

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Crowdex

Dokumentácia k riadeniu projektu

Vedúci práce: Ing. Michal Kompan, PhD.

Členovia tímu: Bc. Dušan Cymorek, Bc. Peter Gašpar, Bc. Vladimír L'alík,
Bc. Michal Polko, Bc. Miroslav Šafárik, Bc. Slavomír Šárik,
Bc. Štefan Šmihla

Akademický rok: 2014/2015

Obsah

1	Úvod	1-1
2	Roly členov tímu a podiel práce	2-1
2.1	Manažérske úlohy	2-1
2.2	Krátkodobé úlohy	2-1
2.3	Podiel práce na jednotlivých častiach dokumentácie	2-2
3	Aplikácie manažmentov	3-1
3.1	Manažment dokumentácie	3-1
3.2	Manažment podpory vývoja a integrácie	3-3
3.3	Manažment plánovania	3-7
3.4	Manažment testovania a prehliadok	3-11
3.5	Manažment rizík	3-13
3.6	Manažment komunikácie	3-14
3.7	Manažment kvality	3-16
3.8	Manažment rozsahu	3-16
3.9	Manažment monitorovania	3-17
4	Sumarizácie šprintov	4-1
4.1	1. šprint	4-1
4.2	2. šprint	4-1
4.3	3. šprint	4-1
4.4	4. šprint	4-1
4.5	5. šprint	4-2
5	Používané metodiky	5-1
6	Globálna retrospektíva - zimný semester	6-1
6.1	Časový prehľad	6-1
6.2	Distribúcia celkového času riešenia úloh medzi autorov	6-2
7	Preberacie protokoly	7-1
8	Zoznam kompetencií tímu	8-1
8.1	Úvod	8-1
8.2	Členovia tímu	8-1
8.3	Kontakt na tím	8-2
8.4	Motivácia	8-2
8.5	Návrh riešenia	8-3
8.6	Príloha A: Aktuálny rozvrh všetkých členov tímu	8-4

9	Metodiky	9-1
9.1	Dokumentovanie kódu	9-1
9.2	Vytvorenie vetvy	9-8
9.3	Vytvorenie commitu	9-13
9.4	Vytvorenie pull requestu	9-19
9.5	Testovanie softvéru	9-22
9.6	Nahlasovanie chýb	9-28
9.7	Prehliadka zdrojového kódu	9-32
9.8	Písanie kódu	9-39
9.9	Návrh a zobrazovanie formulárov	9-45
A	Zápisy zo stretnutí	A-1
A.1	Zápis č. 0 z neformálneho stretnutia tímu	A-1
A.2	Zápis č. 1 zo stretnutia tímu	A-2
A.3	Zápis č. 2 zo stretnutia tímu	A-3
A.4	Zápis č. 3 zo stretnutia tímu	A-4
A.5	Zápis č. 4 zo stretnutia tímu	A-5
A.6	Zápis č. 5 zo stretnutia tímu	A-10
A.7	Zápis č. 6 zo stretnutia tímu	A-12
A.8	Zápis č. 7 zo stretnutia tímu	A-14
A.9	Zápis č. 8 zo stretnutia tímu	A-16
A.10	Zápis č. 9 zo stretnutia tímu	A-19
A.11	Zápis č. 10 zo stretnutia tímu	A-22
A.12	Zápis č. 11 zo stretnutia tímu	A-25
A.13	Zápis č. 12 zo stretnutia tímu	A-29
B	Retrospektívy k šprintom	B-1
B.1	Retrospektíva k 1. šprintu	B-1
B.2	Retrospektíva k 2. šprintu	B-8
B.3	Retrospektíva k 3. šprintu	B-14
B.4	Retrospektíva k 4. šprintu	B-19
B.5	Retrospektíva k 5. šprintu	B-27

1 Úvod

Tento dokument obsahuje dokumentáciu riadenia projektu Manažment experimentov (Crowdex) platnú k druhému kontrolnému bodu. Dokument vznikol v rámci predmetu Tímový projekt v akademickom roku 2014/15 na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave.

Dokument je členený do logických kapitol. Kapitola 2 obsahuje rozdelenie úloh jednotlivých členov tímu, spolu s prehľadom podielu práce na jednotlivých častiach dokumentácie.

Kapitola 3 zahŕňa zodpovednosti manažérskych úloh členov tímu a opis jednotlivých činností vykonávaných pri riadení projektu.

Kapitola 4 obsahuje sumarizáciu jednotlivých šprintov, v ktorej je uvedený postup tímu v šprintoch.

Kapitola 5 obsahuje referencie na používané metodiky uvedené v kapitole 9 na strane 9-1.

Obsahom kapitoly 6 je globálna retrospektíva za zimný semester.

Kapitola 8 obsahuje zoznam kompetencií tímu tak, ako boli odovzdané začiatkom zimného semestra.

Kapitola 9 je venovaná už spomenutým metodikám, ktoré boli používané počas práce na projekte.

Prílohy A a B obsahujú zápisy zo stretnutí a retrospektívy k šprintom.

2 Roly členov tímu a podiel práce

2.1 Manažérske úlohy

Manažérske úlohy v rámci projektu sme si rozdelili nasledovne:

Manažérska úloha	Zodpovedná osoba
Manažment dokumentácie	Dušan Cymorek
Manažment rozvrhu	Peter Gašpar
Manažment rizík	Vladimír L'alík
Manažment rozsahu, manažment monitorovania	Michal Polko
Manažment podpory vývoja a integrácie	Slavomír Šárik
Manažment komunikácie	Miroslav Šafárik
Manažment testovania a prehliadok	Štefan Šmihla

Tabuľka 2.1: Rozdelenie manažérskych úloh

2.2 Krátkodobé úlohy

V tabuľke nižšie uvádzame krátkodobé úlohy jednotlivých členov tímu, ktoré sa vyskytli počas riešenia projektu.

Úloha	Zodpovedná osoba
Tvorba zápisov a retrospektív	Cyklické striedanie všetkých členov
Webová stránka tímu	Peter Gašpar, Michal Polko
Konfigurácia produkčného servera, vývojového a produkčného prostredia, nástrojov pre manažment úloh - YouTrack, pre verziovanie zdrojového kódu – Github, pre kontinuálnu integráciu - Travis CI, pre kolaboráciu - MediaWiki	Slavomír Šárik
Prihláška na TP Cup	celý tím

Tabuľka 2.2: Rozdelenie krátkodobých úloh

2.3 Podiel práce na jednotlivých častiach dokumentácie

2.3.1 Dokumentácia k riadeniu projektu

Kapitola	Autori
Úvod	Dušan Cymorek
Úlohy členov tímu	Dušan Cymorek, Peter Gašpar, Slavomír Šárik
Manažment dokumentácie	Dušan Cymorek
Manažment podpory vývoja a integrácie	Slavomír Šárik
Manažment plánovania	Peter Gašpar
Manažment testovania a prehliadok	Štefan Šmihla, Michal Polko
Manažment rizík	Vladimír L'alík
Manažment komunikácie	Miroslav Šafárik
Manažment kvality	Vladimír L'alík, Peter Gašpar
Manažment rozsahu	Michal Polko
Manažment monitorovania	Michal Polko
Sumarizácia šprintov	Michal Polko
Používané metodiky	Peter Gašpar
Globálna retrospektíva	Michal Polko, Štefan Šmihla
Metodiky	Celý tím
Zápisy zo stretnutí	Celý tím
Retrospektívy k šprintom	Celý tím

Tabuľka 2.3: Autori jednotlivých kapitol dokumentácie k riadeniu projektu

2.3.2 Dokumentácia k inžinierskemu dielu

Kapitola	Autori
Úvod	Michal Polko
Ciele na zimný semester	Peter Gašpar, Michal Polko
Celkový pohľad na systém	Peter Gašpar, Slavomír Šárik, Miroslav Šafárik
Registrácia používateľa	Štefan Šmihla
Autentifikácia používateľa	Slavomír Šárik
Profil používateľa	Miroslav Šafárik
Detail experimentu	Miroslav Šafárik
Pridanie a správa experimentu	Dušan Cymorek, Peter Gašpar, Vladimír L'alík
Pozvanie používateľa do experimentu	Miroslav Šafárik
Notifikácie	Slavomír Šárik
Nastavenia profilu používateľa	Štefan Šmihla
Hodnotenie používateľov	Miroslav Šafárik
Dashboard	Peter Gašpar
Diskusia k experimentu	Slavomír Šárik
Spustenie experimentu	Štefan Šmihla
Realizácia experimentu	Štefan Šmihla
Vyhľadávanie v systéme	Dušan Cymorek
Filtrovanie výsledkov vyhľadávania	Vladimír L'alík
Dizajn a interakcia s používateľom	Peter Gašpar, Michal Polko

Tabuľka 2.4: *Autori jednotlivých kapitol dokumentácie k inžinierskemu dielu*

3 Aplikácie manažmentov

V nasledujúcich podkapitolách uvádzame popis činností a postupov potrebných pre korektné riadenie procesu, projektu a v neposlednom rade aj produktu.

3.1 Manažment dokumentácie

Manažér dokumentácie dohliada na tvorbu dokumentácií vytváraných priebežne v jednotlivých šprintoch, ale i na dokumentovanie zdrojového kódu. Vytvára šablóny a upravuje dokumenty tak, aby bola dodržaná rovnaká identita naprieč dokumentáciou v tíme. Okrem toho v projekte *Crowdex* zodpovedá za údržbu prekladov v aplikácii.

Počas riešenia tímového projektu je vytváraných niekoľko typov dokumentov. V závislosti od typu dokumentu sa líšia autori aj nástroje, v ktorých je dokument vytváraný.

- Textová dokumentácia
 - Špecifikácie
 - Metodiky
 - Návody
 - Zápisy, retrospektívy
 - Dokumentácia riadenia a dokumentácia k inžinierskemu dielu
- Dokumentácia zdrojového kódu
- Webové sídlo

3.1.1 Textová dokumentácia

3.1.1.1 Špecifikácie

Sú vytvárané zodpovednou osobou pri riešení implementačných úloh a slúžia na rozpracovanie analýzy, návrhu, implementácie a testovania danej úlohy. Riešitelia úlohy následne pri implementácii postupujú podľa tejto špecifikácie.

Kvôli potrebe rýchleho zdieľania a poznámkovania špecifikácie sa špecifikácia vytvára na nástroji *Google Drive*. Po dokončení úlohy a zapracovaní zmien vzniknutých pri implementácii je špecifikácia prepísaná do dokumentácie k inžinierskemu dielu. Počas zimného šprintu sme dospeli k záveru, že bude vhodnejšie na kritické dokumenty zaviesť Wiki (rozhodli sme sa pre *MediaWiki*), na ktorej budú všetky tieto dokumenty pohromade a v prípade zmien budú všetci členovia tímu automaticky informovaní prostredníctvom mailu.

3.1.1.2 Metodiky

Vznikali prevažne v rámci zadaní na predmete Manažment v informačných systémoch. Keďže bolo vhodné prezentovať tím jednotne, manažér dokumentácie vytvoril šablónu pre *MS Word*, ktorá bola dostupná všetkým členom tímu.

Podobne ako v predchádzajúcom prípade, i tu sme dospeli k záveru, že metodiky sa budú nachádzať na tímovej Wiki stránke.

3.1.1.3 Návody

Ide o kratšie dokumenty, ktorých účelom je zjednodušiť členom tímu prácu s nejakým nástrojom, špecifickou časťou implementácie, testovania, a pod. Vzhľadom na ich povahu boli tieto dokumenty počas zimného semestra vytvárané a umiestnené v zdieľanom priečinku na *Google Drive*, v letnom semestri je naplánovaný ich presun na Wiki.

3.1.1.4 Zápisy, retrospektívy

Zápisy a retrospektívy popri systéme na manažment úloh zobrazujú priebeh riešenia projektu počas jednotlivých šprintov počas semestra. Pre stretnutia bola určená rotácia osôb v tíme, ktoré mali na starosti riadenie stretnutia, tvorbu zápisu a overenie zápisu. Tieto roly sa každý týždeň posúvali smerom doľava, pričom tvorca zápisu z minulého týždňa sa stal vedúcim stretnutia, overovateľ sa stal zapisovateľom a rolu overovateľa prebral nový člen tímu. Takto bolo zabezpečené rovnomerné rozdelenie písania zápisov v rámci tímu.

Forma zápisov sa počas semestra mierne menila, pretože spočiatku sme mali nepresné informácie o potrebe písania zápisníc. Samotné zápisy sa tvorili na *Google Drive*, pričom následne boli prepísané do *MS Word* a prevedené do formátu *PDF*.

Tvorbu retrospektív mali na starosti vždy minimálne dvaja členovia tímu. V retrospektíve sú uvedené zoznam úloh a stav ich riešenia, časový prehľad práce autorov počas šprintu, zhodnotenie šprintu (spolu s prípadnými úlohami, ktoré vznikli počas riešenia šprintu), Burndown Chart a nakoniec graf distribúcie celkového času riešenia úloh medzi autorov.

3.1.1.5 Dokumentácia riadenia a dokumentácia k inžinierskemu dielu

Tieto dokumentácie okrem iného agregujú vyššie spomenuté typy dokumentácie, ktoré vznikali priebežne počas šprintu. Približne dva týždne pred odovzdaním v kontrolnom bode započal proces prepisu týchto čiastkových dokumentácií podľa typu do Dokumentácie riadenia alebo Dokumentácie k inžinierskemu dielu.

Pre oba dokumenty vytvoril manažér dokumentácie šablónu v nástroji *MS Word*, pričom dokumenty sa nachádzali na webovom úložisku MS OneDrive, vďaka ktorému bola umožnená súčasná kolaborácia viacerých členov tímu pri prepisovaní jednotlivých kapitol. Tento spôsob, ktorý sa spočiatku javil ako vhodný a bezproblémový však počas narastania počtu strán v dokumente čoraz viac spôsoboval problémy s nestabilitou a konfliktmi pri editovaní dokumentu. Preto

bolo po prvom kontrolnom bode jednohlasne rozhodnuté o prechod na tvorbu dokumentácie v nástroji \LaTeX v spojení s distribuovaným nástrojom riadenia revízií *Git*.

Dokumentácia v druhom kontrolnom bode je už vytváraná iba prostredníctvom spomenutej kombinácie nástrojov. Hoci bol prechod spočiatku pre niektorých členov ťažší, po pár dňoch sa ukázalo, že toto rozhodnutie bolo správne.

Tvorba dokumentácie prebieha nasledovne: každý z členov tímu prepíše dokumenty, ktoré vytvoril. Aktuálna verzia dokumentácie sa nachádza na serveri (či už *Git*, alebo predtým *OneDrive*). Manažér dokumentácie priebežne upravoval dokumenty tak, aby bola zabezpečená konzistentnosť v používaných štýloch, odstránené osamotené predložky na konci riadku, a opravené prípadné chyby. S niektorými činnosťami mu pomáhali niektorí členovia tímu, čím sa zabezpečila redukcia rizika zlyhania ľudského faktoru, najmä pri prehliadnutí prípadnej chyby.

3.1.2 Dokumentácia zdrojového kódu

Dokumentácia zdrojového kódu vznikala priebežne počas semestra, predovšetkým však po sprevádzkovaní dokumentačného nástroja *Yardoc*¹, ktorý umožňuje vytváranie štruktúrovanej dokumentácie na základe označkových komentárov v zdrojových súboroch programu.

S počiatku bola nová forma tvorby dokumentácie problémom, no časom sa podarilo všetkým členom tímu dokumentovať zdrojový kód podľa metodiky 9.1 Dokumentovanie kódu na strane 9-1.

3.1.3 Webové sídlo

Webové sídlo slúži okrem zdroju informácií o tíme aj ako miesto, na ktorom je verejnosti prístupná dokumentácia vytváraná počas práce na tímovom projekte. Táto dokumentácia zahŕňa:

- **Plán**, ktorý aktualizuje manažér plánovania,
- **Zápisy zo stretnutí** - po overení na stránku nahráva overovateľ,
- **Retrospektívy šprintov** - publikuje vždy jeden z autorov,
- **Formálna dokumentácia** (*Dokumentácia k riadeniu projektu, Dokumentácia k inžinierskemu dielu*) - zverejňuje manažér dokumentácie po jej formálnej korektúre.

3.2 Manažment podpory vývoja a integrácie

Úloha manažéra podpory vývoja a integrácie pozostáva z manažmentu verzií, konfigurácií softvérového systému a integrácie použitých súčastí do celku. V projekte *Crowdex* je manažér priamo zodpovedný za nasledujúce úlohy:

- správa a údržba aplikačného servera

¹<http://yardoc.org/>

- správa a údržba produkčného prostredia
- nasadzovanie aplikácie na produkčný server
- monitorovanie aktuálneho stavu servera a aplikácie
- riešenie vzniknutých problémov počas behu aplikácie na serveri
- správa a údržba vývojového prostredia
- správa a údržba nástroja YouTrack² pre manažment úloh projektu
- správa a údržba repozitárov v nástroji Github³ pre manažment verziovania zdrojového kódu
- správa a údržba nástroja Travis CI⁴ určeného pre kontinuálnu integráciu
- spravovanie verzií produktu, použitých zásuvných modulov, knižníc a ich vzájomnej kompatibility s aplikáciou

Jeho úlohou je taktiež dohľad nad dodržiavaním metodiky pre verziovanie zdrojového kódu zvyšnými členmi tímu a udržiavať zdrojový kód v repozitári v konzistentnom stave.

Pred začatím práce na úlohach bola každá úloha pridelená niektorému z členov tímu. Nasledujúcim krokom bolo vytvorenie nových vetiev v repozitári, pričom každá implementovaná funkcionálna mala priradenú svoju vlastnú vetvu. Ako nástroj pre manažment zdrojového kódu sme používali *GitHub*, ktorý predstavuje webovú aplikáciu určenú pre pohodlnú správu zdrojového kódu.

Počas implementácie príbehov šprintu sme postupovali v súlade podľa jasne definovaných metodík určených pre oblasť verziovania kódu. Zdrojový kód bol ukladaný do lokálnych repozitárov vytvorením commitov, pričom tieto lokálne repozitáre sme priebežne synchronizovali s centrálnym repozitárom na Githube.

Po ukončení implementácie pridelenej úlohy bolo potrebné implementáciu zvalidovať. Na koľko každá úloha mala pridelenú zodpovednú osobu, táto osoba dohliadala nad kvalitou výslednej práce. Po validovaní úlohy zodpovednou osobou bolo potrebné pripraviť zdrojový kód implementácie pre jeho ďalšie zlúčenie s hlavnou vetvou.

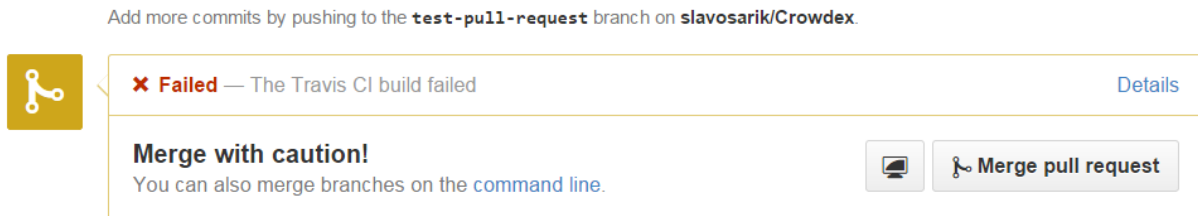
Procesu zlúčenia zdrojového kódu predchádzalo vytvorenie *pull requestu* osobou, ktorá implementovala úlohu. Vytvorenie pull requestu prebiehalo priamo prostredníctvom webového rozhrania Githubu, kde sa vyplnili základné údaje o zdrojovom kóde a implementovaných funkcionálnych. Proces vytvorenia pull requestu bol napojený na automatizovanú kontinuálnu

²<https://www.jetbrains.com/youtrack/>

³<https://github.com/>

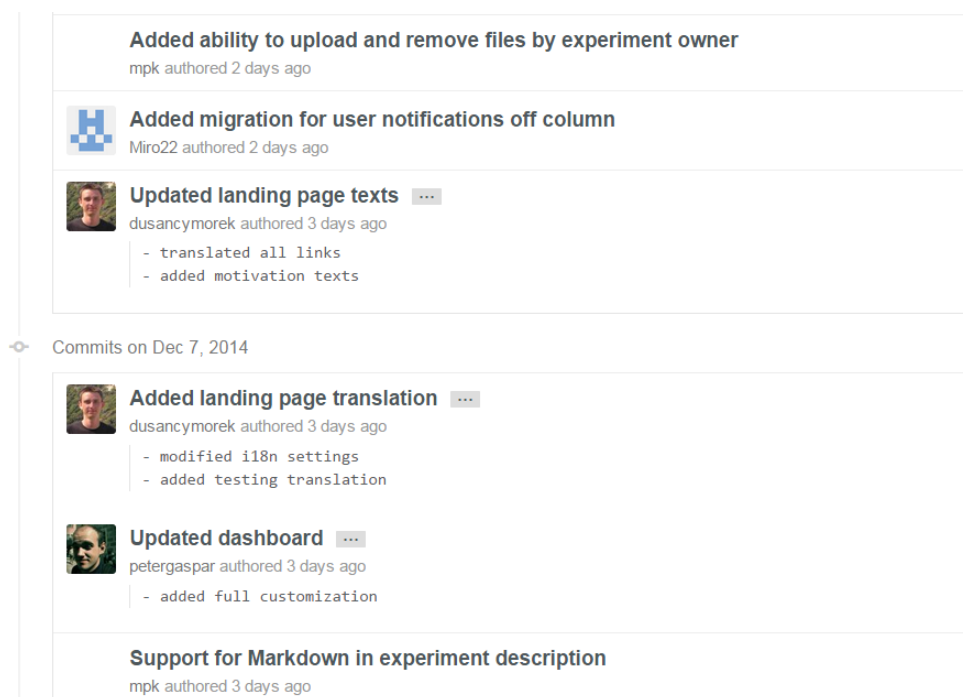
⁴<https://magnnum.travis-ci.com/>

integráciu využitím služby *Travis CI*. Kontinuálna integrácia nám umožnila automatizovať proces verifikácie zdrojového kódu, kde sa najprv zostavil testovací build a následne nad týmto buildom boli vykonané unit testy. V prípade, ak sa vyskytla chyba počas procesu testovania, implementujúca osoba bola povinná odstrániť túto chybu. Na obrázku 3.1 môžeme vidieť ukážku zlyhania procesu kontinuálnej integrácie.



Obr. 3.1: Ukážka nástroja GitHub počas kontroly pull requestu

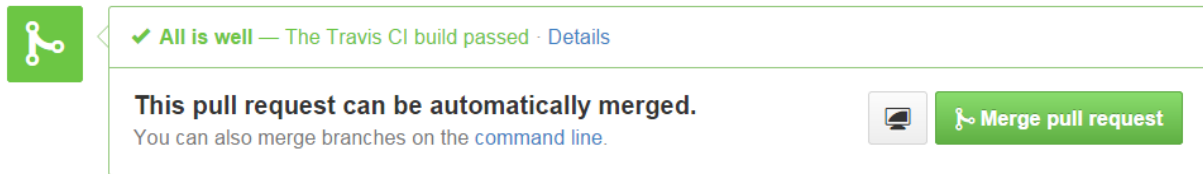
Manažér podpory vývoja a integrácie bol zodpovedný za vykonanie prehliadku kódu z pohľadu použitých zásuvných modulov a knižníc a overiť dodržiavanie metodík počas verziovania zdrojového kódu prehliadkou vykonaných commitov. Na obrázku 3.2 môžeme vidieť zoznam commitov spolu s ich popismi.



Obr. 3.2: Ukážka nástroja GitHub počas kontroly pull requestu

Na základe kladného výsledku procesu kontinuálnej integrácie, ktorý je zobrazený na obrázku 3.3, manažér zlúčil zdrojový kód v pull requeste s hlavnou vetvou a vytvoril v changelogu nový

záznam obsahujúci posledné vykonané zmeny spolu s uvedením autora týchto zmien. Na obrázku 3.4 môžeme vidieť ukážku časti z changelogu.



Obr. 3.3: Ukážka úspešného výsledku procesu kontinuálnej integrácie

0.12.3 (09-12-2014)

- fixed input cursor (@petergaspar)

0.12.2 (09-12-2014)

- fixed issue when no logged user wanted to see experiment detail (@Morzeux)

0.12.1 (09-12-2014)

- added landing page texts (@dusancymorek, @petergaspar)

0.12.0 (08-12-2014)

- added widget personalization (@petergaspar)
- added experiment files uploading (@mpk)
- added markdown support for experiment description (@mpk)
- small comments design changes (@mpk)

0.11.0 (07-12-2014)

- added possibility to finish experiment by uploading files (@Morzeux)
- added new notification types (@Morzeux, @vladolalik)
- fixed language tests and cleaned empty tests (@Morzeux)
- added edit and delete experiment comments (@vladolalik, @slavosarik)

0.10.1 (05-12-2014)

- fixed bugs with redirection and fixed user profile (@petergaspar)

Obr. 3.4: Ukážka changelogu

Po finálnom zlúčení zdrojového kódu do hlavnej vetvy manažér pripravil testovacie prostredie (*staging*) pre nasadenie aktuálnej verzie. V tejto fáze dochádza k testovaniu funkcionality⁵, ktorá

⁵testovacie prostredie na produkčnom serveri sa bude v budúcnosti používať na testovanie

už je nasadená na produkčnom serveri, avšak v testovacom prostredí. Po úspešnom otestovaní manažér pripravil produkčné prostredie a nasadil aplikáciu do produkčného prostredia. Keďže sme pri vývoji produktu postupovali agilnou metódou vývoja softvéru, bolo potrebné byť v kontakte s product ownerom a pravidelne dodávať aktuálnu verziu produktu spolu so správou o stave a poslednými prebehnutými zmenami.

3.3 Manažment plánovania

Manažér plánovania dohliada na pridel'ovanie, plnenie a zaznamenávanie priebehu úloh v rámci celého projektu. V našom tíme sme zaviedli viaceré pravidlá, ktoré zabezpečujú korektné a včasné plnenie úloh a ich náležitú archiváciu. Využívame pritom viaceré podporné nástroje:

- YouTrack – nástroj na manažment projektu a správu úloh,
- Toggl – nástroj na meranie času stráveného nad úlohou,
- Google Drive – nástroj na vzájomnú výmenu súborov s možnosťou kolaboratívneho hodnotenie a poznámkovania.

Pri stanovovaní úloh a plánu sme postupovali v súlade s metodikou vývoja Scrum. V spolupráci s vlastníkom produktu (*angl. Product Owner*) sme zostavili hlavný *Product Backlog* – zoznam všetkých základných úloh, ktoré budeme plniť počas realizácie projektu.

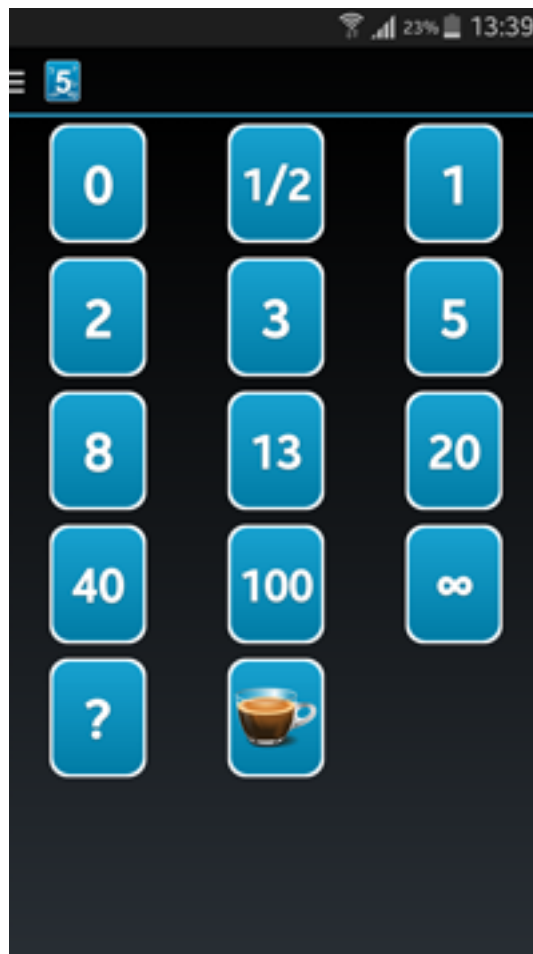
Pred začiatkom každého šprintu vybral vlastník produktu skupinu úloh, ktoré sa riešili nasledujúce obdobie. Postupovali sme pritom v súlade s ohodnotením podľa ich časovej náročnosti. Pri stanovení náročnosti sme používali nasledujúce mobilné aplikácie:

- Scrum Poker⁶ (operačný systém Android) - obrázok 3.5,
- Scrum Card Desk⁷ (operačný systém iOS) - obrázok 3.6.

