jednotkový test skončí neúspechom.
fail	Test v každom prípade skončí neúspechom.

Tabuľka 8.2: Dôležité metódy triedy Assert

Metódy triedy Assume, ktoré dopĺňajú možnosti testovania (Tabuľka č. 8.3):

Metóda	Opis
assumeNoException	Používa sa v prípade, keď chceme, aby sa test dokončil do konca. Ak sa vyskytne výnimka, test sa zastaví a bude ignorovaný.
assumeNotNull	Skontroluje, či objekt nemá hodnotu null. Ak jeho hodnota je null, tak sa test zastaví a bude ignorovaný.
assumeThat	Skontroluje, či argument vyhovuje podmienke. Ak nie, tak sa test zastaví a bude ignorovaný.
assumeTrue	Skontroluje, či je boolean výraz pravdivý. Ak výraz nie je pravdivý, jednotkový test sa zastaví a bude ignorovaný.

Tabuľka 8.3: Dôležité metódy triedy Assume

## 8.5. Použitie JMock testovania

Autor: Igor Slotík

V niektorých prípadoch je nutné na otestovanie funkcionality metódy alebo triedy použiť inštancie iných tried či rozhraní. Problém môže nastať v prípade, keď tieto triedy alebo rozhrania ešte nie sú dokončené. V takomto prípade je pohodlné využiť výhody JMock testovania. Pri JMock testovaní sa nepoužijú inštancie nedokončených tried, ale mock objekty.

Trieda Mockery umožňuje využiť JMock testovanie. Spôsoby vytvárania inštancie triedy Mockery sa líšia v prípade vytvárania mock objektov ako náhrady za inštancie tried a rozhraní.

Vytvorenie inštancie triedy Mockery za účelom vytvárania inštancií tried:

```
private Mockery context = new Mockery() {
    setImposterizer(ClassImposterizer.INSTANCE);
};
```

Vytvorenie inštancie triedy Mockery za účelom vytvárania inštancií rozhraní:

```
private Mockery context = new Mockery() {
};
```

Ďalej uvediem metódy, ktoré poskytuje trieda Mockery.

Nevyhnutne potrebné metódy triedy Mockery sú v tabuľke č. 8.4.

Metóda	Opis
setImposterizer	Umožňuje vytvárať mock objekty ako inštancie tried.
mock	Vytvára mock objekt.
checking	Špecifikuje očakávané správanie pre mock objekty.
assertIsSatisfied	Testy spadnú ak nebolo dodržané čo len jedno očakávané správanie.

Tabuľka 8.4: Dôležité metódy triedy Mockery.

Doplňujúce metódy triedy Mockery sú v tabuľke č. 8.5.

Metóda	Opis
sequence	Vracia novú sekvenciu, ktorá obmedzuje poradie vykonávania očakávaných správanií mock objektov.
setDefaultResultForType	Nastavuje očakávanú návratovú hodnotu pre konkrétny typ, ak očakávané správanie nebolo nastavené v metóde checking.
setExceptionHandlerTranslator	Prekladá chyby do lepšie zrozumiteľnej formy.
setNamingScheme	Nastavuje schému mien takým spôsobom, aby sa generovali také mená pre mock objekty, ktoré nie sú explicitne uvedené v teste.
states	Vracia nové stavy, ktoré obmedzujú poradie vykonávania očakávaných správanií mock objektov.

Tabuľka 8.5: Doplnujúce metódy triedy Mockery.

## 8.6. Zmena tvorby JUnit testu

Autor: Martin Detko a Matej Mihálik

### Zmena pri tvorbe testovacej triedy

Test case-y (testovacie classy) musia rozširovať TestCase abstraktnú triedu. Tá sa postará o to, aby sa pred spustením testov nainicializovala testovacia databáza a následne po skončení testu sa aj vymazala. Funkcie setUpBeforeClass a tearDownAfterClass teda už nemôžu vykonávať inicializáciu databázy. Funkcia setUpBeforeClass sa vykoná pred inicializáciou databázy a funkcia tearDownAfterClass sa vykoná po vymazaní databázy.

Dáta sa musia nachádzať v adresári „recommender/lib/jrecommender/csv“ a csv súbor sa musí volať rovnako ako testovacia trieda. Samotný formát CSV súborov je nasledovný:

```
@TABLE;documents
```

```
id;system_id;identification
```

```
1;'1';'1_1'
```

```
2;'1';'1_2'
```

```
3;'1';'1_3'
```

```
4;'1';'1_4'
```

```
21;'2';'2_1'
```

```
25;'2';'2_5'
```

```
28;'2';'2_8'
```

@TABLE je identifikátor tabuľky. Ďalší riadok obsahuje vymenované stĺpce tabuľky, do ktorých sa bude ukladať. Ostatné riadky potom obsahujú samotné dáta, ktoré sa budú ukladať, jeden riadok zodpovedá jednému riadku tabuľky. Na oddelovanie jednotlivých hodnôt sa používa bodkočiarka. Pre uloženie stringu je nutné ho vymedziť jednoduchými apostrofmi.

### Zmena pri tvorbe testovacej metódy

Všetka testovacia logika sa musí nachádzať vo funkcií s názvom test. V prípade veľkého množstva testovacej logiky, je možné spraviť V podstate veľkého množstva testovacej funkcionality, je možné ju rozdeliť do funkcií ale tieto funkcie musia byť volané metódou test.

Nakoľko HibernateUtil má už protected metódy, všetky custom sql query, ktoré budú volané nad databázou v testoch, sa musia nachádzať v triede TestHelper. Tato trieda má public access a pracuje výhradne s testovou databázou.

## 9. Manažment rizík

**Autor:** Ľudovít Mydla

### 9.1. Roly a zodpovednosti

Roly	Zodpovednosti
Vlastník produktu	<ul style="list-style-type: none"> <li>hlási chybu</li> <li>rozhoduje o prioritě spracovania chýb</li> <li>poskytuje spätnú väzbu</li> </ul>
Scrum master	<ul style="list-style-type: none"> <li>rozhodovanie o relevantnosti opravy chyby podľa špecifikácie</li> <li>kontroluje duplicitu chyby</li> <li>stanovuje časový odhad spracovania chyby</li> </ul>
Vývojár	<ul style="list-style-type: none"> <li>opravuje chybu</li> <li>vyjadruje sa ku časovému odhadu spracovania chyby</li> </ul>
Tester	<ul style="list-style-type: none"> <li>vykonáva testy</li> </ul>
Manažér testovania	<ul style="list-style-type: none"> <li>dohliada nad procesom spracovania chyby</li> <li>dohliada nad testovaním</li> <li>uzatvára chyby ako vyriešené</li> </ul>

Tabuľka 9.1: Roly a zodpovednosti pri testovaní

### 9.2. Spracovanie chyby

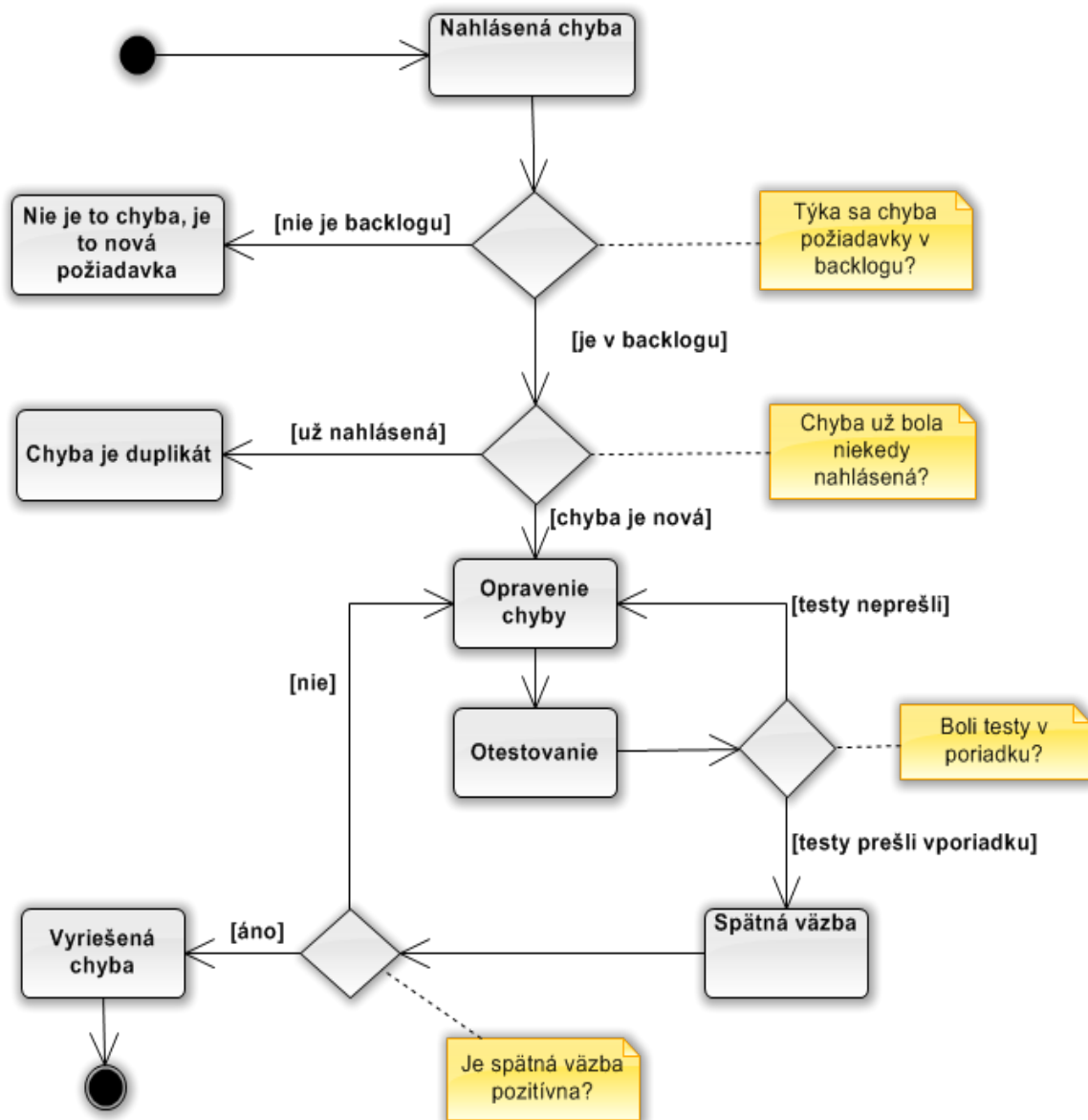
Manažment rizík v tomto dokumente sa sústreďí na manažment chýb. Konkrétne na spracovanie chyby po spustení projektu do prevádzky. Keď zákazník – vlastník produktu – zistí, že niečo nesúhlasí s jeho aktuálnymi požiadavkami.

Vstup	Produkt, ktorý nespĺňa aktuálne požiadavky zákazníka
Výstup	Produkt, ktorý splňa požiadavky, alebo nová požiadavka
Zodpovedný	Manažér testovania

1.	Vytvorenie nového záznamu o chybe.
2.	Skontrolovanie backlogu. Ak sa požiadavka nachádza v backlogu a nie je splnená pokračuje sa krokom 3. Inak je to nová požiadavka, ktorú rieši predchádzajúca kapitola.
3.	Skontroluje sa zoznam nahlásených chýb. Ak chyba ešte nebol hlásená, pokračuje sa krokom 4. Inak je to duplicitná chyba.
4.	Určí sa čas trvania a vývojár(i) pre vykonanie opravy.

5.	Chyba sa testuje. Ak testy boli úspešné, pokračuje sa krokom 6, inak sa opakuje krok 4.
6.	Očakáva sa spätná väzba vlastníka produktu. Ak oprava zodpovedá jeho požiadavkám, pokračuje sa krokom 7, inak sa preskočí naspäť na krok 4.
7.	Chyba je vyriešená a uzavrie sa.

Vizuálne zachytáva tento postup diagram aktivít na obrázku 9.1.



### 9.3. Vytvorenie novej chyby

Proces spracovania chyby začína nahlásením chyby vlastníkom produktu. Chybu hlási podľa

svojich aktuálnych požiadaviek. To znamená, že to, čo hlási ako chybu môže byť požiadavka na novú vlastnosť, ktorá pred tým nebola spomenutá.

Vstup	Nezrovnalosť medzi aktuálnymi požiadavkami vlastníka produktu a aktuálnym stavom produktu.
Výstup	Vytvorený záznam v systéme Redmine.
Zodpovedný	Vlastník produktu

Hlásenie chyby prebieha v nástroji Redmine kliknutím na položku „New Issue“ v menu, ako na obrázku 9.2 Zobrazí sa formulár na obrázku 9.3.

1.	V poli „Tracker“ sa vyberie možnosť „Bug“
2.	Do poľa „Subject“ sa zadá výstižný text opisujúci chybu
3.	Do poľa „Description“ sa môže napísať text bližšie opisujúci chybu. V tomto prípade je potrebné zaznamenať podmienky, kedy sa chyba vyskytla.
4.	Ako „Status“ sa musí uviesť „New“
5.	Ako „Priority“ sa uvedie priorita potreby opravy chyby podľa vlastníka produktu.
6.	Do poľa „Due date“ sa určí dátum, do ktorého by mala byť chyba opravená.
7.	Ostatné políčka zostanú prázdne a záznam o chybe sa vytvorí stlačením tlačidla „Create“

Home My page Projects Administration Help

Personalizovane odporucanie

Overview Activity Backlog Dashboard Roadmap Issues **New issue** Charts Gantt Calenda

## Overview

Obrázok 9.2: Hlavné menu nástroja Redmine so zvýraznenou položkou na vytvorenie záznamu o

**New issue**

Tracker \* Bug  Private

Subject \*

Description **B I U S C H1 H2 H3** **pre** **Text formatting**

Status \* New

Priority \* Normal

Assignee

Category

Target version

Parent task

Start date 2011-12-12

Due date

Estimated time Hours

% Done 0 %

Files **Choose File** No file chosen **Optional description**

Add another file (Maximum size: 5 MB)

Watchers  Igor Slotik  Jakub Sevcach  Ludovit Mydla  
 Martin Detko  Matej Mihalik  Matej Červeňák  
 Michal Cadrik  Michal Kompan

Create Create and continue Preview

Obrázok 9.3: Formulár na vytvorenie záznamu o chybe

## 9.4. Priradenie chyby

Keď sa vytvoril záznam o chybe, nasleduje rozhodovanie, či je vôbec potrebné sa ňou zaoberať. Niekedy totiž to, čo vlastník produktu označí za chybu, je nová požiadavka a tú rieši metodika v predchádzajúcej kapitole. Závisí to od toho, či daná vlastnosť alebo funkcionálna bola dohodnutá v backlogu. V tom prípade to je neimplementovaná alebo nesprávne implementovaná požiadavka a teda chyba a proces jej spracovania pokračuje. Rozhodnutie vykonáva Scrum master podľa znalosti produktu a backlogu.

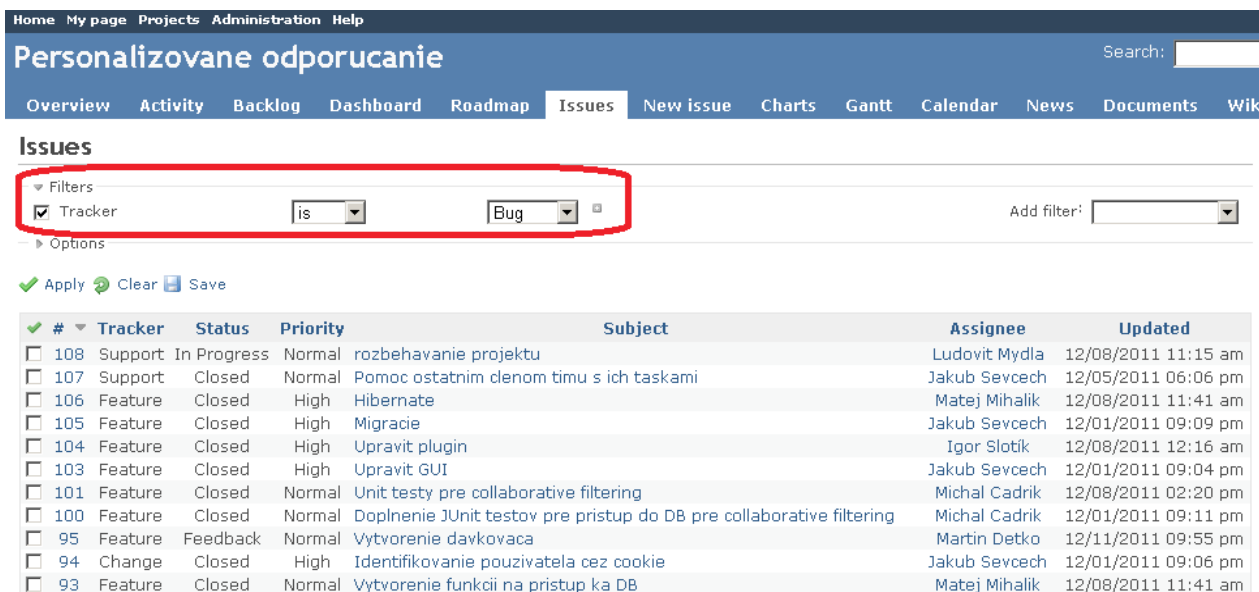
Vstup	Nahlásená chyba
Výstup	Rozhodnutie, či to je chyba alebo nová požiadavka
Zodpovedný	Scrum master

Niekedy sa stáva, že tá istá chyba je hlásená viacerými ľuďmi nezávisle od seba. Duplicitné požiadavky je preto potrebné eliminovať. Po prezretí si zoznamu úloh je Scrum master schopný rozlíšiť, či chyba už nebola hlásená a nepracuje sa na nej. Ak nie, tak sa určí vývojár(i), ktorý má opravu vykonať. Po konzultácii časového odhadu trvania opravy s vývojárom určí Scrum master aj čas trvania.

