

**Dokumentácia k riadeniu projektu**  
*Tvorba "Pahko" sémantického obsahu pre adaptívny  
webový (výučbový) portál*

Tímový projekt

tím č. 13 – Thirteam

*Thirteam*

Bc. Martin Franta  
Bc. Martin Gajdoš  
Bc. Martin Habdák  
Bc. Róbert Kocian  
Bc. Petra Vrablecová  
Bc. Zuzana Zimová

Vedúci pedagóg: Ing. Marián Šimko, PhD.

e-mail: [tim13@googlegroups.com](mailto:tim13@googlegroups.com)

Ak. rok : 2011/2012



# Obsah

---

<b>1</b>	<b>Úvod .....</b>	<b>1-1</b>
<b>2</b>	<b>Predstavenie tímu .....</b>	<b>2-1</b>
2.1	Martin Franta .....	2-1
2.2	Martin Gajdoš .....	2-1
2.3	Martin Habdák .....	2-1
2.4	Róbert Kocian .....	2-1
2.5	Petra Vrablecová .....	2-2
2.6	Zuzana Zimová .....	2-2
2.7	Úlohy členov tímu .....	2-2
<b>3</b>	<b>Manažment kvality .....</b>	<b>3-1</b>
3.1.1	<i>Odlahčovanie kontrolerov .....</i>	<i>3-1</i>
3.1.2	<i>Prispôsobenie Javascriptu .....</i>	<i>3-1</i>
<b>4</b>	<b>Manažment rizík .....</b>	<b>4-1</b>
<b>5</b>	<b>Manažment rozvrhu .....</b>	<b>5-1</b>
5.1	Plán projektu na letný semester .....	5-1
<b>6</b>	<b>Manažment podpory vývoja .....</b>	<b>6-1</b>
<b>7</b>	<b>Monitorovanie projektu .....</b>	<b>7-1</b>
7.1	Výber nástroja .....	7-1
7.2	Metriky .....	7-1
7.2.1	<i>Výber metrík .....</i>	<i>7-1</i>
7.2.2	<i>Popis metrík .....</i>	<i>7-1</i>
<b>8</b>	<b>Manažment komunikácie .....</b>	<b>8-1</b>
8.1	Osobné stretnutia .....	8-1
8.2	Facebook .....	8-2
8.3	Google Groups .....	8-3
8.4	Telefónny kontakt .....	8-3
8.5	Redmine .....	8-3
<b>9</b>	<b>Manažment tvorby dokumentácie .....</b>	<b>9-1</b>
9.1	Dokumentácia k riadeniu a inžinierskemu dielu .....	9-1
9.2	Zápisy zo stretnutí .....	9-1

9.3	Preberacie protokoly .....	9-1
9.4	Webová stránka tímu .....	9-1
<b>Príloha A: Metodika štruktúry zdrojových kódov .....</b>		<b>A-1</b>
A.1	Názvy premenných .....	A-1
A.2	Názvy funkcií.....	A-1
A.3	Odsadzovanie.....	A-2
A.4	Komentovanie zdrojového kódu .....	A-2
<b>Príloha B: Metodika pre prácu s nástrojom Git.....</b>		<b>B-1</b>
B.1	Komentáre pri ukladaní novej verzie kódu .....	B-1
B.2	Vývoj vo vetvách .....	B-1
<i>B.2.1</i>	<i>Pomenovávanie vetiev .....</i>	<i>B-1</i>
<i>B.2.2</i>	<i>Spájanie vetiev.....</i>	<i>B-1</i>
<b>Príloha C: Zápisy zo stretnutí .....</b>		<b>C-1</b>
	Zápisnica č. 24 .....	C-3
	Zápisnica č. 25 .....	C-5
	Zápisnica č. 26 .....	C-7
	Zápisnica č. 27 .....	C-9
	Zápisnica č. 28 .....	C-11
	Zápisnica č. 29 .....	C-13
<b>Príloha D: Preberacie protokoly .....</b>		<b>D-1</b>
	Preberací protokol.....	D-3

# 1 Úvod

---

Tento dokument opisuje postupy riadenia projektu vypracovávaného tímom číslo 13 s názvom *Thirteam* na predmete *Tímový projekt*. Tím sa skladá z absolventov bakalárskeho štúdia: Martin Franta, Martin Gajdoš, Martin Habdák, Róbert Kocian, Petra Vrablecová, Zuzana Zimová. Pedagogickým vedúcim tímu je Ing. Marián Šimko.

V rámci dokumentu boli vypracované nasledujúce časti jednotlivými členmi tímu:

---

Úvod	Bc. Petra Vrablecová
Predstavenie tímu	celý tím
Manažment kvality	Bc. Róbert Kocian
Manažment rizík	Bc. Martin Franta
Manažment rozvrhu	Bc. Zuzana Zimová
Manažment podpory vývoja	Bc. Petra Vrablecová
Monitorovanie projektu	Bc. Martin Habdák
Manažment komunikácie	Bc. Martin Gajdoš
Manažment tvorby dokumentácie	Bc. Petra Vrablecová
<b>Prílohy</b>	
Metodika štruktúry zdrojových kódov	Bc. Martin Franta
Metodika pre prácu s nástrojom Git	Bc. Petra Vrablecová
Zápisy zo stretnutí	autori jednotlivých zápisníc uvedení v prílohe
Preberacie protokoly	Bc. Petra Vrablecová

---

Za finálny vzhľad dokumentácie je zodpovedný manažér dokumentácie – Bc. Petra Vrablecová.



## 2 Predstavenie tímu

---

Náš tím je vedený skvelým vedúcim Ing. Mariánom Šimkom. Pozostáva zo šiestich členov, z ktorých každý má jedinečné a obsiahle znalosti v oblasti informačných technológií. Okrem základných znalostí viacerých programovacích jazykov a skúseností s vyvíjaním webových aplikácií či prácou s databázou, má tím vďaka niekoľkoročným skúsenostiam, nadobudnutých počas štúdia, všetky potrebné predpoklady pre rýchle učenie sa nových zručností. Ďalšou prednosťou nášho tímu je schopnosť efektívne spolupracovať a komunikovať.

### 2.1 Martin Franta

Absolvent bakalárskeho štúdia na FIIT STU, kde väčšinu zadaní vypracovával v jazykoch C a Java. Rád spoznáva nové technológie a rozširuje svoje znalosti, preto si pre svoj bakalársky projekt „Editor dopravných udalostí v prostredí webovej mapy“ zvolil jazyk Python a databázu PostgreSQL. Má niekoľkoročné skúsenosti s vývojom webových aplikácií, ktoré sú reálne nasadené a denne používané stovkami ľudí. Vo svojom súčasnom zamestnaní sa venuje vývoju aplikácii v PHP - ZendFrameworku, v Jave a JavaScripte.

### 2.2 Martin Gajdoš

Počas svojich štúdií sa venoval rôznym informačným technológiám pre tvorbu dynamických web stránok, užitočných desktopových aplikácií ako aj práci s databázami. Približne v strede svojho prvostupňového vysokoškolského štúdia sa rozhodol, že svoju pozornosť zameria na programovací jazyk Java. V ňom vypracoval bakalársku prácu, ktorá sa zaoberala navigáciou a grafovými algoritmi. Počas posledného ročníka bakalárskeho štúdia sa mu taktiež podarilo zamestnať sa vo firme venujúcej sa informačným technológiám, kde svoje znalosti ďalej rozširuje.