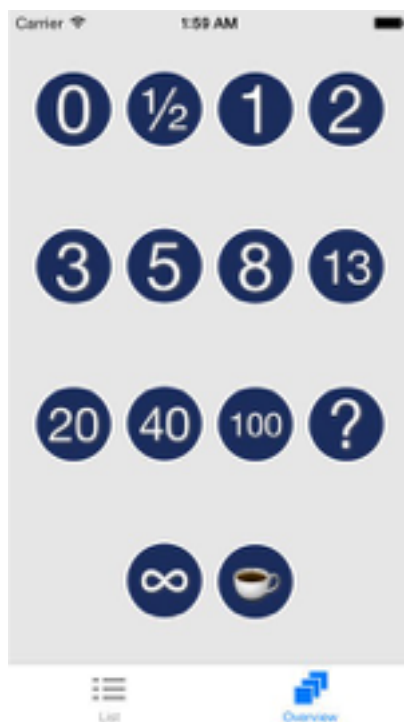
Každý člen tímu ohodnotil úlohu číslom kartičky, ktoré zodpovedalo odhadovanému počtu potrebných človekohodín (8 hodín) na realizáciu danej úlohy. Následne sa všetky odhady ukázali a členovia, ktorí ohodnotili úlohu vyšším, prípadne nižším číslom ako väčšina, musel zdôvodniť, prečo považuje úlohu za náročnejšiu, respektíve jednoduchšiu. Na základe diskusie sa určila finálna náročnosť úlohy, ktorá bola východiskom pri rozdeľovaní úloh opísanom v ďalšom odseku.

⁶<https://play.google.com/store/apps/details?id=artarmin.android.scrum.poker>

⁷<https://itunes.apple.com/us/app/scrum-card-deck/id779873086?mt=8>



Obr. 3.5: Ukážka aplikácie Scrum Poker



Obr. 3.6: Ukážka aplikácie Scrum Card Desk

Pri rozdeľovaní úloh sme sa primárne zamerali aj na svoje vlastné schopnosti a vedomosti v rámci tímu. Platili pritom nasledujúce pravidlá:

- Každý člen tímu sa zapísal na aspoň 1 úlohu.
- Každá úloha nižšieho rozsahu (ohodnotená rozsahom maximálne 1 deň) mala prideleného aspoň 1 člena (riešiteľa).
- Každá úloha vyššieho rozsahu (ohodnotená rozsahom viac ako 1 deň) mala pridelených aspoň 2 členov (riešiteľov).
- Každéj úlohe bola pridelená zodpovedná osoba, ktorá mala za úlohu vytvoriť špecifikáciu pre danú úlohu. Na základe tejto špecifikácie sa následne postupovalo pri implementácii a testovaní.

Tvorba špecifikácie bola časovo ohraničená na základe dĺžky šprintu. Tabuľka č. 3.1 vyjadruje stanovené rozloženie času na šprint. Keďže sme postupovali inkrementálne a iteratívne, vytvorené špecifikácie sa aj v období implementácie a testovania podľa potreby upravovali po dohode so zodpovednou osobou.

Deň / Dĺžka šprintu	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 týždeň	Tvorba špecifikácie			Implementácia/ Testovanie			-							
2 týždne	Tvorba špecifikácie			Implementácia/Testovanie										

Tabuľka 3.1: Rozdelenie činností počas šprintov

Po dokončení špecifikácie zodpovedné osoby vytvorili v nástroji *YouTrack* príslušné podúlohy, ktoré vznikli zvýšením granularity základných úloh z *Product Backlogu*. Tieto podúlohy boli následne pridelené jednotlivým riešiteľom (danej základnej úlohy). Po dokončení každej podúlohy zodpovedná osoba verifikovala stav jej riešenia a na základe toho rozhodla, či bola úspešne splnená alebo či sa vyžadovali ďalšie korekcie. V prípade, že boli splnené všetky podúlohy základnej úlohy, zodpovedná osoba označila základnú úlohu ako verifikovanú a pre daný šprint aj uzavretú.

Na zaznamenávanie času stráveného nad riešením projektu sme používali bezplatný nástroj *Toggl*⁸. Časy sme pritom priebežne ukladali do *YouTracku* k príslušným úlohám. Na záver šprintu sme pre účely tvorby retrospektívy vygenerovali export z nástroja *YouTrack*. Ten zahŕňal najmä stav splnenia úloh, riešiteľov, zodpovedné osoby a celkový čas trvania.

Tabuľka č. 3.2 súhrnne uvádza dĺžky šprintov, ktoré sme doposiaľ úspešne zrealizovali.

Č. šprintu	Dátum začiatku	Dátum ukončenia	Dĺžka trvania
1	14.10.2014	28.10.2014	2 týždne
2	28.10.2014	11.11.2014	2 týždne
3	11.11.2014	18.11.2014	1 týždeň
4	18.11.2014	2.12.2014	2 týždne
5	2.12.2014	9.12.2014	1 týždeň

Tabuľka 3.2: Dĺžky trvania šprintov

Súčasťou nášho projektu boli aj nasledujúce úlohy, ktoré sme nerealizovali v rámci jednotlivých šprintov:

- vytvorenie webového sídla,

⁸<https://www.toggl.com/>

- vytvorenie prihlášky do TP Cupu,
- prepojenie aplikácie s Git službou.

Tieto úlohy sa nachádzajú v kapitole 2 – *Úlohy členov tímu* aj s príslušnými riešiteľmi.

3.4 Manažment testovania a prehliadok

Úlohou manažéra testovania a prehliadok je sledovať kvalitu a funkcionality kódu. Na meranie funkcionality, pokrytia a stability kódu sa používa tzv. *unit testing*. Tento pojem zahŕňa nástroje, metodiky a činnosti spojené s overením funkcionality kódu. Prehliadka kódu (*angl. code review*) zahŕňa kontrolu komplexností a duplicit kódu, nedodržiavanie konvencii a podobne. Nástroje, ktoré pri tom využívame:

- *Code Climate* - externý nástroj na kontrolu kvality kódu, duplicit, komplexností a podobne.
- *RSpec - framework* určený na písanie prehľadných a čitateľných testov.

Vízia z hľadiska testovania bola pracovať princípom testom riadeného vývoja *angl. Test-Driven Development*. To v praxi znamená písanie testov ešte pred implementáciou ľubovolnej elementárnej funkcionality. Pre väčšinu členov však prechod na tento princíp bol adaptačne, aj časovo náročný. Preto sa na tento princíp nekládol až taký dôraz a bol považovaný skôr za odporúčanie. Napriek tomu si s postupom času niektorí členovia zvykli písať počas implementácie testy a pokrytie testov sa s postupom času zlepšovalo. Naším cieľom preto naďalej je viac sa adaptovať na princíp testom riadeného vývoja a zlepšovať tak nielen pokrytie kódu, ale aj jeho stabilitu.

Počas implementácie každý člen dbal nato, aby udržal existujúce testy zelené. *Pull request* sa nemohol vykonať bez toho, aby všetky testy boli zelené. V tomto spolupracoval manažér integrácie, ktorý povolil *pull request* len v prípade, že všetky testy ostali zachované. Úspešný stav vidieť na obrázku 3.7, kde sa zároveň pokrytie kódu 56,36% odosiela priamo do služby *Code Climate*.

```
285 $ bundle exec rake
286 /home/travis/.rvm/rubies/ruby-2.1.1/bin/ruby -S rspec ./spec/config/localization_spec.rb
./spec/controllers/dashboard_controller_spec.rb ./spec/helpers/dashboard_helper_spec.rb
./spec/helpers/notifications_helper_spec.rb ./spec/mailers/user_mailer_spec.rb ./spec/models/address_spec.rb
./spec/models/event_spec.rb ./spec/models/event_type_spec.rb ./spec/models/experiment_condition_spec.rb
./spec/models/experiment_conditions_skill_spec.rb ./spec/models/experiment_spec.rb ./spec/models/experiment_user_reward_spec.rb
./spec/models/experiments_has_tag_spec.rb ./spec/models/notification_spec.rb ./spec/models/skill_spec.rb
./spec/models/tag_spec.rb ./spec/models/user_follower_spec.rb ./spec/models/user_login_spec.rb ./spec/models/user_spec.rb
./spec/models/users_skill_spec.rb ./spec/views/experiment_comments/comment_spec.rb ./spec/views/experiments/show_spec.rb
./spec/views/main_page/authentication_spec.rb ./spec/views/main_page/set_language_spec.rb ./spec/views/search/search_spec.rb
./spec/views/users/show_spec.rb
287
.....
.....
288
289 Finished in 2 minutes 1.1 seconds
290 217 examples, 0 failures
291
292 Randomized with seed 885
293
294 Coverage = 56.36%. Sending report to https://codeclimate.com for branch development... done.
295
296 The command "bundle exec rake" exited with 0.
```

Obr. 3.7: Ukážka úspešného testovania.

Priebeh testovania, tak ako aj zmena kvality kódu sa zaznamenávali na *Code Climate* (obrázok 3.8 a 3.9), z ktorého chodili manažérovi testovania a prehliadok notifikácie o aktuálnom stave.

The screenshot displays the Code Climate interface. On the left, a list of notifications shows test coverage changes: 'Test coverage has improved to 57.2% (+0.2)', 'Test coverage has improved to 57.0% (+0.6)', 'Test coverage has declined to 56.4% (-0.6)', 'Test coverage has improved to 57.0% (+0.6)', and 'Test coverage has declined to 56.4% (-0.6)'. A notification at the bottom states 'One class/module was added' for 'MarkdownHelper'. On the right, a search bar is present above a circular GPA score of 2.86. Below the GPA, there are buttons for 'code climate 2.9' and 'coverage 57%', and a link to 'Link to Code Climate from your README'. At the bottom right, a shield icon indicates '56.6% Test Coverage'.

Obr. 3.8: Ukážka základných správ v Code Climate.

Rating	Name	Complexity	Duplication	Churn	C/M	Coverage	Smells
A	AccountActivationsController	25	0	9	25.2	33.3%	0
A	Address	6	0	1	N/A	100.0%	0
A	AlertsHelper	0	0	1	0.0	100.0%	0
C	DashboardController	145	0	12	20.7	36.1%	3
A	DashboardHelper	0	0	1	N/A	100.0%	0

Obr. 3.9: Ukážka štatistík v Code Climate.

Manažér testovania a prehliadok si miesta so zaznamenanými pachmi v kóde pravidelne prezeral a upozorňoval na chyby, prípadne vykonával ďalšie kroky potrebné k náprave (vytvorenie poznámky *TODO* priamo v kóde, prípadne samostatnej úlohy na *refactoring* kódu).

3.5 Manažment rizík

Manažér rizík zodpovedá za riešenie možných rizík, ktoré môžu ohroziť vývoj projektu. Jeho úlohou v našom tíme je identifikovať možné riziká, následne analyzovať ich následky a definovať kroky potrebné pre minimalizovanie dôsledkov rizika. Konkrétne v našom tíme máme aplikovaných niekoľko postupov ako predchádzať rizikám. Niektoré sa nám podarilo určiť a eliminovať ešte predtým ako sme pocítili ich následky.

Prvé riziko, ktoré sme definovali bolo spojené s kvalitou kódu, pretože v našom tíme sa podieľame všetci siedmi na vývoji. Je teda nutné aby každá funkcionálna mala kvalitnú a hlavne aktuálnu dokumentáciu a kód zodpovedal tejto dokumentácii. Pred začiatok prvého šprintu sme sa dohodli na tom, že každá úloha musí mať okrem riešiteľa pridelenú aj zodpovednú osobu, ktorá vytvorí špecifikáciu a overí splnenie úlohy. Zodpovednú osobu určujeme pri plánovaní šprintu, pričom je označená aj v systéme *Youtrack*, čo znamená, že dostáva notifikácie o tom ako prebieha práca na danej úlohe. Po tom ako riešiteľ dokončí svoju úlohu označí ju ako "*Fixed*" v systéme, zodpovedná osoba môže začať s testovaním kódu. Ak riešiteľ urobil zmeny oproti špecifikácii je nutné buď upraviť špecifikáciu alebo funkcionálnu tak aby kód a dokumentácia boli konzistentné. Až po skontrolovaní a označení úlohy v systéme *Youtrack* ako "*Verified*" môže riešiteľ vytvoriť *Pull Request* svojej vetvy na *Development* vetvu.

Dôležitosť dodržiavanie týchto krokov sme zistili až v druhom šprinte, keď že počas prvého šprintu kontroly zo strany zodpovedných osôb neboli dôkladné, implementovaná funkcionálna nezodpovedala dokumentácii. Pri implementovaní funkcionality nadväzujúcej na tú z prvého šprintu sa vyskytli prvé spory v tíme práve pre nekonzistentnú dokumentáciu. Od druhého šprintu boli kontroly dôkladnejšie a implementovanie novej funkcionality už prebiehalo rýchlejšie, keď že dokumentácia zodpovedala implementovanej funkcionality.

Úloha	Priorita	Náročnosť	Šprint	Zodp. osoba	Riešitelia
Jazykové nastavenia		1	1	Dušan	Slavo
Registrácia nového používateľa	4	2	1	Štefan	Slavo, Štefan
Autentifikácia používateľa (štandardne)	4	2	1	Slavo	Štefan, Slavo
Pridávanie experimentov	4	2	1	Vlado	Miro, Peťo
Detail profilu používateľa (osoba)	3	2	1	Miro	Dušan, Vlado

Obr. 3.10: Product Backlog s úlohami, ktoré majú priradené riešiteľov a zodpovedné osoby

So zavedením predchádzajúcich pravidiel nám vyplynulo ďalšie riziko spojené so splnením úloh v šprinte. Keďže riešiteľ úlohy mohol začať pracovať na svojej úlohe až potom ako dostal špecifikáciu určili sme si termíny, kedy je nutné vytvoriť špecifikáciu. V prípade šprintu, ktorý trval jeden týždeň to je druhý deň šprintu a počas dvoj týždňového šprintu to je tretí deň od začiatku šprintu.

Pri plánovaní šprintu sme vždy brali do úvahy aj ostatné povinnosti jednotlivých členov tímu aby sme eliminovali riziko nesplnenia úloh v šprinte. Ak sa vyskytla úloha, ktorá buď bola náročná na implementáciu svojou komplexnosťou alebo tým, že s daným problémom nemal nikto v tíme skúsenosti, tak sme k nej priradili aspoň dvoch riešiteľov. Vyskytli sa aj prípady úloh, ktoré bolo náročne navrhnuť a vtedy k nim boli pridelené dve zodpovedné osoby, ktoré navrhli špecifikáciu.

3.6 Manažment komunikácie

Manažér komunikácie zodpovedá za spôsoby komunikácie medzi jednotlivými členmi tímu. Zabezpečuje, aby bola komunikácia možná, čo najefektívnejšia a aby sa všetky informácie dostali ku všetkým členom tímu. Komunikácia v tíme sa delí na formálnu a neformálnu.

3.6.1 Formálna komunikácia

V rámci tímu bola hneď po jeho sformovaní vytvorená tajná skupina na sociálnej sieti *Facebook*. Do tejto skupiny boli pridaní všetci členovia tímu, pričom po prvých pár stretnutiach bol do nej pozvaný aj vedúci projektu, Ing. Michal Kompan, PhD., ktorý toto pozvanie prijal. Skupina sa využíva primárne pre rýchlu komunikáciu medzi jednotlivými členmi tímu, keďže všetci členovia majú založený profil na tejto sociálnej sieti a túto sieť pravidelne navštevujú. Komunikujú sa tu hlavne zložitejšie otázky, ktoré chceme mať v rámci komunikácie aj nejakým spôsobom archivované. Keďže sa v skupine nachádza aj vedúci projektu, môže sa vyjadrovať ku všetkým veciam, ktoré v nej rozoberáme, prípadne sme aj my schopní ho priamo osloviť a v relatívne rýchлом čase dostať odpoveď.

Ďalší druh formálnej komunikácie je automatizovaná komunikácia cez emaily s využitím systému *YouTrack*. Na začiatku každého šprintu je pre každú úlohu okrem riešiteľov pridelená aj tzv. zodpovedná osoba. Vo výnimočných prípadoch (krátka či jednoduchá úloha, refaktORIZÁCIA zdrojového kódu a pod.) môže byť zodpovedná osoba zároveň riešiteľom. V ostatných prípadoch

sú riešiteľ a zodpovedná osoba odlišní. Jednou z úloh zodpovednej osoby je vytvoriť príslušné úlohy v systéme *YouTrack* a prideliť týmto úlohám daných riešiteľov. Takto sa zabezpečí, že zodpovedná osoba je prostredníctvom emailu, ktorý má nastavený v systéme *YouTrack*, informovaná o každej zmene na daných úlohách (o zmene stavu, o čase riešenia, o prípadných komentároch a ďalších). Zodpovedná osoba tak presne vie, v akom stave je úloha, koľko práce sa už na nej vykonalo a kedy sú práce na danej úlohe ukončené. Vďaka tejto automatizácii, ktorú prináša systém *YouTrack*, nie je nutné kontinuálne kontaktovať riešiteľov s otázkami o stave úlohy.

Ďalší kanál využívaný na komunikáciu je *Google Drive*, kde sú zdieľané všetky dokumenty členov tímu (zápisy, metodiky, rôzne návody, ...). V rámci tohto systému prebieha okrem zdieľania týchto dokumentov aj ich kolaboratívna tvorba a prípadné pripomienkovanie. Komunikácia pri tvorbe dokumentov zvyčajne prebieha dvomi spôsobmi:

- komunikácia pri tvorbe určitého dokumentu prostredníctvom chatu, ktorý *Google Drive* v dokumentoch poskytuje,
- komunikácia prostredníctvom poznámok (komentárov) v jednotlivých dokumentoch.

Komunikáciu prostredníctvom poznámok (komentárov) využívame najmä na pripomienkovanie konkrétnych častí v dokumentoch. Rovnako takto autor upozorňuje na prípadné nejasnosti pri tvorbe dokumentu. Komunikáciu prostredníctvom chatu využívame najmä vtedy, ak je za vytvorenie daného dokumentu zodpovedných viac osôb - tieto pri tvorbe komunikujú v reálnom čase. Hoci je tento druh komunikácie skôr neformálny, kvôli súvislosti s *Google Drive* sme ho zaradili do tejto časti.

Počas 9. stretnutia bolo diskutované používanie kolaboratívneho nástroja. Rozhodovali sme sa medzi *MediaWiki* a *DocuWiki*, nakoniec sme sa rozhodli pre *MediaWiki*. Aktuálne je tento systém už nainštalovaný na serveri a začne sa používať v letnom semestri.

Okrem internetovej komunikácie je dôležitým zdrojom informácií pre členov tímu aj pravidelné týždenné stretnutie, na ktorých sa okrem členov tímu zúčastňuje aj zadávateľ projektu. Na tomto stretnutí sa

- riešia všetky problémy v tíme,
- diskutujú návrhy jednotlivých členov tímu,
- diskutuje funkcionality systému,
- plánujú šprinty,
- každý člen tímu vyjadruje, v akom stave je funkcionality, ktorej je riešiteľom (čo bolo urobené, čo sa urobí a aké prípadné problémy má),
- vyhodnocujú šprinty.

3.6.2 Neformálna komunikácia

Okrem skupiny využívame na sociálnej sieti *Facebook* aj skupinový chat v dvoch formách. Prvá forma je skupinový chat, v ktorom sa nachádzajú všetci členovia tímu. Tento je používaný na riešenie menších a nie tak významných problémov, prípadne sa cez neho zdieľajú krátke informácie, otázky či názory, ktorých archivovanie je pre nás nepodstatné. Výhodou tejto formy komunikácie je najmä to, že prebieha takmer v reálnom čase vzhľadom na to, že skoro vždy existuje aspoň jedna osoba, ktorá je prihlásená na sociálnu sieť *Facebook* a môže reagovať na prípadnú otázku v chate. Inými slovami, spravidla nenastáva situácia, že ak niektorý z členov tímu napíše do chatu otázku, čakal by na odpoveď, resp. názor od iného člena tímu dlhšie ako 10 minút.

Druhá forma skupinového chatu je skupinový chat medzi riešiteľmi úlohy a zodpovednou osobou. Tento používame najmä pri ukončovaní prác na úlohe, kedy zodpovedná osoba kontaktuje riešiteľov cez novovytvorený skupinový chat, resp. cez existujúci, ak už existoval predtým. Cez tento chat sa rieši prehliadka zdrojového kódu (najmä nájdené chyby či nejasnosti v implementácii).

Mimo internetu sa na neformálnu komunikáciu využívajú aj neformálne stretnutia.

Neformálne stretnutia sú zvyčajne stretnutia na pôde fakulty, na ktorých sa nezúčastňuje zadávateľ projektu a ktoré sa realizujú najčastejšie vtedy, ak je potrebné vypracovať nejaký dokument, kde je potrebná kooperácia a diskusia všetkých (alebo väčšiny) členov tímu, prípadne ak je v tíme potrebné urobiť mozgovú búrku. Príkladom takýchto situácií bola tvorba prihlášky do súťaže *TP Cup* či tvorba biznis modelu. Takéto stretnutia sa zvyčajne konali v pondelok okolo 15:00. Každý člen tímu pri nich komunikoval svoje názory a zvyčajne „online“ prebehla aj tvorba príslušného dokumentu cez *Google Drive*.

Okrem neformálnych stretnutí na fakulte sa nám podarilo zrealizovať aj stretnutie v podniku v meste – *teambuilding* – na ktorom sa zúčastnil aj vedúci projektu. Po skončení zimného semestra je plánovaný ďalší *teambuilding*, ktorým sa zavŕši celosemestrálne úsilie všetkých členov tímu.

3.7 Manažment kvality

Manažment kvality zodpovedá za kvalitu vytvoreného kódu. Je mimoriadne dôležité mať nielen fungujúci kód, ale aj kvalitný kód. Keďže na vývoji sa podieľajú všetci členovia tímu, nutne musíme kód udržiavať dobre štruktúrovaný a prehľadný. V súčasnosti máme vytvorené metodiky na dodržiavanie základných princípov, ktoré nám zjednodušia v budúcnosti rozširovanie funkcionalít a pomôžu udržať kód funkčný. *Git*.

3.8 Manažment rozsahu

Úlohou manažmentu rozsahu v našom tíme je zabezpečiť dve základné požiadavky:

- Produkt by rozsahom mal obsahovať tie vlastnosti, ktoré boli dohodnuté na začiatku projektu.
- Jednotlivé vlastnosti projektu by mali byť implementované v rozsahu, ktorý sa dohodne na stretnutí pri začiatku šprintu.

Dodržiavanie rozsahu na úrovni produktu sme zabezpečili pomocou vytvorenia dokumentu *Product Backlog*, ktorý obsahuje všetky vlastnosti, ktoré by mal výsledný produkt obsahovať. Podľa tohto dokumentu sa riadi aj plánovanie šprintov.

Dodržiavanie rozsahu na úrovni jednotlivých úloh sme zabezpečili pomocou špecifikácií, ktoré boli vytvárané pre každú z úloh. Vytvorená špecifikácia presne definovala, čo sa má v rámci úlohy vykonať, a bolo na zodpovednosti člena tímu, ktorý funkcionality implementoval, aby ju implementoval v celom rozsahu.

Ak však zistil, že z nejakého dôvodu nie je možné implementovať funkcionality tak ako bola uvedená v špecifikácií, bolo jeho povinnosťou zmeniť špecifikáciu tak, aby zodpovedala realite (implementácií).

Ak nebolo implementované všetko, čo bolo uvedené v špecifikácií, zostávajúca funkcionality bola vložená ako ďalšie úlohy do nástroja *YouTrack* a tie boli riešené v niektorom z ďalších šprintov.

3.9 Manažment monitorovania

Manažment monitorovania v našom tíme je realizovaný ako kontrola dodržiavania plánu nasledovným spôsobom.

Na začiatku semestra sme si v dokumente *Product Backlog* stanovili približné rozdelenie úloh (implementácie vlastností produktu) do jednotlivých šprintov. Toto nám umožňuje priebežne monitorovať stav práce na produkte a prípadné omeškanie voči pôvodnému plánu.

Kontrola stavu práce na šprinte je zabezpečená pomocou funkcionalít obsiahnutých v nástroji *YouTrack* - *Burndown Chart* a *Agile Board*. Kontrolujeme najmä krivku v *Burndown Chart*, a v rámci stretnutí sa snažíme analyzovať, prečo sa krivka aktuálneho šprintu (ne)odchyľuje voči jej ideálnemu priebehu.

Monitorovanie stavu jednotlivých úloh sme zabezpečili jednak vzájomnou komunikáciou na spoločnom chate, a jednak na stretnutiach, ktoré sa konajú pravidelne každý týždeň. Každý z členov tímu dostáva priestor, aby oboznámil tím s postupom pri implementácií svojich úloh a prípadnými komplikáciami, ktoré nastali. Takto máme možnosť zistiť problémy vopred a podľa potreby delegovať iného člena tímu, aby pomohol pri ich riešení.

Každý člen si monitoruje postup implementácie úlohy pomocou nástroja na meranie času (najmä *Toggl*). Každá z úloh šprintu má priradený časový odhad, čo umožňuje riešiteľovi porovnať si čas strávený prácou na úlohe s uvedeným odhadom a tak zistiť, či je jeho tempo postačujúce (resp. či bol odhad správny).

4 Sumarizácie šprintov

4.1 1. šprint

V rámci prvého šprintu sme začali s návrhom produktu. Zároveň sme implementovali nasledovnú funkcionálnosť:

- Základnú interakciu s experimentami
- Základnú interakciu s profilmi používateľov
- Registráciu a prihlásenie používateľa v kontexte našej aplikácie
- Navrhli a implementovali sme základný dizajn aplikácie

4.2 2. šprint

Dvojtýždňový druhý šprint obsahoval implementáciu nasledujúcich funkcií:

- Správa experimentu a možnosť prezerat' si detail experimentu
- Možnosť úpravy nastavení používateľského profilu
- Notifikácie o udalostiach v aplikácii
- Vytvorili sme dizajn ďalších častí aplikácie

4.3 3. šprint

V týždňovom treťom šprinte sme sa venovali zvyšovaniu kvality kódu produktu - vykonali sme refactoring vo viacerých oblastiach aplikácie a vytvárali sme jednotkové testy.

Z predchádzajúceho šprintu sme dokončili implementáciu nastavení profilu.

4.4 4. šprint

Dvojtýždňový štvrtý šprint obsahoval implementáciu nasledujúcich funkcií:

- Implementovali sme ďalšie dôležité funkcie v rámci realizácie experimentu - správcovia môžu pozývať používateľov, a experiment je možné spustiť.
- Nasadili sme *ElasticSearch*, ktorý nám umožnil implementovať fulltextové vyhľadávanie medzi experimentami a používateľmi spoločne s filtrovaním výsledkov vyhľadávania.
- Vytvorili a implementovali sme dizajn pre uvítaciu stránku a dashboard.

4.5 5. šprint

V rámci týždňového piateho šprintu sme implementovali nasledovnú funkcionálnosť:

- Dokončili sme základnú funkciu produktu - realizáciu experimentu. Implementovali sme možnosť nahrávania výsledkov (súborov) z pohľadu účastníka experimentu, ktoré si môže vlastník experimentu prezerať v rámci správy experimentu; taktiež je možné nahrávať súbory aj od vlastníka experimentu.
- Komentáre k experimentu, ktoré je možné pridávať, upravovať i zmazať.
- Nadviazali sme na základnú implementáciu dashboardu a dokončili sme jeho funkcionálnosť.
- Upravili sme aj texty na uvítacej stránke, čo nadväzovalo na jej dizajn vytvorený v predchádzajúcom šprinte.

5 Používané metodiky

Pri riadení vývoja nášho projektu a produktu sme aplikovali nasledujúce metodiky:

- **Dokumentovanie kódu** - oblasť dokumentovania zdrojového kódu v prostredí webového rámca *Ruby on Rails*,
- **Vytváranie vetiev** - proces vytvárania vetiev postupne od opisu vytvárania vetvy až po ukážku v editore *RubymMine*,
- **Vytvorenie commitu** - proces vytvárania commit správ postupne od opisu až po ukážku v editore *RubyMine*,
- **Vytvorenie pull requestu** - proces odovzdávania zdrojového kódu vytvorením pull requestu, postupne od opisu až po ukážku v nástroji *GitHub*,
- **Testovanie softvéru** - stanovenie vhodných postupov a noriem pre vytváranie a údržbu automatizovaných testov pri vývoji a údržbe softvéru,
- **Nahlasovanie chýb** - postup pre nahlásenie jednoznačne reprodukovateľných chýb v aplikácii,
- **Prehliadka zdrojového kódu** - zabezpečenie krokov, vykonaním ktorých dôjde k zlepšeniu čistoty a konzistentnosti zdrojového kódu a k nájdeniu skrytých chýb,
- **Písanie kódu** - základné princípy ako správne písať kód v programovacom rámci *Ruby on Rails*,
- **Návrh a zobrazovanie formulárov** - definovanie pravidiel a princípov uplatňovaných pri tvorbe formulárov v celom systéme.