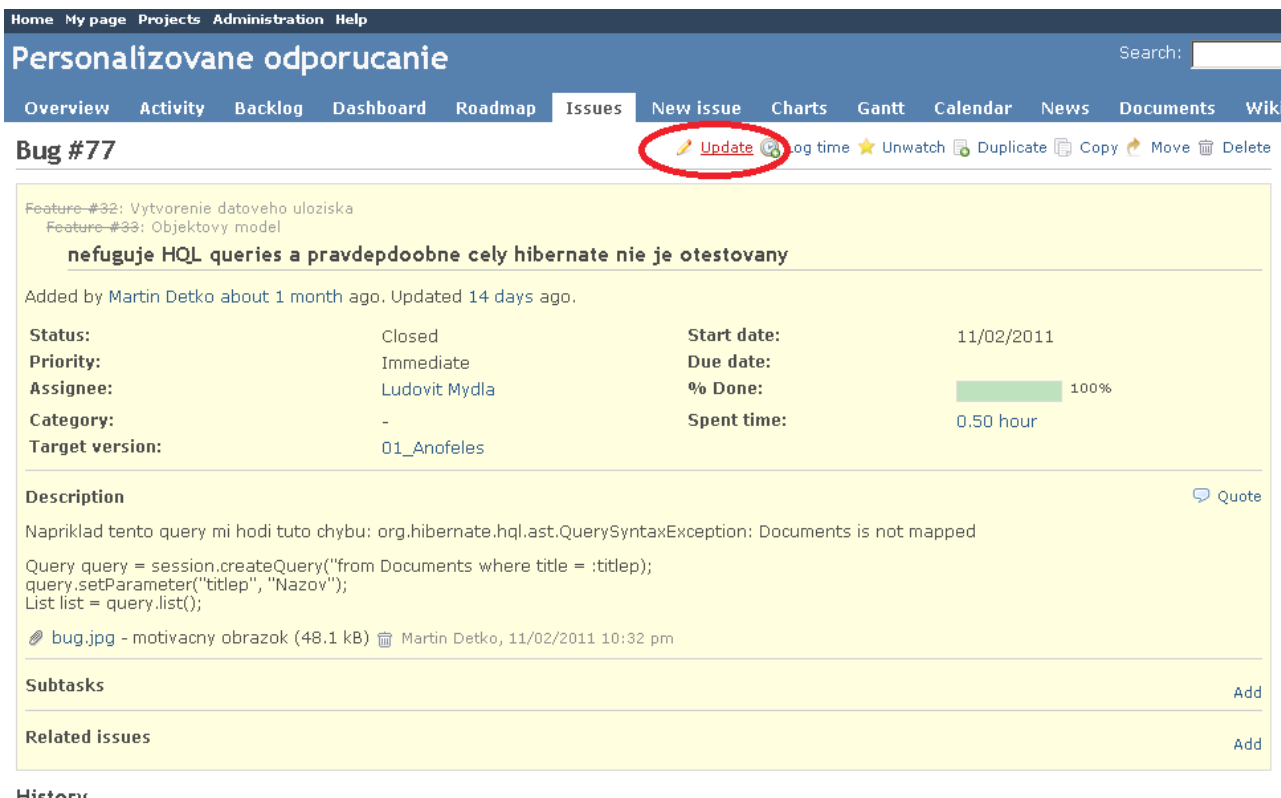
Vstup	Nahlásená chyba
Výstup	Zamietnutá duplicitná chyba alebo chyba priradená vývojárovi s určeným časom na jej opravu
Zodpovedný	Scrum master

V zozname chýb, po kliknutí na „Issues“ v hlavnom menu nástroja Redmine sa nám zobrazia chyby (filtrovanie je nastavené na „Tracker is bug“) ako na obrázku 9.4 Chyba sa zamietá nasledovným spôsobom.

1.	Vyberie sa chyba na zaradenie kliknutím na názov chyby.
2.	Vyberie sa možnosť editácie kliknutím na „Update“ ako na obrázku 9.5.
3.	V poli „Status“ sa vyberie možnosť „Rejected“.
4.	Vyberie sa v poli „Assignee“, osoba Scrum mastera..
5.	Záznam sa uloží stlačením tlačidla „Submit“.



Obrázok 9.4: Zoznam úloh s ukážkou nastavenia filtra



Obrázok 9.5: Ukážka editovania záznamu o chybe

Podobne ako v prípade zamietnutia, aj v prípade priradenia sa vykonávajú prvé dva kroky výberu chyby zo zoznamu. V budúcich postupnostiach krokov to už implicitne predpokladáme.

1. Vyberie sa v poli „Assignee“ osoba, ktorá má opravu chyby na starosti.



2.	Do poľa „ <i>Start date</i> “ sa vloží aktuálny dátum.
3.	Do poľa „ <i>Due date</i> “ sa vloží dátum, do ktorého chyba musí byť opravená
4.	Do poľa „ <i>Estimated</i> “ time sa vloží odhadovaný čas na opravu v hodinách.
5.	Záznam sa uloží stlačením tlačidla „ <i>Submit</i> “.

### 9.5. Oprava chyby

Po priradení času a vývojára na úlohu opravy chyby sa po každom odpracovanom čase upraví záznam o chybe. Na tejto aktivite sa podieľa aj tester, ktorý po oprave chyby vykoná testy, ktoré zaručujú bezchybnosť opraveného kódu. Písanie a vykonávanie testov je opísané v predchádzajúcej kapitole.

Vstup	Zaradená chyba
Výstup	Opravená chyba ponechaná na spätnú väzbu
Zodpovedný	Manažér testovania

Do nástroja Redmine sa činnosť na oprave (oprava aj testovanie) vykáže nasledovne.

1.	V poli „ <i>% Done</i> “ sa vyberie percento úlohy, ktoré doteraz bolo splnené.
2.	V poli „ <i>Status</i> “ sa vyberie možnosť „ <i>In Progress</i> “, ak ešte nebola práca na oprave ukončená. V poli „ <i>Status</i> “ sa vyberie možnosť „ <i>Resolved</i> “, ak sa opravy vykonali a čakajú na testovanie. V poli „ <i>Status</i> “ sa vyberie možnosť „ <i>Feedback</i> “, ak sa testy úspešne vykonali.
3.	Do poľa „ <i>Spent time</i> “ sa vloží počet odpracovaných hodín od posledného zápisu.
4.	V poli „ <i>Activity</i> “ sa vyberie možnosť „ <i>Development</i> “.
5.	V poli „ <i>Notes</i> “ sa môžu uviesť ľubovoľné poznámky.
6.	Pre aktualizáciu údajov sa stlačí tlačidlo „ <i>Submit</i> “.

Tento proces sa opakuje pokiaľ nebudú testy úspešne spustené. Iba vtedy je povolené touto metodikou uviesť v poli „*% Done*“ hodnotu 100% a nastaviť „*Status*“ na „*Feedback*“.

### 9.6. Vyriešenie chyby

Po vyjadrení spätnej väzby od vlastníka produktu sa oprava opäť obnoví alebo sa chyba uzavrie.

Vstup	Chyba čakajúca na spätnú väzbu
-------	--------------------------------

Výstup	Uzavretá chyba alebo chyba vrátená na ďalšiu opravu
Zodpovedný	Manažér testovania

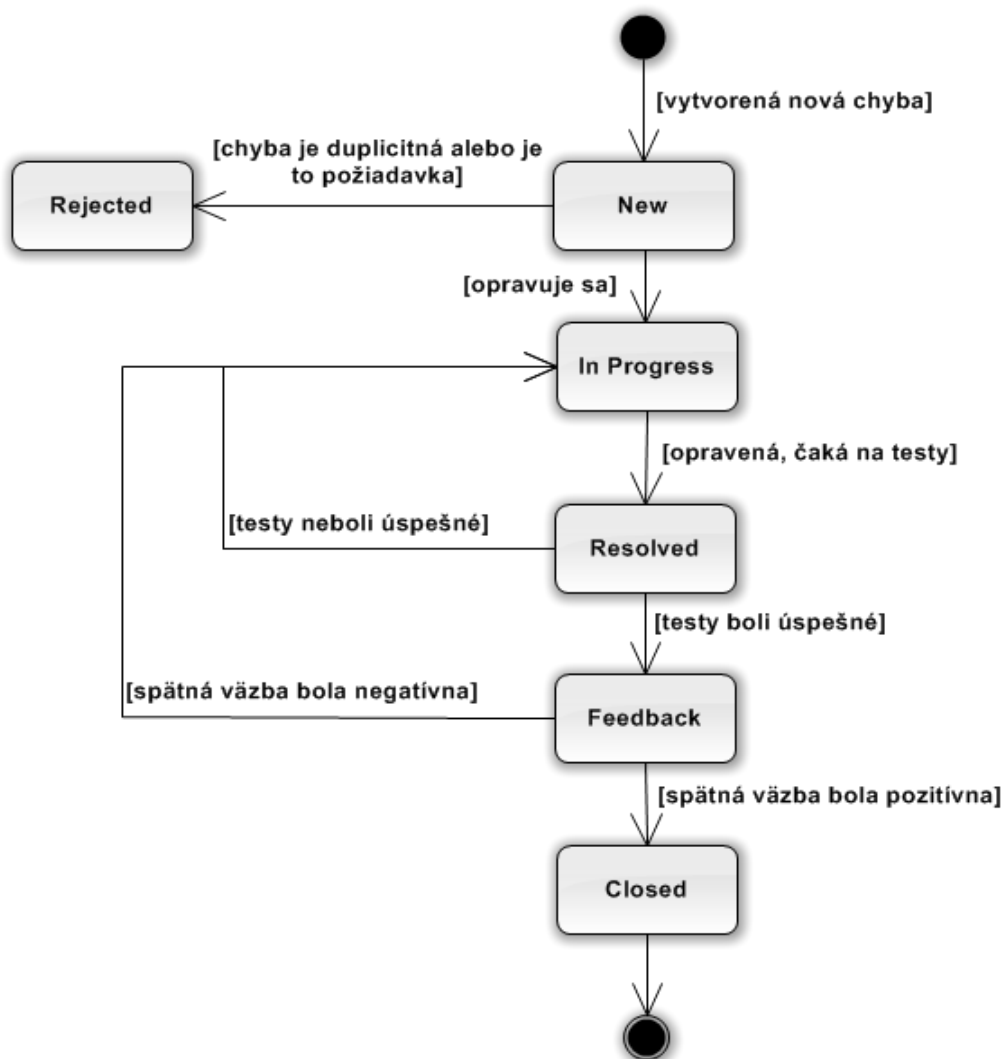
V prípade pozitívnej spätnej väzby sa vykoná:

1.	V poli „Status“ sa vyberie možnosť „Closed“
2.	Pre aktualizáciu údajov sa stlačí tlačidlo „Submit“

Ak je väzba negatívna, teda že vlastník produktu nie je spokojný s opravou, vykoná sa:

1.	V poli „Status“ sa vyberie možnosť „In Progress“
2.	Pre aktualizáciu údajov sa stlačí tlačidlo „Submit“

Celkový stavový diagram správy o chybe je na obrázku 9.6.



### 9.7. Manažment rizík v druhom semestri

Manažment rizík v druhom semestri spočíval najmä v triezvom odhade času na splnenie daných úloh počas stretnutia. Niektoré časové odhady sa zdali príliš krátke na splnenie danej úlohy, preto boli preferované dlhšie časové odhady. V niektorých prípadoch sa aj redukovalo množstvo úloh pre daný šprint alebo sa rozdelili medzi členov s menším množstvom úloh.

Metodika sa od minulého semestra nezmenila. Jej používaním sa však vytvorili niektoré „skratky“. Konkrétne - veľmi sa nepoužíva stav *Feedback*. Keďže vlastník produktu, nemá vždy čas na kontrolu otestovaného kódu, preskakuje sa tento stav a priamo, podľa posúdenia vývojára sa úloha označuje za uzavretú - „*Closed*“ alebo zostáva ako vyriešená - „*Resolved*“. Dá sa to opísať tak, že čakanie na spätnú väzbu je nulové. Inak je metodika rovnaká.

## 10. Manažment rozvrhu

### 10.1. Tvorba rozvrhu

Nakoľko sa pri tvorbe produktu riadime metodikou SCRUM, nemáme dopredu určený rozvrh projektu. Ten sa tvorí vždy pre konkrétny šprint. Táto tvorba prebieha vždy na prvom tímovom stretnutí v rámci šprintu.

### 10.2. Postup tvorby rozvrhu

1. Zhodnotí sa predchádzajúci (končiaci) šprint. Všetky úlohy, ktoré sa členom tímu nepodarilo z tohto šprintu dokončiť, sa im automaticky prenášajú do nasledujúceho šprintu.
2. Vlastník produktu predloží používateľské príbehy (funkcionality systému), ktoré chce zapracovať do projektu.
3. Tím následne ohodnotí časovú náročnosť všetkých používateľských príbehov pomocou „Poker cards“ metódy. Táto zložitost' je vyjadrená v človeko – hodinách.
4. Pre každý používateľský príbeh sa určí jedna zodpovedná osoba. V prípade, že sa v tíme nenájde dobrovoľník, pridelí túto rolu vedúci tímu.
5. Tím následne identifikuje podúlohy jednotlivých používateľských príbehov. Podúlohy musia byť identifikované tak, aby bolo možné ich rozsah jednoznačne definovať a aby mohla byť podúloha pridelená na vypracovanie jednému členovi tímu.
6. Každý podúlohe následne tím opäť definuje náročnosť v človeko – hodinách.
7. Náročnosť všetkých používateľských príbehov sa potom prehodnotí na základne hodnotenia ich podúloh.
8. Vlastník produktu sa na základe tohto upraveného hodnotenia náročnosti a vlastných priorít dôležitosti pre jednotlivé používateľské príbehy rozhodne, ktoré chce nechať v nasledujúcom šprinte vypracovať. Odporúčaný rozsah pre jeden šprint je 50 – 70 človeko – hodín.
9. Zodpovedné osoby za jednotlivé používateľské príbehy následne pridelia podúlohy jednotlivým členom tímu. Pri tomto rozdelení sa dbá na to aby všetci členovia tímu mali celkovo približne rovnaké zaťaženie v človeko – hodinách. Odporúčaný počet človeko – hodín na jedného člen tímu je 8 – 16 hodín na šprint.
10. V prípade nespokojnosti niektorých členov tímu so zadelením úloh sa tím snaží nájsť kompromisné riešenie tak, aby boli spokojní všetci členovia tímu. Posledné slovo má však vždy vedúci tímu.
11. Keď sú všetci členovia tímu s rozvrhom spokojný, tvorba rozvrhu pre daný šprint sa skončila.

### 10.3. Ukončenie šprintu – aktualizovanie stavu úloh v systéme Redmine

Každý z členov tímu je povinný pred stretnutím k projektu aktualizovať stav úloh, ktoré mu boli pridelené na vypracovanie v systéme Redmine, aby bolo možné na stretnutí vyhodnotiť prípadnú potrebu zmeny celkového plánu projektu. Stav, ktorý môžu úlohy v štádiu ich spracovania nadobudnúť sú:

- *Riešená* – úloha sa začala vypracovávať, ale ešte nie je ukončená.
- *Zatvorená* – úloha sa úspešne vypracovala
- *Odmietnutá* – úloha sa v priebehu práce stala neaktuálnou (nie je potrebná, požiadavka na ňu bola zrušená alebo ju momentálne nie je možné vypracovať)

Pri aktualizácií stavu úlohy v systéme Redmine sa používa nasledujúci postup:

1. otvorenie systému na stránke <http://vm28.ucebne.fiit.stuba.sk/Redmine>
2. prihlásenie sa do systému – pravý horný roh → *Sign In*
3. v hornom menu kliknúť na záložku *Issues*
4. v zozname vyhľadať želanú úlohu a rozkliknúť ju
5. v menu pod sumárom úlohy kliknúť na možnosť *Update*.
6. Vyplniť formulár podľa pokynov v nasledujúcich tabuľkách 4, 5, 6 (podľa stavu úlohy)
7. Potvrdiť kliknutím na *Submit*

#### 10.4. Riešenie úlohy

Pole	Hodnota
Status	Možnosť „ <i>In Progress</i> “
Spent Time	Čas strávený prácou na úlohe od jej poslednej aktualizácie v systéme Redmine
Activity	Možnosť „ <i>Development</i> “

Tabuľka 10.1: Formulár pre riešenú úlohu

#### 10.5. Zatvorenie úlohy

Pole	Hodnota
Status	Možnosť „ <i>Closed</i> “
Spent Time	Čas strávený prácou na úlohe od jej poslednej aktualizácie v systéme Redmine
Activity	Možnosť „ <i>Development</i> “

Tabuľka 10.2: Formulár pre zatvorenú úlohu

**10.6. Odmietnutie úlohy**

Pole	Hodnota
Status	Možnosť „ <i>Rejected</i> “
Spent Time	Čas strávený prácou na úlohe od jej poslednej aktualizácie v systéme Redmine
Activity	Možnosť „ <i>Development</i> “
Comment	Stručne (jednou vetou) zhrnutý dôvod zamietnutia úlohy
Notes	Podrobný opis toho, prečo bola úloha zamietnutá

Tabuľka 10.3: Formulár pre odmietnutú úlohu

## 11. Manažment podpory vývoja

**Autor:** Igor Slotík

### 11.1. Systém na správu verzií GIT

#### Role a zodpovednosti účastníkov

V tabuľke č. 11.1 sú identifikované role, vystupujúce v manažmente verzií, spolu s prislúchajúcimi zodpovednosťami.