### 2.3 Martin Habdák

Absolvoval bakalárske štúdium na FIIT STU v odbore Informatika. Téma jeho bakalárskej práce bola „Vizualizácia SQL príkazov“. V jej rámci vytvoril systém, ktorý umožňuje zobrazovať medzivýsledky pri vykonávaní SQL príkazov, a tak uľahčil ich pochopenie. Momentálne pokračuje v inžinierskom štúdiu v odbore Softvérové inžinierstvo. Má skúsenosti s programovacími jazykmi C/C++, C#, PHP, Java, JavaScript. Pracuje aj v oblasti databáz s technológiami MySQL a s jazykmi patriacimi do SGML. Vo voľnom čase sa rád venuje spoznávaniu nových technológií a programovaniu hier na PC a XBox 360 v prostredí MS XNA Framework.

### 2.4 Róbert Kocian

Bakalárske štúdium absolvoval v Českej republike. V priebehu tohto štúdia sa priebežne venoval jazykom ako sú JAVA, C/C++, ABAP, ASP.NET, C#. V poslednom čase sa zaoberá optimalizáciou internetových stránok a vývojom WEB 2.0 aplikácií v jazykoch PHP, MySQL, JavaScript, (X)HTML, CSS. Zaujíma sa o realtime programovanie webových aplikácií založených na technológii AJAX a o interakciu medzi HTML a JavaScriptom pomocou frameworku jQuery.

### 2.5 Petra Vrabecová

Absolventka bakalárskeho študijného programu Informatika na FIIT STU. Počas štúdia sa stretla s viacerými programovacími jazykmi (napr. C, VBA, SQL), ale najväčšiu pozornosť venovala jazyku Java, ktorý využila aj vo svojej bakalárskej práci „Testovanie dynamických aspektov nástrojov na kompozíciu webových služieb“. Momentálne pokračuje v štúdiu na FIIT STU, odbor Softvérové inžinierstvo, kde zbiera nové vedomosti a skúsenosti.

### 2.6 Zuzana Zimová

Absolventka bakalárskeho štúdia na FIIT STU v odbore Informatika, kde počas svojho štúdia rozvíjala najmä znalosti jazykov C a Java. Okrem týchto dvoch jazykov využila vo svojej bakalárskej práci s názvom „Porovnanie vhodnosti paradigiem pre problém z umelej inteligencie“ i jazyky Prolog a Lisp. Momentálne je študentkou prvého ročníka inžinierskeho štúdia na FIIT STU v odbore Informačné systémy.

### 2.7 Úlohy členov tímu

Úlohy členov tímu sa v letnom semestri nezmenili.

Bc. Martin Franta	Manažér vývoja, Manažér rizík
Bc. Martin Gajdoš	Manažér tímu, Manažér komunikácie
Bc. Martin Habdák	Manažér monitorovania, Manažér testovania
Bc. Róbert Kocian	Manažér kvality
Bc. Petra Vrabecová	Manažér podpory vývoja, Manažér dokumentovania
Bc. Zuzana Zimová	Manažér plánovania



## **3 Manažment kvality**

---

### **3.1.1 Odľahčovanie kontrolerov**

Snažili sme sa upraviť kontrolery tak, aby obsahovali čo najmenej kódu. Kód, ktorý sme odstránili obsahoval SQL jazyk, ale aj jazyk ruby. Dôvodom odstránenia bolo zachovanie konvencií jazyka Ruby on rails z pohľadu dodržania architektúry MVC tak, aby bolo čo najviac kódu smerovaného na databázu v modeloch a nie v kontroleroch.

### **3.1.2 Prispôbenie Javascriptu**

Zo začiatku nebol kód napísaný v JavaScripte problémom, pretože aj keď bol neprehľadný, stále bolo možné ustrážiť a opraviť chyby, či neprehľadnosť kódu. Keď ale začalo tohto kódu pribúdať, bol s tým veľký problém, a preto sme sa rozhodli implementovať do systému Coffee script, ktorý nielenže zjednodušuje a zrýchľuje prácu s písaním kódu, ale tento kód sa stáva prehľadnejším, pretože inak by sa kód nespustil a nahlásil by chybu.



## 4 Manažment rizík

---

Kvôli rizikám súvisiacim s nasadením projektu na server a správou GIT repozitára, bola spustená kontinuálna integrácia projektu.

Riziká spočívali najmä v chybnom kóde, ktorý mohol byť do GIT repozitára nedopatrením odovzdaný a následne spustený na serveri. Po viacerých prípadoch znefunkčnenia verzie projektu na serveri sme pristúpili k automatickému vykonávaniu testov priamo na serveri.

Bola vytvorená nová vývojová vetva v GIT repozitári – staging. Vetva master už obsahuje iba stabilnú a overenú verziu projektu, ktorá je na serveri spustená v produkčnom režime.

Na serveri bol vytvorený testovací projekt, v ktorom sa spúšťajú automatické testy. Do ďalšieho kroku integrácie je možné postúpiť iba po úspešnom prebehnutí testov. Testovací projekt tiež slúži na overenie správneho zlúčenia staging vetvy do lokálneho repozitára na serveri.

Ak testovanie dopadlo úspešne, zmeny môžu byť zanesené aj do hlavného vývojového projektu na serveri, ku ktorému je možné pristupovať cez URL adresu.

Takto navrhnutý postup vyžaduje existenciu testov, ktoré pokrývajú kľúčovú funkcionálnosť, aby sa zabránilo vyradeniu už implementovaných funkcií.



## 5 Manažment rozvrhu

V rámci manažmentu plánovania bol zostavený celkový plán na letný semester, ktorý bol počas semestra bližšie špecifikovaný na tímových stretnutiach, kde sa plánovali jednotlivé šprinty.

Na tímovom stretnutí boli jednotlivým členom pridelené úlohy. Jedna úloha mohla byť vypracovaná viacerými členmi tímu, ale mala len jednu zodpovednú osobu, ktorá bola zodpovedná za dokončenie úlohy v danom šprinte. Tím používa pre plánovanie a sledovanie úloh systém Redmine. Beží na školskom serveri *redmine.fiit.stuba.sk*.

### 5.1 Plán projektu na letný semester

Tím zostavil nasledujúci plán projektu na letný semester:

týždeň	Úlohy
0.	20.12.2011– 10.2.2012 5. šprint 13.2.2012 Začiatok 6. šprintu
1.	<b>15.2.2012</b> TP Cup: Odovzdanie abstraktu príspevku na IIT.SRC Zhodnotenie výsledkov zimného semestra, doplnenie a dopracovanie zistených nedostatkov
2.	<b>22.2.2012</b> TP Cup: Odovzdanie priebežnej správy o riešení projektu, vrátane abstraktu na IIT.SRC Zakomponovanie zmien do dokumentácie zimného semestra Návrh, plán integrácie, plán overenia výsledku
3.	27.2.2012 Začiatok 6. šprintu Implementácia, integrácia a overovanie
4.	Implementácia, integrácia a overovanie
5.	16.3.2012 Začiatok 7. šprintu Implementácia, integrácia a overovanie
6.	Implementácia, integrácia a overovanie
7.	30.3.2012 Začiatok 8. šprintu Implementácia, integrácia a overovanie
8.	Integrácia, overovanie, dokumentovanie projektu
9.	Odovzdanie produktu a dokumentácie k produktu
10.	Prevádzka, externé testovanie, kompletizácia dokumentácie, údržba

11.	<b>25.4.2012</b>	TP Cup: Ukážka projektu v rámci študentskej vedeckej konferencie  Prevádzka, externé testovanie, kompletizácia dokumentácie, údržba
12.		Odobranie celkového výsledku projektu (produkt so zmenami v rámci údržby, dokumentácia, dokumentácia k riadeniu)
	máj 2012	Losovanie poradia v semifinále
	jún 2012	Prezentácia a demonštrácia výsledkov projektu

## 6 Manažment podpory vývoja

---

V letnom semestri sme pokračovali v používaní podporných prostriedkov zvolených v predchádzajúcom semestri (Tab. 1).