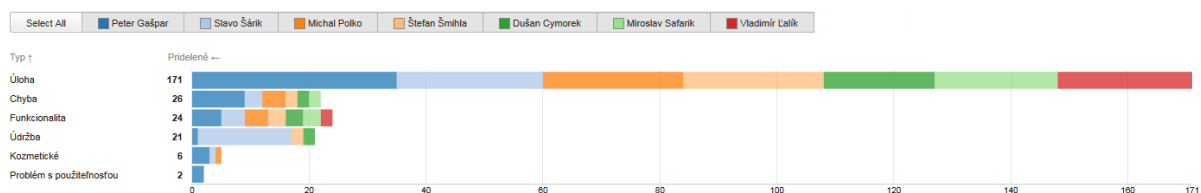
Obsah jednotlivých metodík spolu so všetkými krokmi uvádzame v kapitole č. 9.

6 Globálna retrospektíva - zimný semester

Počas zimného semestra sme sa zamerali na implementáciu základnej a kľúčovej funkcionality pre vytvorenie funkčného prototypu nášho systému. V rámci manažmentu riadenia sme sa taktiež oboznámili s viacerými nástrojmi na skvalitnenie a zlepšenie práce v tíme. Ciele, ktoré sme si stanovili na začiatku zimného semestra (kapitola 2 v Dokumentácii k inžinierskemu dielu), sa nám podarilo úspešne splniť.

Podrobnú dokumentáciu jednotlivých šprintov spolu so zoznamom zrealizovaných úloh uvádzame v prílohe B - *Retrospektívy k šprintom* a v kapitole *Sumarizácie šprintov*. Implementovanú funkcionality opisujeme v dokumente *Dokumentácia k inžinierskemu dielu*.

Na nasledujúcom grafe (obrázok 6.1) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. 6.1: Distribúcia jednotlivých typov úloh medzi autorov

6.1 Časový prehľad

Tabuľka 6.1 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

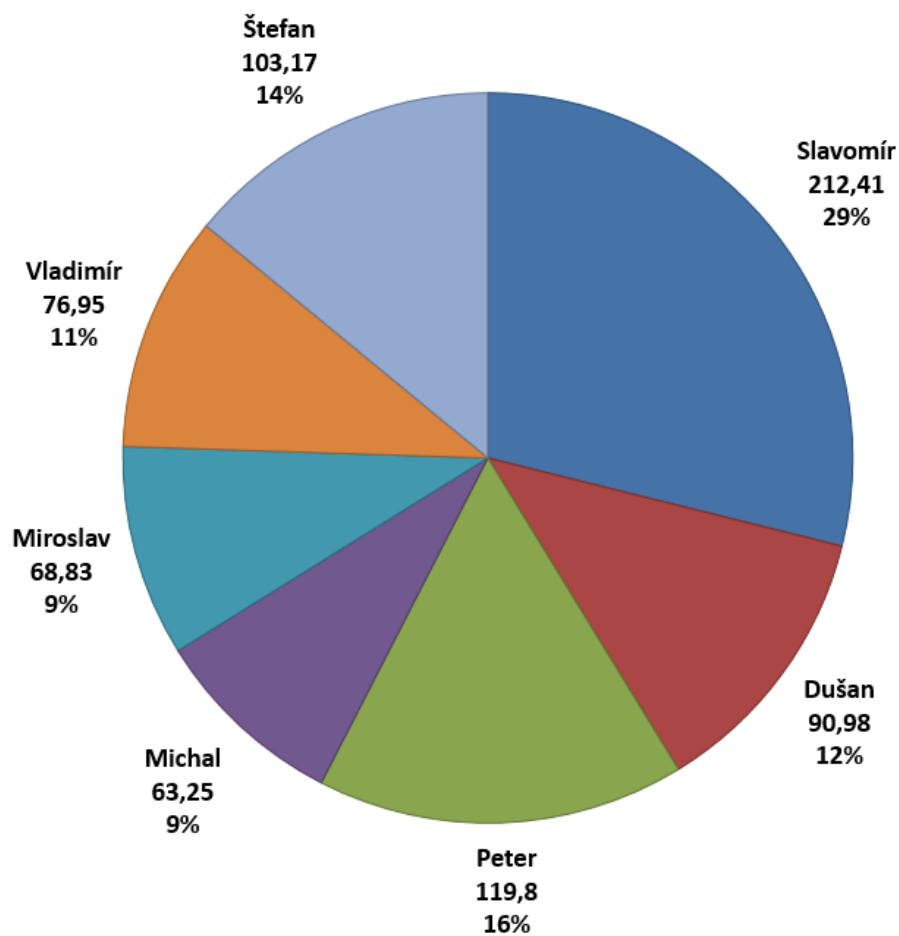
Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	10	41,85
	Documentation	10	44,05
	Testing	1	1,80
	Maintenance	5	3,28
	Celkovo	26	90,98
Michal	Development	13	37,50
	Documentation	5	13,06
	Design	5	12,68
	Celkovo	23	63,25
Miroslav	Development	10	32,92
	Documentation	10	27,00
	Testing	7	7,50

Riešiteľ	Typ	Počet úloh	Počet hodín
	Maintenance	3	1,42
	Celkovo	30	68,83
Peter	Development	29	80,45
	Documentation	8	20,17
	Testing	6	4,4
	Design	6	14,78
	Celkovo	49	119,80
Slavomír	Development	23	94,33
	Documentation	11	50,03
	Testing	4	5,22
	Maintenance	16	62,83
	Celkovo	54	212,41
Vladimír	Development	10	48,52
	Documentation	12	19,90
	Testing	5	8,53
	Celkovo	27	76,95
Štefan	Development	19	72,50
	Documentation	11	24,66
	Testing	3	6,00
	Celkovo	33	103,17

Tabuľka 6.1: Časový prehľad úloh

6.2 Distribúcia celkového času riešenia úloh medzi autorov

V grafe (obrázok 6.2) sa nachádza časové a následne percentuálne vyjadrenie podielu práce pre každého z autorov.



Obr. 6.2: Distribúcia celkového času riešenia medzi autorov

7 Preberacie protokoly

V tejto kapitole sa nachádzajú preberacie protokoly projektu.

PREBERACÍ PROTOKOL

Odovzdávající subjekt: Tím č. 5 - Code Crushers

Preberající subjekt: Ing. Michal Kompan, PhD.

Predmet prebratia: Tímový projekt - Manažment experimentov (Crowdex)

Poznámky:

.....
podpis zástupcu odovzdávajúcej strany

.....
podpis zástupcu preberajúcej strany

V, dňa

PREBERACÍ PROTOKOL

Odovzdávající subjekt: Tím č. 5 - Code Crushers

Preberající subjekt: Ing. Michal Kompan, PhD.

Predmet prebratia: Tímový projekt - Manažment experimentov (Crowdex)

Poznámky:

.....
podpis zástupcu odovzdávajúcej strany

.....
podpis zástupcu preberajúcej strany

V, dňa

8 Zoznam kompetencií tímu

8.1 Úvod

Tento dokument obsahuje predstavenie tímu č. 5 na predmete Tímový projekt v ak. roku 2014/15. Dokument vznikol na Fakulte informatiky a informačných technológií STU v Bratislave. Zároveň sa v ňom nachádza úvodná motivácia a základný návrh riešenia problematiky informačného systému na manažment experimentov.

8.2 Členovia tímu

Bc. Dušan Cymorek – z programovacích jazykov si najviac obľúbil jazyk Java. Menej skúseností pri tvorbe webových aplikácií nahrádza chuťou objavovať a učiť sa nové veci. To sa prejavilo aj pri výbere a následnom vypracovaní bakalárskeho projektu, v ktorom vytvoril automatizovaný softvérový systém založený na RFID technológii v knižnici. V súčasnosti na tomto systéme naďalej pracuje mimo univerzity. Okrem toho sa zaujíma o počítače a počítačové siete a ich správu.

Bc. Peter Gašpar – udomácnil sa vo vodách webových technológií. Skúsenosti s programovaním v jazykoch PHP, MySQL, JavaScript, HTML a CSS uplatnil v bakalárskom projekte Odporúčanie s využitím osobných vyjadrení, ktorého výstupom bola webová aplikácia na personalizované prehrávanie videí. Okrem toho má aktívne skúsenosti s programovacím jazykom Java, v ktorom vytvoril niekoľko študentských projektov. Mimo univerzity sa podieľal na tvorbe komplexného e-shopového informačného systému.

Bc. Vladimír L'alík – má najviac skúseností s tvorbou webových aplikácií najmä v jazyku PHP v kombinácii s MySQL, ktoré nadobudol počas štúdia a popri práci. Ďalej má skúsenosti s vývojom mobilných aplikácií pre operačný systém Android, ktoré nadobudol pri vyvíjaní bakalárskej práce. Okrem toho sa počas štúdia venoval aj jazykom C++, PL/SQL a Java.

Bc. Michal Polko – má skúsenosti s tvorbou desktopových aj mobilných webových aplikácií a najmä s programovaním v jazyku JavaScript, či už na strane klienta alebo servera (Node.js). Okrem webových technológií sa zaujíma aj o počítačovú grafiku. Uvedené znalosti skĺbil aj pri implementácii webovej aplikácie v rámci bakalárskeho projektu Zobrazovanie svetelných polí v prostredí webového prehliadača.

Bc. Miroslav Šafárik – počas predchádzajúceho štúdia nadobudol na rôznej úrovni skúsenosti s jazykmi C++, Java, C# (a vybranými časťami frameworku .NET), SQL a PL/SQL. Trochu sa oboznámil aj s vývojom na platforme Android a pracoval na malom semestrálnom tímovom projekte (5-členný tím) Online zbierka matematických úloh (vyvíjaná s využitím PHP frameworku Yii). Je otvorený novým a moderným technológiám.

Bc. Slavomír Šárik – absolvoval Startup Summer School, kde sa naučil princípy agilného vývoja softvéru, ktoré aplikoval pri práci v startupe. Počas štúdia získal skúsenosti s vývojom

aplikácií v Jave, JSF, Hibernate, taktiež skúsenosti s RubyOnRails webovými aplikáciami a SQL databázami. V bakalárskej práci sa venoval oblasti spracovania prirodzeného jazyka, kde vyvinul a naimplementoval Java aplikáciu schopnú prekladu viet v slovenskom jazyku do SQL databázových dopytov. Zaujíma sa o mobilné technológie a hardware.

Bc. Štefan Šmihla – webovému backendu (ORM, REST API) sa venuje profesne, kde má možnosť rozvíjať znalosti v jazyku Python. Ovláda aj základné znalosti Ruby on Rails, ktoré nadobudol v rámci Výskumne orientovaného seminára. Okrem toho sa rád venuje problémom, pri ktorých je potrebné nejakým algoritmickým spôsobom spracovávať dáta. Toto uplatnil napríklad vo svojej výskumnej bakalárskej práci, pri ktorej skúmal dynamiku písania počas autentifikácie na Android zariadeniach. V tejto práci rozvinul znalosti v jazyku Java. Rád sa učí nové technológie, najmä také, ktoré rozvíjajú spôsob, akým programátor rozmýšľa.

8.3 Kontakt na tím

Tímový mail: fiit.tp05@gmail.com

Kontakty na jednotlivých členov tímu sa nachádzajú v systéme AIS.

8.4 Motivácia

V súčasnej dobe si ako študenti, a zároveň aj výskumníci, uvedomujeme problematiku súvisiacu so získavaním dát potrebných na overenie hypotéz, testovanie riešení a zhrnutie výsledkov. Či už ide o hľadanie participantov pre experimenty alebo zbieranie dokumentov, je potrebné poskytnúť niečo, čo celý tento proces výrazne urýchli a zjednoduší. Našou víziou je poskytnúť svetu nápad, ktorý túto potrebu naplní a daný problém odstráni.

Na základe našich predchádzajúcich skúseností pri vývoji softvérových riešení v rámci bakalárskeho štúdia a ich následnom overovaní sme sa stretli s potrebou zapojiť do tohto procesu i ďalších ľudí. V praxi sme zistili, že získanie potrebného počtu participantov môže byť časovo i technicky náročná úloha. Tento čas by sa však dal efektívnejšie využiť pri ďalšom výskume.

Zvyčajne je pri participantoch dôležitý nielen ich počet, ale aj ich vhodnosť pre daný experiment. Každý človek je jedinečný svojimi osobnostnými črtami, životnými skúsenosťami a ochotou zúčastňovať sa rozličných experimentov. Ďalšími problémami sú nedostatočná propagácia experimentov, prípadne motivácia pre participantov.

Cieľom nášho projektu bude vytvoriť komplexný informačný systém, ktorým sa pokúsime vyriešiť všetky už spomínané problémy vznikajúce pri experimentoch využívajúcich ľudské zdroje. Rozhodli sme sa pre informačný systém, keďže tento druh softvérového produktu skrýva v sebe mnoho nesporných výhod. Medzi inými aj možnosť evidencie a následnej analýzy rozličných údajov, čo sa pri systéme tohto druhu môže odraziť pozitívnym používateľským zážitkom.

8.5 Návrh riešenia

V tejto kapitole uvádzame našu bližšiu predstavu o funkciách, ktoré bude výsledný systémový produkt ponúkať pre koncových používateľov. Ťažiskom informačného systému bude komplexná správa experimentov.

Na jednej strane pôjde o možnosť vytvárať, manažovať a analyzovať experimenty rôzneho druhu zo strany zadávateľov. Na druhej strane budú so systémom aktívne pracovať aj samotní participanti, ktorí si jednoduchou cestou budú môcť vyhľadávať experimenty.

Prvý kontakt so systémom chceme používateľom spríjemniť použitím moderných technológií - akou je napríklad prihlásenie pomocou sociálnych sietí alebo univerzitného konta študenta. Tieto technológie môžu navyše nášmu systému pomôcť získať zaujímavé informácie o vhodných účastníkoch experimentov. Dôležité je pritom zabezpečiť, aby sme mali k dispozícii aktuálne a najmä pravdivé informácie. Nesmieme však zabúdať aj na nevyhnutnosť ochrany osobných údajov, a tiež mieru, do akej bude systém "zvedavý".

Motivátorom pre zadávateľov by mali byť najmä kvalitní participanti, ktorých im náš systém pomôže nájsť. Výhodiskom môžu byť pritom nielen poskytnuté osobné údaje, ale napríklad aj predchádzajúca účasť participantov na experimentoch. S tým úzko súvisí aj kvalitatívne ohodnotenie ich účasti (uvedené nižšie) alebo miera záujmu o účasť na konkrétnom experimente.

Hybnú silu pre participantov budú predstavovať odmeny, ktoré im poskytnú práve zadávateľia experimentov. Našou snahou bude prostredníctvom systému ponechať výber typu a výšky odmeny na zadávateľovi. Príkladom môže byť odmena vo forme finančnej čiastky, alebo darčkového poukazu. Ďalšou možnosťou je motivovanie prostredníctvom interného hodnotiaceho systému (zahŕňajúceho body, odznaky či ocenenia), na základe ktorého bude systém vytvárať zoznamy najlepších používateľov (participantov i zadávateľov).

V neposlednom rade nemožno zabudnúť aj na distribúciu celého systému. Najvhodnejšie riešenie v tomto smere poskytuje použitie webových technológií, a teda vytvorenie webovej aplikácie. Tým zabezpečíme rýchlu a jednoduchú dostupnosť na viacerých platformách, no zároveň môžeme využiť technológie, ktoré čo najviac zvýšia celkový zážitok z používania aplikácie.

8.6 Príloha A: Aktuálny rozvrh všetkých členov tímu

Deň/čas	8.00	9.00	10.00	11.00	12.00	13.00	14.00	15.00	16.00	17.00	18.00	19.00
	-	-	-	-	-	-	-	-	-	-	-	-
	8.50	9.50	10.50	11.50	12.50	13.50	14.50	15.50	16.50	17.50	18.50	19.50
Po												
Ut												
St												
Št												
Pia												

Legenda:

voľný čas nedostupné

Obr. 8.1: Rozvrh hodín celého tímu

9 Metodiky

V tejto kapitole uvádzame všetky metodiky, ktoré vznikli ako výstup z predmetov Manažment v informačných systémoch a Manažment v softvérových systémoch, prípadne vznikli z vlastnej iniciatívy tímu.

9.1 Dokumentovanie kódu

9.1.1 Úvod

Táto metodika je zameraná na oblasť dokumentovania zdrojového kódu v prostredí webového rámca *Ruby on Rails*. Cieľom tejto metodiky je stanoviť jednotné pravidlá a postupy pri dokumentovaní tried, metód a atribútov, uviesť užitočné tipy a prispieť tak k sprehladeniu zdrojového kódu, ktorý je zdieľaný medzi viacerými autormi. Metodika opisuje prácu s dokumentačným nástrojom *YARD* a vývojovým prostredím *RubyMine*. Metodika vychádza z odporúčaných prístupov ku dokumentovaniu zdrojového kódu v jazyku *Ruby* a webovom rámcem *Ruby on Rails*. Je určená pre všetkých členov tímu, ktorí nielen implementujú nové triedy a metódy, ale aj používajú už implementované časti aplikácie.

9.1.2 Zoznam nadväzujúcich metodík a dokumentov

1. <http://www.rubydoc.info/gems/yard/file/docs/GettingStarted.md>

9.1.3 Vymedzenie pojmov a skratiek

- **Dokumentácia** – opis triedy, metódy, atribútu, prípadne časti kódu, ktorý slúži na lepšie pochopenie kódu písaného programátorom.
- **Trieda** – okrem triedy sa tento pojem súhrnne používa aj pre *controller*, *model*, *helper*.
- **Značka** (*angl. tag*) – slúži na definovanie rôznych častí komentárov pre dokumentačný nástroj *YARD*.

9.1.4 Postupy

V tejto kapitole sú uvedené postupy, ktoré je potrebné dodržiavať pri dokumentovaní zdrojového kódu. Nachádzajú sa tu aj užitočné tipy a klávesové skratky, ktoré uľahčujú vytváranie a používanie dokumentácie.

9.1.4.1 Vytvorenie dokumentácie a jej používanie

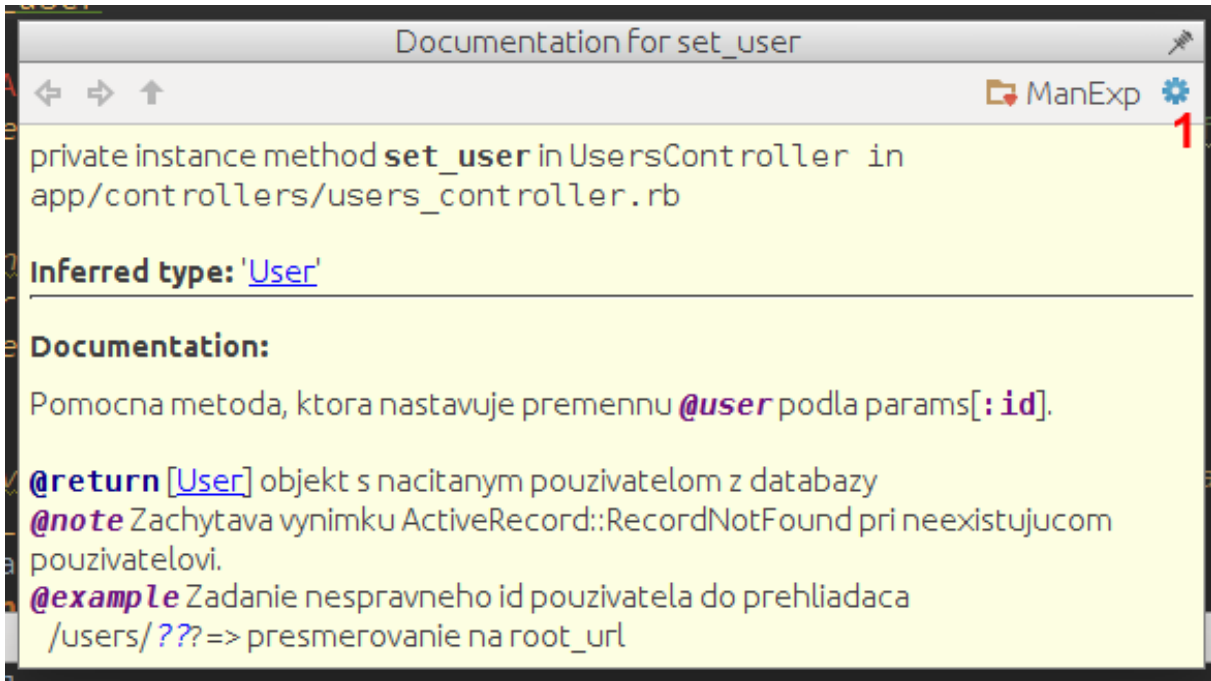
Spôsob vytvorenia dokumentácie závisí od spôsobu jej používania:

- Zobrazenie dokumentácie priamo v prostredí *RubyMine*,

- Zobrazenie formátovanej dokumentácie na lokálnom serveri.

Zobrazenie dokumentácie priamo v prostredí RubyMine

Dokumentácia sa zobrazuje po nastavení kurzora na názov triedy, metódy, prípadne atribútu a stlačení klávesovej skratky *Ctrl+Q*. Dokumentácia sa automaticky obnovuje po úpravách, nie je teda potrebné vykonávať žiadne ďalšie kroky.



Obr. 9.1: Okno s dokumentáciou v prostredí RubyMine

Opis obrazovky:

1. Po kliknutí na ikonu ozubeného kolieska môžeme nastaviť veľkosť textu v okne

Zobrazenie formátovanej dokumentácie na lokálnom serveri

V tomto prípade je dokumentácia zobrazená ako formátovaná webová stránka s možnosťou vyhľadávania. Server sa štandardne nachádza na adrese `http://localhost:8808`. Pre použitie tohto spôsobu zobrazenia je potrebné spustiť server príkazom `yard server -reload`, ktorý zabezpečí automatickú aktualizáciu dokumentácie pri obnovení stránky.

- (User) `set_user` (private) **1** [permalink](#)

Note: Zachytava výnimku ActiveRecord::RecordNotFound pri neexistujúcom používateľovi. **2**

Pomocná metóda, ktorá nastavuje premennú `@user` podľa `params`. **3**

Examples: **4**

Zadanie nesprávneho id používateľa do prehliadača

```
/users/??? => presmerovanie na root_url
```

Returns: **5**

- (User) — objekt s načítaným používateľom z databázy

[Hide source] **6**

```
# File 'app/controllers/users_controller.rb', line 64
64 def set_user
65   @user = User.friendly.find(params[:id])
66   rescue ActiveRecord::RecordNotFound
67     redirect_to root_url, :flash => { :error => t('users.errors.not_found') }
68 end
```

Obr. 9.2: Webová verzia dokumentácie

Opis obrazovky:

1. Názov metódy
2. Poznámka
3. Opis metódy
4. Príklady použitia, v našom prípade ukážka prípadu riešenia požiadavky na zobrazenie neexistujúceho používateľa
5. Návrátová hodnota metódy
6. Odkaz na zobrazenie, prípadne skrytie zdrojového kódu metódy

9.1.4.2 Všeobecné pravidlá pre dokumentáciu kódu

Označenie a forma komentárov:

- Komentáre v zdrojových súboroch sa označujú znakom `#`, za ktorým nasleduje práve jedna medzera.
- Značky pre HTML, CSS a JS komentáre sú od textu komentára oddelené jednou medzerou, forma je nasledovná:
 - HTML komentár:


```
<!-- Komentár -->
<!-- Komentár
  -- na viacero
  -- riadkov
-->
```

– CSS a JS komentár:

```
/* Komentár */
/* Komentár
 * na viacero
 * riadkov
*/
```

- Komentáre umiestňujte vždy nad časťou kódu, na ktorú sa komentár vzťahuje. Toto pravidlo platí celoplošne pre všetky komentáre.
- Komentáre píšete v slovenskom jazyku, s veľkým začiatočným písmenom. V prípade použitia značiek dodržiavajte štýl používaný v uvedených príkladoch.
- Používajte iba povolené značky, pričom dodržiujte ich poradie.

Dokumentácia má vo všeobecnosti nasledovnú štruktúru:

9.1.4.3 Dokumentovanie tried

Pre dokumentovanie tried použite nasledovné značky (v dokumentácii nasledujú v uvedenom poradí):

@author

Význam:

- Uvedenie mena a priezviska autora, @git_username.
- Povinné.

Príklad:

```
# @author Dušan Cymorek, @dusancymorek
```

Ukážka 1: Použitie značky @author

@since

Význam:

- Uvedenie verzie programu, v ktorej bola daná trieda pridaná. Verziu programu určuje manažér integrácie, aktuálna verzia aplikácie (na *Git master vetve*) je uvedená v súbore *README.md*.

- Povinné.

Príklad:

```
# @since 0.2.4
```

Ukážka 2: Použitie značky @since

@note

Význam:

- Poznámka, ktorá môže pomôcť ďalšiemu členovi tímu.

Príklad:

```
# @note Pri vytváraní inicializuje hodnotu atribútu x na 1 (nie nula).
```

Ukážka 3: Použitie značky @note

@deprecated

Význam:

- Vysvetlenie a upozornenie na nahradenie triedy inou triedou, na ktorú je potrebné uviesť referenciu.

Príklad:

```
# @deprecated Zlúčené s controllerom pre triedu X  
# {#controller_pre_trieduX}, z dôvodu deduplikovania kódu.
```

Ukážka 4: Použitie značky @deprecated

@see odkaz popis

Význam:

- Uvedenie odkazu na inú triedu, metódu, atribút, prípadne užitočnú webovú stránku.

Príklad:

```
# @see http://yardoc.org/ YARD
```

Ukážka 5: Použitie značky @see

@todo nadpis

opis

Význam:

- Uvedenie chýbajúcich častí v triede, ktoré je treba implementovať, prípadne opraviť.
- Opis musí byť odsadený o tri medzery od znaku #.

Príklad:

```
# @todo Pridat' prepojenie na tabul'ku user_skills
# Potrebne pred implementaciou vyplnovania skusenosti pouzivatel'a
# v nastaveni profilu.
```

Ukážka 6: Použitie značky @todo

9.1.4.4 Dokumentovanie metód

Dokumentovanie metód realizujte prostredníctvom nasledovných značiek (opäť záleží na poradí):

@param názov [typ] popis parametra

Význam:

- Opisuje vstupný parameter metódy.
- Povinné, ak má metóda vstupné parametre.

Príklad:

```
# @param user_fullname [String] meno a priezvisko pouzivatel'a oddelené
# medzerou
```

Ukážka 7: Použitie značky @param

@option názov Hash parametra [typ] klúč pre parameter (štandardná hodnota) opis parametra

Význam:

- Táto značka sa používa v spojení so značkou @param, ktorý opisuje parameter typu Hash. Značka @option opisuje jednotlivé prvky v množine.
- Povinné, ak má metóda vstupné parametre vo forme Hash.

Príklad:

```
# @param user_params [Hash] parametre triedy User získané z prehliadača
# @option user_params [String] :id friendly-id pouzivatel'a
# @option user_params [String] :firstname meno pouzivatel'a
# @option user_params [String] :surname priezvisko pouzivatel'a
# @option user_params [String] :school ('FIIT') škola pouzivatel'a
```

Ukážka 8: Použitie značky @option

@return [typ] popis návratovej hodnoty

Význam:

- Uvedenie návratovej hodnoty metódy.
- Podobne ako pri značke @param, v prípade viacerých návratových hodnôt je potrebné každú opísať v samostatnej značke @return s vysvetlením, kedy je ktorá hodnota vrátená.

- Povinné.

Príklad:

```
# @return [User] nájdený používateľ podľa id
```

Ukážka 9: Použitie značky @return

@raise [typ] popis výnimky

Význam:

- Uvedenie prípadu, v ktorom metóda vyvoláva výnimku, ktorú neošetruje.

Príklad:

```
# @raise [BudgetBalanceError] ak nemá zadávateľ dostatok kreditu na účte
# na zvýšenie rozpočtu experimentu
```

Ukážka 10: Použitie značky @raise

@note – pozri predchádzajúcu kapitolu

@deprecated – pozri predchádzajúcu kapitolu

@see odkaz popis – pozri predchádzajúcu kapitolu

@todo nadpis

opis – pozri predchádzajúcu kapitolu

@example nadpis

príklad

Význam:

- Uvedenie príkladu použitia metódy, ošetrenia výnimky, prípadne vyvolania výnimky.
- Príklad musí byť odsadený o tri medzery od znaku #.