Rola	Zodpovednosť
Vedúci tímu	<ul style="list-style-type: none"> <li>• Celkový dohľad a kontrola</li> </ul>
Vývojár	<ul style="list-style-type: none"> <li>• Vývoj novej funkcionality</li> <li>• Príprava vytvorenia novej verzie</li> <li>• Oprava vážnych chýb v momentálnej verzií</li> </ul>
Manažér podpory vývoja	<ul style="list-style-type: none"> <li>• Inštalácia systému kontroly verzií</li> <li>• Konfigurácia centrálného úložiska</li> <li>• Dohľad nad centrálnym úložiskom</li> </ul>
Manažment kvality	<ul style="list-style-type: none"> <li>• Testovanie</li> <li>• Hodnotenie výsledkov testovania</li> <li>• Rozhodnutie o nasadení do produkčnej verzie</li> </ul>
Majiteľ produktu	<ul style="list-style-type: none"> <li>• Zadávanie požiadaviek na produkt</li> </ul>
Manažér komunikácie	<ul style="list-style-type: none"> <li>• Zbieranie požiadaviek od používateľa</li> <li>• Komunikácia vývoja produktu s externými pozorovateľmi</li> </ul>

Tabuľka 11.1: Role a zodpovednosti

#### Pravidlá pre prácu s vetvami

Manažment verzií zahŕňa pravidlá pre prácu s týmito vetvami:

- master (kap. 4.1)
- devel (kap. 4.2)

Manažment verzií ďalej zahŕňa pravidlá pre prácu s týmito triedami vetiev:

- feature (kap. 4.3)

- bug (kap. 4.4)

Pravidlá pre prácu s vetvou *master*

<i>Pôvodná vetva:</i>	žiadna
<i>Môže sa „zatlačiť“ do (git push ? master) :</i>	žiadna
<i>Môže sa do nej „zatlačiť“ (git push master ?):</i>	vetva <i>devel</i>
<i>Zodpovedný:</i>	Manažér podpory vývoja

Je v ňom momentálna produkčná verzia. Každá verzia má vlastné číslo. Žiadny vývojár nemôže svojvoľne ukladať zmeny do vetvy *master*. Manažér podpory vývoja má zodpovednosť nad centrálnym úložiskom, a teda aj nad vetvou *master*.

### Pravidlá pre prácu s vetvou *devel*

<i>Pôvodná vetva:</i>	vetva <i>master</i>
<i>Môže sa „zatlačiť“ do (git push ? devel) :</i>	vetva <i>master</i>
<i>Môže sa do nej „zatlačiť“ (git push devel ?):</i>	vetvy tried <i>feature</i> a <i>bug</i>
<i>Zodpovedný:</i> vedúci tímu	Manažér podpory vývoja, Manažér kvality,

Je v ňom aktuálny bod vývoja novej verzie. Keď vývojár potrebuje aktualizovať svoje lokálne úložisko, stiahne si momentálnu verziu z vetvy *devel*. Keď zdrojový kód vo vetve *devel* dosiahne stabilný bod, teda všetky zmeny pre momentálnu verziu sú vyriešené a otestované, vetva *devel* je pripravená na spojenie s vetvou *master*. Vetva *master* sa spojí s vetvou *devel* len vtedy, keď je vo vetve *devel* vytvorená a otestovaná nová produkčná verzia. Zhodnotenie výsledkov testovania a rozhodnutie, či vetva *devel* je pripravená na spojenie s vetvou *master*, má na zodpovednosť manažér kvality a vedúci tímu. Manažér podpory vývoja má zodpovednosť nad centrálnym úložiskom, a teda robí celkový dohľad nad vetvou *devel*.

### Pravidlá pre prácu s triedou vetiev *feature*

<i>Pôvodná vetva:</i>	vetva <i>devel</i>
<i>Môže sa „zatlačiť“ do (git push ? feature):</i>	vetva <i>devel</i>
<i>Môže sa do nej „zatlačiť“ (git push feature ?):</i>	žiadna
<i>Akceptovateľné pomenovania:</i> Feature #12	Feature #[číslo vetvy <i>feature</i> ], napr.:
<i>Zodpovedný:</i>	vývojár

Vetva triedy *feature* (skrátene vetva *feature*) je vetva vytváraná na každom počítači vývojára. Preto sa nazýva aj lokálnou vetvou. Používajú sa na vyvíjanie funkcionality pre nasledujúcu alebo dokonca ešte v budúcnosti vzdialené softvérové verzie. Z tohto dôvodu na začiatku vývoja novej funkcionality nemusí byť nutne známa verzia, pre ktorú sa funkcionality vyvíja. V prípade, že vo vetve *feature* sa vyvíja funkcionality nevyhnutne potrebná pre ďalšiu verziu, je nevyhnutné, aby sa vetva *feature* spojila s vetvou *devel* ešte pred spojením vetiev *devel* a *master*. Pokiaľ vyvíjaná funkcionality je určená pre vzdialené produkčné verzie alebo nie je potrebná pre momentálnu produkčnú verziu, nemusí sa vetva *feature* spojiť s vetvou *devel* pred spojením vetiev *devel* a *master*.



Po dokončení zmien sa zmeny pošlú do lokálneho úložiska s odpovedajúcou správou. Príkaz *commit* vytvára nový *commit* objekt s novým identifikačným číslom. Tým sa udržuje informácia o existencii vetvy *feature* a dajú sa ľahko nájsť všetky *commit* objekty, v ktorým je nová funkcionálna integrovaná.

Pri úspešnom otestovaní sa spoja vetvy *feature* a *devel*. Za správne zhodnotenie výsledkov testovania a za spojenie s vetvou *devel* je zodpovedný vývojár, ktorý novú funkcionálnu vyvíjal. Po spojení s vetvou *devel* sa vetvu *feature* neodporúča mazať, pretože môže byť užitočná pri vývoji ďalšej funkcionality. Vetvu *feature* sa odporúča mazať len pri neúspešnom pokuse o novú funkcionálnu.

### Pravidlá pre prácu s triedou vetiev bug

<i>Pôvod vetvy:</i>	vetvy <i>master</i> a <i>devel</i>
<i>Môže sa „zatlačiť“ do (git push ? bug) :</i>	vetvy <i>master</i> a <i>devel</i>
<i>Môže sa do nej „zatlačiť“ (git push bug ?):</i>	žiadna
<i>Možné pomenovania:</i>	Bug #[číslo vetvy bug], napr.: Bug #12
<i>Zodpovedný:</i>	vývojár

Je vytváraná z okamžitej nutnosti opraviť neželaný stav momentálnej produkčnej verzie vo vetve *master* alebo zmien pre nasledujúcu produkčnú verziu vo vetve *devel*. Pokiaľ je vážna chyba vo vetve *master*, tak sa určí nové číslo verzie. Po vyriešení chyby sa riešenie uloží s odpovedajúcou správou. Vetva *bug* sa spojí s vetvami *master* a s *devel*, aby sa zaručilo, že problém bude vyriešený aj pre nasledujúcu verziu produktu.

Ak sa opravuje vážna chyba vo vetve *devel*, nie je nutné spojenie vetiev *master* a *devel*, pretože sa predpokladá, že vo vetve *master* chyba nenastala.

### Príkaz commit

Používa sa na poslanie vykonaných zmien so správou popisujúcou zmeny do lokálneho úložiska. Bez neho by nebolo možné vykonané zmeny uložiť a poslať do centrálného úložiska. Príkaz *commit* neovplyvňuje úložiská iných vývojárov, v tom zmysle, že do nich nič nevkladá. Pokiaľ by sa hneď po príkaze *commit* našli chyby, je možné ich hneď vrátiť späť pomocou príkazu *git reset*. Taktiež je možné mazať aj existujúce *commit* objekty.

### Parametre pre príkaz commit

Príkaz *commit* sa môže vykonať s rôznymi parametrami:

Parametre pri git commit	Použitie	Komentár
-a (all)	Potvrdenie všetkých zmien v úložisku.	Nutné.
-C [commit objekt]	Použije sa existujúci <i>commit</i> objekt a použije sa z neho správa a autorské informácie.	Odporúča sa vtedy, keď je možné použiť takú istú správu iného <i>commit</i> objektu.
-c [commit objekt]	Ako -C [commit objekt], ale dá sa meniť.	Odporúča sa hlavne pri existencií podobného <i>commit</i> objektu.
-e (edit)	Umožňuje meniť správu, ktorá by sa inak nedala meniť. (-C, -F)	Odporúča sa hlavne pri čítaní správy zo súboru.
-F [súbor]	<i>Správa</i> sa načíta z určeného súboru.	Odporúča sa hlavne pri vytváraní dlhších správ.
-v (verbose)	Na zistenie vykonaných zmien oproti predchádzajúcemu <i>commit</i> objektu pri príkaze <i>commit</i> .	Nutné.
-m (message) [správa]	Napísanie správy, ktorá sa uloží s novým <i>commit</i> objektom.	Odporúča sa pri kratších správach.
-n (no verify)	Nahrádza predchádzajúci <i>commit</i> objekt s odpovedajúcou správou.	Zakazuje sa.
-q (quite)	Zatajenie správy.	Zakazuje sa.
-t [súbor]	Používa obsah daného súboru na inicializáciu správy. Správu je možné ďalej meniť. Ak sú pri príkaze <i>commit</i> nastavené parametre -m alebo -F, tak táto možnosť nemá žiadny účinok.	Iná varianta kombinácie -F [súbor] -s.

Tabuľka 11.2: Parametre pre príkaz Commit

### Ako písať príkaz commit

Bežný odporúčaný formát príkazu *commit*.

- git commit -a -v -m [správa]

Použitie *správy* z iného *commit* objektu s možnosťou zmeny správy. *Správa sa odporúča zmeniť*, aby bolo možné na základe rôznych správ rozlíšiť *commit* objekty.

- git commit -a -v -c [identifikačné číslo commit objektu]

Použitie správy zo súboru s možnosťou zmeny správy. *Správa sa odporúča zmeniť*, aby bolo možné

na základe rôznych správ rozlíšiť *commit* objekty.

- `git commit -a -v -F [súbor] -e`

Ak ešte nebol urobený príkaz *commit* a je potrebné urobené zmeny odstrániť:

- `git reset -hard`

Alebo je možné odstrániť špecifikovaný *commit* objekt:

- `git reset -hard[identifikačné číslo commit objektu]`

### Ako často robiť príkaz *commit*

Pred vývojom novej funkcionality sa napíšu najprv testy. To odpovedá spôsobu vývoja v našom tíme a vývoju riadeného testovaním. Po vývoji novej funkcionality ju treba otestovať. Testy sa musia robiť na konci vývoja funkcionality, ale odporúča sa, hlavne pri vývoji robustnejšej funkcionality, aby sa časť funkcionality testovala aj počas vývoja. Pokiaľ funkcionality prejde úspešne testami, je možné urobiť príkaz *commit*. Za rozhodnutie, že testy sú úspešné je zodpovedný vývojár funkcionality. Vývojár si nemôže dovoliť uložiť také zmeny, ktoré by v prípade spojenia s vetvou *devel* spôsobili nefunkčnosť, poškodili zdrojový kód iných vývojárov alebo spôsobili iné škody.

### Správy pri príkaze *commit*

Sú nevyhnutné pre rýchle pochopenie vykonaných zmien. Sú písané hlavne pre ľudí, ktorí nevykonávali zmeny, preto je dôležité, aby boli napísané jasne a zrozumiteľne. Musia byť pochopiteľné po ľubovoľnom čase od uloženia zmien.

Správy sa budú písať v slovenčine. Je nevyhnutné, aby napísané slová a vety boli gramaticky korektné.

### Ako písať správu

Dobre napísaná správa sa skladá z nasledovných častí:

- Začiatočný riadok
  - Začína sa identifikátorom, ktorý označuje vetvu v *Redmine* za účelom implementácie novej funkcionality alebo vyriešenia chyby.
  - Za identifikátorom je meno vývojára alebo skupiny vývojárov, ktorí sú zodpovední za zmeny.
  - Potom nasleduje dvojbodka a stručný súhrn zmien.
  - Píše sa v prítomnom čase.
  - Stručný opis zmien.
  - Má do 50 znakov.
- Prázdny riadok
  - Nasleduje po začiatočnom riadku.
  - Oddeľuje súhrn zmien od tela.

- Ak sa vynecháva telo, nie je potrebný.
- Telo
  - Píše sa v prítomnom čase.
  - Nasleduje po prázdnom riadku.
  - Detailnejší opis zahrňujúci:
    - motiváciu pre zmeny,
    - zmenu správania, ktoré zapríčinila nová implementácia,
    - aké chyby sa opravili,
    - či a aké rozhrania sa zmenili,
    - možné nedostatky.

## 12. Monitorovanie projektu

Autor: Matej Červeňák

### 12.1. Nástroje na monitorovanie projektu

Na monitorovanie celého projektu sa používa nástroj *Redmine*. Ten slúži aj na manažovanie projektu. Keďže ale obsahuje veľkú paletu grafov a prehľadov, možno ho použiť i na efektívne monitorovanie. Jednotlivé časti v *Redmine* majú rozličné možnosti filtrovania. Či už podľa osoby, ktorej sme úlohu prideliť, alebo podľa časti vývoja alebo časových obmedzení.

#### 12.1.1. Redmine a jeho nástroje

Na sledovanie a monitorovanie projektu disponuje *Redmine* týmito nástrojmi:

##### Roadmap / Plán

- percentuálny prehľad naplňania jednotlivých šprintov
- slúži na prehľad jednotlivých šprintov

Pre každý šprint sa zobrazujú priradené úlohy a celkový percentuálny prehľad vyriešených úloh:

- Tmavá farba - uzatvorené úlohy (status *closed*)
- Bledá farba - vyriešené časti otvorených úloh (status *open*)



### Spent time / Strávený čas

- prehľad stráveného času podľa vybranej skupiny
- slúži na sledovanie stráveného času

**Spent time**

▼ Date range  
 all time  
 From: 2011-10-05 to: 2011-12-08

✓ Apply Clear

Details Report

Details: Month Add: Clear

Total: 415.86 hours

Category	2011-10	2011-11	2011-12	Total
none	121.86	187.50	75.00	384.36
Study	5.00	4.00	6.00	15.00
Documents		16.50		16.50
<b>Total</b>	<b>126.86</b>	<b>208.00</b>	<b>81.00</b>	<b>415.86</b>

Obrázok 12.2: Ukážka obrazovky Strávený čas (*Spent time*)

### Reports / Prehľad

- prehľad stavov úloh z pohľadu rôznych skupín
- slúži na celkový prehľad súčasného stavu otvorených (*open*) a uzavretých (*closed*) úloh

**Reports**

Tracker	open	closed	Total
Bug	-	1	1
Feature	8	54	62
Support	1	13	14
Idea	-	1	1
Study	1	11	12
Change	-	7	7

Version	open	closed	Total
00_product_backlog	-	-	-
01_Anofeles	-	26	26
02_bubo_bubo	6	10	16
03_canis	-	11	11
04_Delphinidae_Delphis	1	3	4
05_Eptesicus_Fuscus	-	4	4

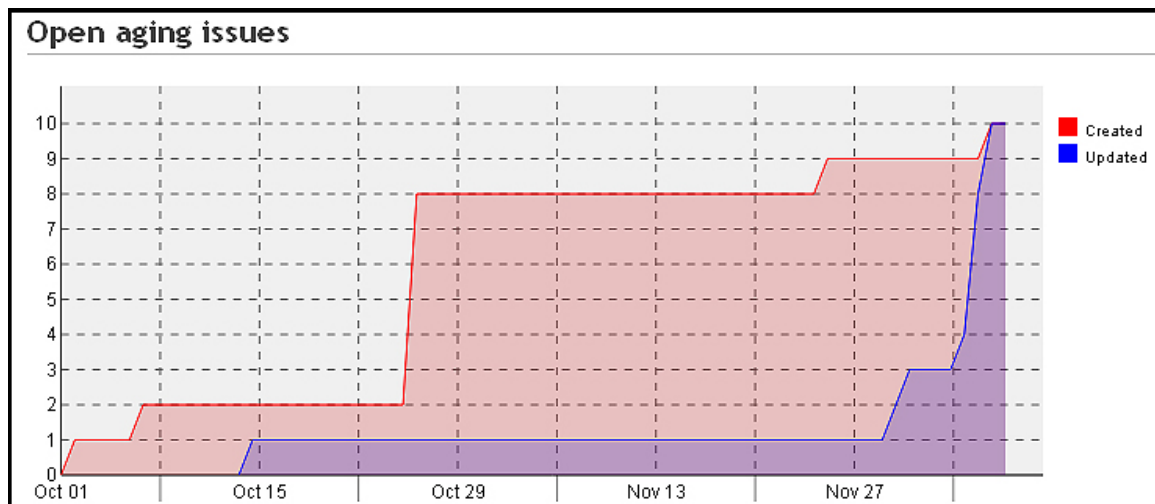
Priority	open	closed	Total
Low	-	-	-
Normal	9	66	75
High	1	18	19
Urgent	-	-	-
Immediate	-	3	3

Category	open	closed	Total
Documents	-	3	3
Study	1	4	5

Obrázok 12.3: Ukážka obrazovky Prehľad (Reports)