**Tab. 1.** Používané podporné prostriedky.

Verziovanie zdrojového kódu	<i>Git</i>
Manažment úloh	<i>Redmine</i>
Kontinuálna integrácia	<i>Jenkins</i>
Zdieľanie dokumentov	<i>Google Docs</i>
Komunikácia	<i>Facebook, Google Groups, IM, telefón</i>

Vyvíjaný systém je počas vývoja neustále spustený na virtuálnom stroji tímu č. 13 (`team13-11.ucebne.fiit.stuba.sk:443`). Jeho kontinuálne integrácia je zabezpečovaná nástrojom Jenkins.

Pri verziovaní zdrojového kódu pomocou nástroja Git sme sa riadili metodikou z predchádzajúceho semestra (Príloha B). Kvôli neustálym problémom s integrovaním všetkých nových funkcií do systému pribudla v rámci vývoja tzv. *staging* vetva, v ktorej tím spája všetky vyvíjané funkcionality pred odovzdaním. Finálna stabilná verzia sa nachádza v *master* vetve.

Na manažment úloh je naďalej úspešne využívaný nástroj Redmine, ktorý je prepojený s úložiskom zdrojového kódu.

Využívanie prostriedkov pre zdieľanie dokumentov a komunikáciu je bližšie opísané v príslušných kapitolách (kap. 8 - Manažment komunikácie, kap. 9 - Manažment tvorby dokumentácie).





## 7 Monitorovanie projektu

---

### 7.1 Výber nástroja

Manažment úloh sa vykonáva pomocou nástroja Redmine. Keďže poskytuje aj funkcionality na monitorovanie projektu, rozhodli sme sa použiť ho na tento účel. Zbiera dáta vkladané pri práci s úlohami a umožňuje ich reprezentáciu pomocou grafov. Medzi ne patrí aj *Ganttov graf* a *Burndown with velocity*, používaný najmä pri metodike SCRUM, ktorú náš tím aplikuje pri vývoji.

### 7.2 Metriky

#### 7.2.1 Výber metrík

Pre monitorovanie sa tím rozhodol používať aspoň niektoré zo základných metrík odporúčaných inštitútom SEI<sup>1</sup>. Z nich boli vybrané:

- počet odpracovaných hodín,
- počet chýb.

Ďalšie metriky získavané nástrojom Redmine:

- počet nedokončených úloh (*New, In progress*),
- počet dokončených úloh,
- počet zostávajúcich hodín.

Tieto údaje tím nepovažuje za úplne postačujúce. Preto sa z nich vypočítavajú zložitejšie ukazovatele. Osvedčujú sa najmä pri plánovaní ďalšieho šprintu a hodnotení kvality projektu. Medzi ne sme zaradili:

- index produktivity – *feature points*<sup>2</sup>/ *počet reportovaných hodín* (jednotlivca a celku),
- index kvality.

#### 7.2.2 Popis metrík

##### Počet odpracovaných hodín

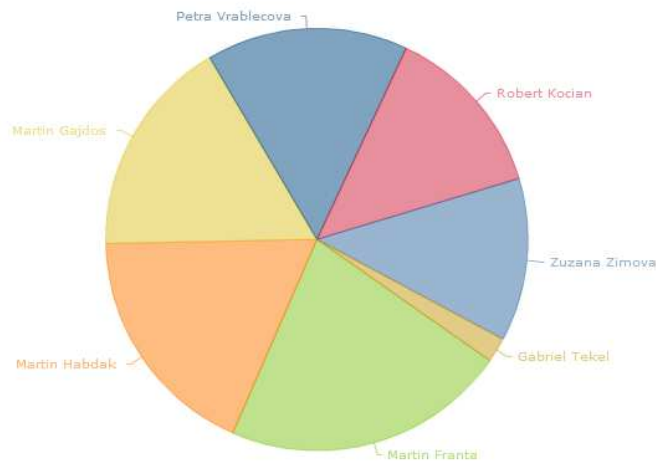
Sledovanie odpracovaných hodín poskytuje základný náhľad na podiel práce na projekte pre jednotlivých členov tímu. Získava sa pomocou nástroja Redmine. Členovia pri práci v ňom evidujú počet odpracovaných hodín. Následne možno v grafe sledovať nazbierané hodnoty. Takýto graf vidieť na obr. 5.

---

<sup>1</sup> <http://www.sei.cmu.edu/library/abstracts/reports/92tr019.cfm>

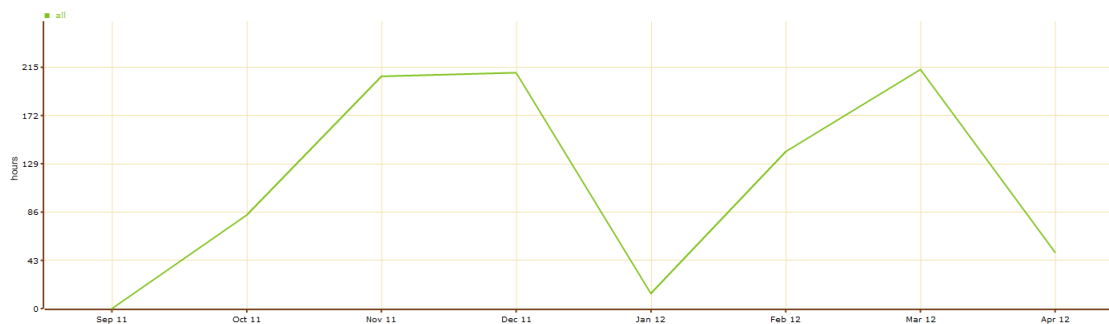
<sup>2</sup> body, ktorými je ohodnotená daná feature v metodike SCRUM

Sledovanie odpracovaných hodín nie je vždy presnou metrikou. Preto tím používa túto hodnotu na výpočet indexu produktivity. Môžu totiž nastať prípady, keď člen s menším počtom odpracovaných hodín vykonal v skutočnosti viac práce na projekte.



**Obr. 1.** Počet odpracovaných hodín na projekte.

Pre sledovanie práce na projekte sa používa aj graf odpracovaných hodín v závislosti od času. Možno ho vidieť na obr. 6. Zobrazuje obdobia, v ktorých sa najviac pracuje na projekte. Riadi sa podľa neho manažment plánovania pri vytváraní rozvrhu úloh. Poznatky z tohto grafu sa aplikujú aj v manažmente rizík.