Príklad:

```
# @example Presmerovanie priradení nesprávneho používateľ'a
# /users/??? =>presmerovanie na root_url, zobrazenie chybovej hlášky
```

Ukážka 11: Použitie značky @example

9.1.4.5 Dokumentovanie atribútov

Pre dokumentovanie atribútov dodržujte nasledovné pravidlá:

- Všetky atribúty musia byť zdokumentované – uveďte na čo atribút slúži.
- Atribúty združujte do logických celkov v prípade, že zastrešujú spoločnú funkcionality. Podobne združujte atribúty spolu so súvisiacimi pomocnými funkciami. Napr.:

```
# Adresy - fakturacna, dodacia
has_many :addresses, dependent: :destroy
accepts_nested_attributes_for :addresses
```

Ukážka 12: Dokumentovanie atribútov

9.1.4.6 Dokumentovanie iných častí kódu

Rozsah ďalších komentárov v kóde sa snažte obmedziť. V prípade, že je nutné vysvetliť úsek kódu, myšlienku, prípadne upozorniť na niečo špeciálne, použite štandardný komentár. Ak je potrebné niečo upraviť, doimplementovať, použite na začiatku komentára slovo **TODO**.

Príklad:

```
def metoda_x
  ...
  # Overenie hash v prípade, že je experiment skryty vo vyhľadavani
  # TODO Refactor - odstranenie zloženej podmienky
  ...
end
```

Ukážka 13: Dokumentovanie iných častí kódu

9.1.4.7 Dokumentovanie HTML, CSS a JS kódu

HTML súbory komentujte nasledovne:

- Na začiatku súboru (pred samotným kódom) uveďte krátky opis obsahu HTML súboru – na čo slúži, čo zobrazuje.
- Nekomentujte jednotlivé prvky kódu, ale logické celky – napríklad formuláre, karty, rozbaľovacie menu.
- Iné komentáre používajte len v krajných prípadoch – ošetrovanie chýb, prípadne potreby dodatočnej úpravy (*TODO*).

Pre **CSS** platia analogické pravidlá. Pre logické celky (jednotlivé bloky so štýlmi) uveďte, na čo sa daný blok používa.

Pre **JS** metódy použite rovnaký štýl dokumentovania ako pre metódy v jazyku *Ruby* (kapitola 9.1.4.4 Dokumentovanie metód).

9.2 Vytvorenie vetvy

9.2.1 Úvod do metodiky

Táto metodika má za cieľ poukázať na problematiku manažmentu zdrojového kódu spolu s jeho verziovaním v projekte Crowdex. Dokument sa bližšie zameriava na proces vytvárania

vetiev postupne od opisu vytvárania vetvy až po ukážku vo vybranom nástroji Git⁹ a postup uvádzaný v editore RubyMine¹⁰. Metodika je určená pre každého člena v tíme, ktorý sa podieľa na implementácii kódu. Cieľom je ukázať a zaviesť jednotný postup pre zahájenie práce na rôznych vetvách počas implementácie, či už nových, existujúcich funkcií alebo ich opravy.

9.2.2 Zoznam nadväzujúcich metodík a dokumentov

Existuje taktiež oficiálna dokumentácia k práci s vetvami počas verziovania softvéru prostredníctvom nástroja Git, bližšie uvedená v podkapitolách 3.3 a 3.4¹¹.

9.2.3 Vymedzenie pojmov a skratiek

- **Vetva** (Branch) – predstavuje istú fázu vývoja zdrojového kódu v Gite
- **Git** – predstavuje voľne dostupný nástroj pre správu verzií
- **Github** – je to služba webová služba, ktorá umožňuje spravovanie gitovských repozitárov, pričom umožňuje ďalšie pokročilé funkcionality spravovania zdrojového kódu úlohu zostavenie buildov a spustenia automatických unit testov nad týmto buildom
- **Produkcia** – stav aplikácie dostupnej pre používateľov
- **Pull request** – predstavuje overený stav vetvy pripravený pre zlúčenie s inou vetvou
- **Repozitár** – úložisko, na ktoré sa nahráva zdrojový kód

9.2.4 Postupy

V tejto kapitole sa nachádza uvedený postup, ako vytvárať vetvy a následne napojenie sa do vetvy. Jednotlivé kroky v metodike sú usporiadané postupne v časovej následnosti a sú záväzné pre každého člena tímu.

9.2.4.1 Pravidlá pre názov novej vetvy

- formát pomenovania vetvy - [typ]-[názov funkcionality]
- typ - predstavuje typ implementovanej funkcionality, napr. feature, fix, refactor
- celý názov vetvy musí byť uvedený v angličtine
- názov musí byť čo najstručnejší, aby vhodne reprezentoval prácu, ktorá sa vykonáva na vetve, najlepšie 1-2 kľúčové slová, žiadne vety alebo súvetia

⁹<http://git-scm.com/>

¹⁰<https://www.jetbrains.com/ruby/>

¹¹<http://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

- názov vetvy obsahuje iba malé písmená
- oddel'ovanie slov v názve pomocou spojovníkov

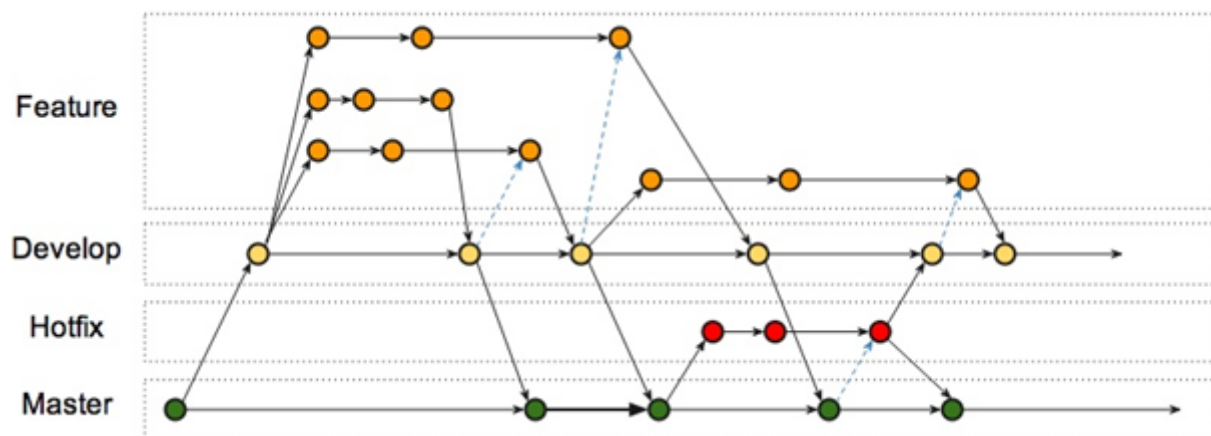
9.2.4.2 Typy vetiev

- **development**
 - vetva slúži ako hlavné prepojenie implementovaných funkcionalít a vetiev, ktoré sa s touto vetvou zlučujú
 - vetva vznikla oddelením od vetvy *master*
 - priebežne sa zlučuje s vetvou pre implementáciu funkcionality
 - zlučuje sa do vetvy *master*, resp. *staging*
- **feature**
 - ide o vetvu, kde prebieha implementácia novej funkcionality softvéru
 - názov vetvy pozostáva z kl'účových slov obsahujúcich danú funkcionalitu
 - vetva vzniká oddelením od vetvy *development*
 - zlučuje sa do vetvy *development*
- **fix** (bug, refactor)
 - táto vetva slúži na opravenie zdrojového kódu malého rozsahu, väčšinou ide o opravenie preklepov, drobných chýb alebo refaktorovanie zdrojového kódu
 - vetva vzniká oddelením od vetvy *development* alebo *master*
 - vetva zaniká zlúčením do vetvy *development* alebo tiež do vetvy *master* pre priamu opravu chyby v produkcii
- **shared**
 - táto vetva vznikla odčlenením od *development* vetvy
 - slúži na vytváranie commitov pre zdieľané časti zdrojového kódu, ako sú migrácie databázy alebo knižnice
 - v prípade, ak došlo v migrácii k úprave existujúcej tabuľky alebo stĺpec, je potrebné upraviť aj model
 - priebežne sa zlučuje do *feature* alebo *fix* vetiev
- **staging**
 - na tejto vetve prebieha testovanie aplikácie a novo pridaných alebo opravených funkcionalít

- vetva sa zlučuje s vetvou *development*
- po otestovaní funkcionality je zlúčená do vetvy *master*
- *master*
 - ide o vetvu, ktorá obsahuje zlúčené a prekontrolované a otestované commity z *development* a *fix* vetiev
 - táto vetva slúži pre nasadenie aplikácie do produkcie
 - vetva sa zlučuje s vetvou *staging*

9.2.4.3 Princíp vetvenia z existujúcej vetvy

Na základe predchádzajúceho rozdelenia typov vetiev je potrebné si pred samotným vytvorením novej vetvy rozmyslieť, ako sa bude vytvárať nová vetva. Treba zvoliť jednu z existujúcich vetiev ako východiskový základ novej vetvy. Po výbere a prepnutí na existujúcu vetvu sa môže oddeliť od tejto vetvy a vytvoriť úplne novú vetvu, ktorá už bude mať základy zvolenej existujúcej vetvy. Napríklad pre implementovanie novej funkcionality sa treba prepnúť na existujúcu vetvu *develop* a následne vytvoriť novú vetvu, ktorá bude vychádzať z vetvy *develop*, pričom treba dodržať konvencie pre pomenovanie vetvy. Po vytvorení commitov v novej vetve a ukončení implementácie sa vytvorí Pull Request a dôjde k zlúčeniu novej vetvy do vetvy *develop*. Názorná ukážka vetvenia v Gite je zobrazená na obr. 9.3.

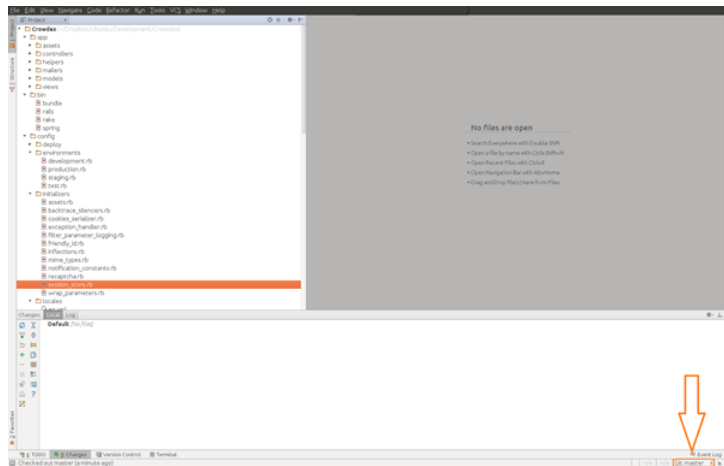


Obr. 9.3: Ukážka schémy vetvenia v Gite¹²

9.2.4.4 Postup vytvorenia vetvy

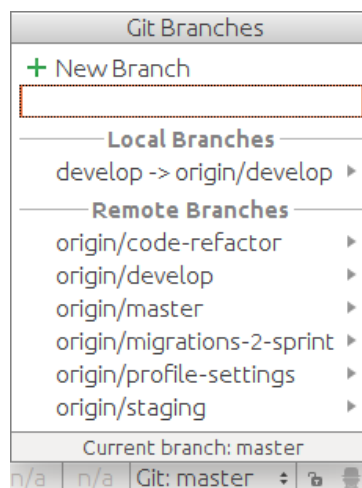
1. Kliknite v editore do pravého dolného rohu pre zobrazenie zoznamu vetiev na obr. 9.4.

¹²<https://blog.networld.to/963/the-beauty-of-git-and-how-it-affects-our-day-to-day-work>



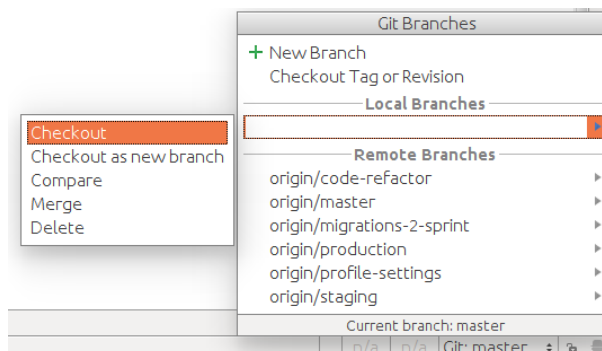
Obr. 9.4: Ukážka prostredia RubyMine

2. Kliknite na jednu z existujúcich vetiev v zozname *Local* alebo *Remote Branches* pre voľbu východiskovej vetvy na obr. 9.5.



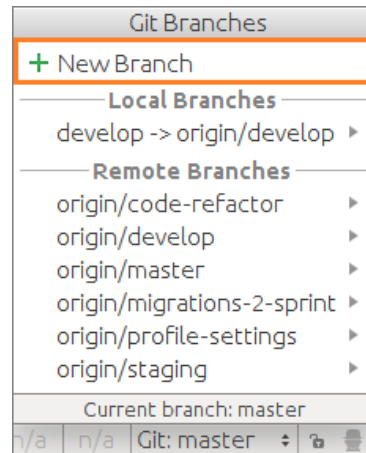
Obr. 9.5: Zoznam existujúcich vetiev

3. Zvoľte možnosť *Checkout* na obr. 9.6.



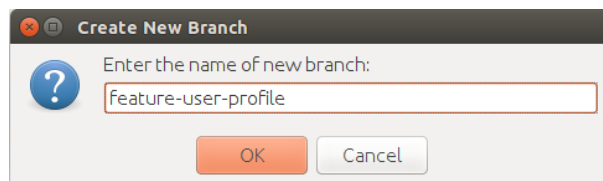
Obr. 9.6: Výber možnosti pre napojenie na existujúcu vetvu

4. Kliknite opäť do dolného pravého rohu na obr. 9.4.
5. Zobrazí sa zoznam vetiev na obr. 9.5.
6. Kliknite na prvý riadok na značku + *New Branch* pre vytvorenie novej vetvy na obr. 9.7.



Obr. 9.7: Výber možnosti pre vytvorenie vetvy

7. Vypĺňte formulár na obr. 9.8 podľa podľa pravidiel nachádzajúcich sa v podkapitole 9.2.4.1.



Obr. 9.8: Formulár pre názov novej vetvy

8. Potvrďte vytvorenie novej vetvy.

9.3 Vytvorenie commitu

9.3.1 Úvod do metodiky

Táto metodika má za cieľ poukázať na problematiku manažmentu zdrojového kódu spolu s jeho verziovaním v projekte *Crowdex*. Dokument sa bližšie zameriava na proces vytvárania commit správ postupne od opisu až po ukážku vo vybranom editore *RubyMine*. Metodika je určená pre každého člena v tíme, ktorý sa podieľa na implementácii kódu. Cieľom je ukázať a zaviesť jednotný postup pre uloženie zdrojového kódu do lokálneho repozitáru.

9.3.2 Zoznam nadväzujúcich metodík a dokumentov

Existuje taktiež oficiálna dokumentácia k práci s commit správami počas verziovania softvéru prostredníctvom nástroja *Git*, bližšie uvedená na oficálnej stránke ¹³.

9.3.3 Vymedzenie pojmov a skratiek

- **Commit** – predstavuje uloženie aktuálneho stavu zdrojového kódu vo vetve
- **Git** – predstavuje voľne dostupný nástroj pre správu verzií
- **Github** – je to služba webová služba, ktorá umožňuje spravovanie gitovských repozitárov, pričom umožňuje ďalšie pokročilé funkcionality spravovania zdrojového kódu úlohu zostavenie buildov a spustenia automatických unit testov nad týmto buildom s inou vetvou
- **Repozitár** – úložisko, na ktoré sa nahráva zdrojový kód

9.3.4 Postupy

V tejto kapitole sa nachádza uvedený postup, ako vytvárať commit správy a následne uloženie zdrojového kódu do lokálneho repozitáru.

9.3.4.1 Pravidlá pre formát správy

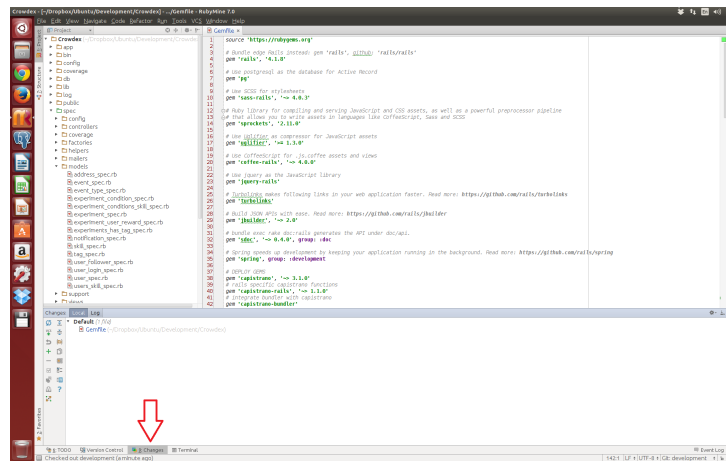
- obsah musí byť uvedený v angličtine
- políčko *title*
 - musí obsahovať názov vykonanej zmeny, napr. vo forme *Updated / Added / Removed / Fixed XY*
 - názov musí byť čo najstručnejší, aby vhodne reprezentoval prácu, ktorá sa vykonala pred vytvorením commitu
 - ide o povinný atribút
 - maximálna dĺžka políčka *title* je obmedzená na 69 znakov
- medzi políčkami *title* a *commit information* sa nachádza jeden prázdny riadok
- políčko *commit information*
 - ide o nepovinný atribút
 - ak z názvu správy nie je jasné, je nutné pridať dodatočnú informáciu o pridanom commite

¹³<http://git-scm.com/docs/git-commit>

Obr. 9.9: Formulár pre commit správu

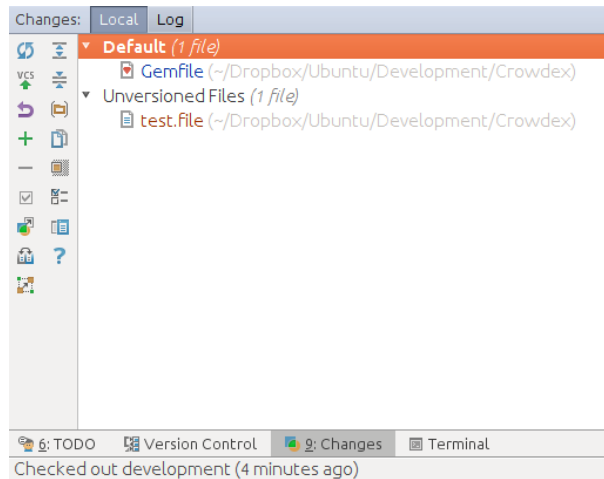
9.3.4.2 Postup vytvorenia commitu

1. Kliknite na tlačidlo *Changes* v editore podľa obr. 9.10.



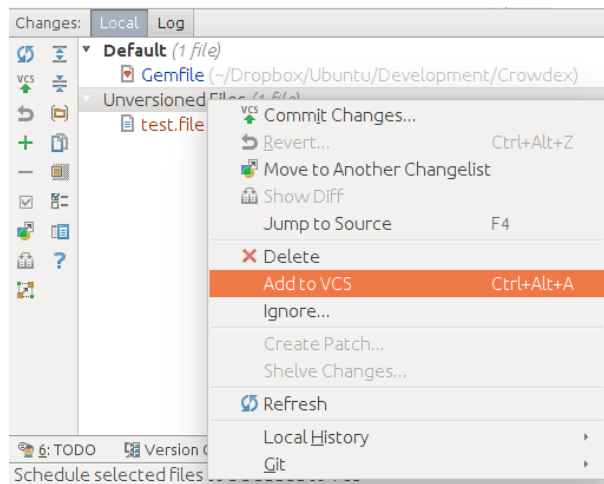
Obr. 9.10: Ukážka prostredia RubyMine

2. Zobrazí sa vám rozhranie pre manažment zmien - obr. 9.11.



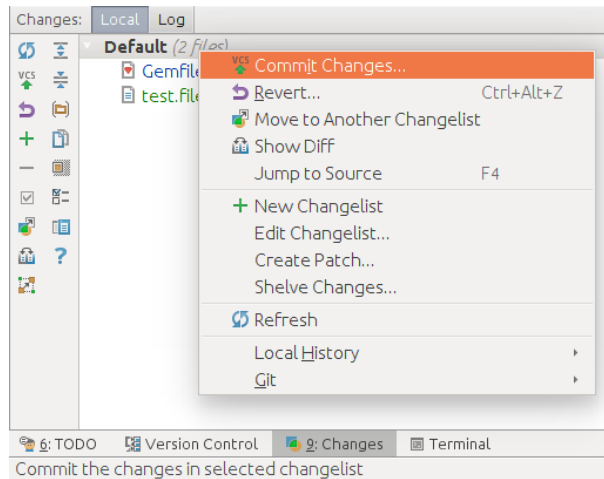
Obr. 9.11: Rozhranie pre manažment zmien

3. Ak ste pridávali nové súbory, kliknite pravým tlačidlom na *Unversioned files* a zvol'te možnosť *Add to VCS* - obr. 9.12.



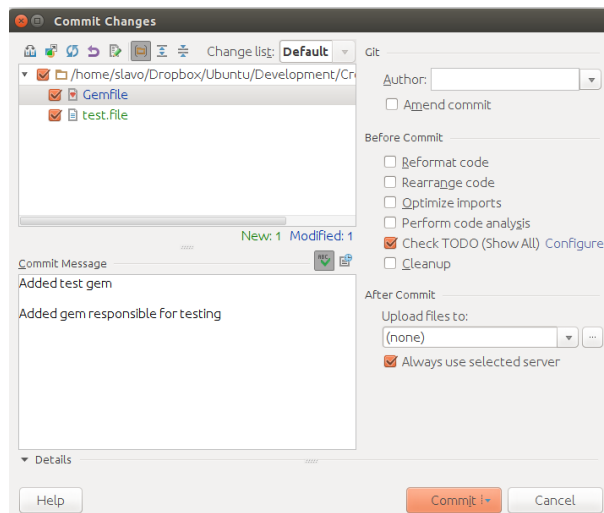
Obr. 9.12: Pridanie súborov do repozitára

4. Kliknite pravým tlačidlom na *Default* a zvol'te možnosť *Commit Changes* - obr. 9.13.



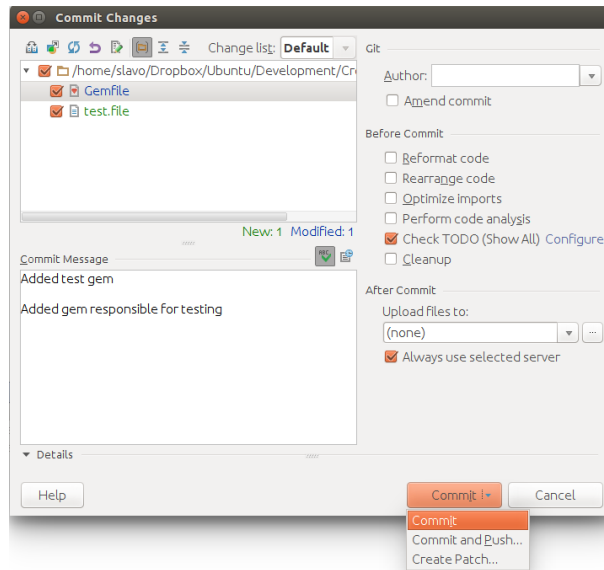
Obr. 9.13: Pridanie zmien do repozitára

5. Vyplňte fomulár na obr. 9.14 podľa a podľa pravidiel nachádzajúcich sa v podkapitole 9.3.4.1.



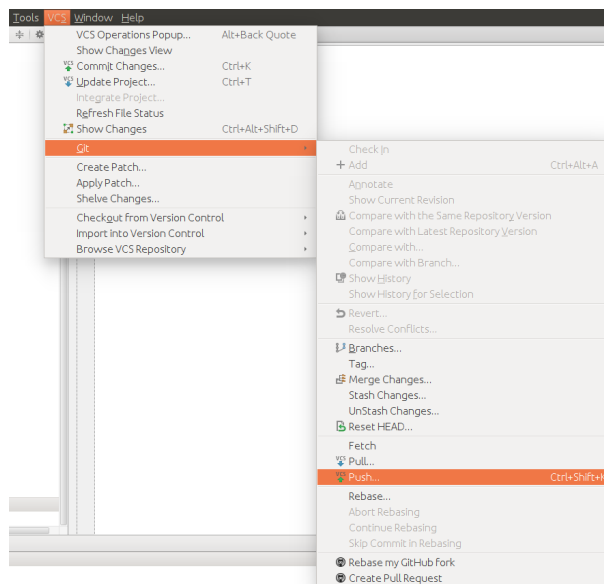
Obr. 9.14: Vyplnenie formulára pre commit správu

6. Kliknite na tlačidlo *Commit* a zvol'te možnosť *Commit* a vytvorte commit - obr. 9.15.



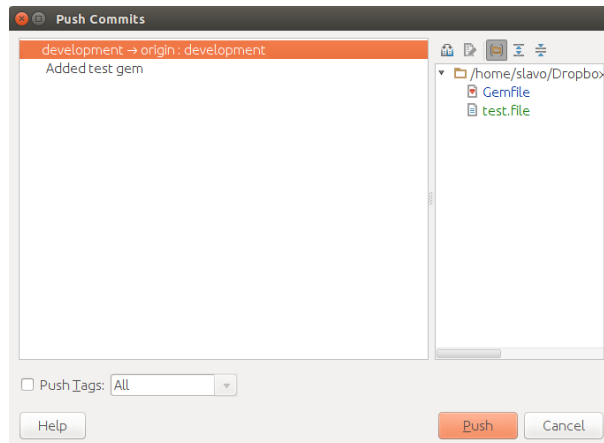
Obr. 9.15: Vytvorenie commitu

7. Kliknite v hornej lište na tlačidlo *VCS* v menu, zvol'te možnosť *Git* a kliknutím na *Push* vytvorte požiadavku na odoslanie commitu do centrálneho repozitára - obr. 9.16.



Obr. 9.16: Odoslanie commitu do centrálneho repozitára

8. Kliknutím na *Push* synchronizujete lokálny repozitár s centrálnym repozitárom - obr. 9.17.



Obr. 9.17: Synchronizácia repozitára

9.4 Vytvorenie pull requestu

9.4.1 Úvod do metodiky

Táto metodika má za cieľ poukázať na problematiku manažmentu zdrojového kódu spolu s jeho verziovaním v projekte *Crowdex*. Dokument sa bližšie zameriava na proces odovzdávania zdrojového kódu vytvorením pull requestu, postupne od opisu až po ukážku vo vybranom nástroji *Github*. Metodika je určená pre každého člena v tíme, ktorý sa podieľa na implementácii kódu. Cieľom je ukázať a zaviesť jednotný postup pre vytvorenie pull requestu.

9.4.2 Zoznam nadväzujúcich metodík a dokumentov

Existuje taktiež oficiálna dokumentácia k pull requestom počas verziovania softvéru prostredníctvom nástroja *Git*, bližšie uvedená na oficiálnej stránke ¹⁴.

9.4.3 Vymedzenie pojmov a skratiek

- **Vetva** (Branch) – predstavuje istú fázu vývoja zdrojového kódu v Gite
- **Build** – zostavenie aktuálnej verzie produktu
- **Commit** – predstavuje uloženie aktuálneho stavu zdrojového kódu vo vetve
- **Git** – predstavuje voľne dostupný nástroj pre správu verzii
- **Github** – je to služba webová služba, ktorá umožňuje spravovanie gitovských repozitárov, pričom umožňuje ďalšie pokročilé funkcionality spravovania zdrojového kódu
- **Kontinuálna integrácia** (Continuous integration) – metóda, ktorá má za úlohu zostavenie buildov a spustenia automatických unit testov nad týmto buildom

¹⁴<http://git-scm.com/docs/git-request-pull>

- **Pull request** – predstavuje overený stav vetvy pripravený pre zlúčenie s inou vetvou
- **Repozitár** – úložisko, na ktoré sa nahráva zdrojový kód

9.4.4 Postupy

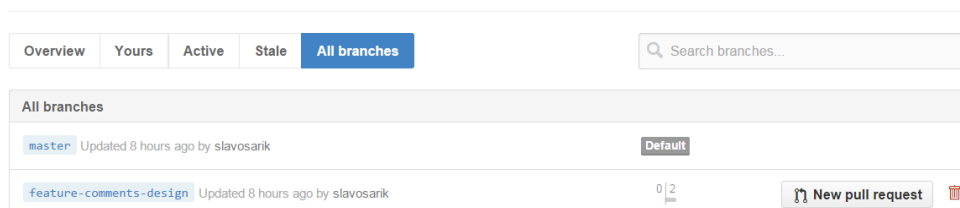
V tejto kapitole sa nachádza uvedený postup, ako vytvárať vytvoriť pull request.