### Open aging issues

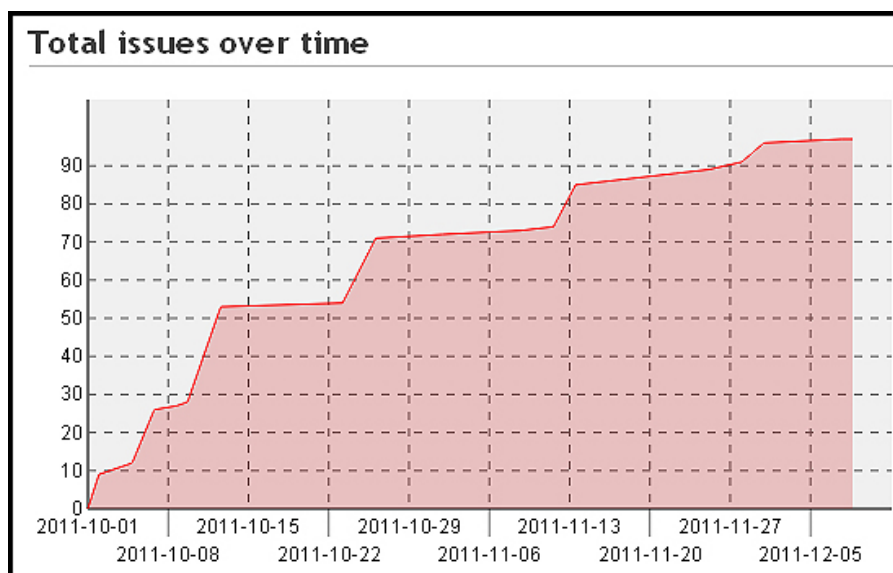
- graf, ktorý zobrazuje otvorené a aktualizované úlohy v časovom úseku
- slúži na sledovanie vývoja úloh



Obrázok 12.4: Ukážka grafu Open aging issues

### Total issues over time

- graf, ktorý zobrazuje všetky vytvorené úlohy v danom časovom úseku
- slúži na sledovanie celkového počtu úloh v čase

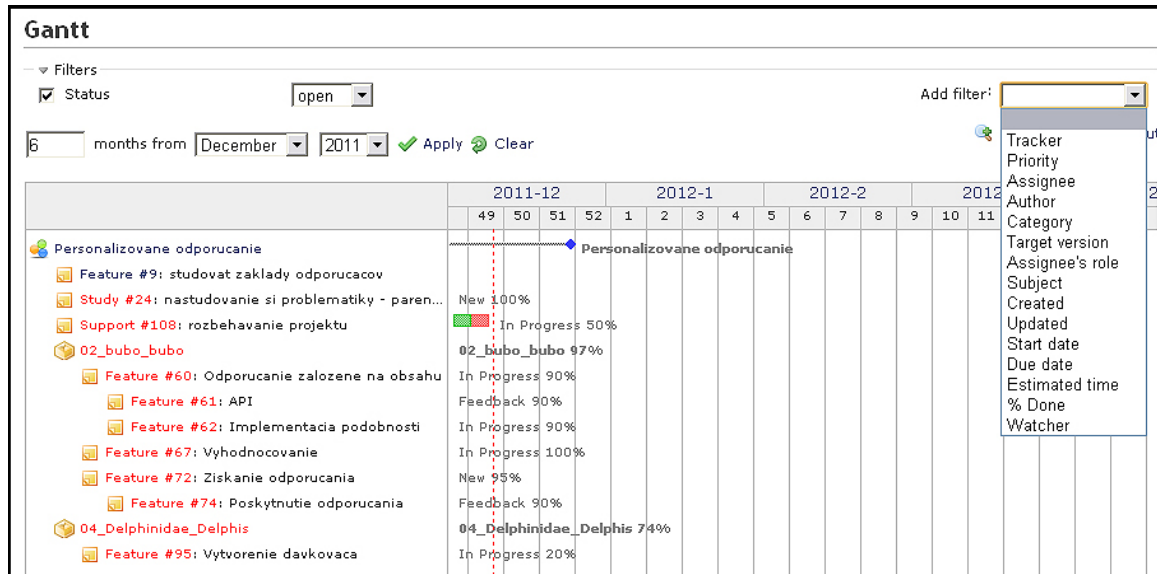


Obrázok 12.5: Ukážka grafu Total issues over time

## Gantt chart / Gantov graf

- trvanie jednotlivých úloh a ich nadväznosť pre zadanú skupinu
- slúži na sledovanie dokončených častí projektu a umožňuje ich lepšie plánovanie

Zobrazuje dátumy na jednej osi a úlohy spolu so zdrojmi na druhej. Graf znázorňuje kedy má byť úloha splnená a jej aktuálny stav (*progress*). Graf umožňuje sledovať, čo bolo už v procese dokončenia projektu urobené a odhadnúť, či je daná činnosť popredu, pozadu alebo prebieha načas.

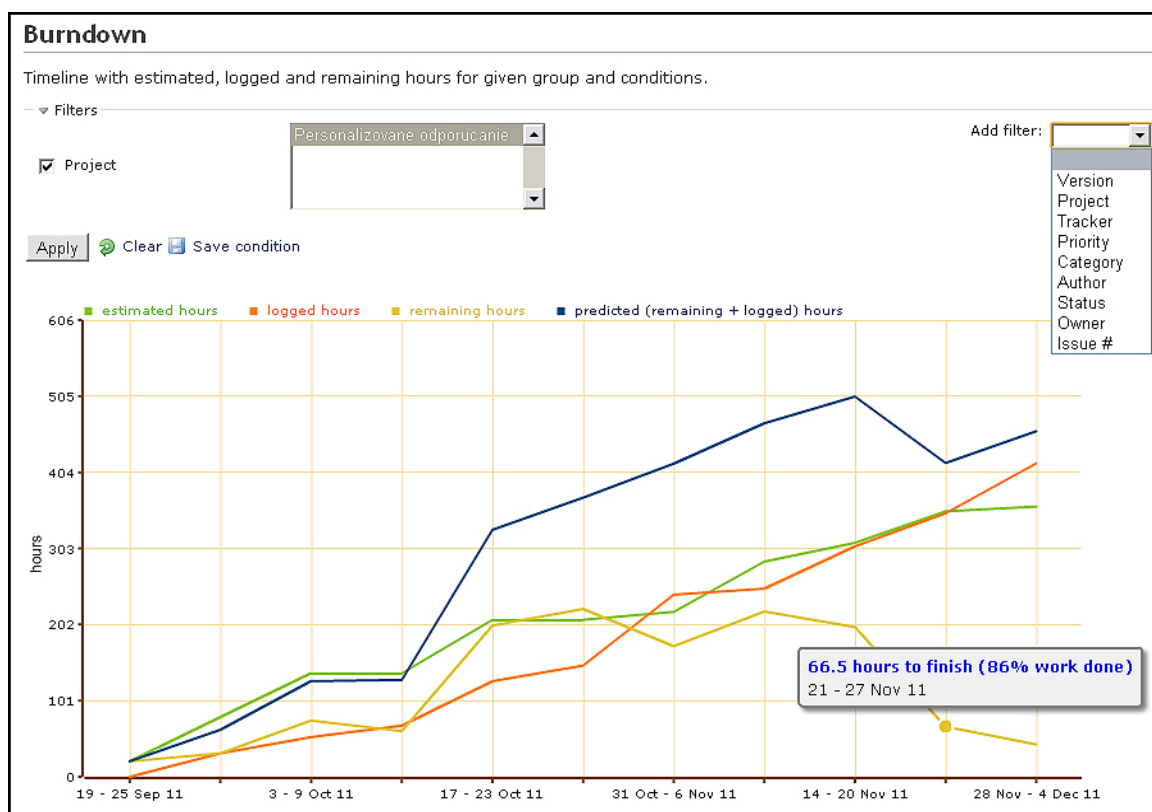


Obrázok 12.6: Ukážka obrazovky Gantov graf (*Gantt chart*)

## Burndown chart

- časová os s odhadmi, evidovanými a ostávajúcimi hodinami pre zadanú skupinu a podmienky
- slúži na sledovanie odhadov a ich porovnanie so skutočným vývojom.





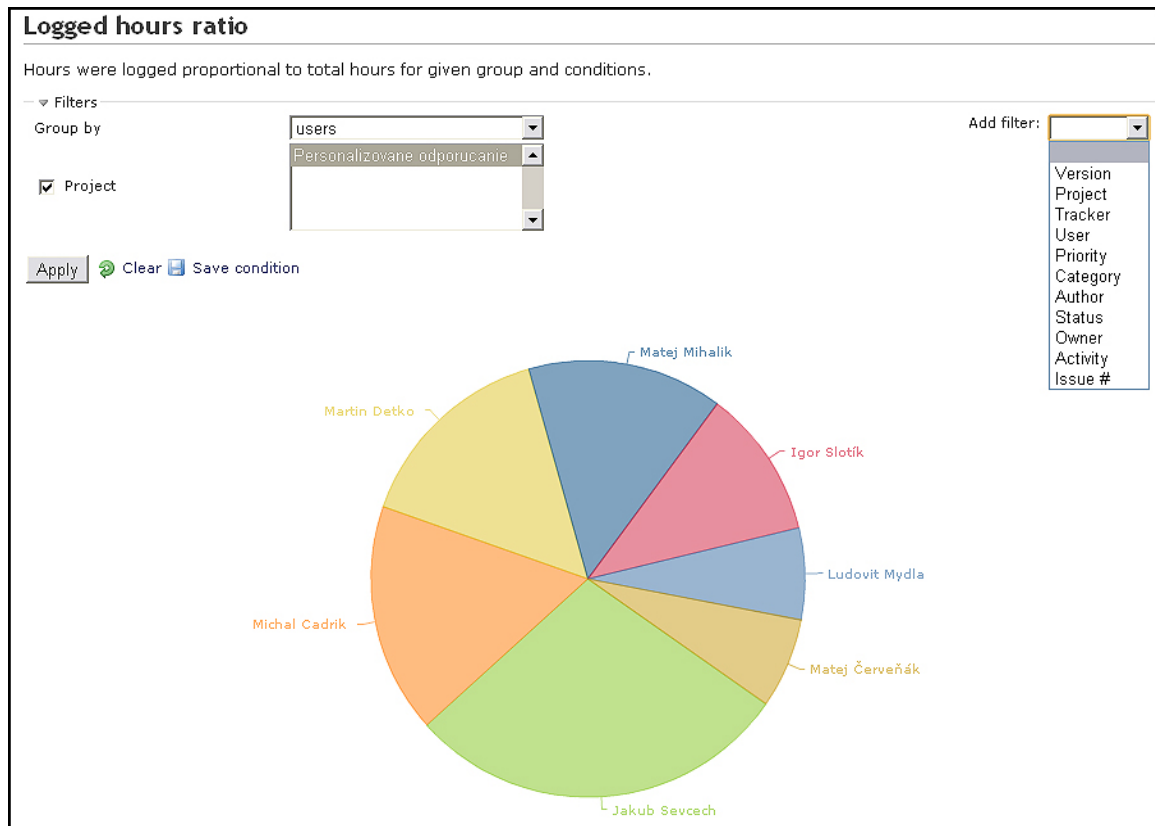
Obrázok 12.7: Ukážka obrazovky Burndown chart

### Logged hours timeline

- časová os s počtom evidovaných hodín pre zadanú skupinu a podmienky
- Slúži na sledovanie spotrebovaného času (hodín) pre jednotlivé aspekty vývoja

### Logged hours ratio

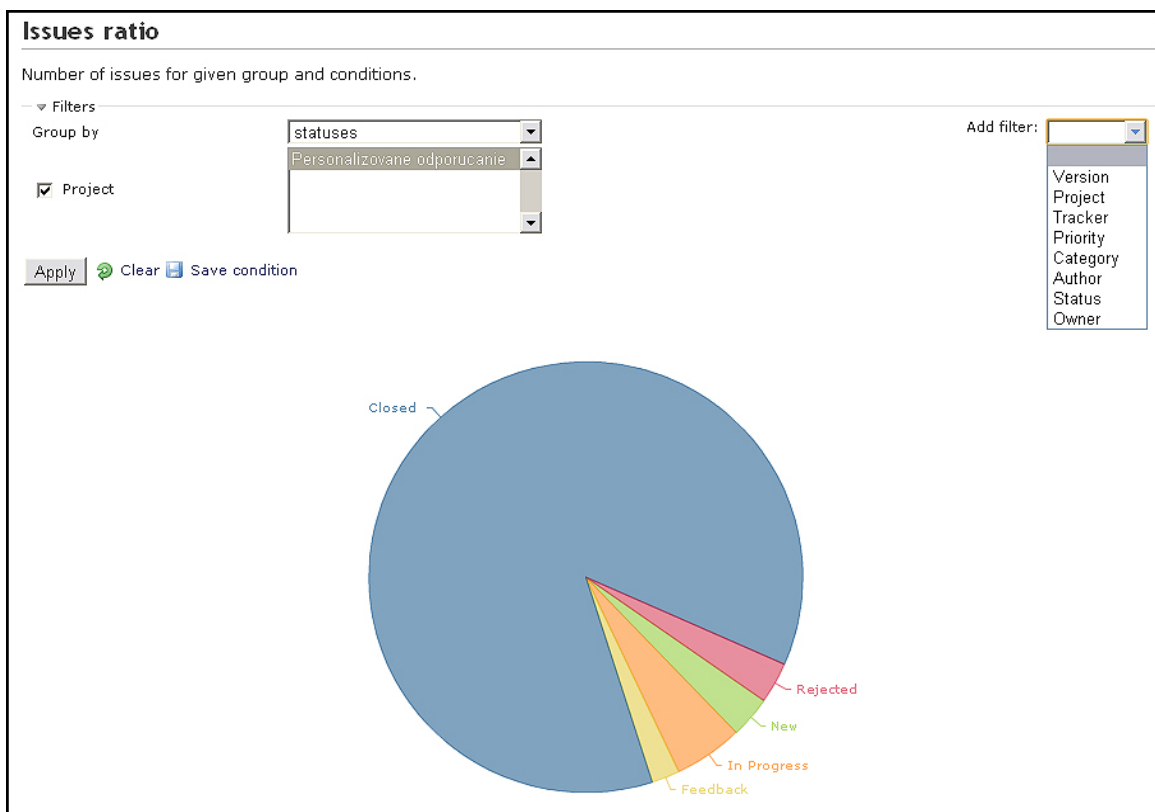
- graf zobrazujúci počet evidovaných hodín a ich proporčné rozdelenie pre zadanú skupinu a podmienky
- Slúži na sledovanie rozdelenia času (hodín) pre jednotlivé aspekty vývoja.



Obrázok 12.8: Ukážka obrazovky Logged hours ratio

### Issues ratio

- graf zobrazujúci počet úloh a ich proporčné rozdelenie pre zadanú skupinu a podmienky
- slúži na sledovanie stavov úloh a ich proporčné rozdelenie

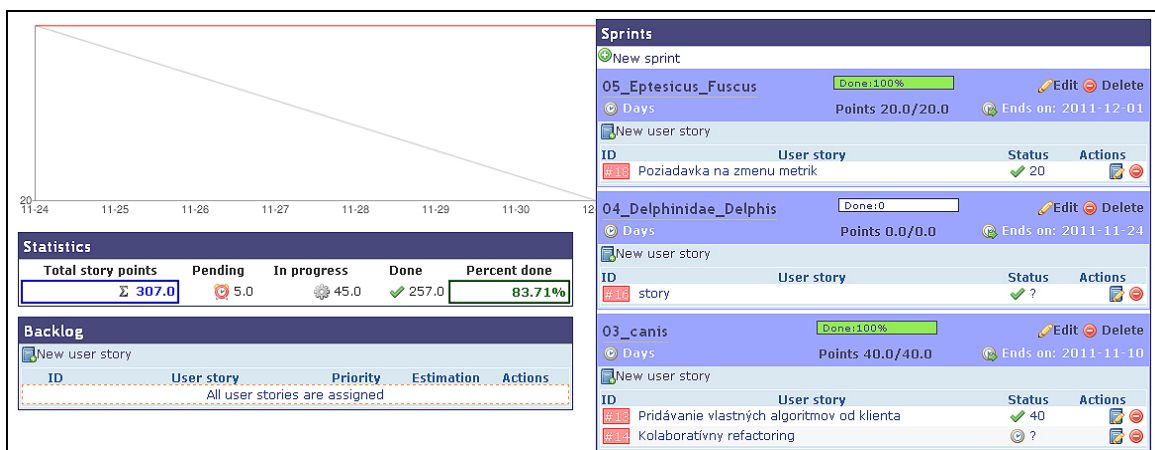


Obrázok 12.9: Ukážka obrazovky Issues ratio

### Backlog

- prehľad jednotlivých šprintov a k nim priradených užívateľských príbehov
- slúži na sledovanie naplňania šprintov a k nim priradených užívateľských príbehov

Pre každý šprint sa zobrazuje jeho pokrok vo vývoji, celkový stav a stav užívateľských príbehov. Po kliknutí na daný šprint sa zobrazia jeho detaily (viď. 8).



Obrázok 12.10: Ukážka obrazovky Backlog

## Dashboard

- zobrazuje jednotlivé užívateľské príbehy zaradené do daného šprintu a rozdelenie užívateľských príbehov na úlohy
- slúži na sledovanie statusov jednotlivých úloh užívateľských príbehov zaradených do šprintov

Graficky znázorňuje úlohy zaradené do daného šprintu, informácie o danej úlohe (názov, priradenie, stav) a užívateľský príbeh, pod ktorý úloha spadá.

The screenshot displays a Jira dashboard for a sprint named "02\_bubo\_bubo". At the top, there is a progress bar indicating "Done: 76%". Below this, the sprint description is visible, along with "Days" and "Points 145.0/190.0". The sprint ends on "2011-10-27".