**Obr. 2.** Počet odpracovaných hodín v závislosti od času.

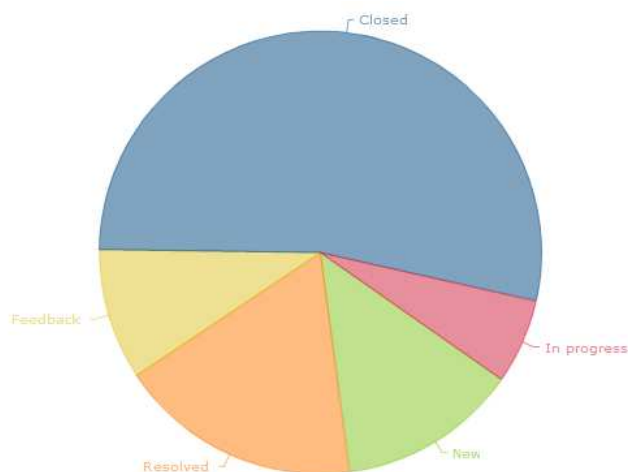
### Počet chýb

Hodnota sa určuje podľa množstva problémov v nástroji Redmine, ktoré sú označené ako *Bug*. Vyjadruje počet chýb, ktoré sa vyskytli v projekte. Hlavným cieľom tejto hodnoty je jej použitie vo vzťahu na výpočet indexu kvality. Počet chýb treba udržiavať na najnižšej úrovni. V ideálnom prípade je nulový. Pri výskyte veľkého množstva chýb zasiahne manažér kvality do procesu vývoja, aby odstránil zdroj tohto problému.

### Počet nedokončených úloh

Metrika meria počet úloh, ktoré neboli dokončené. Získané poznatky sa spracovávajú v manažmente plánovania, najmä pri vytváraní rozvrhu a definovaní nových úloh. Medzi nedokončené úlohy sa zaraďujú tie, ktoré na konci šprintu nie sú označené ako *Resolved*

alebo *Closed*. Potrebné údaje sa získavajú nástrojom Redmine. Na obr. 7 vidieť ich znázornenie grafom. Meranie počtu nedokončených úloh slúži aj na včasnú identifikáciu rizík v projekte.



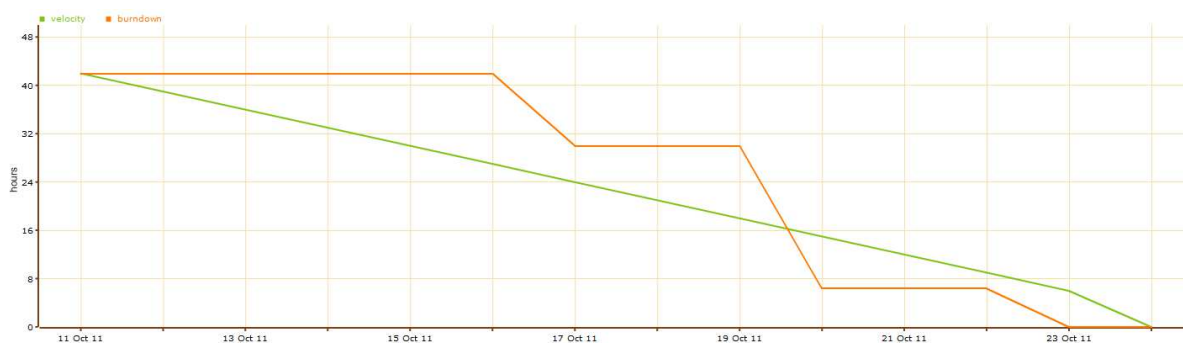
Obr. 3. Graf nedokončených úloh.

### Počet dokončených úloh

Metrika reprezentuje počet úloh, ktoré boli dokončené v danom období. Ako dokončené sa považujú tie úlohy, ktoré sú označené ako *Resolved* alebo *Closed*. Nazbierané údaje sa používajú v procese výpočtu indexu kvality a na spočítanie feature points pri výpočte indexu produktivity.

### Počet zostávajúcich hodín

Pri vytváraní úloh v nástroji Redmine sa udáva odhadovaný čas na ich dokončenie. Z tohto odhadu sa vytvára krivka zostávajúcich hodín na dokončenie projektu respektíve úloh šprintu. Zakresľuje sa do grafu, ktorý sa v metodike SCRUM nazýva *Burndown chart*. Takýto graf je na obr. 8. Obsahuje aj krivku odhadovaného dokončenia úloh, ktorá je vytvorená výpočtom tzv. *Velocity*. Zakreslená je zelenou farbou. Ak sa krivka zostávajúcich hodín nachádza nad odhadovaným stavom, znamená to, že projekt mešká. Indikuje to riziko nedokončenia všetkých úloh načas. Poznatky z tohto grafu sleduje najmä manažment plánovania a rizík. V prípade, že sa úlohy nestíhajú dokončiť v danom šprinte, zisťuje sa, čo tento nedostatok spôsobuje. Zobrazí sa burndown pre každého člena, aby sa určil pôvod problému. Chybu rieši príslušný manažment, podľa jej povahy.



Obr. 4. Burndown graf pre prvý šprint.

### Index produktivity

Meria produktivitu jednotlivých členov tímu. Predstavuje množstvo práce vykonané za časovú jednotku. Vypočíta sa ako podiel feature points a počtu odpracovaných hodín. Čím je tento index väčší, tým je väčšia produktivita. Feature points sú pre každý feature stanovené kartami použitím metodiky SCRUM. Počítajú sa len feature points dokončených úloh. Index produktivity získame dosadením hodnôt do vzťahu:

$$P = \frac{\varphi}{H}$$

, kde P je index produktivity,  $\varphi$  je počet feature points, H je počet odpracovaných hodín. Tento údaj je použitý v manažmente plánovania a pri rozdeľovaní jednotlivých úloh medzi členov tímu. Index možno využiť aj v manažmente ľudských zdrojov pri priradovaní členov tímu k úlohe. Vynásobením indexu so 100% získame produktivitu osoby vyjadrenú v percentách. Index je mierne skreslený, pretože tím vyvíja projekt pomocou technológie, s ktorou predtým nemal skúsenosti.

**Tab. 2.** Index produktivity pre šprinty 1 - 4.

	1	2	3	4	Spolu
<b>FP</b>	50	42	41	71	204
<b>Hodiny</b>	52	99.8	78.1	107	336.9
<b>Index</b>	0.961538	0.420842	0.524968	0.663551402	0.605521
<b>Percentuálne</b>	96.15385	42.08417	52.4968	66.35514019	60.55209

**Tab. 3.** Index produktivity pre šprinty 5 - 8.

	5	6	7	8	Spolu
<b>FP</b>	49	23	45	49	166
<b>Hodiny</b>	42	46.8	57.5	58.5	204.8
<b>Index</b>	1.166666	0.491452	0.782608	0.8376068	0.810546
<b>Percentuálne</b>	116.6666	49.14529	78.26086	83.760683	81.05468

Z tabuliek vidno, že index produktivity postupne stúpala. Zaujímavý je fakt, že najväčší bol vždy na začiatku semestra a následne dosiahol minimum, po ktorom sa vždy len zvyšoval. Celková produktivita tímu teda postupne stúpala. Ak porovnáme obidva semestre, všimneme si, že v druhom semestri dosiahla priemerná produktivita oveľa vyššiu úroveň, kde ich rozdiel činí približne 20,5%. Najvyššiu produktivitu dosiahol tím počas piateho šprintu. Najnižšiu produktivitu tím vykázal v druhom šprinte.

## Index kvality

Index poskytuje informácie pre manažment kvality. Predstavuje pomer chýb v projekte a počtu dokončených úloh. Index kvality získame dosadením hodnôt do vzťahu:

$$\sigma = \frac{E}{T}$$

, kde  $\sigma$  je index kvality, E je počet chýb v projekte, T je počet dokončených úloh. Čím je index menší, tým väčšia je kvalita projektu. Vynásobením indexu so 100% získame percentuálny podiel chýb v dokončených úlohách.