9.4.4.1 Pravidlá pre formát správy pull requestu

- obsah musí byť uvedený v angličtine
- *názov pull requestu*
 - obsahuje názov vetvy, pre ktorú sa vytvára pull request
 - ide o povinný atribút
- *obsah správy*
 - formát správy - [názov funkcionality] (@[autor1], @[autor2])
 - ide o povinný atribút
 - obsahuje záznam o implementovaných funkcionalitách
 - ku každej funkcionalite sa vyžaduje uviesť autora, ktorý implementoval funkcionalitu
 - v prípade, ak ide o viacerých autorov, uvedení sú spoločne pri funkcionalite

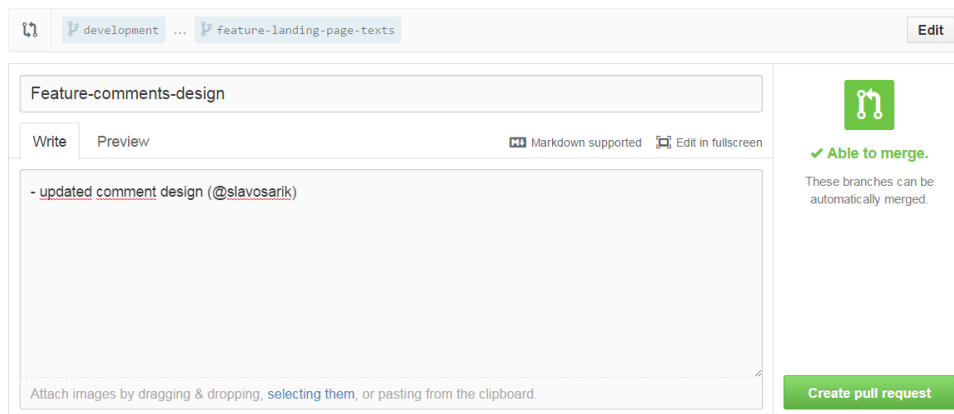
9.4.4.2 Postup vytvorenia Pull requestu

1. Stiahnite si z centrálného repozitára aktuálnu verziu zdrojového kódu vetvy, s ktorou chcete zlúčiť aktuálnu vetvu a vyriešte potenciálne konflikty.
2. Otvorte si webovú stránku služby *Github* - <https://github.com/>.
3. Zobrazte si zoznam dostupných vetiev repozitára.
4. Kliknite na tlačidlo *New pull request* nachádzajúce sa vedľa názvu vetvy, ku ktorej chceme vytvoriť pull request - obr. 9.18.



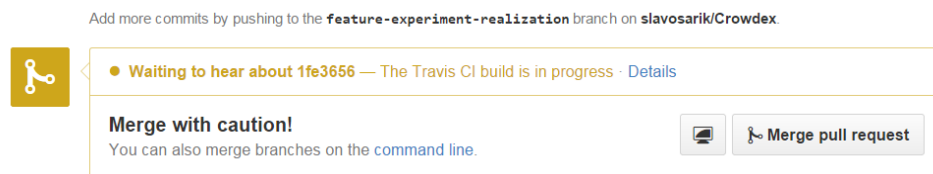
Obr. 9.18: Zoznam dostupných vetiev

5. Vyplňte nasledujúci formulár na obr. 9.19 podľa pravidiel nachádzajúcich sa v podkapitole 9.4.4.1.



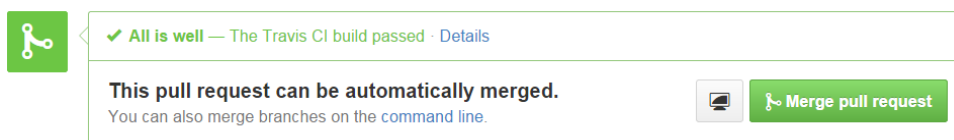
Obr. 9.19: Formulár pre vytvorenie pull requestu

6. Kliknite na tlačidlo *Create pull request*.
7. V prípade, ak automatizovaný proces kontinuálnej integrácie odhalí chybu, musíte túto chybu opraviť. Kliknutím na tlačidlo *Details* - obr. 9.20 zobrazíte správu z testovania obsahujúcu popis chyby. Po opravení chyby sa proces kontinuálnej integrácie spustí odznova.



Obr. 9.20: Priebeh kontinuálnej integrácie

8. V prípade, ak kontinuálna integrácia prebehne úspešne, dôjde k zobrazeniu správy o úspešnom prebehnutí procesu - obr. 9.21.



Obr. 9.21: Úspešné ukončenie procesu kontinuálnej integrácie

9. O zlúčení vetvy do hlavnej vetvy budete notifikovaný. Následne musíte odstrániť zlúčenú vetvu v prípade, ak už na vetve ďalej pracovať už nebudete. Vetvu odstránite vykonáte kliknutím ikonu koša pri vetve označenej popisom *Merged*- obr. 9.22.

Overview	Yours	Active	Stale	All branches	Search branches...
All branches					
development	Updated 9 minutes ago by slavosarik	✓	Default		
experiment-comment-design	Updated 18 minutes ago by mpk	✓	2 0	#42	Merged
feature-landing-page-texts	Updated 27 minutes ago by dusancymorek		8 3		New pull request

Obr. 9.22: Zoznam vetiev po zlúčení vetvy

9.5 Testovanie softvéru

9.5.1 Úvod

Táto metodika sa zaoberá testovaním vyvíjaného softvérového projektu *Crowdex* zameraného na manažment experimentov. Cieľom tejto metodiky je stanoviť vhodné postupy a normy pre vytváranie a údržbu automatizovaných testov pri vývoji a údržbe softvéru, tak ako aj objasniť technológie, ktoré tvorbu testovacích prípadov zjednodušujú. Metodika nepokrýva výukový návod ako písať jednotlivé testy, iba stručné ukážky.

Metodika je určená všetkým osobám, ktoré sú priamo zainteresované vo vývoji projektu *Crowdex*, predovšetkým serverovej časti.

9.5.2 Zoznam nadväzujúcich metodík a dokumentov

Tento dokument nenadväzuje na žiadne metodiky. Pre efektívne písanie testov je však dôrazne odporúčané naštudovať si nasledujúce dokumenty:

- Výukový kurz na prácu s gemom *RSpec*¹⁵
- Konvencie ako správne komentovať a opisovať testy¹⁶

9.5.3 Vymedzenie pojmov a skratiek

V dokumente sú obsiahnuté slovenské pojmy, avšak väčšina zahraničných materiálov používa anglické pojmy. Pre lepšiu orientáciu je v tejto časti obsiahnutý zoznam pojmov a skratiek, aj s ich anglickým ekvivalentom.

- **Testovací prípad** (*angl. Test Case*) - testovací prípad, ktorý zahŕňa drobnú funkcionálnu úroveň určenú na testovanie.
- **Jednotkový test** (*angl. Unit Test*) - drobná porovnávacia podmienka, ktorá overuje správanie daného testovacieho prípadu.

¹⁵<http://rspec.codeschool.com/>

¹⁶<http://betterspecs.org/>

- **Testom riadený vývoj** (*angl. Test Driven Development – TDD*) - spôsob vývoja softvéru, pri ktorom sa najprv napíše testovací prípad a následne sa vyskúša či zlyhá. Po tejto etape nasleduje implementácia funkcionality, po ktorej by testovací prípad mal prejsť a overiť túto funkcionality.
- **Pokrytie kódu** (*angl. Code Coverage*) - údaj, ktorý udáva percentuálne množstvo pokrytia kódu testami. Ukazuje ktoré riadky kódu sú pokryté testami a ktoré nie. Užitočné pri kontrole kvality a vyhodnocovaní rizík vzniku neodhalenej chyby.
- **Code Climate** - služba poskytujúca prehliadku a analýzu kódu.

9.5.4 Postupy

Metodika opisuje použité technológie na testovanie, pokyny na udržiavanie priečinku s testami a v závere udáva pokyny na testom riadený vývoj, ktoré sú všeobecne záväzné.

9.5.4.1 Používané technológie

Na testovanie sú použité nasledujúce technológie (gemy):

- **RSpec** - knižnice určené na testovanie aplikácií v *Ruby on Rails*. Krátku ukážku s vysvetlením predstavuje Obrázok 9.23.
- **FactoryGirl** - pomôcka na zjednodušené vytváranie modelov, ktoré budú následne predmetom testovania. Krátku ukážku s vysvetlením predstavuje Obrázok 9.24.
- **codeclimate-test-reporter** - gem slúžiaci na prepojenie testov so službou *Code Climate* a hodnotenie pokrytia kódu testami.

```

require 'spec_helper'

# Opis triedy, ktorá je určená na testovanie
describe Address do
  # Blok before sa vykonáva pred každým unit testom
  before do
    @address = Address.new(
      street: 'King Alley 7',
      postal_code: 7777,
      city: 'Gondor',
      country: 'Middle-earth'
    )
  end

  # Inštancia triedy, ktorá sa testuje
  subject { @address }

  # Jednotlivé unit testy. Testujú správanie inštancie, ktorá je v bloku
  # subject. Treba mať napamäti, že jednotlivé unit testové bloky sa
  # vykonávajú v náhodnom poradí.
  it { should respond_to(:street) }
  it { should respond_to(:postal_code) }
  it { should_not respond_to(:state) }
  it { should be_valid }

  # Opis testovacieho prípadu
  describe 'empty fields' do
    it { @address.street = ' ' }
    it { @address.postal_code = ' ' }

    # Vykonáva sa po každom unit teste nachádzajúcom sa v príslušnom bloku.
    after { should_not be_valid }
  end
end
end

```

Obr. 9.23: Ukážka s vysvetleniami základných prvkov RSpec

```

require 'spec_helper'

# Ukážka definovania 'továrne' na vytváranie modelu pre adresu.
# Nachádzajú sa module 'spec/factories.rb'.
factory :address do
  # Sequence pri každom volaní metódy inkrementuje premennú 'n'.
  sequence(:street) { |n| "King Alley #{n}" }
  postal_code 7777
  city 'Gondor'
  country 'Middle-earth'
end

describe Address do

  before do
    # Ukážka vytvárania modelu bez pomoci FactoryGirl.
    @address = Address.new(
      street: 'King Alley 7',
      postal_code: 7777,
      city: 'Gondor',
      country: 'Middle-earth'
    )

    # Ukážka vytvárania ekvivalentného model s pomocou FactoryGirl.
    @address = FactoryGirl.build(:address)
  end
end

```

Obr. 9.24: Ukážka vytvárania modelov pomocou FactoryGirl

9.5.4.2 Štruktúra rspec priečinku

Na udržanie prehľadnosti priečinku *rspec* bolo nevyhnutné stanoviť určitú štruktúru. Je podstatné aby ste ju dodržiavali:

- **config/***
Testy v tomto priečinku sú určené na testovanie konfigurácie. Užitočnou utilitou v tomto priečinku je testovací skript *localization_spec.rb* testujúci lokalizačné súbory, najmä či ich štruktúra a jednotlivé kľúče sú identické.
- **helpers/***
Priečinok obsahuje testy určené na overenie funkcionality komplexnejších pomocných metód.
- **mailers/***
Priečinok obsahujúci testy pre automatické odosielanie emailov.

- **models/***
Jeden z najdôležitejších priečinkov. Obsahuje testy, ktoré sú určené na validáciu modelov. Testuje ich správanie, limity, obmedzenia, funkcionálnosť, a podobne.
- **support/***
Priečinkov obsahujúci pomocné metódy, ktoré majú zjednodušovať samotné testovanie. Jedným príkladom takejto metódy je metóda *wait_for_ajax*, ktorá ošetruje čakanie na asynchrónne *ajax* volania.
- **views/***
Tento priečinkov obsahuje testy, ktoré priamo testujú funkcionálnosť stránky (vyplňovanie formulárov, prihlasovanie, funkčnosť odkazov, funkčnosť tlačidiel, a podobne). Taktiež sem patria testy validujúce samotné html súbory (správnosť štýlov, elementov, a podobne).
- **factories.rb**
Súbor obsahujúci “továrne” na vytváranie rôznych modelov. Zjednodušuje vytváranie modelov pri ostatných testoch.
- **spec_helper.rb**
Konfiguračný súbor pre knižnicu *RSpec*. Obsahuje rôzne nastavenia pre testovanie.

9.5.4.3 Postup písania testov

Všeobecne zaužívaný princíp testom riadeného vývoja je písanie jednotlivých prípadov testovania ešte pred samotnou implementáciou. Tento postup je záväzný pre celý tím. Stručný príklad je ilustrovaný na nasledovnom príklade implementácie validácie dĺžky hesla:

1. Napíšete test, pri ktorom sa otestuje pripravovaná funkcionálnosť. Obrázok 9.25 zobrazuje napísaný test, telo triedy *User* je prázdne.
2. Overíte, či daný test spadne. Obrázok 9.26 zobrazuje, že test na validáciu prítomnosti hesla prešiel, ale test, ktorý kontroluje minimálnu dĺžku spadol.
3. Naimplementujete funkcionálnosť. V triede *User* na obrázku 9.27 pribudol riadok, na validovanie minimálnej dĺžky hesla.
4. Overíte funkcionálnosť, pričom testy musia prejsť. Na obrázku 9.28 vidieť, že táto funkcionálnosť funguje a prešla testami.

```

user.rb
class User < ActiveRecord::Base
end

user_spec.rb
class User < ActiveRecord::Base
end

require 'spec_helper'

describe User do
  before { @user = FactoryGirl.build(:user) }

  subject { @user }

  describe 'when password is short' do
    before { @user.password = 'passwd' }
    it { should_not be_valid }
  end
end

```

Obr. 9.25: Napísanie testu pre pripravovanú funkcionálnosť

```

morzeux@morzeux-virtual: ~/Rails/Crowdex
morzeux@morzeux-virtual:~/Rails/Crowdex$ rspec spec/models/user_spec.rb
.F

Failures:

  1) User when password is short should not be valid
     Failure/Error: it { should_not be_valid }
       expected #<User id: nil, email: "bilbo2@baggins.me", firstname: "Bilbo2", surname: "Baggins",
password: "passwd"> not to be valid
       # ./spec/models/user_spec.rb:11:in `block (3 levels) in <top (required)>'

Finished in 0.13905 seconds
2 examples, 1 failure

Failed examples:

rspec ./spec/models/user_spec.rb:11 # User when password is short should not be valid

Randomized with seed 40518
morzeux@morzeux-virtual:~/Rails/Crowdex$

```

Obr. 9.26: Overenie spadnutého testu

```

user.rb
class User < ActiveRecord::Base
  validates :password, length: { minimum: 8 }
end

user_spec.rb
require 'spec_helper'

describe User do
  before { @user = FactoryGirl.build(:user) }

  subject { @user }
  it { should respond_to(:password) }

  describe 'when password is short' do
    before { @user.password = 'passwd' }
    it { should_not be_valid }
  end
end

```

Obr. 9.27: Implementácia funkcionality


```
morzeux@morzeux-virtual: ~/Rails/Crowdex
morzeux@morzeux-virtual:~/Rails/Crowdex$ rspec spec/models/user_spec.rb
**
Finished in 0.20123 seconds
2 examples, 0 failures

Randomized with seed 16659
morzeux@morzeux-virtual:~/Rails/Crowdex$
```

Obr. 9.28: Overenie implementovanej funkcionality

9.5.4.4 Údržba existujúcich testov

Počas vývoja sa môže stať, že existujúce testy popadajú. Je na zodpovednosti každého riešiteľa úlohy, aby po sebe skontroloval testy a opravil tie, ktoré spadli.

Testy môžu spadnúť z viacerých dôvodov:

- Funkcionalita je rozbitá - v tomto prípade je potrebné aby ste funkcionality opravili.
- Funkcionalita je zmenená, ale funguje - v tomto prípade je potrebné opraviť test.
- Funkcionalita je neaktuálna, alebo zrušená - v tomto prípade môžete neaktuálny test odstrániť.

Nie vždy je možné test hneď opraviť. Niekoľko prípadov, kedy test nie je možné z vašej strany opraviť:

- *Ak sa vám nedarí test opraviť, pretože v RSpec nie ste natoľko zdatní a neviete sa tam pohnúť nech robíte čokoľvek.* Obráťte sa na radu na manažment testovania a vyriešite situáciu spoločne.
- *Ak sa vám nedarí test opraviť, pretože funkcionality je stále implementovaná alebo redizajnovaná.* V tomto prípade napíšte *TODO* a pripíšte tam človeka (seba, alebo toho, koho sa to týka), kto by ten test mal opraviť až funkcionality bude funkčná.

9.6 Nahlasovanie chýb

9.6.1 Úvod

Táto metodika sa zaoberá nahlasovaním chýb v nástroji *YouTrack*. Jej cieľom je presne zadefinovať, čo má byť obsahom hlásenia o chybe a ul'ahčiť tak prácu vývojárom.

Metodika pokrýva postup pre nahlásenie jednoznačne reprodukovateľných chýb v aplikácii. Naopak, metodika nepokrýva nahlasovanie vylepšení a požiadaviek na vlastnosti aplikácie, taktiež nepokrýva ďalší životný cyklus chyby od jej nahlásenia – spracovanie a jej uzatváranie.

Metodika je určená pre všetkých vývojárov aplikácie. Nie je určená pre verejnosť, ktorá aplikáciu používa, nakoľko ide o prácu s interným nástrojom.

9.6.2 Zoznam nadväzujúcich metodík a dokumentov

Táto metodika nenadväzuje na žiadnu inú metodiku.

9.6.3 Vymedzenie pojmov a skratiek

- **aplikácia** - webová aplikácia Crowdex.
- **regresia** - chyba, ktorá bola do aplikácie zavedená po jej aktualizácií (v pôvodnej verzii sa nevyskytujúca).
- **vizuálna chyba** - chyba týkajúca sa vzhľadu používateľského rozhrania aplikácie.

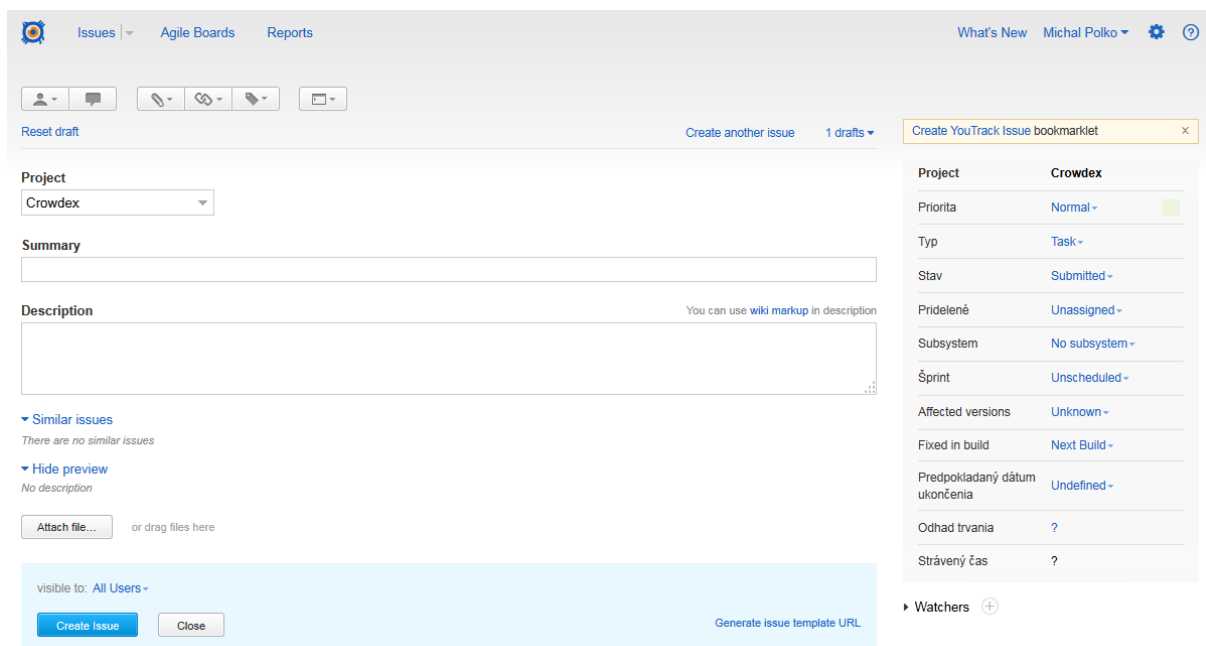
9.6.4 Postupy

V tejto kapitole sú popísané kroky, ktoré musíte vykonať pre nahlásenie chyby v nástroji *YouTrack*.

9.6.4.1 Nahlásenie chyby

Pre nahlásenie chyby:

1. Prihláste sa do nástroja *YouTrack*.
2. Prepnite sa na zoznam chýb pomocou kliknutia na tlačidlo *Issues*.
3. Pred nahlásením chyby:
 - (a) Skontrolujte, či vami nájdená chyba už nebola nahlásená.
 - (b) Ak je to možné, skontrolujte, či vo vývojovej verzii už nebola chyba opravená.
4. Ak chyba ešte nebola nahlásená, kliknite na tlačidlo *Create Issue* vo vrchnom paneli.



Obr. 9.29: Rozhranie pre nahlásenie chyby

V nástroji sa zobrazí obrazovka ako na obr. 9.29. Polia formuláru vyplňte v slovenskom jazyku nasledovným spôsobom:

- **Summary**

- Zadajte názov chyby podľa nasledujúceho pravidla:
 - * Názov komponentu, v ktorom sa chyba vyskytuje (napr. "Prihlásenie alebo "Prídanie experimentu")
 - * Pomlčka
 - * Výstižný názov chyby
 - * Príklad dobrého názvu: "Registrácia – chybné zobrazené tlačidlo pre odoslanie formuláru".
- Po vložení názvu sa v sekcii *Similar issues* objaví zoznam chýb, ktoré majú podobný názov. Každú z nich otvorte a opäť skontrolujte, či sa nejedná o tú istú chybu (duplikát). Ak je už chyba nahlásená, kliknutím na tlačidlo Close nahlasovanie predčasne ukončíte.

- **Description**

- Vložte detailný popis chyby.
- Do tohto pol'a zaznamenajte:
 - * Aké akcie treba v aplikácii vykonať, aby sa prejavila nahlasovaná chyba. Kroky vložte ako očíslovaný zoznam.

- * Čo sa malo udiat' po vykonaní uvedených krokov.
- * Čo sa namiesto toho udialo.
- * Ak ide o vizuálnu chybu, uveďte názov a verziu webového prehliadača a názov a verziu operačného systému.
- Ak sú vám informácie známe, zaznamenajte aj:
 - * Či ide o regresiu, alebo sa chyba prejavovala vždy.

• **Description**

- Prílohu vložte pomocou kliknutia na tlačidlo *Attach file...*
- Ak ide o vizuálnu chybu, priložte k hláseniu aj snímku obrazovky, na ktorom je viditeľná.
- Ak nastáva interná chyba aplikácie, je vhodnejšie vyhotoviť snímku obrazovky namiesto jej kopírovania ako text do popisu.
- Pre ostatné chyby priložte relevantné súbory podľa vlastného uváženia.

• **Priorita**

- Prioritu zvol'te podľa závažnosti chyby a nasledovného kľúča s príkladmi:
 - * *Show-stopper*
 - Nefunkčný server, resp. aplikácia.
 - * *Critical*
 - Bezpečnostná chyba.
 - Chyba ovplyvňujúca väčšinu používateľov.
 - Chyba spôsobujúca stratu údajov používateľov.
 - * *Major*
 - Chyba, ktorá ovplyvňuje malé množstvo používateľov.
 - Problémy s rýchlosťou aplikácie.
 - * *Normal*
 - Všetky ostatné chyby, pre ktoré nie je vhodná iná priorita.
 - * *Minor*
 - Vizuálne chyby, ktoré neovplyvňujú používanie aplikácie.
 - Kozmetické chyby kódu (komentáre a pod.).

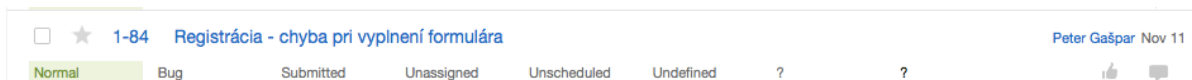
• **Typ**

- Zo zoznamu vyberte *Bug*.

• **Pridelené**

- Ak ide o vizuálnu chybu, zaškrtnite políčka pri používateľoch *Peter Gašpar* a *Michal Polko*.
 - Ak ide o chybu servera, zaškrtnite políčko pri používateľovi *Slavo Šárik*.
 - V ostatných prípadoch ponechajte nevyplnené (hodnota *Unassigned*).
- Ostatné polia ponechajte na ich pôvodných hodnotách, resp. nevyplnené.

Nahlásenie chyby potvrdíte kliknutím na tlačidlo *Create Issue*. Po úspešnom vložení sa chyba zobrazí v zozname chýb na hlavnej stránke nástroja, ako na obr. 9.30.



Obr. 9.30: Úspešne nahlásená chyba

9.7 Prehliadka zdrojového kódu

9.7.1 Úvod

Cieľom tejto metodiky je zabezpečiť kroky, vykonaním ktorých dôjde k zlepšeniu čistoty a konzistentnosti zdrojového kódu a k nájdeniu skrytých chýb. Čistý kód je jeden zo základných predpokladov jeho udržateľnosti, znižuje sa pri ňom počet chýb, znižujú sa riziká v projekte a výrazne zvyšuje šance pre jeho úspech. Hoci riešitelia úloh majú k dispozícii metodiku, ktorá popisuje konvencie pre písanie zdrojového kódu, mnohokrát dochádza k ich porušovaniu z rozličných dôvodov (časový stres, komplexnosť úlohy, zložitosť riešenia a pod.). Z tohto dôvodu je podstatné, aby človek, ktorý danú úlohu neriešil, skontroloval zdrojový kód a rozhodol, či sa v kóde nenachádzajú tzv. pachy, či nie sú v kóde skryté chyby a či je dané riešenie v súlade so špecifikáciou. Metodika nepokrýva spôsoby a kroky, akými vykonať nápravu problematických častí zdrojového kódu (ako vykonať refaktORIZÁCIU).

Táto metodika je určená pre zodpovedné osoby. Okrem iných činností, ktoré tieto osoby vykonávajú (písanie špecifikácie, dohľadnutie na stav plnenia úlohy, vytvorenie úloh v systéme YouTrack), je teda ich zodpovednosťou aj vykonať prehliadku kódu v súlade s touto metodikou. Riešitelia úlohy nevykonávajú prehliadku zdrojového kódu, t.j. priamo sa ich táto metodika nedotýka, no dodržiavaním konvencií v tejto metodike môžu výrazne urýchliť proces ukončenia prác na úlohe.

9.7.1.1 Zoznam nadväzujúcich metodík

Táto metodika sa na určitých miestach odvoláva na nasledujúce metodiky, ktoré vytvorili autori tímu Code Crushers a sú uvedené v tomto dokumente:

- Dokumentovanie zdrojového kódu

- Testovanie softvéru
- Nahlasovanie chýb
- Písanie kódu

9.7.1.2 Vymedzenie pojmov a skratiek

- **Šprint** – časové obdobie (zvyčajne jeden alebo dva týždne) v metodike Scrum, počas ktorého sa pracuje na dohodnutých úlohách.
- **Zodpovedná osoba (ZO)** – osoba, ktorá v danom šprinte zodpovedá za splnenie príslušnej úlohy, resp. príslušných úloh.
- **Riešiteľ** – osoba, ktorá rieši príslušnú úlohu, resp. príslušné úlohy v danom šprinte.

9.7.2 Postupy

V tejto časti opíšeme jednotlivé postupy, ktoré treba vykonať pri prehliadke kódu.