The dashboard is organized into columns representing different stages of the sprint:

- User story**: This column contains two user stories:
  - "Odporúčanie založené na obsahu(P1)" with a priority of #5 and 40 points. The description is: "Získať odporúčanie content-based prístupmi, aby bolo možné porovnávať metódy."
  - "Výpočet podobnosti(P1)" with a priority of #8 and 100 points. The description is: "Počítať podobnosť používateľov a dokumentov rôznymi spôsobmi, aby sa dalo skúmať vplyv podobnosti na odporúčanie."
- Pending**: This column is currently empty.
- In progress**: This column contains three tasks:
  - "API" assigned to "Misiak" with a status of "Feedback".
  - "Implementacia podobnosti" assigned to "ludo" with a status of "In Progress".
  - "Spravit API" assigned to "sevo" with a status of "Closed".
  - "Implementacna logika pouzivatelov" assigned to "xmihalikm7" with a status of "Closed".
  - "Implementacna logika dokumentov" assigned to "xmihalikm7" with a status of "Closed".
- Done**: This column is currently empty.

Obrázok 12.11: Ukážka obrazovky Dashboard

## 12.2. Prehliadky modelov

### Použité nástroje

Názov	Použitie
Software Ideas Modeler	<ul style="list-style-type: none"> <li>• tvorba a úprava modelov</li> <li>• prehliadky modelov</li> </ul>
LibreOffice Writer	<ul style="list-style-type: none"> <li>• vypracovanie záznamu o vykonanej prehliadke</li> </ul>
Redmine	<ul style="list-style-type: none"> <li>• spoločné úložisko tímu</li> </ul>

Tabuľka 12.1: Použité nástroje

### Roly a zodpovednosti

Rola	Zodpovednosť
Návrhár	<ul style="list-style-type: none"> <li>• Návrh modelov</li> <li>• tvorba modelov</li> <li>• úprava modelov po prehliadke</li> <li>• vypracovanie záznamu o úprave</li> </ul>
Kontrolór	<ul style="list-style-type: none"> <li>• Prehliadky modelov</li> <li>• vypracovanie záznamu zo stretnutia</li> </ul>
Scrum master	<ul style="list-style-type: none"> <li>• Plánovanie prehliadok</li> </ul>
Product owner	<ul style="list-style-type: none"> <li>• Schvaľovanie záverečných zmien</li> </ul>

Tabuľka 12.2: Roly a zodpovednosti

### Procesy prehliadky

#### Inicializácia prehliadky

*Vstup: požiadavka na vykonanie prehliadky*

*Výstup: zoznam prítomných na prehliadke a vypísanie počiatočných informácií do dokumentu*

*Zodpovednosť: scrum master*

Manažér prehliadok predloží požiadavku na vykonanie prehliadky modelu a určí, kto sa danej prehliadky zúčastní. Jeho úlohou je najmä výber osoby, zodpovednej za funkciu kontrolóra. Táto osoba by nemala byť rovnaká ako tá, ktorá zastáva funkciu návrhára. Funkcia kontrolóra môže byť tiež pridelená aj zákazníkovi. V takom prípade sa musí zúčastniť prehliadky. Návrhár zapíše do prázdneho záznamu o vykonanej prehliadke údaje o navrhnutom modeli a informácie spolu s tým súvisiace.

### **Prezentácia podkladov**

*Vstup: podklady súvisiace s prehliadkou*

*Výstup: prezentácia podkladov a kontrolovaného modelu*

*Zodpovednosť: návrhár*

Návrhár, ktorý je autorom kontrolovaných modelov, si pripraví krátku prezentáciu o navrhnutom modeli. Vysvetlí jednotlivé časti modelu a zdôvodní svoje rozhodnutia pri návrhu. V prípade potreby, aby neprišlo k nedorozumeniam, môže nastať diskusia, ktorú riadi a usmerňuje scrum master.

### **Vykonanie prehliadky**

*Vstup: podklady potrebné k prehliadke*

*Výstup: skontrolované podklady modelu*

*Zodpovednosť: kontrolór*

Kontrolór vykoná prehliadku modelu v programe *Software Ideas Modeler*, v ktorom bol model aj vytvorený. Riadi sa pri tom údajmi, ktoré je potrebné poznačiť v zázname o vykonanej prehliadke.

### **Vypracovanie záznamu o vykonanej prehliadke**

*Vstup: skontrolované podklady modelu a dokument o vykonanej prehliadke*

*Výstup: skontrolovaný model a vypracovaný dokument o vykonanej prehliadke*

*Zodpovednosť: kontrolór*

Kontrolór po vykonaní prehliadky zaznamená potrebné údaje do dokumentu o vypracovaní prehliadky. Prípadne navrhne potrebné zmeny a úpravy. Ak je kontrolovaný model v poriadku a nie sú navrhnuté žiadne úpravy, pokračuje sa až procesom *Ukončenie prehliadky*.

Tento proces prehliadky je detailne vysvetlený v časti *II*.

### **Zmeny po prehliadke**

*Vstup: dokument o vykonanej prehliadke s prípadnými zmenami*

*Výstup: správa o vykonaní zmien*

*Zodpovednosť: návrhár*

Ak kontrolór odhalí chyby v návrhu modelu, alebo navrhne prípadné vylepšenia, musí ich návrhár zapracovať do svojho modelu. Riadi sa pri tom zaznamenanými údajmi v dokumente o prehliadke. Po zmene modelu návrhár zapracuje správu o jej vykonaní do dokumentu o prehliadke.

### **Kontrola zmien**

*Vstup: prezentácia zmenených podkladov*

*Výstup: vyhodnotenie kontroly a potvrdenie o kontrole*

*Zodpovednosť: scrum master*

Po zapracovaní prípadných zmien do modelu, návrhár urobí krátku prezentáciu, kde objasní

prevedené úpravy. Po ukončení prezentácie, a prípadnej diskusie, scrum master vyhodnotí kontrolu a vypíše potvrdenie o kontrole v dokumente o prehliadke. Ak nie sú zmeny vykonané správne alebo ich nebolo vôbec treba uskutočniť, navrhne sa nová úprava modelu a postup sa vracia späť na proces *Zmeny po prehliadke*.

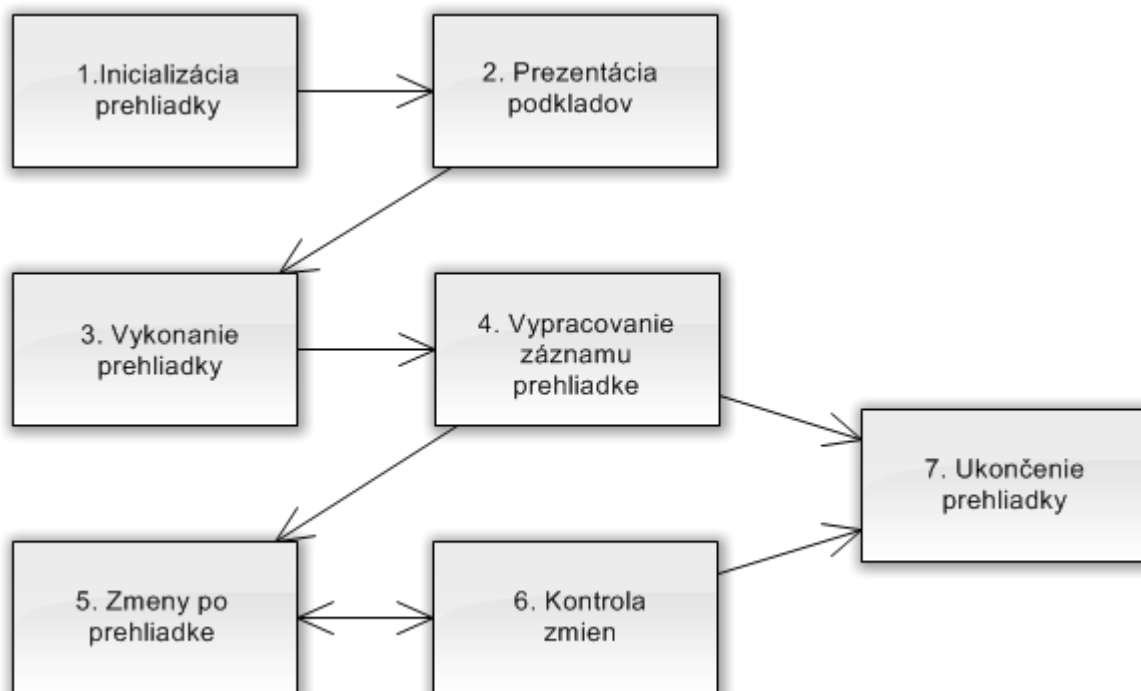
### Ukončenie prehliadky

*Vstup: všetky vypracované materiály pri prehliadke*

*Výstup: ukončenie prehliadky a schválenie materiálov*

*Zodpovednosť: product owner*

Product ownerovi sú po vykonaní prehliadky, prípadných zmenách a potvrdení od scrum mastera poskytnuté vypracované záverečné správy z jednotlivých procesov. Ak je všetko v poriadku, tak product owner schváli a ukončí prehliadku. Pokiaľ nastanú nejaké nezrovnalosti, tak po konzultácii a dohode so scrum masterom rozhodne o ďalšom postupe.



#### 12.2.1. Vypracovanie záznamu o vykonanej prehliadke

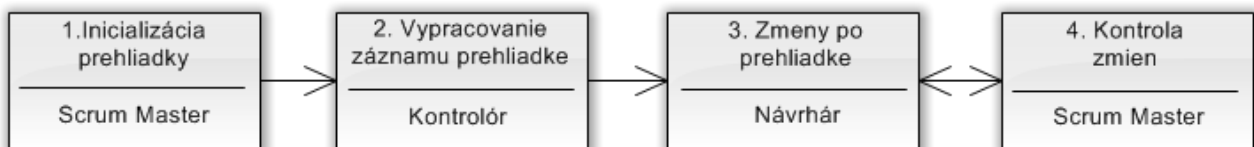
##### Členenie formulára

Formulár je rozdelený na štyri hlavné časti a každá z nich sa vyplňa v inom procese prehliadky.

V 12 je ku každej časti dokumentu priradený proces, v ktorom sa s danou časťou pracuje a zodpovedná osoba, ktorá je poverená túto časť vyplniť. Každý z procesov uvedených v 12 končí vyplnením alebo potvrdením záznamu o vykonanej prehliadke.

Časť dokumentu	Názov procesu	Vypĺňa
1.	Inicializácia prehliadky	Scrum master
2.	Vypracovanie záznamu o prehliadke	Kontrolór
3.	Zmeny po prehliadke	Návrhár
4.	Kontrola zmien	Scrum master

Tabuľka 12.3: Časti dokumentu a k nim priradené procesy a zodpovedné osoby



Obrázok 12.13: Postup pri vyplňaní formulára

### Vypĺňanie formulára

1. Vopred pripravenú šablónu formulára si stiahnuť zo systému *Redmine*.
2. Na základe zistených skutočností pri prehliadke modelu vyplniť požadované prázdne polia dokumentu.
  - Na vyplnenie formulára implicitne použiť *LibreOffice Writer* a výstupný súborový formát *\*.odt*.
  - Vo výnimočnom prípade je možné použiť iný textový editor s podporou exportu do súborového formátu *\*.odt*.
3. Vyplnený záznam o vykonanej prehliadke nahrať do systému *Redmine*.
4. O vyplnení dokumentu informovať kompetentnú osobu, ktorá preberá ďalšiu zodpovednosť za dokument:
  - Návrhára – ak sú navrhnuté nejaké zmeny alebo sa v modeli vyskytli nezrovnalosti
  - Scrum mastera – ak je všetko v poriadku a prehliadka prebehla bez navrhnutých úprav

### Návod na vyplňanie polí formulára

Formulár obsahuje vopred pripravené polia, do ktorých zodpovedné osoby zapíšu požadované údaje. Je členený na štyri samostatné časti. Každú časť vyplňa iná osoba v rôznych procesoch (viď 12).



**Názvy polí a návod na ich vyplnenie:****1. Časť dokumentu**

- **Názov:** Inicializácia prehliadky
- **Vypĺňa:** návrhár

Názov poľa	Popis
<i>Návrhár</i>	meno autora modelu
<i>Projekt</i>	názov projektu, ku ktorému bol model vytvorený
<i>Časť projektu</i>	lokalizácia časti projektu, ku ktorej patrí
<i>Dátum vytvorenia</i>	dátum vytvorenia modelu
<i>Použitý softvér</i>	názov nástroja, ktorým bolo model vytvorený
<i>Typ modelu</i>	špecifikácia modelu (príklady: Use case, Diagram tried, Diagram postupnosti)
<i>Verzia</i>	verzia vytvoreného modelu
<i>Názov súboru</i>	názov súboru, v ktorom je model uložený

**2. Časť dokumentu**

- **Názov:** Vypracovanie záznamu o vykonanej prehliadke
- **Vypĺňa:** kontrolór

Názov poľa	Popis
<i>Meno kontrolóra</i>	meno revízora modelu
<i>Dátum vykonania prehliadky</i>	dátum, kedy bola prehliadka vykonaná
<i>Číslo kontrolovanej časti</i>	poradové číslo kontrolovanej časti
<i>Kontrolovaná časť</i>	popis kontrolovanej časti
<i>Vyhodnotenie</i>	status určujúci, či sa nachádza v kontrolovanej časti chyba
<i>Popis chyby (nepovinné)</i>	popis nájdenej chyby v modeli. Ak sa chyba nevyskytuje, nevyplňa sa
<i>Návrh na zmenu (nepovinné)</i>	popis navrhnutej zmeny. Ak kontrolór nepozná riešenie problému, nevyplňa sa
<i>Čas na opravu (nepovinné)</i>	navrhnutý čas vyriešenia problému. Ak kontrolór nevie určiť, nevyplňa sa

### 3. Časť dokumentu

- **Názov:** Zmeny po prehliadke
- **Vyplňa:** návrhár

Názov poľa	Popis
<i>Meno návrhára</i>	meno autora zmien
<i>Dátum zmeny</i>	dátum, kedy bola zmena vykonaná
<i>Zpracovanie zmeny</i>	obsahuje údaj o tom, či bola navrhnutá zmena uskutočnená
<i>Dôvod nevykonania zmeny (nepovinný údaj)</i>	popis dôvodu nevykonania zmeny. Ak bola zmena vykonaná, nevypĺňa sa

### 4. Časť dokumentu

- **Názov:** Kontrola zmien
- **Vyplňa:** scrum master

Názov poľa	Popis
<i>Meno scrum mastera</i>	meno autora kontroly zmien
<i>Dátum kontroly</i>	dátum, kedy bola kontrola zmien vykonaná
<i>Schválenie/Neschválenie zmien</i>	údaj o tom, či bola zmena schválená alebo nie
<i>Návrh nových zmien (nepovinný údaj)</i>	ak nebola zmena schválená, môže sa navrhnúť úprava
<i>Potvrdenie o vykonaných zmenách</i>	záverečné potvrdenie zmien
<i>Potvrdenie o prehliadke</i>	záverečné potvrdenie celej prehliadky

## 13. Manažment komunikácie

**Autor:** Jakub Ševcech

### 13.1. Nástroje na podporu komunikácie

#### Email

- Po osobnej komunikácii najčastejšie používaný prostriedok komunikácie. V spojení s emailovou adresou vytvorenou pomocou nástroja Google group sa dá používať na komunikovanie s celým tímom súčasne.
- Po dokončení nejakej časti, ktorá sa týka viacerých členov tímu, treba poslať hromadný email na adresu skupiny ([tim12tp@googlegroups.com](mailto:tim12tp@googlegroups.com))
- Rovnako treba posielat' hromadný email pri ďalších udalostiach (vytvorenie metodiky, otázka ...), ktoré sa týkajú celého tímu

#### ICQ / Facebook Chat / Jabber

- Komunikácia v reálnom čase
- Posielanie súborov
- Tento nástroj sa využíva na komunikáciu v reálnom čase medzi členmi tímu. Je určený na rýchle riešenie problémov a na rýchle odpovedanie na otázky.
- V prípade použitia tohto nástroja je potrebné kontaktovať toho, kto je zodpovedný za úlohu, ktorú riešime alebo toho, kto je najschopnejší pomôcť v riešení aktuálneho problému.
- Tento nástroj slúži na podporu diskusie, ktorá sa ťažko vytvára prostredníctvom emailu.

### 13.2. Nástroje na podporu kolaborácie

#### Git

Nástroj na manažment verzii. Používame na manažment verzii zdrojových kódov.

- Vývojové vetvy projektu
- Uchovávanie stavov projektu
- podpora distribuovaného vývoja
- Zdieľané úložisko

Ku každému uloženému stavu projektu je možné priradiť krátku správu o zmenách v tejto verzii spolu s informáciou o autorovi zmeny. Pomocou týchto správ sa dá sledovať aktivita jednotlivých členov tímu a oznamuje sa tak celému tímu stav, v akom je v súčasnosti projekt.

Nástroj git dokonca umožňuje vystopovať autorstvo konkrétnych riadkov zdrojového kódu projektu prostredníctvom týchto správ. Takto sa dá nie len sledovať postup projektu, ale sa dá aj rýchlo nájsť človek, ktorý je zodpovedný za konkrétny úsek kódu a s prípadnými otázkami je možné obrátiť sa na konkrétneho zodpovedného človeka.

Vytváranie vetiev umožňuje pracovať na jednej funkcii oddelene od zvyšku projektu a to nie len jednému členovi tímu, ale zdieľaním tejto vetvy na servery je možné, aby sa do nej zapojil každý.

## Redmine

Nástroj na manažovanie projektu, ktorý umožňuje:

- Manažovanie úloh v projekte
- Zdieľanie súborov
- Wiki
- Grafy vývoja projektu

Tento nástroj sa používa na všetky úlohy spojené s pridelovaním úloh a s monitorovaním projektu. Na začiatku každého šprintu ten, kto viedol stretnutie vloží úlohy do tohto nástroja, a potom každý člen tímu upravuje stav svojich úloh podľa toho, ako na nich pracuje.