Tab. 4. Index kvality pre šprinty 1 - 4.

Šprint	1	2	3	4	Spolu
Počet úloh	6	26	20	41	93
Počet chýb	1	0	1	12	14
Index	0.166667	0	0.05	0.29	0.1505
Percentuálne	16.66667	0	5	29	15.05

Tab. 5. Index kvality pre šprinty 5 - 8.

Šprint	5	6	7	8	Spolu
Počet úloh	11	17	8	7	43
Počet chýb	1	1	0	0	2
Index	0.0909	0.0588	0	0	0.0465
Percentuálne	0.9090	0.5882	0	0	0.4651

Tab. 4 a tab. 5 znázorňujú vývoj indexu kvality. Tento index rástol do 4 šprintu. Tu dosiahol maximum. Ako dôvod vidíme veľké množstvo úloh, ktoré tím musel dokončiť v krátkom čase. Priemerný index dosiahol hodnotu približne 0,15. V tab. 5 vidíme, že index sa v nasledovnom období výrazne znížil. Za posledné dva šprinty sa nevyskytli žiadne chyby v dokončených úlohách. Index tu nadobúda najlepšiu hodnotu – 0. Dôvodom tohto zlepšenia je výrazné zvýšenie úsilia na dodržanie kvality produktu. Počas celého semestra tak index neprekročil hodnotu 0,0909. Priemerná hodnota v druhom semestri dosiahla 0,0465. Tím vylepšil index takmer o 0,11.



## 8 Manažment komunikácie

---

Tím pri prácach na projekte využíva niekoľko komunikačných prostriedkov. Patria medzi ne:

- osobné stretnutia,
- Facebook,
- Google Groups,
- telefónny kontakt,
- Redmine.

### 8.1 Osobné stretnutia

Najčastejšia forma tohto druhu komunikácie je realizovaná pravidelnými tímovými stretnutiami s vedúcim projektu (zákazníkom) v rámci predmetu Tímový projekt. Tieto stretnutia prebiehajú v softvérovom štúdiu FIIT. Ich hlavnou náplňou bývajú diskusie o aktuálnom stave projektu. Tie by sa dali ďalej rozdeliť na *medzišprinové* a *začiatko/koncošprintové*, v závislosti, od aktuálneho stavu šprintu.

Čo sa týka komunikácie, účastníci týchto stretnutí sa dajú rozdeliť do troch skupín:

- vývojársky tím,
- zákazník/vedúci projektu,
- externý poradcovia/hodnotitelia.

*Vývojársky tím* je reprezentovaný členmi tímu číslo 13, na predmete Tímový projekt 2011/2012. Jeho úlohou je v prvom rade komunikovať so zákazníkom. Táto forma komunikácie môže mať nasledovné podoby:

- neformálna komunikácia,
- podávanie hlásení o stave projektu,
- podávanie návrhov,
- odpovedanie na dotazy,
- hodnotenie nových požiadaviek pre fungovanie systému.

Neformálna komunikácia spočíva hlavne v uvoľnení atmosféry pri komunikácií so zákazníkom. To je dôležité, kvôli odstráneniu napätia, ktoré môže brzdiť tvorivý potenciál tímu. Taktiež pripravuje pôdu pre prechod k pracovnej diskusii.

Podávanie hlásení o stave projektu býva pravidelne každý týždeň pri stretnutí tímu s vedúcim projektu. Podľa aktuálneho stavu šprintu môže nadobúdať rozdielne stupne formálnosti. Počas prebiehajúceho šprintu podávajú členovia tímu priebežné informácie. Tieto hlásenia nie sú hodnotené a slúžia najmä ako konzultácia prípadných problémov, návrhov a postrehov. Kvôli tomu je úroveň ich formálnosti nižšia, ako pri konečnom vyhodnocovaní šprintu. To býva pravidelne, každé dva týždne. Vtedy členovia tímu prezentujú výsledky úloh, ktoré im boli pridelené. Táto časť je už hodnotená vedúcim projektu.

Návrh na upravenie aplikácie môže byť podaný ľubovoľným členom tímu. Po jeho vyslovení prebieha diskusia so zákazníkom, ktorému je objasnené, ako by realizovanie návrhu mohlo zlepšiť funkčnosť aplikácie. Po jeho zhodnotení ho môže zákazník zamietnuť, alebo zaradiť do nasledujúceho šprintu.

Pri nejasnosti, návrhu alebo nespokojnosti zákazníka nad niektorým z aspektov prác na projekte, môže požiadať o vysvetlenie. Odpovedať by mal vždy ten člen tímu, ktorý bol za danú časť prác zodpovedný, alebo o nej má najviac informácií.

Hodnotenie nových požiadaviek na systém je uskutočňované pri začatí nového šprintu, ktoré býva po ukončení šprintu starého. Prebieha metódou bodov použitia (story points). Každý člen tímu pri tejto metóde hodnotí novú požiadavku nezávisle od ostatných. Ak niekto z tímu odhad nad/podhodnotí, požiadavka sa konzultuje medzi členmi navzájom, ako aj s vedúcim projektu. V konečnom štádiu by sa mal dosiahnuť konsenzus.

*Zákazník/vedúci projektu* je v našom prípade reprezentovaný jednou osobou. Podľa toho, z ktorej pozície účinkuje, môže viesť komunikáciu s tímom vo viacerých smeroch. Ako zákazník navrhuje základné vlastnosti, ktoré by mal produkt nadobúdať. Z pozície vedúceho projektu podáva návrhy na architektúru a spôsob fungovania systému, hodnotí stav prác a navrhuje rozdelenie úloh do jednotlivých šprintov.

*Externý poradcovia* nie sú príležitostnými účastníkmi tímových stretnutí. Ich rola by sa dala prirovnať k *odborníkom v danej oblasti (subject matter experts)*, ktorí by mali byť súčasťou SCRUM tímov. Ich komunikácia s tímom prebieha hlavne vo forme odporúčaní, ako pri riešení úloh postupovať.

Všetky dôležité časti diskusie na stretnutí sú zaznamenávané zvoleným členom tímu. Jeho poznámky sú následne prevedené do digitálnej podoby, kde ich môžu ostatní členovia skontrolovať a v prípade potreby zrevidovať.

## 8.2 Facebook

Výber sociálnej siete *Facebook* (FB) ako jedného z nástrojov komunikácie bol uskutočnený na základe viacerých faktorov:

- podpora skupín,
- elektronická nástenka,
- možnosť okamžitej komunikácie (instant messaging),
- možnosť zasielania správ,
- všetci členovia tímu už boli zaregistrovaný.

Aby mohol tím efektívne komunikovať, je vhodné, aby mohol vo virtuálnom priestore komunikovať oddelene od ostatných záležitostí. Pre tento účel je užitočná podpora skupín. To nám umožňuje poslať príspevky na akúsi elektronickú nástenku (wall) tímu, kde si ho môžu všetci ostatní členovia ihneď prečítať. Táto možnosť je preferovaná, ak potrebujeme zdieľať s tímom informáciu, ktorá má dlhodobejšiu hodnotu.