9.7.2.1 Začiatok prehliadky

Stav úlohy

Zodpovedná osoba začne vykonávať prehliadku zdrojového kódu v momente, keď riešiteľ označí v systéme *YouTrack* danú úlohu ako *Fixed*. Informáciu o stave úlohy je možné zistiť dvomi spôsobmi:

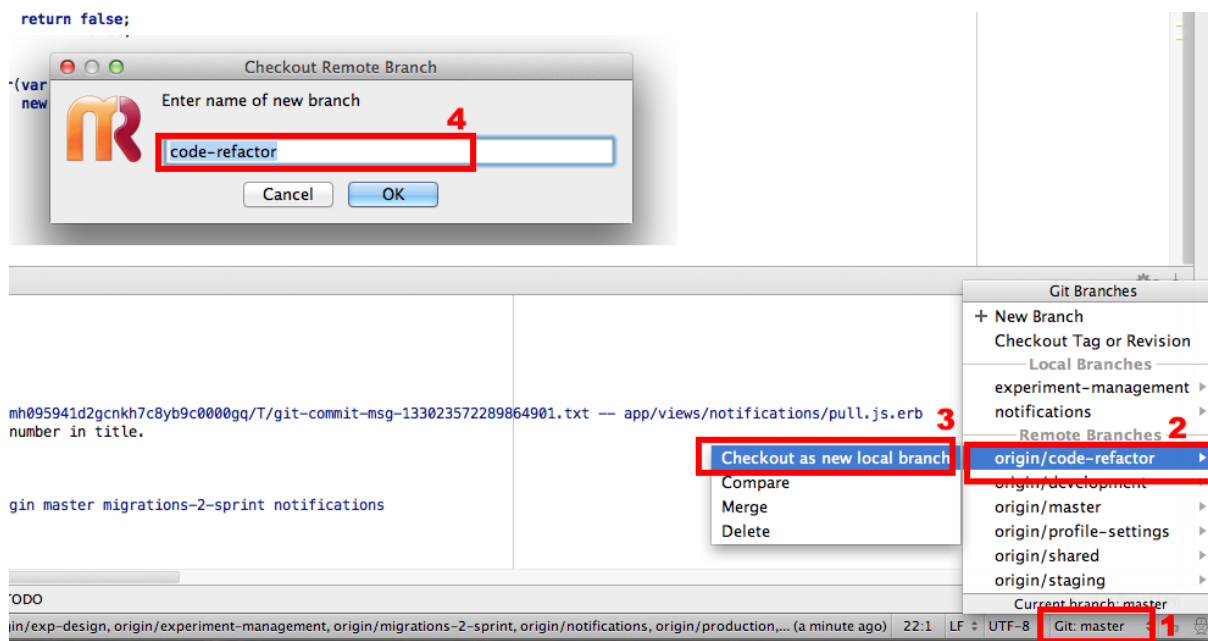
- priebežnou kontrolou emailovej schránky, na ktorú zodpovednej osobe prichádzajú všetky informácie o úlohe (čas riešenia, zmeny stavu úlohy),
- priebežnou kontrolou systému *YouTrack*:
 1. prihláste sa do systému *YouTrack* s Vašimi prihlasovacími údajmi,
 2. kliknite v hornom menu na odkaz *Agile Boards*,
 3. skontrolujte, či je príslušná úloha umiestnená v stĺpci *Fixed*.

Stiahnutie príslušnej vetvy

V ďalšom kroku je potrebné stiahnuť lokálnu kópiu príslušnej Git vetvy, v ktorej bola daná úloha vyvíjaná. Stiahnutie vetvy prebieha v nástroji *RubyMine* (obr. 9.31). Postup stiahnutia vetvy je nasledujúci:

1. kliknite na aktuálnu vetvu v nástroji *RubyMine* v pravom dolnom rohu,
2. kliknite na vzdialenú vetvu, nad ktorou chcete vykonať prehliadku,
3. kliknite na *checkout as new local branch*,

4. zadajte názov pre lokálnu kópiu vetvy, prípadne ponechajte odporúčaný a potvrd' te.



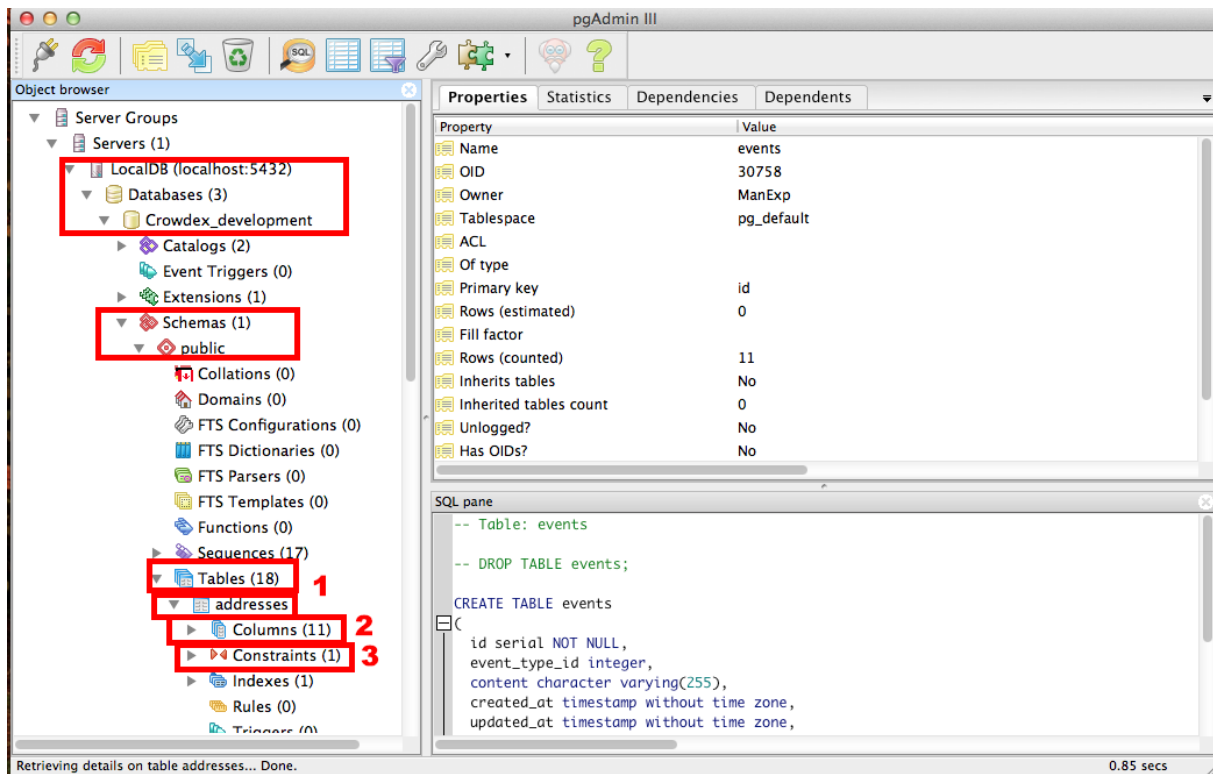
Obr. 9.31: Stiahnutie príslušnej vetvy

9.7.2.2 Súlad so špecifikáciou

V rámci samotnej prehliadky je ako prvé potrebné skontrolovať súlad implementácie so špecifikáciou. Sústrediť sa pritom treba na dátový model – konkrétne názvy tabuliek, názvy stĺpcov a existenciu primárnych kľúčov. Túto kontrolu vykonajte prostredníctvom nástrojov *RubyMine* (ak je to potrebné) a *pgAdmin* (obr. 9.32) v týchto krokoch:

- vykonajte prípadné nevykonané zmeny nad Vaším dátovým modelom prostredníctvom migrácií:
 1. v nástroji *RubyMine* kliknite v kontextovom menu na *Tools*,
 2. vyberte položku *Run Rake Task...*,
 3. vyberte alebo zadajte príkaz *db:migrate*.
- v prípade, že mala databáza obsahovať aj predpripravené statické údaje (uvedené v špecifikácii), postupujte ako v predchádzajúcom bode, pričom namiesto príkazu *db:migrate* spust' te príkaz *db:seed*,
- otvorte si nástroj *pgAdmin* a prihláste sa s Vašimi prihlasovacími údajmi,
- zvol' te si databázový server (zvyčajne *LocalDB*), databázu *Crowdex_development*, rozbal' te príslušné schémy kliknutím na *Schemas* a vyberte schému *public*,
- kliknite na položku *Tables*, čím si si zobrazíte všetky tabuľky, a postupne:

1. skontrolujte, či sa v databáze nachádzajú tie tabuľky, ktoré boli uvedené v špecifikácii a najmä, či ich názvy sú v súlade so špecifikáciou,
2. kliknutím na príslušnú tabuľku v zozname zobrazíte ďalšie možnosti – vyberte možnosť *columns* a skontrolujte, či názvy stĺpcov sú v súlade so špecifikáciou,
3. kliknutím na možnosť *constraints* môžete overiť existenciu primárneho kľúča – musí existovať pre každú tabuľku.



Obr. 9.32: Kontrola dátového modelu v nástroji pgAdmin

9.7.2.3 Čistý kód

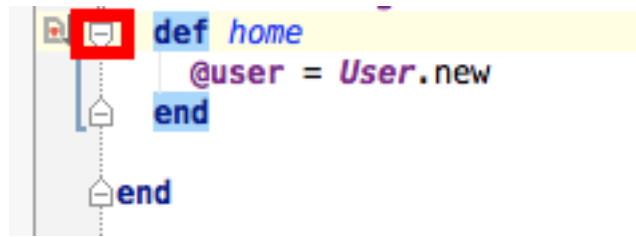
Po kontrole súladu so špecifikáciou treba pristúpiť ku kontrole čistoty kódu. Táto sa vykonáva v programovacom prostredí *RubyMine*. Pri tejto kontrole sa treba zamerať najmä na body uvedené v tabuľke 9.1, no všímať si treba aj prípadné nedodržanie konvencií uvedených v metodike pre písanie zdrojového kódu. V tabuľke je uvedené poradové číslo, ďalej tzv. pach – problematcké miesto v zdrojovom kóde, ktoré vyžaduje pozornosť – podmienky použitia a náprava. Náprava je realizovaná dvomi spôsobmi:

- odstránenie zodpovednou osobou, ak je táto náprava rýchla a neexistuje priestor pre vznik chyby,
- označenie pachu v zdrojovom kóde zodpovednou osobou vo forme komentáru so skratkou *TODO* – zoznam *TODO* komentárov eviduje prostredie *RubyMine*. Nápravu následne vykoná riešiteľ.

Por. číslo	Názov	Podmienky použitia	Náprava
1.	Príliš dlhý riadok v kóde	Riadok presahujúci deliacu čiaru (120 znakov) v nástroji Rubymine.	ZO: zalomte príslušný riadok
2.	Dlhý názov metódy	Názov metódy má dĺžku viac ako 25 znakov.	TODO metoda s dlhým názvom
3.	Metóda s veľkým počtom parametrov	Metóda obsahuje viac ako 3 parametre.	TODO metoda s veľkým počtom parametrov
4.	Príliš dlhá metóda	Metóda pozostáva z viac ako 10 riadkov kódu.	TODO dlhá metoda
5.	Zložitá podmienka	Podmienka obsahuje aspoň 3 logické spojky.	TODO zložitá podmienka
6.	Metóda s logickým parametrom	Metóda obsahuje aspoň 1 logický (boolean) parameter.	TODO metoda s logickým parametrom
7.	Duplikovaný kód	Rovnaké alebo podobné časti kódu na viacerých miestach v zdrojovom kóde.	Všade, kde sa vyskytuje duplikát: TODO duplikovaný kód
8.	Zakomentovaný zdrojový kód	Zdrojový kód obsahuje zakomentované časti.	ZO: odstráňte zakomentovaný kód
9.	Zdokumentovanie metód	Metóda neobsahuje dokumentačný komentár alebo je tento v nesprávnom formáte. Formát určuje metodika pre dokumentovanie zdrojového kódu.	TODO dokumentácia
10.	Komentáre problematických častí kódu	Zložité a problematické časti kódu neobsahujú komentáre.	TODO chýbajúci komentár
11.	Neznáme konštanty	V kóde sa vyskytujú nepomenované konštanty.	TODO – neznáma konštanta

Tabuľka 9.1: *Problematické časti zdrojového kódu*

V prípade kontroly nedostatkov č. 2, 3 a 6 odporúčame skryť telá metód kliknutím na trojuholník v červenom ráme (obr. 9.33).



Obr. 9.33: Skrytie tela metódy vo vývojovom prostredí Rubymine

V prípade nesúlady s konvenciami pre písanie zdrojového kódu treba dané miesto označiť ako *TODO konvencia*.

9.7.2.4 Funkčný kód

Po kontrole čistoty kódu je potrebné skontrolovať aj jeho funkčnosť za účelom odhalenia prípadných chýb. Kontrolu funkčnosti vykonajte všetkými spôsobmi, ktoré povaha úlohy umožňuje – za úplné minimum sa považuje vykonanie kontroly funkčnosti cez zdrojový kód.

Kontrola funkčnosti cez vizuálne testovanie

V rámci špecifikácie sú uvedené akceptačné testovacie prípady. Pokiaľ to charakter úlohy dovoľuje (boli zrealizované príslušné *views*), je potrebné tieto scenáre vizuálne overiť na zrealizovanej implementácii v nasledujúcich krokoch:

- spust' te si na Vašom počítači Rails server:
 - v nástroji *RubyMine* kliknite v kontextovom menu na položku *Run*,
 - kliknite na položku *Run...* a vyberte položku *Development: Crowdex*.
- po naštartovaní servera choď te na adresu *http://localhost:3000/*,
- vykonajte scenáre uvedené v špecifikácii a všímajte si reálny stav s očakávaným stavom.

Kontrola funkčnosti cez zdrojový kód

Ďalší spôsob odhalenia chýb je skontrolovať vybrané časti zdrojového kódu, ktoré sú náchylné na chybovosť. Postupne skontrolujte nasledujúce časti kódu:

- cykly:
 - sústreď te sa pritom najmä na cykly, ktorých telo obsahuje viac ako 4 riadky.
- riadiace podmienky cyklu:
 - dôraz dajte najmä na okrajové hodnoty podmienky a podmienky spĺňajúce bod č. 5 v tabuľke 9.1

- vetvenie:
 - skontrolujte logiku vetviacich podmienok (špeciálne v prípade, že podmienka spĺňa bod č. 5 v tabuľke 9.1),
 - skontrolujte logiku vetvenia – najmä či môže dôjsť k vykonaniu každej z vetiev.
- zamerajte svoju pozornosť na premenné reprezentujúce inštancie triedy, nad ktorými sa vykonáva prístup k atribútom:
 - skontrolujte hlavne fakt, či je ošetrovaná situácia, keď je inštancia triedy *nil*.

Kontrola funkčnosti cez automatizované testy

V prípade, že k danej úlohe boli vytvorené automatizované testy, je možné funkčnosť kódu overiť ich spustením. Spustenie automatizovaných testov vykonajte spôsobom, ktorý nájdete v metodike pre písanie testov.

9.7.2.5 Ukončenie prehliadky

Kontakt s riešiteľmi

Po skončení prehliadky je potrebné kontaktovať riešiteľ a, resp. riešiteľov úlohy a oboznámiť ich s výsledkom prehliadky. V prípade, že pri prehliadke neboli odhalené žiadne nedostatky, nie je nutné tento bod vykonať. Riešiteľov kontaktujte prostredníctvom sociálnej siete Facebook vytvorením skupinového chatu. **Náprava nedostatkov**

V prípade nájdených nedostatkov je potrebné s riešiteľmi vykonať ich nápravu.

Náprava nedostatkov bude vykonaná nasledujúcim spôsobom:

- v prípade odhalenia nesúlady špecifikácie a implementácie, alebo odhalenia chýb počas kontroly funkčnosti opísanej v predchádzajúcej kapitole sa vyžaduje od riešiteľa vykonanie bezodkladnej nápravy, ak to povolia časové možnosti šprintu. V opačnom prípade bude v systéme *YouTrack* do ďalšieho šprintu zaevidovaná úloha typu *bug* v súlade s metodikou pre nahlasovanie chýb,
- pri nedostatkoch č. 1 a č. 8 v tabuľke 9.1 len informujte riešiteľa o tejto skutočnosti a vykonanej náprave,
- pri zvyšných nedostatkoch zostáva záznamom o nich informácia v zdrojovom kóde o *TODO* úlohe. V prípade, že budú od riešiteľa uvedené relevantné dôvody, ktoré ho viedli k danému nedostatku, odstráňte príslušné *TODO* úlohy. V opačnom prípade sa dohodnite s riešiteľom, kedy dané nedostatky odstráni. V prípade, že bude všeobecný súhlas celého tímu spolu s vlastníkom produktu na zrealizovaní špeciálneho šprintu určeného na refaktorizáciu zdrojového kódu, budú tieto nedostatky odstránené v danom šprinte.

Zmena stavu úlohy

Po náprave nedostatkov, resp. po dohode na ich odstránení v ďalšom šprinte, zmeňte stav úlohy na *Verified*. Prehliadka zdrojového kódu sa týmto končí.

9.8 Písanie kódu

9.8.1 Úvod

Metodika popisuje základné princípy ako správne písať kód, konkrétne v programovacom rámci *Ruby on Rails*. Hlavným cieľom je zjednotenie štýlu písania kódu, používanie osvedčených postupov. Dodržiavanie týchto praktík nám pomôže udržať kód prehľadný, zrozumiteľný. Keďže problematika je rozsiahla a samotný rámec *Ruby on Rails* má množstvo zaužívaných konvencií, preto táto metodika popisuje vybrané postupy a situácie, ktoré sa môžu najčastejšie vyskytnúť a nie sú úplne štandardné. Metodika poslúži celému tímu pretože všetci členovia sa podieľajú na vývoji. Momentálne nenadväzuje na žiadnu inú metodiku. Vychádza z odporúčaných postupov, ktoré postupne vytvorili samotní programátori používajúci rámec *Ruby on Rails*. Metodika opisuje vývoj aplikácie v prostredí nástroja *RubyMine*.

9.8.2 Zoznam nadväzujúcich metodík a dokumentov

1. <https://github.com/bbatsov/ruby-style-guide>
2. http://guides.rubyonrails.org/active_record_basics.html#naming-conventions
3. <http://www.toptal.com/ruby-on-rails/top-10-mistakes-that-rails-programmers-make>
4. <http://rubyglasses.blogspot.sk/2007/08/actsasgoodstyle.html>
5. https://github.com/styleguide/ruby?utm_source=rubyweekly&utm_medium=email

9.8.3 Vymedzenie pojmov a skratiek

- **Session** – permanentné sieťové spojenie medzi klientom a serverom.
- **Cookie** – permanentné sieťové spojenie medzi klientom a serverom.

9.8.4 Postupy

9.8.4.1 Rozdelenie funkcií do tried

Keď vytvárate alebo dopĺňate funkcionality je dôležité zachovávať nasledujúce princípy.

Controller

Dbajte na to aby obsahoval len tieto typy aplikačnej logiky:

- **Spracovanie Session a Cookie** – môže zahŕňať autentifikáciu a autorizáciu, alebo spracovanie cookie.
- **Práca s modelom** – logika, ktorá zahŕňa vyhľadanie správneho modelu pomocou parametrov z požiadavky, následne zobrazenie výsledku.

- **Správa parametrov požiadaviek** – spracovanie parametrov požiadavky a následne vyvolanie odpovedajúcej funkcie v modeli, ktorá ich ďalej spracuje (aktualizácia, uloženie).
- **Zobrazenie/presmerovanie** – zobrazenie výsledkov (html, xml, json) alebo presmerovanie.

View

- Snažte sa vyhnúť použitiu vetvených podmienok, aby ste zachovali view prehľadný.
- Vždy keď sa nejaká časť kódu opakuje používajte **Partials** a **Layouts**.

Model

V modeli môžete spracovávať len nasledujúcu aplikačnú logiku:

- **Nastavenie *ActiveRecord*** – relácie, validácia atribútov modelu.
- **Jednoduché metódy** slúžiace na aktualizáciu údajov v databáze alebo uloženie, ktoré by sa mohli zlúčiť s využitím parametrov zlúčte do jednej metódy.
- **Zložité dopyty**, napríklad tie, ktoré zahrňujú viac ako jednoduchý *find* vytvorte ako metódy v modeli. Nepoužívajte zložitejšie dopyty mimo modelu.
- **Obmedzenie prístupu** – skryť interné informácie modelu.

Helpers

Keď pri vytváraní nových funkcionalít a neviete ich zaradiť do predchádzajúcich kategórií môžete vytvoriť triedu odvodenú od **Helper**, kde zoskupíte podobné funkcie a podľa toho túto triedu pomenujete.

9.8.4.2 Tvorba názvov

Premenné

V Ruby môžu premenné obsahovať dáta rôzneho typu. Rozsah ich použitia určuje ich názov. Názov definujte v anglickom jazyku tak aby vysvetľoval význam, pričom slová obsahujú iba malé písmená a na spájanie použijete podtržník. Preto pri vytváraní nových premenných postupujte nasledovne:

- **Premenná inštancie** - názov začína znakom **@** a pokračuje názvom rovnako ako lokálna premenná. Je dôležité ich používať len v prípade posielania dát z *Controller* do *View*, inak ich nepoužívajte.
- **Premenné triedy** – názov začína dvomi znakmi **@@** a za nimi nasleduje meno. Snažte sa vyhnúť použitiu tohto typu premenných.
- **Konštanty** - Názov konštanty obsahuje len veľké písmená a slová sa oddeľujú pomocou podtržníku.

```

# nesprávny názov lokálnej premennej - nevysvetl'uje význam
d = 5

# správne vytvorené názvy
# lokálna premenná
current_user = User.find(1)

# premenná inštancie
@all_users = User.all

# premenná triedy
@@user = user

# konštanta
EXPERIMENT_CREATE = 'new_experiment'

```

Metódy

V názve metódy môžeme použiť špeciálne znaky na konci ako `?`, `!`. Otáznik na konci použite vtedy, keď metóda vracia booleovskú hodnotu napríklad `User#is_logged_in?`. Názvy potenciálne nebezpečných metód zakončíte znakom `!`. Napríklad vtedy ak metóda mení objekt, pre ktorý je definovaná. Nesmiete vytvoriť metódu s `!` na konci ak nie je definovaná metóda bez `!`.

9.8.4.3 Formátovanie zdrojového kódu

Na formátovanie kódu v prostredí *RubyMine* používajte klávesové skratky. `Ctrl+Alt+L` pre automatické formátovanie kódu a `Ctrl+Alt+I` pre automatické odsadenie riadkov. Ak pred použitím týchto skratiek neoznačíte presnú časť kódu, ktorú treba formátovať *RubyMine* automaticky vykoná úpravy pre celý súbor. Nie všetko dokážu vyriešiť predchádzajúce skratky. Pre nasledujúce situácie použite tieto pravidlá:

- Keď vytvárate metódu s preddefinovanými parametrami vložte medzery okolo `=`.

```

def some_method(arg1 = :default, arg2 = nil, arg3 = [])
  # do something...
end

```

- V metódach používajte na oddelenie logických celkov jeden prázdny riadok.
- Ak vytvoríte riadok, ktorý má viac ako 100 znakov, musíte ho rozdeliť.
- Pri zret'azenom volaní metód, ktoré presiahne rozsah jedného riadku postupujte tým spôsobom, že za poslednou metódou v prvom riadku necháte .

```

User.find(1).followers.first.
  is_active?

```

- Ak voláte metódu, ktorá má veľa parametrov a ich dĺžka prekročí maximálnu dĺžku jedného riadku tak uveďte každý parameter na nový riadok. Parametre odsadíte od začiatku volania metódy dvomi medzerami.

```
params.require(:experiment).permit(
  :user_id,
  :string_hash,
  :slug,
  :title,
  :desc,
  :budget,
  :allow_comments,
  :delete_flag,
)
```

Triedy

Keď vytvárate triedu alebo ju dopĺňate o nové metódy, atribúty je nutné aby ste dodržiavali jednotnú štruktúru tried. Ako prvé po hlavičke triedy doplňte triedy, ktoré používate v danej triede. V modeloch je nutné ako prvé deklarovať vzťahy modelu k ostatným. Potom doplňte definície konštánt a hneď za nimi nasledujú atribúty triedy. Potom definujte makrá pre danú triedu a za nimi nasledujú verejné metódy triedy, potom chránené metódy a na koniec súkromné metódy triedy. Na oddelenie jednotlivých definícií používajte jeden prázdny riadok.

```
class UsersController < ApplicationController
  # include ide ako prvý
  include NotificationHelper

  # konštanty
  EXPERIMENT_CREATE = 'new_experiment'

  # atribúty
  attr_reader :education_lvl

  # nasledujú makrá
  before_action :correct_user, only: [:edit, :update]
  before_action :admin_user, only: :destroy

  # metódy triedy
  def index
    @users = User.paginate(page: params[:page])
  end

  # chránené a súkromné metódy su uvedené na konci
  protected

  def admin_user
```

```

    redirect_to(root_url) unless current_user.admin?
end

private

def user_params
  params.require(:user).permit(:name, :email, :password,
                                :password_confirmation)
end
end

```

```

class Micropost < ActiveRecord::Base
  belongs_to :user
  has_many :comments
  # ... nasleduje zvyšok modelu rovnako ako v iných triedach

```

9.8.4.4 Syntax

Podmienky

- Keď vytvárate podmienku, ktorá kontroluje rozsah používajte metódu *Comparable#between* namiesto vytvárania viacnásobných podmienok.

```

# nesprávny spôsob
do_something if User.age > 18 && User.age < 35

# správny spôsob
do_something if User.age.between?(18, 35)

```

- Nepoužívajte *and* a *or* keď vytvárate booleovské výrazy a tak isto ani na riadenie toku, ale použite tieto znaky *&&* a *||*.
- V podmienkach nepoužívať zbytočne *then* aj keď má časť za podmienkov viac riadkov.

```

# nesprávny spôsob
current_user = if is_active then User.find(id) else nil end

# správny spôsob
current_user = is_active ? User.find(id) : nil

```

- Používajte podmienky v jednom riadku všade, kde je to možné.

```

# nesprávny spôsob
if is_logged_in?
  @experiment = Experiment.new
end

```



```
# správny spôsob
@experiment = Experiment.new if is_logged_in?
```

- V podmienkach používajte *unless* namiesto záporných podmienok. Nepoužívajte však *unless* spolu s *else* vetvou. V tomto prípade obráťte podmienku a použite ako prvý pozitívny prípad.

```
# správny spôsob
redirect_to root_url unless is_logged_in?

# nesprávny spôsob
unless is_logged_in?
  redirect_to root_url
else
  @experiment = Experiment.new
end

# správny spôsob
if is_logged_in?
  @experiment = Experiment.new
else
  @experiment = Experiment.new
end
```

- Nepoužívajte zátvorky okolo podmienok.
- Nepoužívajte operáciu priradenia v podmienkach.

Ostatné

- Na inicializáciu premenných, pri ktorých je možnosť, že už boli inicializované predtým používajte `||=`. Inicializácia sa v tomto prípade vykoná iba ak je hodnota premennej *false* alebo *nil*. Nepoužívať v žiadnom prípade na inicializáciu booleovských premenných.

```
@current_user ||= User.find_by(id: session[:user_id])
```

- V triedach nepoužívajte *self* keď to nie je nutné.
- Pri definovaní validačných podmienok sa vyhnite opakovaniu tých istých podmienok.

```
# nesprávny spôsob
validates_presence_of :field_one
validates_presence_of :field_two
validates_presence_of :field_three
validates_presence_of :field_four

# správny spôsob
```

```
validates_presence_of :field_one, :field_two, :field_three, :  
  field_four
```

- Ak budete musieť použiť *SQL* v podmienkach pre vyhľadávanie alebo čokoľvek iné, argumenty pre túto podmienku vždy vkladajte ako premenné.

```
User.where(["name=_?_and_email=_?", name, email])
```

- Blok `{}` používajte keď ide o blok kódu v jednom riadku inak v prípade, že tento blok obsahuje viac riadkov kódu použite blok *do-end*.

Ret'azce

- Keď potrebujete skladať ret'azec z niekoľkých premenných použite spôsob interpolácie ret'azcov namiesto spájania.

```
# nesprávny spôsob  
user_name = user.first_name + '_' + user.last_name  
  
# správny spôsob  
user_name = "#{user.first_name}_#{user.last_name}"
```

- Keď nepotrebujete použiť interpoláciu ret'azcov alebo niektorý zo špeciálnych znakov použite jednoduché úvodzovky v opačnom prípade dvojité úvodzovky.

```
name = 'Jan_Slovak'
```

- Pri vytváraní veľkých ret'azcov nepoužívajte na skladanie ret'azca znakov metódu *String#+*, ale použite metódu *String#<*.

```
table << '<table_class="users">'  
  
users.each { |user| table << "<tr><td>#{user.name}</td></tr>" }
```

9.9 Návrh a zobrazovanie formulárov

Predmetom tejto metodiky je definícia implementácie dizajnových prvkov tímovej webovej aplikácie. Táto metodika sa zaoberá definovaním pravidiel a princípov, ktoré budú členia tímu uplatňovať pri tvorbe formulárov v celom systéme. Kládne sa pritom dôraz na dizajnovú stránku, a to najmä štruktúrne usporiadanie prvkov v jazyku *HTML* s využitím rámca *Ruby on Rails*. Cieľom tejto metodiky nie je objasniť, ako spracovávať výstupy z formulárov a reagovať na chyby spôsobené nekorektným vyplnením takýchto formulárov. Metodika tiež striktné nestanovuje, aké konkrétne formulárové prvky sú adekvátne na zobrazenie určitej požadovanej

informácie. Predpokladá sa základná znalosť programovacieho jazyka Ruby a základná znalosť značkovacieho jazyka *HTML*.