Takto sa dá sledovať stav celého projektu a to, kto je zodpovedný za jednotlivé úlohy.

The screenshot shows the 'New issue' form in Redmine. At the top, there is a navigation bar with 'Home', 'My page', 'Projects', 'Administration', and 'Help'. The user is logged in as 'sevo'. The main navigation bar includes 'Overview', 'Activity', 'Roadmap', 'Issues', 'New Issue' (selected), 'Charts', 'Gantt', 'Calendar', 'News', 'Documents', 'Wiki', 'Forums', 'Files', 'Repository', and 'Settings'. The form itself has a 'Tracker' dropdown set to 'Change' and a 'Private' checkbox. The 'Subject' field is empty. The 'Description' field has a rich text editor with various formatting options. Below the description are fields for 'Status' (New), 'Priority' (Normal), 'Assignee', 'Category', and 'Target version' (00\_product\_backlog). To the right, there are fields for 'Parent task', 'Start date' (2011-11-05), 'Due date', 'Estimated time' (Hours), and '% Done' (0%). At the bottom, there is a 'Files' section with a 'Browse...' button and an 'Optional description' field. A 'Watchers' section lists several users with checkboxes: Igor Slotik, Martin Detko, Michal Cadrik, Jakub Sevcch, Matej Mihalik, Michal Kompan, Ludovit Mydla, and Matej Červeňák. At the very bottom, there are buttons for 'Create', 'Create and continue', and 'Preview'.

## Google Docs

Kolaboratívna tvorba a zdieľanie dokumentov. Zdieľanie dokumentov. Exportovanie do formátu *pdf* a do formátov kompatibilných s Microsoft Office a OpenOffice.

Tento nástroj sa používa vtedy, ak má viacero členov tímu pracovať súčasne na jednom dokumente. Vtedy je nepraktické vytvárať verzie dokumentu a dopisovať nové zmeny do aktuálnej verzie, pretože väčšina času sa stratí tým, že sa drží krok s neustále sa meniacimi verziami.

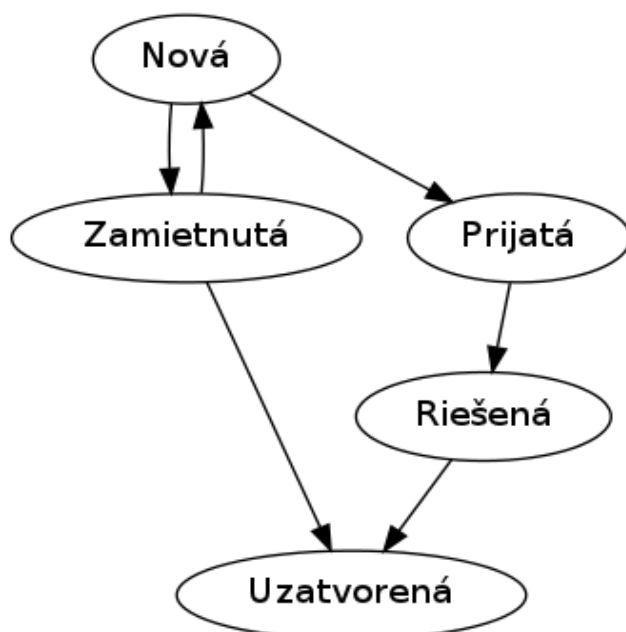
### 13.3. Manažment požiadaviek na zmenu

#### 13.3.1. Roly a ich zodpovednosti

Rola	Zodpovednosť
Zákazník (Product owner)	<ul style="list-style-type: none"> <li>• Vytvára požiadavky na funkcionality produktu do product backlogu</li> <li>• Na začiatku šprintu vytvára požiadavky na zmenu</li> <li>• Hodnotí či bude požiadavka na zmenu prijatá alebo nie</li> </ul>
Vedúci tímu (Scrum master)	<ul style="list-style-type: none"> <li>• Spolu so zákazníkom hodnotí či bude požiadavka na zmenu prijatá alebo nie</li> <li>• Kontroluje plnenie požiadavok</li> <li>• Vytvára zoznam požiadaviek pre nasledujúci šprint (sprint backlog)</li> </ul>
Vývojár	<ul style="list-style-type: none"> <li>• Spoločný názov pre roly programátora, testera, dokumentaristu ...</li> <li>• Implementuje jednotlivé funkcie projektu</li> <li>• Navrhuje, implementuje a spúšťa testy</li> <li>• Píše dokumentáciu</li> <li>• Rieši úlohy a chyby</li> </ul>

#### 13.3.2. Životný cyklus zmeny

Stav	Popis
Nová	Nová požiadavka na zmenu zaradená do product backlogu. Požiadavka na zmeny vzniká hocikedy v priebehu šprintu a vytvára ju zákazník alebo hociktorý člen tímu.
Prijatá	Požiadavka je prijatá a zaradená do sprint backlogu. Požiadavku prijímajú zákazník a vedúci tímu spoločne. Pri prijatí požiadavky na zmenu sa vytvorí detailná špecifikácia zmeny.
Zamietnutá	Požiadavka na zmenu je zamietnutá. Požiadavka však zostáva v product backlogu, pre prípadné prehodnotenie.
Riešená	V závislosti od výsledku procesu prijímania požiadavky sa spustí proces vytvárania úlohy alebo proces vytvárania záznamu o chybe.
Uzatvorená	V prípade, že bola požiadavka na zmenu prijatá, tak úloha alebo chyba, ktorá z nej vznikla bola vyriešená a uzatvorená. V prípade že bola požiadavka zamietnutá, tak sa Zákazník dohodol s vedúcim tímu na tom, že sa nebude realizovať.



Obrázok 13.2: Životný cyklus požiadavky na zmenu

### 13.3.3. Proces identifikácie a vytvorenia požiadavky na zmenu nástroji Redmine

**Vstup:** Potreba vykonania zmeny

**Výstup:** Požiadavka na zmenu

**Zodpovedná osoba:** Zákazník, vedúci tímu, vývojár

Vznikla potreba na vykonanie zmeny. Zákazník, vedúci tímu alebo vývojár vytvára požiadavku na zmenu a zaraďuje ju do product backlogu.

- Po zvolení voľby „New Issue“ v nástroji Redmine sa zobrazí formulár z obrázku 13.3. Na vytvorenie novej požiadavky na zmenu.
- Je potrebné zvoliť hodnotu „Change“ v selectboxe „Tracker“.
- Je potrebné vyplniť všetky povinné atribúty, menovite subjekt (Subject), stav (State), prioritu (Priority), popis (Description).
- Všetky novo vytvorené požiadavky na zmenu sú najskôr zaradené do product backlogu, preto je potrebné nastaviť hodnotu „00\_product\_backlog“ v selectboxe „Target version“.
- Pole „Assignee“ zostáva nevyplnené. Toto pole určuje, komu bude pridelené riešenie požiadavky na zmenu. Toto pole sa vyplňa v procese pridelovania požiadavky na zmenu.
- Vyplnenie ostatných voliteľných atribútov
- Potvrdenie vytvorenia požiadavky na zmenu stlačením tlačidla „Create“ alebo „Create and continue“.

Home My page Projects Administration Help Logged in as sevo My account Sign out

Personalizované odporúčanie Search: Personalizované odporúčanie

Overview Activity Roadmap Issues **New Issue** Charts Gantt Calendar News Documents Wiki Forums Files Repository Settings

**New issue**

Tracker \* Change  Private

Subject \*

Description B I U S C H1 H2 H3  Text formatting

Status \* New Parent task

Priority \* Normal Start date 2011-11-05

Assignee Due date

Category Estimated time Hours

Target version 00\_product\_backlog % Done 0%

Files Browse... Optional description

Add another file (Maximum size: 5 MB)

Watchers  Igor Slotik  Jakub Sevcch  Ludovit Mydla  
 Martin Detko  Matej Mihalik  Matej Červeňák  
 Michal Cadrik  Michal Kompan

Create Create and continue Preview

### 13.3.4. Proces prijatia požiadavky na zmenu

Proces prijatia požiadavky na zmenu sa skladá z dvoch podprocesov: procesu ohodnotenia požiadavky na zmenu a procesu zaradenia požiadavky na zmenu. Celý proces opisuje graf na obrázku číslo 13.4.

#### Proces ohodnotenia požiadavky na zmenu

**Vstup:** Požiadavka na zmenu

**Výstup:** Prijatá požiadavka na zmenu s určenou prioritou

**Zodpovedná osoba:** Zákazník, vedúci tímu, vývojári

Tento proces je zložený z troch hlavných krokov:

- Špecifikácia zmeny
- Určenie trvania realizácie zmeny
- Určenie priority požiadavky

#### Špecifikácia zmeny

Špecifikáciu na zmenu vytvára vedúci tímu spolu so zákazníkom. Špecifikáciu spíšu do textového súboru a pridajú ako prílohu k požiadavke na zmenu.

Vedúci tímu spolu so zákazníkom určia či ide o zmenu spôsobenú chybou pri riešení projektu, alebo ide o novú úlohu. Podľa toho zvolia hodnotu v poli „*Category*“ vo formuláre na úpravu detailu požiadavky na zmenu. Pre chybu zvolia „*Bug*“, pre novú úlohu „*Feature*“.

### Určenie trvania realizácie zmeny

Určenie trvania realizácie požiadavky na zmenu prebieha za účasti všetkých vývojárov.

Trvanie sa odhaduje na základe ohodnotenia náročnosti požiadavky jednotlivými vývojármi. Na toto sa používajú očíslované karty. Každý vývojár má balíček kariet, kde každá reprezentuje ohodnotenie zložitosti úlohy. Hodnoty tvoria Fibonacciho postupnosť (1,2,3,5,8,13,21,34,55,89). Každý vývojár vyberie jednu kartu, ktorá reprezentuje jeho odhad a naraz ich všetci ukážu. Tí, ktorí vybrali najvyššiu a najnižšiu hodnotu, povedia argumenty prečo ich zvolili. Nasleduje druhé kolo výberu ohodnocovania. Za výsledné ohodnotenie trvania sa zvolí priemer hodnôt kariet po druhom kole ohodnocovania. V poli „*Estimated time*“ vo formuláre na upravovanie detailu požiadavky sa vyplní hodnota odhadovaného času.

### Určenie priority

Prioritu určuje zákazník po odhadnutí času potrebného na realizáciu zmeny.

### Proces zaradenia požiadavky do sprint backlogu

**Vstup:** Požiadavka na zmenu

**Výstup:** Požiadavka na zmenu v sprint backlogu

**Zodpovedná osoba:** Zákazník, vedúci tímu

Proces zaradenia požiadaviek do sprint backlogu je zložený z dvoch krokov:

- Výber požiadavky
- Určenie začiatku a konca pre spracovanie zmeny

V product backlogu sú vytvorené požiadavky na zmeny, ktoré sa majú realizovať. Z týchto požiadavok sa na začiatku šprintu vyberú tie, ktoré sa v danom šprinte budú realizovať. Na tom, ktoré požiadavky sa vyberú do sprint backlogu sa dohodne zákazník s vedúcim tímu. Tento krok je ekvivalentný s vyberaním úloh do sprint backlogu.

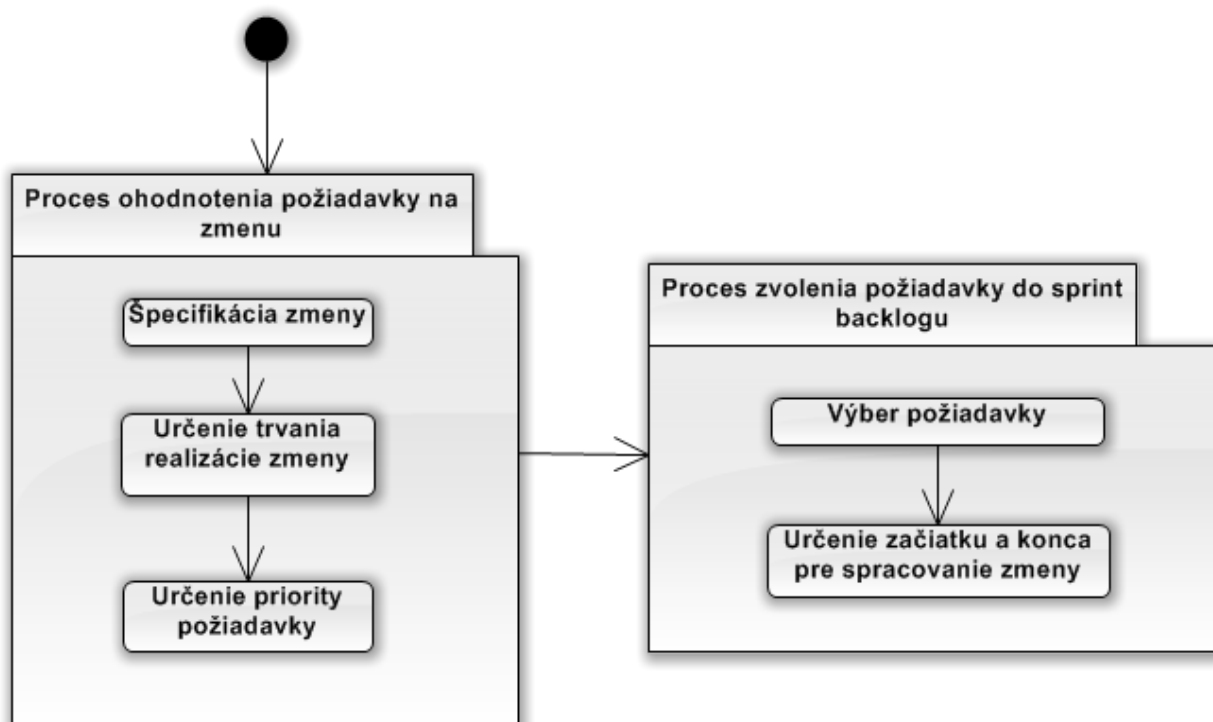
Na základe odhadu trvania realizácie a priority požiadavky na zmenu určí vedúci tímu čas, kedy sa musí začať požiadavka realizovať a čas, do ktorého musí byť požiadavka ukončená.

Vedúci tímu určí vývojára zodpovedného za menu. V poli „*Assignee*“ vo formuláre na nastavenie detailu požiadavky na zmenu zvolí jedného z vývojárov.

Vývojár nastaví v selectboxe „*Target version*“ vo formuláre na nastavenie detailu požiadavky na zmenu označenie šprintu, do ktorého chce úlohu zaradiť.

Po tomto procese nasleduje proces na vytvorenie záznamu o chybe alebo proces na vytvorenie úlohy v závislosti na kategórii požiadavky na zmenu, ktorý bol určený v procese Proces ohodnotenia požiadavky na zmenu.





### 13.3.5. Proces zamietnutia požiadavky na zmenu

**Vstup:** Požiadavky v product backlogu

**Výstup:** Zamietnuté požiadavky

**Zodpovedná osoba:** Zákazník, vedúci tímu

Do product backlogu sa môžu zadávať všetky požiadavky na zmeny. Niektoré z týchto požiadavky sa nikdy nebudú realizovať.

Vedúci tímu spolu so zákazníkom na začiatku šprintu postupne prechádzajú požiadavky na zmenu v product backlogu. Pri vyberaní požiadaviek na zamietnutie môžu zákazník spolu s vedúcim tímu vykonať tri operácie:

- prehodnotenie priority požiadavky na zmenu
- zamietnutie požiadavku na zmenu

Pri prehodnotení priority požiadavky sa zákazník s vedúcim tímu dohodnú na novej prioritě požiadavky na zmenu. Upravia prioritu a nechajú požiadavku v product backlogu.

Pri zamietnutí požiadavky na zmenu sa požiadavka označí ako zamietnutá, ale naďalej zostáva v product backlogu, pre prípadné prehodnotenie.

### 13.3.6. Proces prehodnotenia zamietnutia požiadavky na zmenu

**Vstup:** Zamietnutá požiadavka na zmenu

**Výstup:** Nová požiadavka na zmenu

**Zodpovedná osoba:** Zákazník, vedúci tímu

Vedúci tímu spolu so zákazníkom na začiatku šprintu postupne prechádzajú zamietnuté požiadavky na zmenu v product backlogu.

- Pre každú požiadavku sa dohodnú či naďalej zostane v product backlogu ako zamietnutá požiadavka, alebo ju posunú na opätovné posúdenie.
- Ak sa dohodnú na opätovnom posúdení požiadavky na zmenu, tak v nástroji Redmine zmenia jej stav na „Nová“ (New).

### **13.3.7. Proces uzatvorenia zamietnutej požiadavky na zmenu**

**Vstup:** Zamietnutá požiadavka na zmenu

**Výstup:** Uzatvorená požiadavka na zmenu

**Zodpovedná osoba:** Zákazník, vedúci tímu

Vedúci tímu spolu so zákazníkom na začiatku šprintu postupne prechádzajú zamietnuté požiadavky na zmenu v product backlogu.

- Pre každú požiadavku sa dohodnú či naďalej zostane v product backlogu ako požiadavka, ktorú možno raz prehodnotia alebo ju odstránia z product backlogu.
- Ak sa dohodnú na odstránení požiadavky, tak v nástroji Redmine zmenia jej stav na „Uzatvorená“ (Closed) a zmenia hodnotu v selectboxe „Target version“ na prázdnu hodnotu.

## 14. Manažment tvorby dokumentácie

Autor: Michal Cádrik

### 14.1. Nástroje na tvorbu dokumentácie

#### Libre Office Writer 3.4

Nástroj na písanie dokumentov ohľadom riadenia projektu a používateľskej príručky. Používa sa aj na písanie projektovej dokumentácie.