Instant messaging je medzi členmi tímu využívaný hlavne pri spoločnej koordinácii v reálnom čase. Príkladom je spájanie (merge) jednotlivých vetiev programu dohromady. Vtedy je vhodné viesť tento proces koordinovať ako v prípade nutnosti riešenia konfliktov. Možnosť preposlania konfliktného kódu kolegovi, ktorý na ňom pracoval je v takejto situácii veľmi užitočná.



Zasielanie správ cez aplikáciu FB môže do istej miery nahradiť e-mailové správy. Prostredníctvom nich si členovia tímu medzi diskutujú na témy, ktoré nemajú pre nezainteresovaných členov tímu hodnotu.

### 8.3 Google Groups

Vytvorenie skupiny na *Google Groups* bolo užitočné kvôli dvom hlavným dôvodom:

- zdieľanie jednej e-mailovej adresy,
- zdieľanie dokumentov medzi členmi tímu.

Vďaka nej je možné celý tím kontaktovať z externého prostredia zaslaním správy na jednu e-mailovú adresu. Taktiež, ak jednotlivец zasiela správu, ktorá sa dotýka celého tímu, je vhodné, aby tak učinil použitím tejto skupiny.

### 8.4 Telefónny kontakt

Spôsob komunikácie preferovaný pri nemožnosti využitia internetu. Môže prebiehať buď zaslaním SMS, alebo telefonickým rozhovorom. Zaslanie SMS býva používané, ak nemá zasielaná informácia kritickú prioritu, alebo nepotrebujeme okamžitú odpoveď. Telefónny kontakt býva využívaný ako posledná možnosť kontaktovania, ak je požadovaná osoba prostredníctvom iných prostriedkov nedostupná. Tomu ako odôvodnenie môže slúžiť fakt, že sa jedná o pomerne invazívny spôsob nadviazania komunikácie, ako aj jeho spoplatnenie.

### 8.5 Redmine

Jedná sa o veľmi špecifický spôsob komunikácie. Je používaný ako čisto asynchrónna výmena informácií o stave prác na projekte. Každý z členov pravidelne aktualizuje dosahovaný pokrok, nie však hlavne za účelom komunikácie, ale z dôvodov plánovania a kontroly. Všetci členovia tímu ale majú k týmto informáciám prístup, preto tento nástroj môže byť zaradený medzi komunikačné prostriedky.



## 9 Manažment tvorby dokumentácie

---

Dokumentácia k projektu tímu sa skladá z

- dokumentácie k riadeniu,
- dokumentácie k inžinierskemu dielu,
- zápisov zo stretnutí,
- preberacích protokolov,
- informácií na stránke tímu.

### 9.1 Dokumentácia k riadeniu a inžinierskemu dielu

Pri tvorbe dokumentácie bol využitý textový editor *MS Word*. Na tvorbe tejto dokumentácie sa podieľali všetci členovia tímu. Každý člen mal manažérom dokumentácie pridelenú časť dokumentácie, ktorú mal napísať. Úlohou manažéra dokumentácie bolo následne jednotlivé časti integrovať do finálnej podoby a navrhnuť konečný vzhľad a štruktúru dokumentu. Manažér mal za úlohu aj jasne vyznačiť v dokumente, kto sa podieľal na tvorbe akej časti.

### 9.2 Zápisy zo stretnutí

Zápisy zo stretnutí boli vytvárané pomocou služby *Google Docs*. Manažér dokumentácie vytvoril šablónu pre písanie zápisníc, ktorá zaručuje jednotný formát zápisníc a uľahčuje písanie zápisníc všetkým členom tímu. Pri písaní zápisníc sa postupne striedali všetci členovia tímu.

### 9.3 Preberacie protokoly

Formát a obsah preberacích protokolov boli ponechané na manažérovi dokumentácie, ktorý dokumenty vytváral. Preberací protokol bol dodaný s dokumentáciou k projektu pri každom kontrolnom bode počas semestra.

### 9.4 Webová stránka tímu

Webová prezentácia tímu bola vytvorená spoluprácou viacerých členov tímu. O vzhľad prezentácie a aktualizovanie dokumentov na stránke sa stará manažér dokumentácie. Na stránke sa nachádzajú aj ďalšie dokumenty, ktoré vznikli počas semestra, napr. ponuka tímu, plagát tímu, návrh obrazoviek systému.



## **Prílohy**



## Príloha A: Metodika štruktúry zdrojových kódov

---

Metodika slúži na určenie konvencií a zjednotenie spôsobu písania a úprav zdrojových kódov projektu.

Je určená členom tímu, ktorí sa podieľajú na vývoji.

V aplikácii používame jazyk Ruby a aplikačný rámec Ruby on Rails 3. Väčšina nariadení vychádza z konvencií zaužívaných v jazyku Ruby a v Ruby on Rails.

### A.1 Názvy premenných

- ✦ sú celými a plnovýznamovými slovami (prípadne s významom v rámci problematiky, ktorú projekt rieši)
- ✦ sú uvedené v angličtine
- ✦ viacslovné výrazy sú rozdelené znakom „\_“
- ✦ premenné, ktoré sú inštanciami objektov musia byť pomenované podľa triedy objektu
- ✦ dlhé názvy sú skracované bez straty významu v kontexte okolitého kódu

Príklady názov premenných:

```
@repository
rdt_relationship
name
@older_version
annotation_binding skrátene na binding
```

### A.2 Názvy funkcií

- ✦ označujú čo funkcia vracia alebo čo vykonáva
- ✦ uvedené v angličtine
- ✦ viacslovné výrazy sú rozdelené znakom „\_“

Príklady názvov funkcií:

```
def all_annotations
def check_rdt
def unique_name_in_variant
def text
```

Ruby umožňuje uzavretie blokov kódu zátvorkami „{ }“ alebo kľúčovými slovami „do“ a „end“ viacriadkové bloky kódu, ktoré spravidla obsahujú viacero príkazov sú uzatvorené medzi „do“ „end“ jednoriadkové bloky s jedným príkazom sú uzavreté medzi „{ }“

```
asset_list.each do |entry|
  # viac príkazov
end

File.open(filename, 'w') {|f| @document.write_xml_to f}
```

### A.3 Odsadzovanie

Každý vnorený blok je odsadený o dva znaky

```
rdts.each do |rdt|
  concept = Nokogiri::XML::Node.new "concept", @doc
```

### A.4 Komentovanie zdrojového kódu

- ⤴ sú písané v angličtine
- ⤴ komentárom je označená každá metóda, ktorá bola vytvorená vývojárom. Obsah komentára opisuje jej účel, návratovú hodnotu a parametre
- ⤴ nie je potrebné komentovať preddefinované akcie controllerov, ak sa ich funkcionality zásadne nezmenila.

```
# deletes bindings to annotations with specified IDs
# param annotations - array of IDs
def unbind (annotations)
  ...
```

Komentáre je treba napísať pred podmienkami, príkazmi a cyklami, vykonávajúcimi funkcionality, ktorá si vyžaduje vysvetlenie.



## Príloha B: Metodika pre prácu s nástrojom Git

---

Táto metodika slúži na zjednotenie písania komentárov k verziám kódu pre sprehľadnenie práce s nástrojom Git a pre umožnenie efektívneho prepojenia tohto nástroja s nástrojom na manažment úloh Redmine. Je určená všetkým, ktorí vystupujú v roli vývojára.