Metodika vychádza zo štandardov webového rámca Bootstrap¹⁷ a základných pravidiel pre tvorbu formulárov využívaných vo webovom rámci Ruby on Rails¹⁸. Základným nástrojom pre jej realizáciu je ľubovoľný **editor kódu Ruby a HTML**. Určená je pritom pre každého člena tímu.

9.9.1 Zoznam nadväzujúcich metodík a dokumentov

1. Problematika tvorby formulárov v Rails aplikácii - http://guides.rubyonrails.org/form_helpers.html

9.9.2 Vymedzenie pojmov a skratiek

- **Asynchrónne volanie** – volanie, ktoré sa vykoná bez nutnosti opätovného úplného načítania a vykreslenia webovej stránky.
- **Entita** – v kontexte tejto metodiky sa pod entitou myslí entita databázového modelu.
- **Náhradný popis** (angl. *placeholder*) – popisuje textový formulárový prvok formou krátkeho pomocného textu; popis sa nachádza na mieste, kde sa očakáva textový vstup zadaný používateľom.
- **Značka** – otváracie (a koncové) pomenovanie prvku zo značkovacieho jazyka *HTML*.

9.9.3 Postupy

Základným prvkom, prostredníctvom ktorého vykreslíte v kóde formulár je pomocná metóda *form_for*. Pri tejto metóde budeme okrem základných parametrov využívať aj pomocný atribút *role* s hodnotou *form*.

V prípade, že vytvárame formulár, ktorého odosielanie sa spracúva cez **asynchrónne volanie**, ďalším potrebným parametrom je *remote* s hodnotou *true*. Pokiaľ to explicitne nevyžaduje špecifikácia, v aplikácii používajte primárne formuláre spracúvajúce vzdialené volania. Výnimkou môžu byť nasledujúce situácie:

- formulár určený na autentifikáciu používateľa,
- formulár spracúvajúci rozsiahle množstvo údajov (napr. textový editor),
- spracovanie finančných alebo akýchkoľvek iných transakcií využívajúcich služby tretích strán.

V týchto situáciách sa rozhodnutie ponecháva na autorovi formulára.

¹⁷<http://www.getbootstrap.com>

¹⁸http://guides.rubyonrails.org/form_helpers.html

9.9.3.1 Základná štruktúra formulára

Vychádzajúc z predchádzajúcich pravidiel, základná štruktúra každého formulára v aplikácii bude nasledovná:

```
<%= form_for(<ENTITA>, html: { role: "form" }) do |f| %>
  ...prvky formulara...
<% end %>
```

Formulár spracúvajúci sa cez vzdialené volanie bude rozšírený o parameter *remote*:

```
<%= form_for(<ENTITA>, html: { role: "form", remote: true }) do |f| %>
  ...prvky formulara...
<% end %>
```

kde *<ENTITA>* je zodpovedajúca entita alebo pomenovanie formulára.

9.9.3.2 Interakcia s používateľom

Pri spracovávaní formulára dochádza k interakcii s používateľom, a preto dodržiavajte nasledujúce pravidlá:

- v prípade, že si spracovanie formulára v aplikácii vyžaduje podporu zapnutého Javascriptu, informujte používateľa o tejto povinnosti pred odoslaním samotného formulára,
- v prípade nesprávneho odoslania formulára informujte používateľa o chybe, a to jedným z nasledujúcich spôsobov:
 - výpisom chyby pri príslušnom problematickom prvku,
 - výpisom zoznamu chýb súhrnne pre celý formulár,
 - zvýraznením príslušného problematického prvku.

Zobrazovanie súhrnných chýb pre celý formulár realizujte prostredníctvom nasledujúceho kódu:

```
<% if <ENTITA>.errors.any? %>
  <div id="error_explanation">
    <h2><%= pluralize(@<ENTITA>.errors.count, "error") %> prohibited this
      <ENTITA> from being saved:</h2>
    <ul>
      <% @<ENTITA>.errors.full\_messages.each do |message| %>
        <li><%= message %></li>
      <% end %>
    </ul>
  </div>
<% end %>
```

kde <ENTITA> je entita, ktorá sa prostredníctvom formulára mení. Výpis chýb pri príslušnom problematickom prvku zahŕňajú ukážky kódov v kapitole **Prvky formulára**.

Pri **zostavovaní chybových textov pre formuláre** sa riad' te nasledujúcimi odporučeniami:

- každá chybová hláška musí mať pre používateľ'a nielen informatívnu hodnotu, ale cieľom je poskytnúť aj možnosť zotavenia z chyby vzniknutej nesprávnym vyplnením formulárového prvku,
- poskytnite informácie o tom, ktorý formulárový prvok bol nesprávne vyplnený,
- v popise chyby vždy uvádzajte názov korešpondujúci s popisným názvom formulárového prvku (angl. *label*).

Príklad:

Popisný názov formulárového prvku: Priezvisko

Korektná chybová hláška: Ľutujeme, ale priezvisko môže obsahovať maximálne 30 znakov.

Nekorektná chybová hláška: Ľutujeme, ale vo formulári sa nachádza nesprávne vyplnený prvok.

9.9.3.3 Prvky formulára

Každý prvok formulára predstavuje samostatný ohraničený podkomponent. **Poradie**, v ktorom sú prvky vo formulári umiestnené nemá pevne stanovené pravidlá, postupujte preto v súlade s nasledujúcimi odporučeniami:

- všetky prvky zorad' te najmä podľa dôležitosti v kontexte konkrétneho formulára:
 - príklad: pri registrácii sa najskôr vyplňa meno, priezvisko a až potom dátum narodenia,
- prvé v poradí uveď te najmä textové políčka, následne zaškrŕavacie a výberové políčka,
- textové plochy umiestnite ako posledné,
- formulár uzatvorte uvedením potvrdzovacích tlačidiel, ktoré slúžia na odoslanie formulára alebo jeho zneplatnenie (obnovu),
- v prípade formulárov vyžadujúcich si informovanie / odsúhlasenie určitých podmienok (napríklad Podmienky používania, Ochrana osobných údajov) neuvádzajte túto informáciu za potvrdzovacími tlačidlami.

Pre každý formulárový prvok uveď te **popis** v rozsahu 1-3 slová:

- umiestnite ho vždy pred (teda naľavo alebo nad) samotný formulárový prvok,
- popis začnite veľkým začiatočným písmenom a zakončite dvojbodkou „:“.

Príklady: *Meno:*, *Priezvisko:*, *Dátum narodenia:*

9.9.3.4 Definícia formulárových prvkov

Nasledujúce ukážky definujú štruktúru pre jednotlivé obsahy formulárových prvkov aplikácie. Vo všeobecnosti platí:

- **<ID>** = identifikátor prvku formulára; v prípade, že o formulár určený pre vytvorenie/úpravu entity, použite názov príslušného atribútu,
- **<POPIS>** = popisný názov formulárového prvku.

Každý prvok musí byť obalený značkou *div* s triedou *form-control*:

```
<div class="form-group">
  ... formularovy prvok ...
</div>
```

9.9.3.5 Textové políčko

Textové políčko používajte pre jednoduché a stručné texty s dĺžkou maximálne 50 znakov.

```
<%= f.label <ID>, <POPIS> %>
<%= f.text_field <ID>, class: "form-control" %>
```

9.9.3.6 Textové políčko využívajúce ikonku

Špeciálnym prípadom je textové políčko, ktoré namiesto štandardného popisu využíva iba náhradný popis, ktorý sa nachádza priamo v textovom políčku. Súčasťou tohto prvku je aj ikonka. Textové políčka s ikonkou používajte iba vo formulároch určených na autentifikáciu.

```
<span class="form-icon_form-icon-<ID_IKONKY>"></span>
<%= f.text_field <ID>, { class: "form-control", placeholder: <POPIS> } %>
```

Pre voľbu *<ID_IKONKY>* si vyberáte z nasledujúcich možností pre význam textového políčka:

- *email* – pre e-mailovú adresu,
- *lock* – pre zadávanie citlivých údajov (heslo, kód overenia),
- *user* – pre meno a priezvisko používateľa.

9.9.3.7 Textová plocha

Textovú plochu používajte pre dlhšie texty, pri ktorých sa očakáva dĺžka viac ako 50 znakov.

```
<%= f.label <ID>, <POPIS> %>
<%= f.text_area <ID>, class: "form-control" %>
```

9.9.3.8 Výberové políčko (zoznam)

Výberové políčko slúži na výber jednej z viacerých možností. Tieto možnosti sú pritom spočiatku skryté a používateľovi sa zobrazia až po kliknutí na políčko.

```
<%= f.select(<ID>, <MOZNOSTI>, { }, { :class => "form-control" }) %>
```

<MOŽNOSTI> nahraďte pol'om možností v tvare: [<POPIS>, <ID_MOZNOSTI>] ako parameter metódy *options_for_select*.

Príklad:

```
<%= f.select(:pohlavie, options_for_select([ ["muz", 1], ["zena", 2] ])) %>
```

9.9.3.9 Výberové políčko (vymenovaním prvkov)

Výberové políčko slúži na výber jednej z viacerých možností. Tieto možnosti sú pritom vždy viditeľné.

```
<%= f.label <ID>, <POPIS> %>
<div class="radio_radio-default">
  <%= f.radio_button <ID>, <HODNOTA_1>, :checked => true %>
  <%= f.label <ID>, <POPIS_HODNOTY_1>, :value => <HODNOTA_1> %>
  ...
  <%= f.radio_button <ID>, <HODNOTA_N> %>
  <%= f.label <ID>, <POPIS_HODNOTY_N>, :value => <HODNOTA_N> %>
</div>
```

Pri rozhodovaní medzi výberovým políčkom formou vymenovania prvkov alebo formou zoznamu sa riad'te pravidlom:

- počet prvkov pri políčku formou vymenovania prvkov môže byť najviac 5 prvkov (N najviac 5),
- **inak** sa použite políčko formou zoznamu.

Toto pravidlo sa **neuplatňuje** v nasledujúcich situáciách:

- v kontexte formulároveho prvku sa vyžaduje, aby boli všetky možnosti vždy viditeľné,
- formulárový prvok generuje systém na základe používateľských nastavení, ktoré si explicitne vyžadujú použitie zoznamu (príklad: používateľsky generovaný dotazník).

9.9.3.10 Zaškrtávacie políčko

Zaškrtávacie políčko používajte na potvrdenie výroku alebo výber aspoň jednej z viacerých možností.

```
<div class="checkbox_checkbox-default">
  <%= f.check_box <ID> %>
  <%= f.label (<ID>, <POPIS>) %>
</div>
```

9.9.3.11 Tlačidlo

```
<%= f.submit <POPIS>, class: "btn_btn-<TYP>" %>
```

Pri tlačidle rozlišujte *<TYP>* podľa dôležitosti akcie, ktorú tlačidlo vykonáva:

- *primary* – fatálna zmena v systéme - uloženie údajov, odstránenie položky,
- *success* – vyjadruje úspešný prechod do nového stavu,
 - tento typ používajte v prípade, že tlačidlo umožňuje vykonanie dvojstavovej činnosti (napr. sledovať používateľa, označiť položku ako „páči sa mi to“) a došlo k úspešnému prechodu do druhého stavu (napr. používateľ je sledovaný, položka sa používateľovi páči),
- *default* - presmerovanie na inú stránku, vymazanie formulára a všetky ostatné nešpecifikované aktivity.

9.9.3.12 Zobrazenie chýb a pomocný text

Pomocný text používajte v prípade, že potrebujete používateľa informovať:

- o ohraničeniach vyplývajúcich z formulároveho prvku (maximálna dĺžka, obmedzenia z hľadiska použitia číslíc, písmen),
- o podrobnostiach súvisiacich s vyplnením formulároveho prvku (príklad: „zaškrtnutím tejto možnosti sa vám zablokuje prístup k experimentu“).

```
<span class="help-block">
  <POMOCNY_TEXT>
</span>
```

Pre **výpis chýb** pre jednotlivé formulárove prvky použite nasledujúci kód:

```
<div class="form-control-errors" id="errors-<ID>" style="display:_none"><
  div class="arrow"></div><div class="errors"></div></div>
```

Uvedené kódy umiestnite pred uzatváraciu značku `</div>` s triedou *form-control*.

Príklad:


```
<div class="form-control">
  ... formularovy prvok s ID password...
  <span class="help-block">Heslo musi obsahovat 4 - 12 znakov.</span>
  <div class="form-control-errors" id="errors-password" style="display:
    none"><div class="arrow"></div><div class="errors"></div></div>
</div>
```

A Zápisy zo stretnutí

A.1 Zápis č. 0 z neformálneho stretnutia tímu

Dátum stretnutia: 23.9.2014

Autor: Bc. Dušan Cymorek

Overovateľ: -

A.1.1 Priebeh stretnutia

- Diskusia o prvotnom návrhu riešenia
- Práca na dokumente Zoznam kompetencií tímu

A.2 Zápis č. 1 zo stretnutia tímu

Dátum stretnutia: 24.9.2014

Autor: Bc. Dušan Cymorek

Overovateľ: -

A.2.1 Priebeh stretnutia

- Poznámka: na nultom stretnutí, ktoré sa konalo dňa 22.9.2014 po úvodnej prednáške z predmetu Tímový projekt, vedúci tímu Ing. Michal Kompan, PhD. priblížil svoju predstavu o riešení projektu.
- Úvod do povinností na predmete Tímový projekt a ukážka dokumentácie projektu z minulých ročníkov.
- Diskusia o prvotnom návrhu riešenia.
 - Tím sa zhodol na potrebe motivácie používateľov, ktorá môže byť okrem finančnej odmeny aj vo forme bodového ohodnotenia, na základe ktorého sa budú vytvárať zoznamy najlepších používateľov.
 - Načrtnutie potreby odolnosti systému voči útokom (napr. zvyšovaniu získaných bodov).
 - Návrh zakomponovať do systému dotazníky, ktoré by umožnili zadávateľom získať dodatočné informácie o participantoch. Systém by mal ponúkať možnosť použiť preddefinované i vytvoriť vlastné dotazníky.
- Určenie frekvencie a termínu stretnutí. Stretnutia sa štandardne uskutočnia v týždňových intervaloch v utorok o 13:30.

A.2.2 Úlohy do nasledujúceho stretnutia

- Stiahnuť a sfunkčniť nejaký systém na podporu vývoja a management projektov (odporúčané Redmine alebo Jira).
- Porovnať existujúce riešenia (napr. Amazon Mechanical Turk) a vypracovať špecifikáciu požiadaviek (najmä v čom sa chceme odlíšiť od konkurencie).

A.3 Zápis č. 2 zo stretnutia tímu

Dátum stretnutia: 29.9.2014

Autor: Bc. Slavomír Šárik

Overovateľ: -

A.3.1 Priebeh stretnutia

- Kontrola úloh z predošlého týždňa – zistenie informácií o konkurencii
- Diskusia o možných funkcionalitách systému
 - priame oslovenie participantov prostredníctvom emailových notifikácií
 - vyplácanie participantov vd' aka kreditu
 - zvýšiť motiváciu – súť až pre najlepšieho
 - spôsob provízie pri nefinančných odmenách
 - systémom vypočítaná automatická výška odmeny participantov
 - odporúčanie výšky odmeny na základe podobných experimentov
 - import výsledkov z iných systémom
 - štatistika výkonnosti participantov
 - zahrnutie dotazníkov v systéme
 - nahrávanie súborov – výsledkov experimentu
- Diskusia o dodatočných funkcionalitách systému
 - sledovanie pohľadu participantov pri vykonávaní experimentu
 - sledovanie biometriky participantov

A.3.2 Úlohy do nasledujúceho stretnutia

- Vymyslieť názov tímu
- Vypracovať plagát tímu – zahrnúť názov, logo a číslo tímu
- Dokončiť stránku tímu
- Načrtnúť obrázkový prototyp systému
- Vytvoriť dokument so špecifikáciou požiadaviek
- Vytvoriť predbežný plán úloh
- Diskusia k TP CUP
- Vytvoriť projektový denník

A.4 Zápis č. 3 zo stretnutia tímu

Dátum stretnutia: 7.10.2014

Autor: Bc. Vladimír L'alík

Overovateľ: -

A.4.1 Priebeh stretnutia

- Kontrola úloh z predošlého týždňa
 - vytvorený názov tímu
 - vytvorená stránka tímu a doplnená o reálne texty a obrázky
 - každý sa priebežne oboznamuje s rámcom *Ruby on Rails*
- Diskusia o návrhu používateľského prostredia, spoločne sme vytvorili prototypy dizajnu
 - úvodná stránka
 - používateľský profil
 - detail experimentu
 - detail používateľa
 - detail zaregistrovanej firmy
 - výsledky vyhľadávania experimentov
- Diskusia o dodatočných funkcionalitách systému
 - notifikácie používateľov pri niektorých udalostiach v systéme
 - správy v systéme slúžiace na kontakt používateľov
 - vytvoriť možnosť používateľovi sledovať činnosť iných používateľov
 - vytvoriť rozhranie pre administrátora systému
- Spoločne sme vytvorili backlog s úlohami, ktorým sme postupne určili prioritu
- Diskusia o softvére pomáhajúceho s manažovaním projektu
 - odhlasovali sme si používanie softvéru *YouTrack*

A.4.2 Úlohy do nasledujúceho stretnutia

- Navrhnuť dátový model
- Vytvoriť časový plán
- Pokračovať v učení sa rámca *Ruby on Rails*

A.5 Zápis č. 4 zo stretnutia tímu

Dátum stretnutia: 14.10.2014

Autor: Bc. Miroslav Šafárik

Overovateľ: Bc. Štefan Šmihla

A.5.1 Priebek stretnutia

- Kontrola úloh z predchádzajúceho týždňa
 - vytvorený „papierový“ prototyp (príloha A)
 - vytvorený predbežný dátový model (diagram v prílohe B a na Google Drive, textová verzia na Google Drive)
 - vytvorený produktový backlog spolu s prioritou jednotlivých úloh a približným rozdelením do jednotlivých šprintov
- Riešené úlohy
 - diskusia o rozdelení manažérskych rolí v tíme
 - * pravdepodobne bude súvisieť s MSI/MIS, o čom sa budeme informovať
 - príprava na 1. šprint
 - * dekompozícia kl' účových úloh z produktového backlogu zaradených do 1. šprintu na jednotlivé podúlohy
 - * hlasovanie o náročnosti jednotlivých úloh
 - * pridelenie náročnosti jednotlivým úlohám
 - * zavedenie zodpovedných osôb za jednotlivé úlohy
 - * rozdelenie úloh medzi jednotlivých členov tímu (riešiteľ'ov)
 - * rozdelenie úloh uvádza tabuľka č. A.1
 - * tieto informácie boli zapísané aj do produktového backlogu
 - M. Kompan bol pozvaný na Google Drive a tiež do skupiny na Facebooku
 - prebehla krátka diskusia o používaní Perconiku

Úloha	Zodpovedná osoba	Riešitelia
Jazykové nastavenia	Dušan	Slavomír
Registrácia nového používateľ'a	Štefan	Slavomír, Štefan
Autentifikácia používateľ'a (štandardne) (+ obnovenie zabudnutého hesla, overenie prekročenia maximálneho počtu pokusov)	Slavomír	Štefan

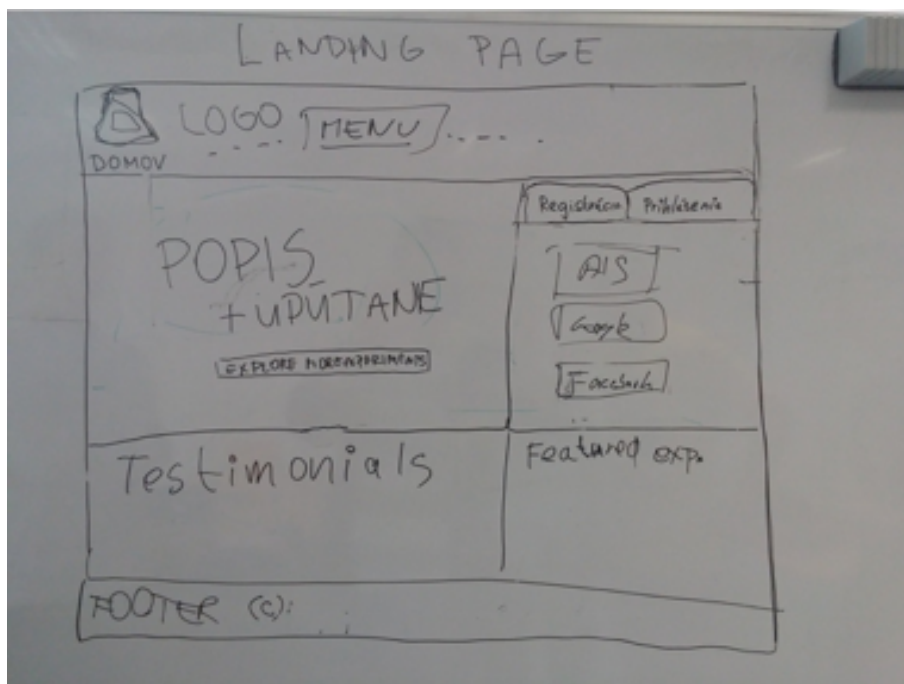
Úloha	Zodpovedná osoba	Riešitelia
Pridávanie experimentov	Vladimír	Miroslav, Peter
Detail profilu používateľa	Miroslav	Dušan, Vladimír
Šablóna - farby, základné prvky	Peter	Michal, Peter
Šablóna - horný panel, bočný panel	Michal	Michal, Peter
Šablóny - footer	Peter	Michal, Peter

Tabuľka A.1: Rozdelenie úloh pre 1. šprint

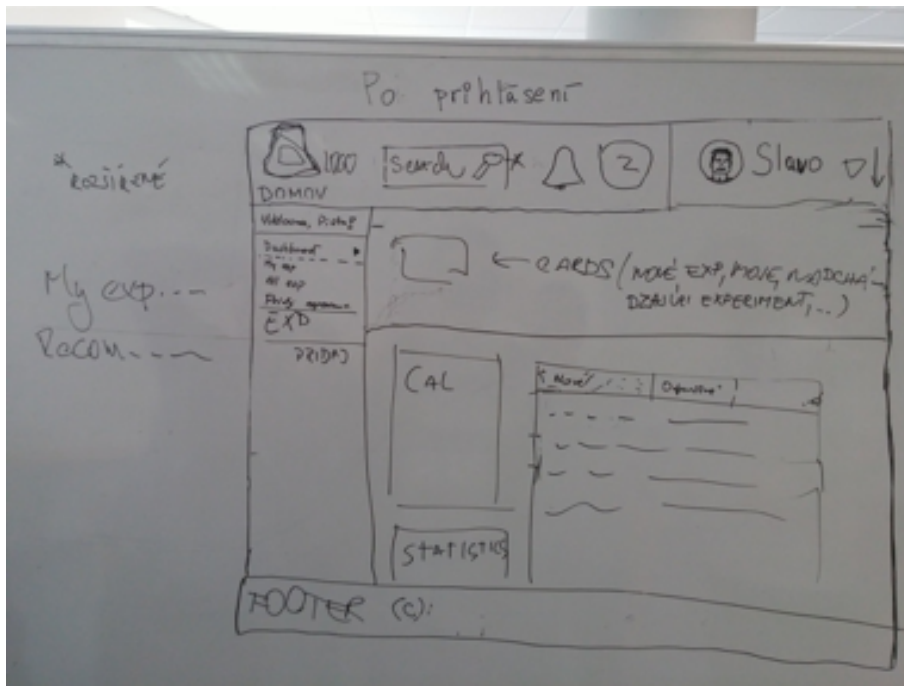
A.5.2 Úlohy do nasledujúceho stretnutia

- Rozdeliť jednotlivé podúlohy z úloh medzi daných riešiteľov
- Zapísať jednotlivé úlohy a podúlohy do *YouTracku*
- Začať pracovať na jednotlivých úlohách
- Prihláška do TP Cupu

A.5.3 Príloha A - papierový prototyp



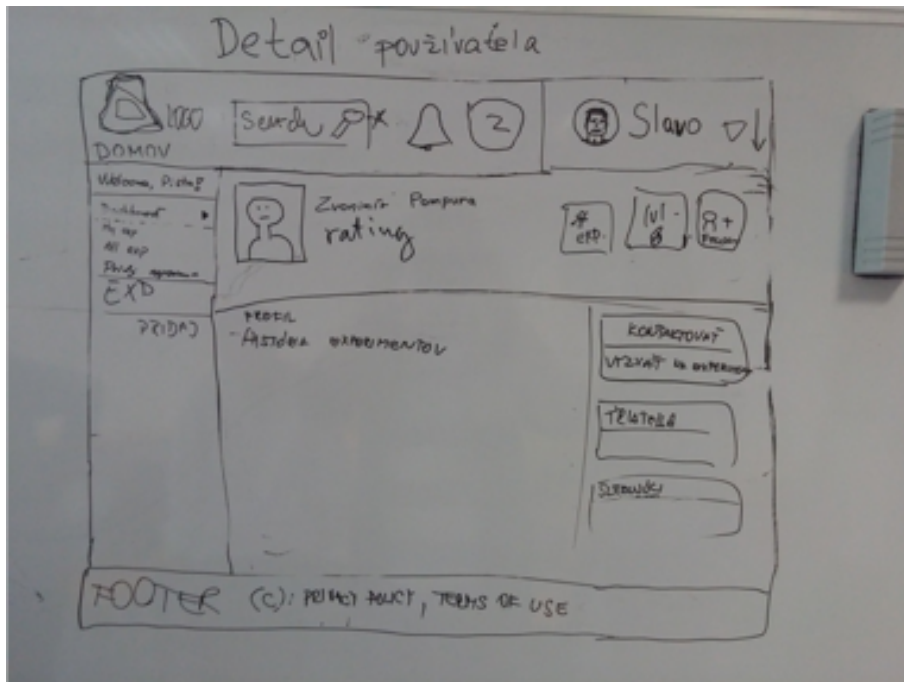
Obr. A.1: Uvítacia stránka



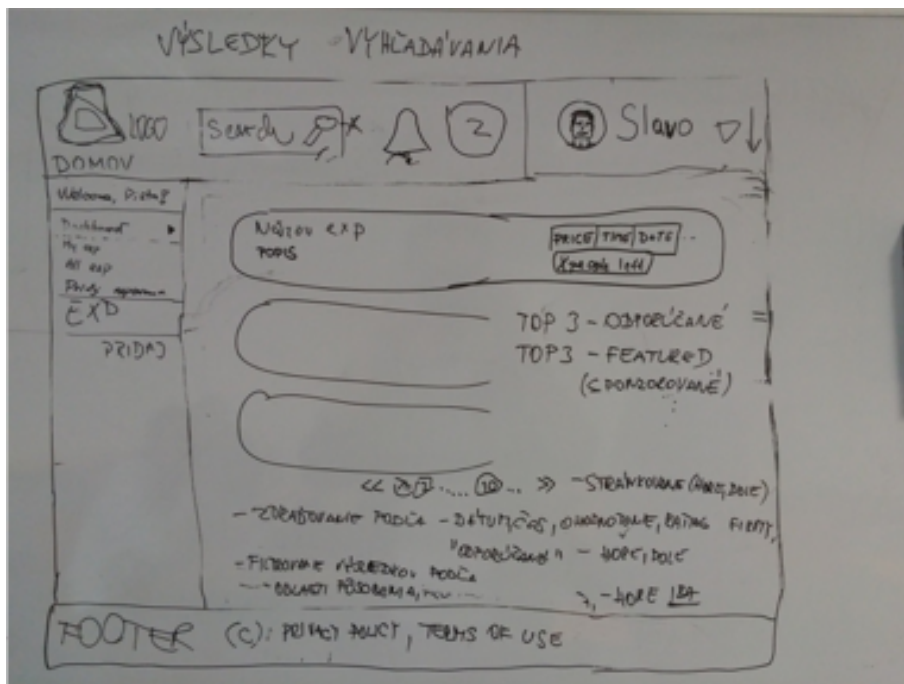
Obr. A.2: Dashboard



Obr. A.3: Detail experimentu

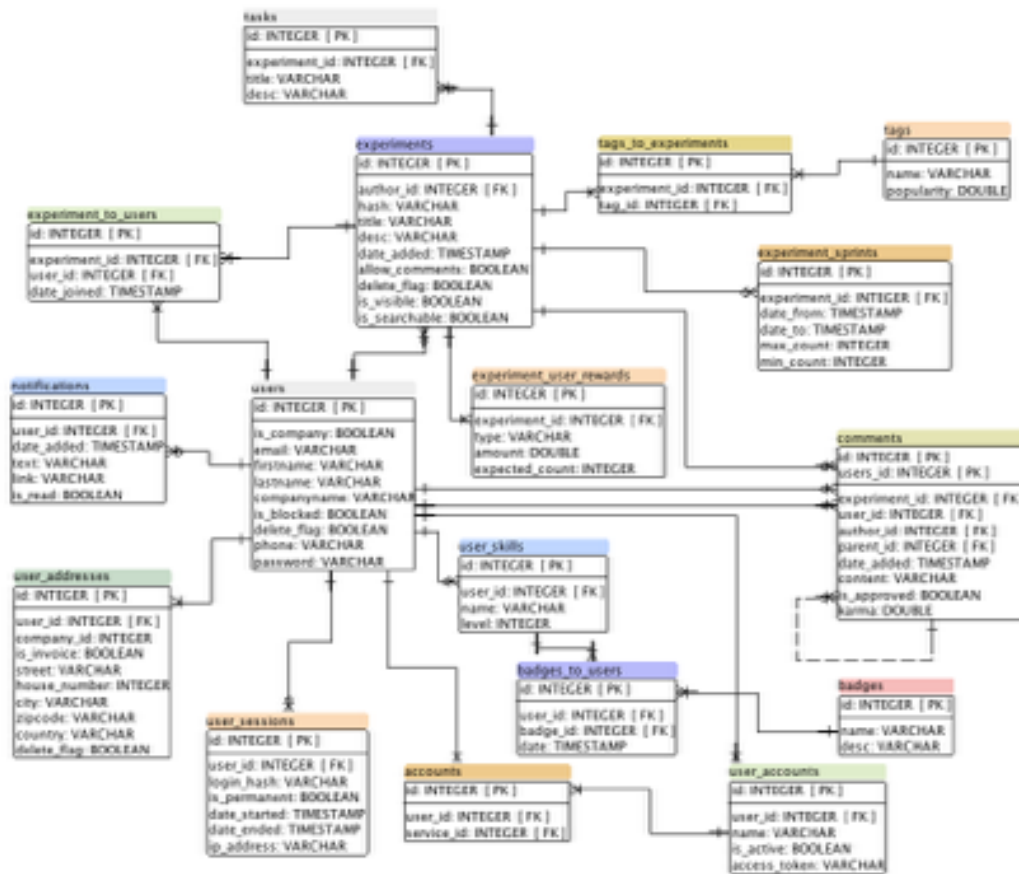


Obr. A.4: Profil používateľa



Obr. A.5: Výsledky vyhľadávania

A.5.4 Príloha B - predbežný dátový model



Obr. A.6: Predbežný dátový model

A.6 Zázpis č. 5 zo stretnutia tímu

Dátum stretnutia: 21.10.2014

Autor: Bc. Štefan Šmihla

Overovateľ: Bc. Peter Gašpar

A.6.1 Priebch stretnutia

- Kontrola úloh z predošlého týždňa
 - Prihláška na TP Cup
 - Zhodnotenie stavu projektu (Tabuľka A.3)
- Riešené úlohy
 - Diskusia k prihláške na TP Cup
 - * Treba doladiť posledné detaily, vytlačiť
 - Konkurencia, ktorá robí to isté www.gesis.org/iirpanel
 - Manažérske role (Tabuľka A.2)
 - Informácie z AIS cez ldap.stuba.sk
 - Time Tracking
 - * Značiť si čas, koľko venujeme projektu
 - * Viest' si projektový denník
 - Názov aplikácie, doména
 - * CrowdTasker – problém s doménou
 - * Crowdex – zatiaľ pracovný názov
 - * Crowdex – joke (Crowdex Premium)
 - Stav projektu
 - * Dokončiť Captchu pre registráciu a zabudnuté heslo
 - * Dokončiť unit testy
 - * Pridávanie experimentu, follow používateľov je takmer dokončené
 - * Zobrazenie profilu používateľa je takmer dokončené