#### Software ideas modeler

Nástroj na tvorbu UML diagramov, v ktorom sa dajú vytvoriť rôzne diagramy a podporuje export v rôznych formátoch.

#### Javadoc

Priamo podporovaný nástroj pre podporu technickej dokumentácie, ktorý je priamo zahrnutý aj v prostredí Eclipse IDE. Každá funkcia alebo trieda musí mať definované parametre, popis a autora pomocou javadoc v anglickom jazyku. Komentáre musia byť napísané tiež v anglickom jazyku.

### 14.2. Používateľskú príručku

Dokumentarista	Osoba zodpovedná za napísanie dokumentácie: vedúci používateľského príbehu alebo osoba ním poverená.
Umiestnenie	<meno_projektu>\Doc\<ID_používateľského_príbehu> <stručný_názov> <verzia>
Editor	LibreOffice (aktuálne verzia 3.4)

Tabuľka 14.1: Zodpovedná osoba a umiestnenie pre používateľské príručky

Zodpovedná osoba za tvorbu používateľskej príručky je vedúci používateľského príbehu alebo osoba ním poverená. Začatie tohto procesu signalizuje používateľský príbeh v stave *akceptovaný*. Používateľská dokumentácia sa tvorí v podadresári koreňového adresára projektu, ktorý je prepojený s verziovacím systémom GIT. Podrobnosti o verziovaní sú v metodike *Manažment verzií*.

Kroky tvorby používateľskej príručky opisuje tabuľka 14.2.

Č. kroku	Činnosť
1.	<ul style="list-style-type: none"> <li>Vytvoriť úlohu do manažovacieho nástroja Redmine</li> </ul>
2.	<ul style="list-style-type: none"> <li>Vytvoriť súbor pre dokumentáciu</li> </ul>
3.	<ul style="list-style-type: none"> <li>Napísanie samotnej príručky</li> </ul>
3.1	<ul style="list-style-type: none"> <li>Úprava titulnej strany</li> </ul>
3.2	<ul style="list-style-type: none"> <li>Vytvorenie zoznamu skratiek a použitej notácie</li> </ul>
3.3	<ul style="list-style-type: none"> <li>Vysvetlenie používateľského prostredia</li> </ul>
3.3.1	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>Popis obrazoviek</li> </ul> </li> </ul>
3.3.2	<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>Popis častých scenárov</li> </ul> </li> </ul>
3.4	<ul style="list-style-type: none"> <li>Často sa pýtané otázky</li> </ul>
3.5	<ul style="list-style-type: none"> <li>Vzorové príklady</li> </ul>
3.6	<ul style="list-style-type: none"> <li>Kontrola dokumentácie manažérom dokumentácie a vlastníkom produktu</li> </ul>
3.7	<ul style="list-style-type: none"> <li>Upload dokumentu</li> </ul>

Tabuľka 14.2: Postupné kroky procesu tvorby používateľskej dokumentácie

### Vytvorenie úlohy v Redmine

Na adrese <http://vm28.ucebne.fiit.stuba.sk/Redmine/projects/recommend/issues/new> vytvorí zodpovedný človek novú úlohu, ktorej nastaví jednotlivé parametre. Dôležité je aby bolo pole *Tracker* nastavené na *Support* a vo *Watchlist* zaškrtnutý vedúci tímu. Názov úlohy bude „*Tvorba dokumentácie pre <názov>*“.

### Vytvorenie súboru pre dokumentáciu

Stiahnuť šablónu *SABLONA\_USERSTORYDOC\_XXX\_názov-používateľského-príbehu\_YY.Y.odt* z <http://vm28.ucebne.fiit.stuba.sk/Redmine/projects/recommend/documents>. Súbor musí byť uložený podľa tabuľky 14.1 pod názvom vytvoreným podľa kapitoly .

### Názov dokumentu

Názov dokumentu je v tvare:

*XXX\_názov-používateľského-príbehu\_YY.Y.odt*

kde: *XXX* je ID používateľského príbehu, *názov-používateľského-príbehu* je stručný, ale dostatočne špecifikujúci názov používateľského príbehu, a *YY.Y* je verzia dokumentu. Zmeny v rámci jedného šprintu inkrementujú číslo za znakom '!'. A pri tvorbe novej verzie alebo rozšírenia verzie v inom šprinte inkrementuje číslo pred znakom '!'.

### Titulná strana

V titulnej strane je potrebné zmeniť *<Téma dokumentu>* na aktuálnu tému príručky a napísať (aj v hlavičke strán) alebo zmeniť (pri doplnení) číslo verzie namiesto *<X.X>*.

### Formát zoznamu skratiek, použitej notácie

Tieto položky sú vo formáte dvojstĺpcových tabuliek kde pre zoznam skratiek platí, že prvý stĺpec sa nazýva „Skratka“ a druhý „Plné znenie“. Pre použitú notáciu platí, že prvý stĺpec sa nazýva „Pojem“ a druhý „Význam“. Pre formátovanie tabuľky pozri kapitolu

### Popis obrazoviek

Popis obrazoviek sa skladá z upraveného obrázka používateľského rozhrania a popisom jednotlivých funkčných častí rozhrania. Obrázok upravíme pomocou nástroja na úpravu obrázkov (mspaint alebo gimp), v ktorom použijeme na zvýraznenie jednotlivých ovládacích prvkov zaoblené obdĺžniky v základných farbách. Pre lepšiu orientáciu môžeme pridať aj číslo k ohraničenej oblasti rovnakou farbou ako je obdĺžnik.

Následne všetky tieto časti popíšeme jednotlivo pričom sa odkazujeme na použitú farbu alebo číslo oblasti, na ktorú sa odkazujeme. Odporúča sa použiť tabuľka.

### Popis častých scenárov

Uviesť presný postup pre každý scenár aplikácie s popisom možných chybových hlásení, ktoré môžu nastať a spôsoby ako dané chyby odstrániť. Používať odkazy na jednotlivé obrázky z kapitoly *Popis obrazoviek*.

### Často sa pýtané otázky

Kapitola by mala obsahovať otázky a odpovede na dané otázky. Tieto otázky sa snažíme identifikovať ešte pred vytvorením dokumentácie. Otázka má štýl *Otázka*. Odpoveď je napísaná štýlom *Telo textu*. Pri revízií sa jednotlivé otázky dopĺňajú.

### Označenie príloh, vzorové prípady

Prílohy sa vkladajú na koniec dokumentu a môžu obsahovať konkrétne príklady pre jednotlivé použitia systému. Označujú sa ako „Príloha X - <názov prílohy>“, kde X značí veľké písmeno abecedy. Štýl je rovnaký ako pre nadpis prvej úrovne, avšak bez čísla. V šablóne je to štýl *Príloha*.

#### 14.2.1. Formát dokumentu

Pre celý dokument platí, že písmo je čiernej farby. Sú zakázané všetky animácie a rotácie textu. Zakázané je aj nastavovať akékoľvek pozadie stránok.

### Formát obsahu

Kostra obsahu je predgenerovaná v šablóne dokumentu. Po skončení úprav nezabudnúť aktualizovať obsah kliknutím pravým tlačidlom a vybratím položky *Update Index/Table*.

### Formát odsekov

Pre odsek textu sa použije štýl zo zoznamu *Paragraph styles* s názvom *Telo textu*.

### Formát nadpisov

- Nadpisu prvej úrovne dokumentu zodpovedá štýl *Heading 1*.

- Nadpisu prvej úrovne dokumentu zodpovedá štýl *Heading 2*.
- Nadpisu prvej úrovne dokumentu zodpovedá štýl *Heading 3*.
- Nadpisu prvej úrovne dokumentu zodpovedá štýl *Heading 4*.
- Nadpisu prílohy dokumentu zodpovedá štýl *Heading 1 - Príloha*.

Ostatné úrovne sa neodporúča používať a ak áno, tak sa pre rovnaké celky použije rovnaká konvencia. Nemôže sa stať, aby sa nadpis pre celok nachádzal na jednej strane samotný bez aspoň riadku z celku. Ak sa to stane, treba nadpis premiestniť na ďalšiu stranu k danému celku.

### Formát číslovania

Pre číslovanie v texte sa použije štýl zo zoznamu *List styles* s názvom *Číslovanie*.

### Formát odrážiek

Na tvorbu odrážiek v texte sa použije štýl zo zoznamu *List styles* s názvom *Odrážky*.

### Formát tabuliek

Tabuľky majú hlavičkový riadok (alebo aj stĺpec) vyznačený 10% šedou farbou pozadia. Obsah buniek má štýl *Obsah tabuľky* (pre sprehl'adnenie je možné zmeniť zarovnanie, ale nie v hlavičke tabuľky). Popis tabuľky sa vytvorí rovnako ako pre obrázok s tým rozdielom, že sa v *Category* vyberie možnosť *Tabuľka*.

### Formát popisu obrázkov, diagramov

Na popis obrázkov a diagramov sa použijú štýly *Obrázok* a *Diagram*. Obrázky a diagramy (obrázok diagramu) sa popisujú nasledovne:

1. Pravým tlačidlom klikneme na objekt a zvolíme *Caption...*
2. Otvorí sa okno a do poľa *Caption* vložíme náš popis.
3. Vyberieme kategóriu z poľa *Category* (*Obrázok* alebo *Diagram*).
4. Stlačíme tlačidlo *OK*.

Každý diagram a aj obrázok musí obsahovať odkaz na súbor (viď nižšie), z ktorého bol vytvorený. Tieto odkazy budú smerovať na súbory (zdrojové súbory diagramov) v priečinku označenom rovnakým názvom ako názov dokumentu, ktorý sa nachádza v priečinku, kde sa nachádza dokument: `<meno_projektu>\Doc\<názov dokumentu>\`.

Odkaz sa vytvorí pravým kliknutím na objekt, zvolením *Picture*, záložka *Hyperlink*, tlačidlo *Browse*, kde si nájdete zdrojový súbor k danému objektu.

### Tvorba diagramov

Na tvorbu diagramov používame nástroj *Software Ideas Modeler*. Riadime sa konvenciou štandardu UML 2.0. Diagram je vhodné exportovať pomocou *Súbor>Exportovať>Obrázok* a zvolíme formát *png* a na záložke *Obsah* v časti *pozadie* necháme zaškrtnuté políčko *Priehl'adné*.

### 14.2.2. Konvencia označovania v texte

V texte sa použijú nasledovné štýly a značky písaného textu podľa významu.

#### Príklad

Príklad musí byť označený nasledovne:

Napíše sa slovo „Príklad:“, na ktorý sa použije štýl *Príklad*. Následne sa popíše príklad. Príklad musí byť stručný čo najviac samo-vysvetľujúci.

#### Položka zodpovedajúca programu

Odkaz na nejakú položku programu (text labelu nejakého EditText poľa) musí byť vyznačené kurzívou od ostatného textu.

#### Spôsob navigácie

Spôsob značenia navigácie v aplikácií rozlišujeme tri základné:

Typ navigácie	Oddeľovací znak	Príklad
Rolovacie menu	>	<i>File &gt; Open</i>
Adresárovej štruktúre	\	<i>MyDocuments\Pictures</i>
Tlačidlá	=	<i>Next = Accept = OK</i>

Tabuľka 14.3: Značenie typov navigácie

#### Odkazovanie na obrázok, diagram, tabuľku

V texte sa odkazujeme pomocou referencie na daný typ objektu cez *Insert>Cross-reference...*. Na záložke *Cross-reference* vyberieme:

- v poli *Type* vyberieme jednu z možností *Tabuľka, Obrázok, Diagram*,
- v poli *Selection* vyberieme objekt na ktorý chceme referencovať,
- a v poli *Insert reference to* vyberieme buď *Numbering* pre iba číslovanie, alebo *Category and number* pre označenie typu objektu s prideleným číslom.

#### Viacero verzií programu

Viacero verzií súčasne, ktoré majú odlišnosti vo funkcionalite, sa tieto uvedú hneď pod odsek, ktorý sa odlišuje tak, že sa napíše štýlom *Verzia* „*Od verzie X.X*“, kde X.X je číslo verzie, kde nastala zmena a do nového odsek sa uvedie zmena štýlom *Telo textu*.

### 14.2.3. Ukončenie tvorby dokumentu

Po ukončení práce na dokumente zmení riešiteľ stav úlohy v Redmine na *resolved* a pošle dokumentáciu na kontrolu manažérovi dokumentácie a produktovému vlastníkovi. Pri schválenej dokumentácii manažér dokumentácie zmení stav úlohy na *resolved* a pošle riešiteľovi informačný e-mail. Ak nie je, tak riešiteľovi pošle e-mail so spísanými nedostatkami a riešiteľ nedostatky odstráni

a znovu pošle na kontrolu.

#### 14.2.4. Odovzdanie dokumentu a zdrojových súborov

Riešiteľ musí odovzdať (ak to doteraz neurobil) aj prípadné zdrojové súbory diagramov a obrázkov (aj upravené aj neupravené), kvôli zmenám v budúcnosti podľa kapitoly a nastaviť referencie objektov. Výsledný dokument nahrá aj na <http://vm28.ucebne.fiit.stuba.sk/Redmine/projects/recommend/documents>.

### 14.3. Dokumentácia k riadeniu

Dokumentáciu k riadeniu tvorí každý člen tímu. Každý je zodpovedný za svoju časť manažmentu projektu a tú vypracuje do dokumentu. Formátovanie jednotlivých častí dokumentu je rovnaké ako pri používateľskej príručke, ale štruktúru dokumentu určuje vytvorená šablóna.

Rola	Zodpovednosť
Manažér dokumentácie	<ul style="list-style-type: none"> <li>• Kontrola dokumentu pred odovzdaním</li> <li>• Vytváranie potrebných štýlov a štruktúry dokumentu</li> </ul>
Ostatní manažéri	<ul style="list-style-type: none"> <li>• Zodpovedajú za vypracovanie dokumentácie k svojim častiam manažmentu a dodržiavanie štýlov</li> </ul>

Tabuľka 14.4: Tabuľka zodpovedností pri tvorbe dokumentácie k riadeniu projektu

#### 14.3.1. Postup úpravy dokumentácie k riadeniu

1. Stiahnuť dokument najnovšej verzie z <http://vm28.ucebne.fiit.stuba.sk/Redmine/documents/1>.
2. Upraviť v ňom svoju časť (ak už bola daná verzia odovzdaná, tak musíte pridať vašu zmenu ako revíziu vo forme prílohy).
3. Doplniť zmeny do tabuľky na konci dokumentu podľa už zadaných zmien
4. Doplniť použité skratky a pojmy použité v texte.
5. Doplniť autorstvo kapitol, ak sa nejaká nová vytvorila.
6. Doplniť autorstvo z dokumentácie k inžinierskemu dielu, ak ste tak neurobili.
7. Zvýšiť verziu dokumentu. (zvyšuje sa číslo za . Teda nová verzia bude mať číslo  $X.<Y+1>$ , kde predošlá verzia mala číslo  $X.Y$ ).
8. Uploadovať dokument späť na server s krátkym popisom.

### 14.4. Dokumentácia k inžinierskemu dielu

Dokumentácia k inžinierskemu dielu popisuje všetky doterajšie šprinty a dokumentuje vývoj diela. Formátovanie jednotlivých častí dokumentu je rovnaké ako pri používateľskej príručke, ale štruktúru dokumentu určuje vytvorená šablóna.



Rola	Zodpovednosť
Manažér dokumentácie	<ul style="list-style-type: none"> <li>• Kontrola dokumentu pred odovzdaním</li> <li>• Vytváranie potrebných štýlov a štruktúry dokumentu</li> </ul>
Vedúci používateľského príbehu	<ul style="list-style-type: none"> <li>• Zodpovedajú za vypracovanie dokumentácie k svojim častiam manažmentu a dodržovanie štýlov</li> </ul>

Tabuľka 14.5: Tabuľka zodpovedností pri tvorbe dokumentácie inžinierskeho diela

#### 14.4.1. Postup tvorby dokumentácie k inžinierskemu dielu

1. Stiahnuť dokument najnovšej verzie z <http://vm28.ucebne.fiit.stuba.sk/Redmine/documents/3>.
2. Upraviť v ňom svoju časť (ak už bola daná verzia odovzdaná, tak musíte pridať vašu zmenu ako revíziu vo forme prílohy).
3. Doplniť zmeny do tabuľky zmien v dokumente k inžinierskemu dielu v dokumentácii riadenia na začiatku dokumentu.
4. Doplniť použité skratky a pojmy použité v texte.
5. Zvýšiť verziu dokumentu. (zvyšuje sa číslo za . Teda nová verzia bude mať číslo  $X.<Y+1>$ , kde predošlá verzia mala číslo  $X.Y$ ).
6. Uploadovať dokument späť na server s krátkym popisom zmien.

#### 14.5. Javadoc

Pri každej vytvorenej funkcii by mal byť Javadoc komentár s popisom metódy a uvedeným autorom danej metódy (pre rýchlejšie mergeovanie súborov).

#### 14.6. Dokumentácia – všeobecné ustanovenia

##### 14.6.1. Spôsob editovania dokumentov

1. Stiahnutie najaktuálnejšej verzie z úložiska dokumentov.
2. Samotné editovanie dokumentu.
3. Ak sa v dokumente nachádza tabuľka so zmenami, tak pripísanie vykonaných zmien.
4. Ak sa v dokumente nachádza tabuľka s autormi jednotlivých častí, tak dopísanie autorstva novej časti.
5. Ak sa medzitým na úložisku objavila nová verzia, tak si ju treba stiahnuť nanovo a prepísať zmeny do novej verzie.
6. Inkrementovať číslo verzie za znakom '!'.  
 6.1. Ak sa objavila nová verzia, tak sa číslo verzie zvyšuje o 1.
7. Ak sa neobjavila nová verzia, tak upload, inak sa pokračuje krokom 5.