### B.1 Komentáre pri ukladaní novej verzie kódu

Pri ukladaní zmien, uveďte vždy komentár opisujúci vykonané zmeny v nasledujúcom tvare:

```
<(refs|closes|fixes) #<cisloUlohyVRedmine> - stručný opis zmien - 1  
riadok - povinný opis>
```

```
<podrobnejší opis zmien v kóde - nepovinný opis>
```

Príklad:

```
refs #1255 - doplneny model a controller Document
```

*vytvoreny model dokumentu s atributmi nazov, obsah, verzia a funkcionalita na vytvorenie,  
prezeranie, editovanie a vymazanie dokumentu*

### B.2 Vývoj vo vetvách

Pred začatím práce na vám pridenej časti projektu vytvorte v úložisku novú vetvu vývoja, v ktorej budete ukladať nové zdrojové kódy. Vetvu vytvoríte príkazom:

```
git checkout -b <názov-vetvy>
```

V prípade, že vetvu vytvárate z IDE RubyMine, vytvoríte ju pomocou príkazu „Git → Checkout branch...“, ktorý nájdete buď v kontextovom menu po kliknutí pravým tlačidlom na priečinok s projektom v navigačnom paneli vľavo alebo v ponuke menu „Version Control“. V dialógu zadajte do položky „New Branch“ názov vetvy a potvrdte. Aktuálnu vetvu vývoja nastavíte v menu „Current Branch...“, ktorá sa nachádza v obidvoch menu nad možnosťou „Checkout Branch...“

#### B.2.1 Pomenovávanie vetiev

Názov vetvy sa bude vyzeráť nasledovne:

```
<(feature|bug)-cisloUlohyvRedmine-nazov-ulohy>
```

napr. *bug-1249-vypis-mena-repozitarov*.

#### B.2.2 Spájanie vetiev

Pred spojením vašej vetvy s hlavnou vetvou, musíte svoju vetvu najskôr aktualizovať a aplikovať lokálne všetky zmeny, ktoré nastali v hlavnej vetve odkedy ste vytvorili vašu vetvu. Pri aplikovaní zmien môžu nastať konflikty v kóde, ktoré musíte vyriešiť. Pri riešení konfliktov ponechajte v konfliktných súboroch kód, ktorý ste vytvorili vy. Ak nastane konflikt v súbore, s ktorým ste nepracovali, uprednostnite kód zo serveru. V prípade že by ste uprednostnili svoj lokálny kód, je možné, že prepíšete prácu niekoho iného a v aktuálnej verzii kódu sa ocitne zastaraný kód.



## Príloha C: Zápisy zo stretnutí

---

V tejto prílohe sa nachádzajú zápisnice zo stretnutí tímu, ktoré sa konali počas druhej polovice letného semestra každý štvrtok 15:00-17:00 a každý druhý piatok 13:00-16:00 v Softvérovom štúdiu. Termíny stretnutí a autori zápisníc sú nasledovní:

12.04.2012 Bc. Petra Vrablecová  
13.04.2012 Bc. Róbert Kocian  
19.04.2012 Bc. Martin Gajdoš  
26.04.2012 Bc. Zuzana Zimová  
03.05.2012 Bc. Martin Gajdoš  
11.05.2012 Bc. Martin Franta



# Zápisnica č. 24

**Dátum:** 12.4.2012

**Trvanie:** 15:00 – 17:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Petra Vrablecová

1. Martin F. prezentoval dizajn obrazoviek systému. Každý sa vyjadrí mailom Mariánovi.  
Ďalej prezentoval úpravy obrazovky DE1 (doplnený properties tab, výpis konfliktov...), zobrazovanie zmien v dokumente po update, konflikty v anotáciach. Marián chce, aby konfliktné anotácie boli farebne odlíšené.
2. Pri práci s anotáciami boli odhalené bugy, ktoré má Robo spísať do Redmine.
3. Petra nemala hotovú svoju obrazovku. Má ju dorobiť do konca víkendu a informovať Mariána mailom.
4. Martin H. prezentoval import repozitára a metadát, kde dorobil validáciu metadát. V rng schéme musel odstrániť namespace xmlns. Takisto dorobil históriu importov. Martin povedal, že jeho obrazovka RDV1 je hotová.
5. Martin G. prezentoval registráciu zmien v grafe. Marián navrhol iné zobrazenie vzťahov uzla ako feature do ďalšieho šprintu.
6. Robo prezentoval prácu s anotáciami na serveri. Bola odprezentovaná aj Zuzanina feature zobrazenia viacerých anotácií na rovnakom riadku v anotačnom pásiku. Nedostatky majú byť odstránené do ďalšieho stretnutia.
7. Bola odovzdaná dokumentácia k šprintom 5 až 8.



# Zápisnica č. 25

**Dátum:** 13.4.2012

**Trvanie:** 13:00 – 16:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Róbert Kocian

Stretnutie v na konci šprintu.

1. kreslenie poster predstavy na IITSRC , rozhodovanie o tom, ktorý program na vytváranie postera použiť. Vyberali sme z programov ako Photoshop, Inkscape, Powerpoint.
2. kontrola obrazoviek
  - Home - potvrdenie neexportovateľných dokumentov - nová obrazovka
    - import skryť a zobrazíť až po kliknutí na nový button
    - buttony import export oddeliť od tabuliek repositories a variants
  - RD2
    - search all columns dať preč zo všetkých tabuliek
    - pridať (aj do iných obrazoviek) checkbox pre hromadné vybratie dokumentu
    - odobrať API kľúč
    - upraviť pozíciu publish buttonov
  - Variant
    - zjednotiť výber tak ako je v obrazovke Varianty s RDV1
    - upraviť farebnú schému
    - upraviť pop-up, aby sa nezobrazoval v tabuľke, ak sa na ňu prepne, ale len v grafe
    - add relationship - treba zvýrazniť warning červenou
    - zvýrazňovať susedov v grafe
    - dorobiť cyklus ukladania v grafe
  - RDV1
    - dorobiť autocomplete
  - Dokument edit,
  - Document view - dorobiť tabuľky v tabkách, aby nebol horizontálny scrollbar





# Zápisnica č. 26

**Dátum:** 19.4.2012

**Trvanie:** 15:00 – 17:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Martin Gajdoš

1. Martin F. odprezentoval navrhované pozadie pre plagát IIT SRC.
2. Prebehla dodatočná diskusia o podobe plagátu.
3. Rozhodlo sa že sa tím dohodne na podobe a realizovať ho bude Petina sestra.
4. Prebehlo kolečko, čo treba dorobiť na jednotlivých obrazovkách.
5. Martin F. urobil návrh dizajnu, ktorý má každý implementovať vo svojej obrazovke.
6. Požiadavky na prezentáciu IIT SRC:
  - a. Plátno
  - b. Projektor
  - c. Stolíky
  - d. Pripojenie na internet
  - e. Pripojenie do siete
7. Zuzka spomenula stav jej obrazovky.
8. Robo nemal problémy vo svojej obrazovke.
9. Maťo F. rozmýšľal nad možnosťou zobrazovania tabiek pomocou vysúvania.
10. Peťa mala dotaz k tomu ako ukladať vzťahy cez tabuľku do grafu.