##### 14.6.2. Zmena úložiska dokumentov

Kvôli problémom s uploadom súborov väčších ako 5 MB na stránku Redmine-u sa zaviedlo pre dokumenty od verzie 2.0 nové úložisko. Súbory sú uložené v zdieľanom priečinku služby Dropbox

s názvom *Timovy projekt*.

Priečinkok obsahuje niekoľko podpriečinkov.

Názov	Popis
Dokumentacia k inz dielu	Verzie dokumentácie k inžinierskému dielu
Dokumentacia k riadeniu	Verzie dokumentácie k riadeniu
IIT SRC	Verzie k rozšírenému abstraktu IIT SRC
Prezentacia po prvom semestri	Finálna verzia prezentácie na TP Cup po prvom semestri
Priebežna sprava	Priebežná správa na TP Cup v druhom semestri

Pred samotnou úpravou dokumentu je potrebné si ho najprv skopírovať zo zdieľaného priečinka a potom začať úpravy. Ďalej sa postupuje už podľa známeho postupu opísaného v Spôsob editovania dokumentov.

**15. Zmeny v dokumente****Autor:** Michal Cádrik

Zmena	Vykonaná zmena	Platná od verzie	Vykonal	Dátum zmeny
1.	Pridané zápisy zo stretnutí 6 až 8	1.1	M. Cádrik	28.11.2011
2.	Upravená metodika manažmentu dokumentácie	1.1	M. Cádrik	28.11.2011
3.	Pridaná metodika na správu verzií GIT	1.1	M. Cádrik	28.11.2011
4.	Pridaná metodika na prehliadky modelov	1.2	M. Červeňák	04.12.2011
5.	Pridaná metodika na tvorbu testov	1.3	M. Detko	05.12.2011
6.	Pridanie zoznamu skratiek a pojmov	1.4	M. Cádrik	06.12.2011
7.	Pridanie zápisu zo stretnutia	1.4	M. Cádrik	06.12.2011
8.	Oprava štýlov v kapitole Monitorovanie projektu	1.4	M. Cádrik	06.12.2011
9.	Oprava chýb v kapitole Plán projektu	1.4	M. Cádrik	06.12.2011
10.	Pridané chýbajúce štýly podľa metodiky	1.4	M. Cádrik	06.12.2011
11.	Upravené pokazené štýly strán	1.5	M. Cádrik	09.12.2011
12.	Zmena dokumentu na lepšie pridávanie nových častí	1.6	M. Cádrik	10.12.2011
13.	Doplnenie a úprava metodík – Monitorovanie projektu	1.7	M. Červeňák	10.12.2011
14.	Doplnenie tabuľky autorov častí v dokumentácií k inžinierskemu dielu a informáciou k Javadoc	1.8	M. Cádrik	10.12.2011
15.	Doplnenie podkapitoly Manažment požiadaviek na zmenu do kapitoly Manažment komunikácie	1.9	J. Ševcech	12.12.2011
16.	Pridaný preberací protokol k odovzdávaniu dokumentácie k riadeniu projektu po prvom semestri	1.10	J. Ševcech	12.12.2011
17.	Výpis autorov pri jednotlivých častiach dokumentácie. Údaje z Úvodnej časti dokumentácie.	1.11	J. Ševcech	12.12.2011
18.	Doplnenie častí do podkapitol Nástroje na podporu komunikácie a Nástroje na podporu kolaboráciu v kapitole Manažment komunikácie.	1.12	J. Ševcech	12.12.2011
19.	Vloženie desiateho zápisu zo stretnutia a	1.13	M. Cádrik	12.12.2011

	pridanie prihlášky na TP Cup do príloh			
20.	Doplnená metodika rozvrhu	1.14	M. Mihalik	12.12.2011
21.	Pridané plány jednotlivých šprintov	1.14	M. Mihalik	12.12.2011
22.	Upravené zhodnotenie plánovania	1.14	M. Mihalik	12.12.2011
23.	Rozšírený product backlog	1.14	M. Mihalik	12.12.2011
24.	Pridané názvy šprintov k plánov ak ich začiatkom	1.14	M. Mihalik	12.12.2011
25.	Pridané grafické plány jednotlivých šprintov + celkový plán projektu	1.14	M. Mihalik	12.12.2011
26.	Upravený diagram celkového plánu	1.15	M. Mihalik	12.12.2011
27.	Oprava chybných štýlov, pravopisu, zlých tvarov slov, formátovania	1.16	M. Cádrik	13.12.2011
28.	Pridaný preberací protokol pre vedúceho tímu	1.16	M. Cádrik	13.12.2011
29.	Pridaná kapitola Manažment rizík	1.17	Ľ. Mydla	13.12.2011
30.	Oprava chýb a formátovania v kapitole Manažment rizík	1.18	M. Cádrik	13.12.2011
31.	Všeobecné ustanovenia písania dokumentácie pre letný semester	2.1	M. Cádrik	23.03.2012
32.	Pridané zápisy zo stretnutí 11 - 17 , pridany nový preberací protokol pre 3. odovzdanie	2.2	M. Cádrik	31.03.2012
33.	Oprava číslovania strany pre nový preberací protokol	2.3	M. Cádrik	31.03.2012
34.	Zmeny v úlohách členov tímu. Kapitola 3.	2.4	J.Ševcech	06.04.2012
35.	Pridania informácií o zmenách v testovaní	2.6	M. Detko	07.04.2012
36.	Aktualizácia plánu projektu, kapitola 5.	2.7	M. Mihalik	07.04.2012
37.	Doplnenie podkapitoly Manažment rizík v druhom semestri.	2.8	Ľ. Mydla	07.04.2012
38.	Doplnenie autorstva do tabuliek k zápisom zo stretnutí	2.9	M. Cádrik	09.04.2012
39.	Doplnenie autorstva k dokumentácií k inžinierskemu dielu	2.11	M. Mihalik	09.04.2012
40.	Doplnenie autorstva k dokumentácií k inžinierskemu dielu	2.12	M. Detko	09.04.2012

Tabuľka 15.1 Zoznam vykonaných zmien

## **Príloha 1 - Preberacie protokoly**

**Autor:** Jakub Ševcech

# **Preberací protokol**

*Projekt*

Personalizované odporúčanie

*Produkt*

Priebežný prototyp odporúčacieho systému

*Predmet odovzdania*

Tlačená dokumentácia k inžinierskemu dielu v rozsahu 33 strán

Tlačená dokumentácia k riadeniu projektu v rozsahu 41 strán

Zdrojové kódy systému

*Odovzdávajúci tím*

7 out of 10 precision (tím č. 12)

*Preberajúci*

Ing. Michal Kompan (pedagóg)

*Dátum odovzdania*

10.11.2011

---

**Ing. Michal Kompan**  
(pedagóg)

---

**Jakub Ševcech**  
(tím č.12)

## **Príloha 2 - Preberacie protokoly**

**Autor:** Jakub Ševcech

# **Preberací protokol**

*Projekt*

Personalizované odporúčanie

*Produkt*

Dokumentácia k riadeniu projektu po prvom semestri

*Predmet odovzdania*

Tlačená dokumentácia k riadeniu projektu v rozsahu 133 strán

*Odovzdávajúci tím*

7 out of 10 precision (tím č. 12)

*Preberajúci*

Ing. Andrej Danko, PhD. (pedagóg)

*Dátum odovzdania*

13.12.2011

---

**Ing. Andrej Danko, PhD.**  
(pedagóg)

---

**Jakub Ševcech**  
(tím č.12)

## **Príloha 3 - Preberacie protokoly**

**Autor:** Michal Cádrik

# **Preberací protokol**

*Projekt*

Personalizované odporúčanie

*Produkt*

Priebežný prototyp odporúčacieho systému

*Predmet odovzdania*

Tlačená dokumentácia k inžinierskemu dielu v rozsahu 63 strán

Tlačená dokumentácia k riadeniu projektu v rozsahu 133 strán

Zdrojové kódy systému

*Odovzdávajúci tím*

7 out of 10 precision (tím č. 12)

*Preberajúci*

Ing. Michal Kompan (pedagóg)

*Dátum odovzdania*

13.12.2011

---

**Ing. Michal Kompan**  
(pedagóg)

---

**Jakub Ševcech**  
(tím č.12)

## Príloha 4 - Prihláška na TP Cup

**Autor:** Jakub Ševcech, Michal Cádrik

### Téma: Personalizované odporúčanie

Názov tímu: 7 out of 10 Precision  
Vedúci tímu: Ing. Michal Kompan  
Kontakt: tim12tp@googlegroups.com

### Predstavenie tímu

**Bc. Michal Cádrik:** (Manažér dokumentácie)

Programuje v jazyku Java (IDE Eclipse), C/C++ (IDE Visual Studio, Qt). Bc. Michal Cádrik má skúsenosti s používaním operačných systémov Windows a Ubuntu. Javadocs používa ako jeden z nástrojov na vytváranie dokumentácie. V bakalárskej práci sa venoval riešeniu hry Sudoku pomocou logických pravidiel.

**Bc. Martin Detko:** (Manažér kvality)

Ovláda programovacie jazyk C/C++ a Java. S týmito jazykmi robí v prostrediach Eclipse a Visual Studio. Zadaním jeho bakalárskej práce bolo vytvoriť inteligentného agenta do prostredia TORCS, čo hovorí o jeho záujme o umelú inteligenciu, na ktorú zameriava aj svoje ďalšie štúdium na fakulte.

**Bc. Ľudovít Mydla:** (Manažér rizík)

Používa najmä jazyk Java a prostredie Eclipse. Používa tiež C/C++, Ruby, MSSQL, ObjectiveC (v IDE Xcode) pri vývoji aplikácie pre iPhone, ale momentálne pracuje s PHP + MySQL + JS + CSS pri vývoji webových stránok s použitím CMS systému WordPress alebo MVC frameworku CodeIgniter. Má skúsenosť s SVN a Git ako systémami na verziovanie, javadocom a systémom na sledovanie chýb Bugzilla. Momentálne sa zaujíma o vývoj pre mobilné platformy.

**Bc. Igor Slotík:** (Manažér podpory vývoja)

Má skúsenosti s programovaním v jazyku Java a C. Bakalárska práca Bc. Igor Slotíka má názov Automatická vizuálna detekcia inteligentnou kamerou. Získal hlboké skúsenosti z rôznych prístupov na detekciu ľudí, farebných modelov, ktoré sa používajú na detekciu ľudí a AdaBoost klasifikátora. Svoje riešenie implementoval pre inteligentnú kameru a pre kameru notebooku.

**Bc. Jakub Ševcech:** (Manažér komunikácie)

Má skúsenosti s programovacími jazykmi Ruby, Java a C/C++. Používa systém Redmine, ako systém na spravovanie projektov a Git ako nástroj na správu verzií. Má skúsenosti s test-driven-development a testovacími frameworkami Rspec a Cucumber. Ako bakalársku prácu navrhol a realizoval metódu na pridávanie poznámok do webových stránok.

**Bc. Matej Mihalik:** (Manažér rozvrhu a plánovania)

Má skúsenosti z prostredia tvorby webových stránok a aplikácií, kde používa technológie PHP+SQL (PostgreSQL)+CSS+JavaScript+formát SVG s využitím frameworku CodeIgniter. Skúsenosti má s programovaním v jazyku Java (prostredia Eclipse a NetBeans) a s jazykmi C/C++



(VisualStudio, CodeBlocks). Bakalársku prácu vypracoval na tému Podpora vyhodnocovania webových dotazníkov.

**Bc. Matej Červeňák:** (Manažér monitorovania)

Má skúsenosti z návrhu informačného systému, s prácou v jazyku UML a s nástrojom CASE studio. Ďalej má znalosti jazykov Java (prostredie NetBeans) a MySQL. Na výbornej úrovni ovláda grafické programy Photoshop, Illustrator a Flash. V bakalárskej práci sa sa zamerával práve na túto problematiku, napodobňovanie fotografických filmových materiálov pomocou grafických editorov.

### **Motivácia**

Množstvo webových stránok a dokumentov robia web neprehľadným. Ľudia častokrát spotrebujú veľa času na nájdenie relevantných dokumentov, o ktoré majú záujem. Pomoc pri hľadaní zaujímavých článkov a šetrenie času, čím prispievame ku redukovaniu informačného zahltenia používateľov, poskytne navrhované riešenie.

Viacerí členovia tímu sa zaujímajú o témy strojového učenia a veríme, že pri implementácii metód odporúčania využijeme ich záujem o podobné algoritmy. Takisto máme skúsenosť s prácou na skrytých Markovových modeloch, ktoré sú využívané pri niektorých metódach na odporúčanie. Máme základnú znalosť problémovej oblasti, na ktorej môžeme ďalej stavať a radi si ju rozšírime. Niektorí z členov tímu pracovali na príbuzných témach vo svojich bakalárskych prácach a chceli by ich rozvíjať ďalej. Tento projekt je na to vhodná príležitosť.

Motivuje nás hlavne:

1. Výzva, ako zladiť požiadavky od množstva ľudí na fakulte, ktorí pracujú s metódami na odporúčanie.
2. Výzva, ako spracovať veľké objemy údajov a poskytovať odporúčanie pre veľké množstvo používateľov v reálnom čase.
3. Naučenie sa niečoho, čo vieme využiť aj mimo projektu, napríklad regulárne výrazy, metódy na spracovanie dokumentov, metódy odporúčania, rôzne štatistické metódy na vyhodnocovanie úspešnosti odporúčania a podobne.
4. Chceme pracovať na zlepšení kvality vyhľadávania a odporúčania, a tak pomôcť pri riešení súčasného zahltenia informáciami.
5. Vytvoríme aplikáciu, ktorá sa má šancu reálne používať pri výskume na fakulte a môže tak pomôcť k zlepšeniu výsledkov študentov a fakulty samotnej.

### **Koncepcia riešenia**

Vytvoríme webovú službu na získavanie odporúčania pre rôznych používateľov. V tejto službe sprístupníme sadu najpoužívanejších algoritmov na kolaboratívne odporúčanie a odporúčanie na základe obsahu. Riešenie je implementované ako REST webová služba v jazyku JRuby. Samotné algoritmy odporúčania a hlavná logika spojená s odporúčaním je implementovaná v jazyku Java s ohľadom na požiadavky na výkon riešenia.

Vytvoríme modulárnu architektúru tak, aby bolo možné jednoducho pridávať nové metódy odporúčania do existujúcej služby v podobe pluginov. Vytvoríme jadro aplikácie, ktoré bude poskytovať najzákladnejšie služby ako správu pluginov alebo prístup k databáze. Toto jadro sprístupníme pre používateľov, ktorí si chcú vytvoriť vlastné metódy odporúčania a budú ho môcť použiť na prepojenie s aplikáciou. Všetky metódy odporúčania ako aj rôzne podporné metódy,

implementujeme vo forme spomínaných pluginov. Medzi takéto podporné metódy patrí napríklad výpočet podobnosti. Takto sa bude dať knižnica jednoducho rozširovať a bude možné držať krok s vývojom v oblasti odporúčania.

Vytvoríme jednoduchý spôsob, ako nakonfigurovať odporúčací systém z existujúcich metód na odporúčanie implementovaných v systéme. Pri vytváraní konfigurácií bude možné kombinovať rôzne metódy odporúčania s rôznymi podpornými metódami. Rovnako bude možné kombinovať aj samotné metódy odporúčania.

Vytvorené konfigurácie bude môcť používateľ striedať a porovnávať tak ich úspešnosť. Pre podporu vyhodnocovania úspešnosti metód odporúčania poskytneme rozhranie na zaznamenávanie spätnej väzby k poskytnutému odporúčaniam. Na základe tejto spätnej väzby budeme zobrazovať štatistiky o úspešnosti odporúčania pre konkrétnu konfiguráciu na základe viacerých metrík.

Vytvoríme všeobecný dátový model na reprezentáciu používateľov, ktorým bude poskytované odporúčanie, ako aj na reprezentáciu rôznych druhov odporúčaných dokumentov. Vytvoríme jednoduché rozhranie na importovanie údajov, na základe ktorých sa bude vytvárať odporúčanie. Rozhranie na interakciu s aplikáciou bude rozdelené na dve časti: grafické prostredie v podobe webovej služby, ktoré bude slúžiť napríklad na podporu správy konfigurácií alebo importovanie údajov a REST API, ktoré bude využívať XML súbory na komunikáciu.

Všetky informácie sa budú ukladať v PostgreSQL databáze. Na prístup k týmto údajom z jazyka Java použijeme objektové mapovanie pomocou nástroja Hibernate. Pre prístup z jazyka JRuby použijeme objektové mapovanie ActiveRecord. Budeme nasledovať metodológiu vývoja riadeného testami (TDD), pričom pre Javu použijeme testovací framework JUnit a pre JRuby to bude kombinácia frameworku Rspec a knižnice Capybara.

## Príloha 5 - Preberací protokol pre priebežné odovzdanie v 2. semestri

**Autor:** Michal Cádrik

### Preberací protokol

*Projekt*

Personalizované odporúčanie

*Produkt*

Priebežný prototyp odporúčacieho systému

*Predmet odovzdania*

Tlačená dokumentácia k inžinierskemu dielu v rozsahu 126 strán

Tlačená dokumentácia k riadeniu projektu v rozsahu 163 strán

Zdrojové kódy systému

Funkčný prototyp systému

*Odovzdávajúci tím*

7 out of 10 precision (tím č. 12)

*Preberajúci*

Ing. Michal Kompan (pedagóg)

*Dátum odovzdania*

12.04.2011

---

**Ing. Michal Kompan**  
(pedagóg)

---

**Jakub Ševcech**  
(tím č.12)