# Zápisnica č. 27

**Dátum:** 26.4.2012

**Trvanie:** 15:00 – 17:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Zuzana Zimová

1. Vo všetkých tabuľkách
  - odstrániť horizontálne scrollbary,
  - rovnaká šírka stĺpcov “Akcie” podľa ikon,
  - zjednotiť názvy stĺpcov (#, Changed / Has changed, ...)
  - zjednotiť šírku stĺpcov na stránke
  - tooltips k názvom stĺpcov, tooltips k ikonám
  - našťýlovať comboboxy
2. Variants v repozitári
  - pridať stĺpcové filtrovanie
  - zameniť ‘v.’ za ‘Version’,
  - Name stĺpec - širší, Actions stĺpec užší
  - zameniť ikonu edit variantov za ‘Assign’ (reťaz)
  - zameniť ‘Publish’ za ikonu wifi
  - Odstrániť Manage RDTs
  - add variant to repository otvára viac dialógov, má otvárať len jeden
3. Vyriešiť problém s rôznym rozlíšením
4. Documents
  - #Conflicts nemá mať hodnoty true a false
  - presúvanie dokumentov nefunguje
  - zameniť add document za create new document
5. Nový dokument
  - validate false -> odstrániť false
  - prehodiť preview a import
  - preview nefunguje, ak dokument nie je validný/uložený
6. Edit documentu
  - tooltip na šípku dole, zameniť ikony
  - textový kurzor na zvolenej tabke, mala by byť ručička
  - properties - pridať version do novej tabky History
  - RDTs - autocomplete combobox, asynchrónne dotiahnutie tabuľky, pridávanie rdt podľa obrazovky RDV1
  - Files - pridávanie súborov nefunguje
  - Document relationships - nelinkuje, typ vzťahu autocomplete
  - Zodpovednosť za tabky: Properties: MF, Annotations : ZZ, RDTs: PV, Files: RK, Relationships: MG

- zobrazenie rozdielu medzi verziami, prekopať celý dialóg, plus konflikty, zmeniť font
- filtrovanie zoznamu anotácií
- edit anotácií v tabke nefunguje
- validácia?

#### 7. RDV1

- select prvý dokument v zozname
- preview - pridať border, zaoblené tvary preč
- náhľad vložiť do stĺpca s názvom (samostatný span)
- opraviť filtrovanie
- doplniť nadpisy tabuliek
- chýba stĺpec Actions
- pri edite prepojenia sa zmenší ikona, vráti chybu
- nastaviť rôzne zarovnanie checkboxov
- pridať akciu pre usera pri pridávaní vzťahu

#### 8. Variant

##### tabuľka

- zarovnať pridávanie pod stĺpce
- javascript chyba pri edite vzťahu
- nefunguje filtrovanie po akcii v grafe
- tooltip v tabuľke dvojriadkový

##### graf

- graf - opraviť neighbour list (nastaviť polopriehľadnosť), pásik s tlačidlami dať zhora
- prezoomovanie
- zameniť cancel a OK pri editovaní uzla
- odstrániť alert váhy pri vytváraní uzla
- učesať pop up-y
- ikony do pop up v grafe s tooltipmi
- apply changes - pridať info o nových vzťahoch

#### 9. Document view

- Upraviť Context, aby zaberol menej miesta
- zakázať vypnutie property panelu
- zakázať otváranie edit a new dialógu naraz
- edit anotácie zo zoznamu
- našťýlovať zoznam

# Zápisnica č. 28

**Dátum:** 3.5.2012

**Trvanie:** 15:00 – 17:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Martin Gajdoš

1. Robo s Peťou zisťovali prečo nefunguje správne zobrazovanie datatables
2. Dohadovali sme termín stretnutia pre budúce stretnutie
3. Dostali sme úlohu kontaktovať kontrolóra kvality Tomáša Kramára a dohodnúť sa s ním čo treba robiť
4. Vo všeobecnosti sme nerobili vzájomne kontroly obrazoviek - nedostatok.
5. Úloha je dokončiť všetky veci, urobiť refaktoring
6. Urobiť dokumentáciu

## Úlohy

Robo	assigned metadata
Zuzka	naštýlovať zoznam anotácií change selection
Morte	RDV1 - rovanko veľké div názvy tabuliek - ako urobiť zobrazenie názvu
Peťa	v tabkách Document view - urobiť RDTs zarovnat pridavanie pod stlpec posledný riadok v tabuľke presahuje graf - zobrazovanie Options v rámci grafu
Maťo F.	v tabkách Properties - história komunikácia s Alefom bude na prezentačnej úrovni ALEF bude nasadený kedy do pondelka (14.5) Dať preč div actions z dialogu
Maťo G.	V inner details - odstrániť Neighbours, zmenšiť ľavé zarovnanie Presunúť buttony v grafe vedľa Apply changes



# Zápisnica č. 29

**Dátum:** 11.5.2012

**Trvanie:** 13:00 – 15:00

**Miesto stretnutia:** Softvérové štúdio

**Prítomní:**

Pedagóg – Ing. Marián Šimko, PhD.

Členovia – Bc. Martin Franta, Bc. Martin Gajdoš, Bc. Martin Habdák, Bc. Róbert Kocian, Bc. Petra Vrablecová, Bc. Zuzana Zimová

**Autor:** Bc. Martin Franta

MH reportuje stav RDV1, väčšina vecí je spravená, menší problém je s posúvaním šírky stĺpcov pri klikaní na dokumenty v tabuľke dokumentov. Je to známy problém z ostatných obrazoviek pri sortovaní, ale riešenie sa doposiaľ nepodarilo nájsť.

Označenie dokumentu pri inicializácii screenu vyberie prvý dokument v DOME ale nie je prvý v tabuľke

Poslať kvalitérovi mail s adresou nášho projektu.

Do dokumentácie by malo ísť zhrnutie toho čo sme spravili - technický popis, komponenty, architektúra.

MG upravil dizajn ovládacích prvkov v grafe.

Dizajn popupov treba zjednotiť, prípadne zjednodušiť.

Treba dorobiť tabky pre dokument - súbory, relationships.

Subsumed-by typ document-document vzťahov dorobiť do importu. Zipy sú v ALEF wiki.

Pri kliknutí na tlačidlo akcie sa nemá označiť riadok v tabuľke (Repository detail).

V dialógoch sú stále akcie so šedým pozadím, treba to v CSS odstrániť.

Tabuľka prekrývajúcich sa anotácií potrebuje mierny redizajn.

(Future work - change to selection v edit dokumentu)

Úprava anotácie - musí sa upraviť binding anotácie na dokument, nie len samotná anotácia, inak nikdy nezmiznú konflikty.

Pri publishi repozitára zistiť, či tam nie sú dokumenty s konfliktmi.

ZZ upravovala nested annotations.

Prečítať si e- mail od prof. Bielikovej, aké sú pokyny - zásady hodnotenia na stránke, pozrieť si, naštudovať si.





## **Príloha D: Preberacie protokoly**

---

Počas trvania projektu na predmete Tímový projekt II bol jeden kontrolný bod, v ktorom sa odovzdávali nasledujúce dokumenty:

**14.5.2012      Záverečná dokumentácia**

- dokumentácia k inžinierskemu dielu
- dokumentácia k riadeniu





Slovenská technická univerzita

**Fakulta informatiky a informačných technológií**

**Tím č. 13 – Thirteam**

e-mail: tim13@googlegroups.com

---

## Preberací protokol

Svojím podpisom vyučujúci potvrdzuje, že dokumentácia k predmetu *Tímový projekt II* bola v danom kontrolnom bode riadne odovzdaná a spĺňa všetky náležitosti potrebné pre akceptáciu odovzdania tohto dokumentu.

vyučujúci: Ing. Marián Šimko, PhD.

kontrolný bod: 4. kontrolný bod (14.5.2012)

dňa

, v Bratislave

---

podpis vyučujúceho