A.6.2 Úlohy do nasledujúceho stretnutia

- Dokončiť prihlášku na TP Cup
- Dokončiť úlohy vyšpecifikované do šprintu 1

A.6.3 Prílohy

A.6.3.1 Príloha 1 – Rozdelenie rolí

Meno	Rola
Dušan Cymorek	Manažment dokumentácie
Peter Gašpar	Manažment rozvrhu
Vladimír L'alík	Manažment rizík
Michal Polko	Manažment rozsahu
Miroslav Šafárik	Manažment komunikácie
Slavomír Šárik	Manažment podpory vývoja a integrácie
Štefan Šmihla	Manažment testovania, prehliadok

Tabuľka A.2: Rozdelenie rolí

A.6.3.2 Príloha 2 - Stav úloh

Úloha	Zodpovedná osoba	Stav úlohy
Jazykové nastavenia	Dušan	Akceptovaná
Registrácia nového používateľa	Štefan	Akceptovaná
Autentifikácia používateľa (štandardne + obnovenie zabudnutého hesla, overenie prekročenia maximálneho počtu pokusov)	Slavomír	Vyriešená
Pridávanie experimentov	Vladimír	Vyriešená
Detail profilu používateľa (osoba)	Miroslav	Vyriešená
Šablóna - farby, základné prvky	Peter	Akceptovaná
Šablóna - horný panel, bočný panel	Michal	Rozpracovaná
Šablóny - footer	Peter	Rozpracovaná

Tabuľka A.3: Stav úloh pre šprint 1

A.7 Zázpis č. 6 zo stretnutia tímu

Dátum stretnutia: 28.10.2014

Autor: Bc. Peter Gašpar

Overovateľ: Bc. Michal Polko

A.7.1 Priebch stretnutia

- Zhodnotenie predchádzajúcich úloh: všetky úlohy boli úspešne splnené.
- Odovzdali sme prihlášku na TP Cup.
- Úspešne sme dokončili 1. šprint:
 - všetky požadované úlohy boli splnené,
 - chceme vytvoriť automatické premazávanie neaktívnych e-mailov,
 - diskutovali sme problém enormného vytvárania záznamov v databáze pri každom prihlásení (riešenie bolo odložené na neskoršie fázy projektu),
 - záznamy v súbore prekladov budeme zorad'ovať podľa abecedy kvôli lepšej prehľadnosti,
 - prechádzali sme si jednotlivé úlohy a vyhodnocovali sme, ktoré sme stihli načas a s ktorými sme, naopak, mali problémy.
- Musíme si zadefinovať proces určený na prehliadky kódu:
 - koľko % kódu sa náhodne pozrie,
 - stanovíme si členov tímu, ktorí budú mať túto úlohu na starosti a budú sa striedať,
 - nesmieme zabúdať aj dokumentáciu,
 - dohodli sme sa, že prehliadky kódu bude vykonávať osoba zodpovedná za danú úlohu.
- Mali by sme si rozdeliť úlohy pre manažment projektu a produktu.
- Budeme tlačiť iba niektoré časti dokumentácie, a to po semestri.
- Riešili sme vytváranie a mergovanie branchov na GitHube. Slavo napísal stručný návod ako postupovať v tomto procese.
- Stanovili sme si metodiky potrebné pre predmet MIS/MSI, ktoré použijeme aj v dokumentácii k Tímovému projektu.
- Prebrali sme dokumentáciu, ktorá sa bude odovzdávať po 3. šprinte.

- Brainstormovali sme myšlienku oprávnení na zobrazenie a pridávanie sa do experimentov. Usúdili sme, že nie je potrebné skrývať experimenty a zakazovať ich zobrazenie.
- Diskutovali sme potrebu nastudovania si protokolu na prihlasovanie do AIS v našom projekte (ldap.stuba.sk).
- Musíme vyriešiť problém prístupu na web cez pevnú sieť.
- Definovali sme si príbehy druhého šprintu, ktorý bude dvojtýždňový (28.10. - 11.11.).
- Dohodli sme sa, že úlohy v *YouTracku* bude vždy vytvárať osoba zodpovedná za danú úlohu.

A.7.2 Úlohy do nasledujúceho stretnutia

- Riešenie príbehov druhého šprintu.
- Rozdelenie a priebežná práca na metodikách tímu.

A.7.3 Prílohy

A.7.3.1 Príloha 1 – Rozdelenie úloh

Úloha	Zodpovedná osoba	Riešitelia
Správa experimentu	Vlado, Dušan	Slavo, Peťo
Detail experimentu	Miro	Štefan
Nastavenia – môj profil + vyplnenie	Štefan	Dušan, Mišo
Notifikácie	Slavo	Vlado, Miro
Šablóna – profil používateľ a/spoločnosti	Peťo	Mišo, Peťo
Šablóna – stránka experimentu	Mišo	Mišo, Peťo
Šablóna – registrácia/prihlásenie	Peťo	Mišo, Peťo

Tabuľka A.4: Rozdelenie úloh pre druhý šprint

A.8 Zápis č. 7 zo stretnutia tímu

Dátum stretnutia: 4.11.2014

Autor: Bc. Michal Polko

Overovateľ: Bc. Dušan Cymorek

A.8.1 Priebeh stretnutia

- Dohodli sme sa na deadline pre tvorbu špecifikácií:
 - pre dvojtýždňový šprint - do troch dní od začiatku (piatok),
 - pre jednotýždňový šprint - do dvoch dní od začiatku (štvrtok).
- Musíme priebežne dokumentovať kód:
 - zatiaľ jednoduché komentáre - čo metóda robí, aké má parametre, ošetruje výnimky?, čo je výstupom,
 - neskôr aj prostredníctvom nástroja *YARD*.
- Dohodli sme sa, že budeme komentovať účel gemov v *Gemfile*.
- Slavo nám predstavil ideu, ako budeme medzi sebou zdieľať migrácie databázy a zredukujeme výskyt konfliktov:
 - vytvorí sa *migration* vetva, do ktorej budú primárne vkladané migrácie,
 - v jednotlivých vetvách treba pravidelne s'ahovať zmeny z vetiev *master* a *migration*.
- Dohodli sme si spôsob ukončovania jednotlivých úloh.
- Prešli sme si stav aktuálneho šprintu.

A.8.2 Prílohy

A.8.2.1 Príloha 1 – Stav úloh

ID	Názov	Stav	Riešitelia
1-27	Profilový obrázok	Znovuotvorená	Dušan
1-40	Vytvoriť inštrukcie pre návrh formulárov	Riešená	Peter
1-44	Správa experimentu	Riešená	Peter, Slavo
1-45	Šablóna - stránka experimentu	Vytvorená	Michal, Peter
1-46	Notifikácie	Riešená	Miro, Vlado
1-47	Šablóna - profil používateľ a/spoločnosti	Riešená	Michal, Peter

ID	Názov	Stav	Riešitelia
1-48	Šablóna - registrácia/prihlásenie	Riešená	Michal, Peter
1-49	Detail experimentu	Riešená	Štefan
1-50	Pridanie neaktívovaného používateľa do experimentu	Akceptovaná	Peter
1-51	Nastavenia profilu	Riešená	Štefan
1-52	Úprava prekladov	Riešená	Dušan
1-54	Upgrade Youtracku	Akceptovaná	Slavo
1-55	Pridať roly tímu na hlavnú stránku	Akceptovaná	Peter
1-56	Vytvoriť retrospektívu za 1. šprint	Akceptovaná	Dušan, Mišo, Miro, Štefan, Peter, Slavo, Vlado
1-59	Prepojenie nástrojov pre správu kódu s projektom	Akceptovaná	Slavo
1-62	Nastavenia profilu - príprava view + napojenie na controller	Vytvorená	Dušan
1-63	Vytvorenie udalosti v systéme	Riešená	Vlado
1-64	Nastavenie monitorovania servera a aplikácie	Akceptovaná	Slavo
1-65	Zobrazenie používateľského rozhrania pre administrátora experimentu (admin UI)	Riešená	Slavo
1-66	Nastavenia detailov experimentu	Riešená	Slavo
1-67	Nastavenie podmienok pre vstup používateľa do experimentu	Vytvorená	Slavo
1-68	Detail experimentu - zobrazenie informácií	Vytvorená	Štefan
1-69	Detail experimentu - prihlásenie do experimentu	Vytvorená	Štefan
1-71	Po vyžiadaní obnovy hesla sa podarí prihlásiť aj neaktívovanému používateľovi	Akceptovaná	Peter, Slavo
1-72	Vytváranie zdieľaných migrácií	Akceptovaná	Slavo
1-73	Zobrazenie notifikácií	Riešená	Miro
1-74	Rekonfigurácia SMTP	Akceptovaná	Slavo

Tabuľka A.5: Stav úloh 2. šprintu

A.9 Zápis č. 8 zo stretnutia tímu

Dátum stretnutia: 11.11.2014

Autor: Bc. Dušan Cymorek

Overovateľ: Bc. Slavomír Šárik

A.9.1 Priebeh stretnutia

- Konzultovali sme stav úloh. Dušan nestihol spraviť svoju úlohu – zostáva doladiť adresy (fakturačná a dodacia), doplniť pridávanie skúseností používateľ a.
- Slavo vysvetlil zavedenie zoznamu zmien na službe *Github*, aby bol product owner lepšie informovaný o priebehu vývoja aplikácie. Používa sa sémantické číslovanie verzií.
- Slavo upozornil na aktualizovanie metodiky používania *Git practices*, ktorá popisuje prácu so systémom *Git* (vytváranie vetiev, práca na úlohách, postup pri *pull requestoch*, odstraňovaní vetiev).
- Opätovne sme sa dohodli na záväznom dodržiavaní pravidiel a postupov opísaných v metodikách.
- Slavo navrhol zmenu oznamovacieho textu pri obnovovaní hesla. Pri zadaní správneho, aj neprávneho emailu sa zobrazí informácia o zaslaní obnovovacieho odkazu, ak email existuje. Týmto zamedzíme špekulantom zisťovanie emailov, ktoré sú v systéme zaregistrované.
- Slavo predniesol požiadavku o použitie aplikácie na monitorovanie serveru od tímu č. 10. Dohodli sme sa na tom, že Slavo sa spýta, akú funkcionálnosť poskytujú a či je to stabilné (aby to neohrozilo vývoj našej aplikácie).
- Dohodli sme sa na priebehu tretieho šprintu – tento šprint bude “upratovací” a dokumentačný. Nebudeme pridávať novú funkcionálnosť, ale refactorovať a vylepšovať existujúcu, popri tom sa bude finalizovať dokumentácia.
- Dohodli sme sa na požiadavke o zvýšenie dostupných prostriedkov na serveri – pamäť +1GB a disk +15GB.
- Na návrh prof. Bielikovej hlbšie preskúmame možnosti konkurenčnej služby *CrowdFlower* a ich *API*.
- Je potrebné upraviť pridávanie podmienok v experimente tak, aby systém umožňoval ich voľné zadávanie, pričom napríklad pri podmienke na vek rozpozná, že ide o rozsah.
- Diskutovali sme o značkách a skúsenostiach používateľ a. Skúsenosti budú riešené podobne ako značky experimentu – voľný text + automatické dopĺňanie.

- Odporúčanie experimentov by malo byť realizované okrem priradovania značiek aj prostredníctvom informácií o predchádzajúcich experimentoch používateľa.
- Dohodli sme sa na premenovaní stĺpca „*Stav zverejnenia*” na „*Stav*” pri správe experimentu.
- Rozdelili sme si úlohy na ďalší šprint, ktorý bude trvať 1 týždeň.

A.9.2 Prílohy

A.9.2.1 Príloha 1 – Stav úloh

ID	Názov	Stav	Riešitelia
1-27	Profilový obrázok	Akceptovaná	Michal
1-44	Správa experimentu	Akceptovaná	Peter, Slavo
1-45	Šablóna - stránka experimentu	Akceptovaná	Michal, Peter
1-46	Notifikácie	Akceptovaná	Miro, Vlado
1-47	Šablóna - profil používateľa/spoločnosti	Akceptovaná	Michal, Peter
1-48	Šablóna - registrácia/prihlásenie	Akceptovaná	Michal, Peter
1-49	Detail experimentu	Akceptovaná	Štefan
1-50	Pridanie neaktívovaného používateľa do experimentu	Akceptovaná	Peter
1-52	Úprava prekladov	Akceptovaná	Dušan
1-54	Upgrade Youtracku	Akceptovaná	Slavo
1-55	Pridať roly tímu na hlavnú stránku	Akceptovaná	Peter
1-56	Vytvoriť retrospektívu za 1. šprint	Akceptovaná	Všetci
1-59	Prepojenie nástrojov pre správu kódu s projektom	Akceptovaná	Slavo
1-60	Nastavenie profilu – upload obrázka	Akceptovaná	Michal
1-63	Vytvorenie udalosti v systéme	Akceptovaná	Vlado
1-64	Nastavenie monitorovania servera a aplikácie	Akceptovaná	Slavo
1-65	Zobrazenie používateľského rozhrania pre administrátora experimentu (admin UI)	Akceptovaná	Slavo
1-66	Nastavenia detailov experimentu	Akceptovaná	Slavo
1-67	Nastavenie podmienok pre vstup používateľa do experimentu	Akceptovaná	Slavo
1-68	Detail experimentu - zobrazenie informácií	Akceptovaná	Štefan
1-69	Detail experimentu - prihlásenie do experimentu	Akceptovaná	Štefan
1-71	Po vyžiadaní obnovy hesla sa podarí prihlásiť aj neaktívanému používateľovi	Akceptovaná	Peter, Slavo

ID	Názov	Stav	Riešitelia
1-72	Vytváranie zdieľaných migrácií	Akceptovaná	Slavo
1-73	Zobrazenie notifikácií	Akceptovaná	Miro
1-74	Rekonfigurácia SMTP	Akceptovaná	Slavo
1-75	Správa experimentu – Značky	Akceptovaná	Peter
1-76	Správa experimentu – Budget	Akceptovaná	Peter
1-77	Správa experimentu – Správa participantov	Akceptovaná	Peter
1-79	Príprava unit testov pre model v šprinte 2	Vyriešená	Štefan
1-80	Nastavenie servera na produkciu a staging	Akceptovaná	Slavo
1-81	Integrácia systému – 2. šprint	Akceptovaná	Slavo

Tabuľka A.6: Stav úloh 2. šprintu

A.9.2.2 Príloha 2 – Úlohy pre 3. šprint

ID	Názov	Stav	Riešitelia
1-51	Nastavenia profilu	Riešená	Dušan, Michal
1-57	Vytvorenie a odovzdanie dokumentácie - inžinierske dielo	Vytvorená	Všetci
1-58	Vytvorenie a odovzdanie dokumentácie - riadenie	Vytvorená	Všetci
1-62	Nastavenia profilu - príprava view + napojenie na controller	Riešená	Dušan
1-82	Integrácia systému – 3. šprint	Riešená	Slavo
1-83	Refactoring - pridávanie experimentu	Akceptovaná	Peter
1-84	Chyba pri vyplnení registračného formulára	Vytvorená	
1-85	Preformulovanie textov	Vytvorená	Peter
1-86	Refactoring zdrojového kódu	Riešená	Všetci
1-87	Refactoring – experiment	Riešená	Slavo, Štefan
1-88	Refactoring – participanti	Riešená	Peter
1-89	Refactoring – dizajn	Vytvorená	Michal, Peter
1-90	Refactoring – preklady	Vytvorená	Dušan, Peter
1-91	Predbežná príprava unit testov	Vytvorená	Štefan
1-92	Refactoring – follow	Vytvorená	Vlado
1-93	Refactoring – notifikácie	Vytvorená	Miro, Vlado

Tabuľka A.7: Úlohy pre 3. šprint

A.10 Zázpis č. 9 zo stretnutia tímu

Dátum stretnutia: 18.11.2014

Autor: Bc. Slavomír Šárik

Overovateľ: Bc. Vladimír L'alík

A.10.1 Priebeh stretnutia

- Konzultovali sme stav úloh po skončení 3. šprintu.
- Úloha k úprave profilu bola úspešne dokončená (nedokončená úloha z 2. šprintu).
- Dokumentácia k riadeniu a inžinierskemu dielu bola vypracovaná.
- Dohodli sme sa na použití nástroja LaTeX pre písanie dokumentácie.
- Pridávanie experimentu je riešene cez modálne okno – JavaScript free.
- Texty na stránke boli preformulované, odstránené smajlíky.
- Prešli sme si úlohy pre refactorovanie z 3. šprintu.
- Niektoré funkcionality stále nie sú pokryté testami.
- Števo navrhol používať wiki collaborators tool - odsúhlasili sme a na budúcom stretnutí sa určí nástroj.
- Diskusia o nepriradených issue - pridávať ich do backlogu?
- Diskusia o resources k serveru - bude vyriešené osobným dohovorom so správcom školských serverov.
- Diskusia - build serveru – návrh používať continuous integration tool - bude nasadený v budúcnosti.
- Diskusia ku granularite jednotlivých úloh:
 - Písať podiel % k úlohám? Ostáva to tak ako doteraz.
 - Otázka k burndown chart pre delenie úloh - úlohy sa rozdelia na špecifikácie a implementačné úlohy ako podúlohy.
 - Úloha do ďalšieho šprintu - prezrieť nastavenia YouTracku
- Diskusia o použití API Crowdflower:
 - Rozšírenie používateľskej základne - viac ľudí pre vykonávanie experimentov.
 - Spojiť to so sledovaním pohľadu používateľa.

- Diskusia, ci sa nezamerat' na jednu konkrétну oblasť - testovanie
- Treba nájsť pridanú hodnotu.
- Diskusia k návrhom k zlepšeniu GUI aplikácie.
- Rozdelili sme si úlohy na ďalší šprint.

A.10.2 Prílohy

A.10.2.1 Príloha 1 – Stav úloh

ID	Názov	Stav	Riešitelia
1-51	Nastavenia profilu	Akceptovaná	Dušan, Michal
1-57	Vytvorenie a odovzdanie dokumentácie - inžinierske dielo	Akceptovaná	Všetci
1-58	Vytvorenie a odovzdanie dokumentácie - riadenie	Akceptovaná	Všetci
1-62	Nastavenia profilu - príprava view + napojenie na controller	Akceptovaná	Dušan
1-82	Integrácia systému - 3. šprint	Akceptovaná	Slavo
1-83	Refactoring - pridávanie experimentu	Akceptovaná	Peter
1-84	Chyba pri vyplnení registračného formulára	Akceptovaná	Peter
1-85	Preformulovanie textov	Akceptovaná	Peter
1-87	Refactoring - experiment	Akceptovaná	Slavo
1-88	Refactoring - participanti	Akceptovaná	Peter
1-89	Refactoring - dizajn	Akceptovaná	Peter, Michal
1-90	Refactoring - preklady	Akceptovaná	Dušan, Peter
1-91	Priebežná príprava unit testov	Akceptovaná	Štefan
1-92	Refactoring - follow	Akceptovaná	Vlado
1-93	Refactoring - notifikácie	Akceptovaná	Miro, Vlado
1-95	Vytvoriť retrospektívu za 2. šprint	Akceptovaná	Všetci

Tabuľka A.8: Stav úloh 3. šprintu

A.10.2.2 Príloha 2 – Úlohy pre 4. šprint

ID	Názov	Stav	Riešitelia
1-96	Oprava dizajnu nastavení profilu	Vytvorená	Peter

ID	Názov	Stav	Riešitelia
1-97	Vytvoriť retrospektívu za 3. šprint	Akceptovaná	Michal, Štefan
1-98	Editácia profilu - chýbajúce preklady pri nahrávaní nesprávnej profilovej fotografie	Vytvorená	Michal
1-99	Editácia profilu - pridať texty o maximálnej veľkosti profilovej fotografie a type súboru	Vytvorená	Michal
1-100	Šablóna - vyhľadávanie / filtrovanie výsledkov	Vytvorená	Peter, Michal
1-101	Šablóna - landing page	Vytvorená	Peter, Michal
1-102	Šablóna - dashboard	Riešená	Peter, Michal
1-103	Dynamické bočné menu	Vytvorená	Peter, Michal
1-104	Datagrid pre zoznamy	Vytvorená	Peter, Michal
1-105	Plátno biznis modelu	Vytvorená	Všetci
1-106	Vyhľadávanie v systéme	Vytvorená	Vlado, Slavo
1-107	Vyhľadávanie v systéme - špecifikácia	Riešená	Dušan
1-108	Integrácia ElasticSearch	Riešená	Slavo
1-109	Spustenie experimentu	Vytvorená	Miro
1-110	Spustenie experimentu - špecifikácia	Akceptovaná	Štefan
1-111	Integrácia 4. šprint	Riešená	Slavo
1-112	Filtrovanie výsledkov vyhľadávania	Vytvorená	Štefan
1-113	Filtrovanie výsledkov vyhľadávania - špecifikácia	Riešená	Vlado
1-114	Spustenie experimentu - príprava migrácie	Vyriešená	Štefan
1-115	Spustenie experimentu - príprava cron úlohy	Vytvorená	Miro
1-116	Spustenie experimentu - aktualizácia nastavení experimentu	Vytvorená	Miro
1-117	Spustenie experimentu - aktualizácia notifikácií	Vytvorená	Miro
1-118	Spustenie experimentu - aktualizácia detailu experimentu	Vytvorená	Miro
1-119	Dashboard - špecifikácia	Riešená	Peter

Tabuľka A.9: Úlohy pre 4. šprint

A.11 Zázpis č. 10 zo stretnutia tímu

Dátum stretnutia: 25.11.2014

Autor: Bc. Vladimír L'alík

Overovateľ: Bc. Miroslav Šafárik

A.11.1 Priebeh stretnutia

- Diskutovali sme o použití nástroja na správu dokumentov. Rozhodovali sme sa medzi DokuWiki a MediaWiki, nakoniec sme sa dohodli, že začneme používať jeden z týchto nástrojov až v letnom semestri.
- Peťo vytvoril návod ako používať pluralizáciu slov v rôznych jazykoch. Dohodli sme sa, že to budeme využívať.
- Spoločne sme vymýšľali ako pomenovať a rozlíšiť v aplikácii experimenty, ktoré som ako používateľ vytvoril a experimenty, v ktorých som zúčastnený ako participant. Dospeli sme k názvu Experimenty, Úlohy.
- Slavo opravil upload súborov na server.
- V zdrojovom kóde sa nám nahromadili úlohy označené TO DO, dohodli sme sa, že budú riešené v piatom šprinte.
- Padol návrh aby sme zorganizovali teambuilding na vianočných trhoch vo Viedni.
- Diskutovali sme o možných spôsoboch odmeny pre participantov. Jedno z navrhovaných riešení bolo vytoriť v systéme vlastnú menu a obchod s rôznymi predmetmi.
- Je potrebné definovať životný cyklus experimentu.
- Na záver sme diskutovali o hodnotení používateľov a škálovaní hodnotenia.

A.11.2 Prílohy

A.11.2.1 Príloha 1 – Stav úloh

ID	Názov	Stav	Riešitelia
1-97	Vytvoriť retrospektívu za 3. šprint	Akceptovaná	Michal, Štefan
1-98	Editácia profilu - chýbajúce preklady pri nahrávaní nesprávnej profilovej fotografie	Vyriešená	Michal
1-99	Editácia profilu - pridať texty o maximálnej veľkosti profilovej fotografie a type súboru.	Vyriešená	Michal
1-100	Šablóna - vyhľadávanie / filtrovanie výsledkov	Riešená	Peter, Michal

ID	Názov	Stav	Riešitelia
1-101	Šablóna - landing page	Akceptovaná	Peter, Michal
1-102	Šablóna - dashboard	Akceptovaná	Peter, Michal
1-103	Dynamické bočné menu	Vytvorená	Peter, Michal
1-104	Datagrid pre zoznamy	Riešená	Peter, Michal
1-105	Plátno biznis modelu	Akceptovaná	Všetci
1-106	Vyhľadávanie v systéme	Akceptovaná	Vlado, Slavo
1-107	Vyhľadávanie v systéme - špecifikácia	Akceptovaná	Dušan
1-108	Integrácia ElasticSearch	Akceptovaná	Slavo
1-109	Spustenie experimentu	Vytvorená	Miro
1-110	Spustenie experimentu - špecifikácia	Akceptovaná	Štefan
1-111	Integrácia 4. šprint	Riešená	Slavo
1-112	Filtrovanie výsledkov vyhľadávania	Vytvorená	Štefan
1-113	Filtrovanie výsledkov vyhľadávania - špecifikácia	Riešená	Vlado
1-114	Spustenie experimentu - príprava migrácie	Vyriešená	Štefan
1-115	Spustenie experimentu - príprava cron úlohy	Vytvorená	Miro
1-116	Spustenie experimentu - aktualizácia nastavení experimentu	Riešená	Miro
1-117	Spustenie experimentu - aktualizácia notifikácií	Vytvorená	Miro
1-118	Spustenie experimentu - aktualizácia detailu experimentu	Riešená	Miro
1-119	Dashboard - špecifikácia	Akceptovaná	Peter
1-122	Vyhľadávanie v systéme - vyhľadávanie experimentov	Akceptovaná	Slavo
1-123	Vyhľadávanie v systéme - vyhľadávanie používateľov	Akceptovaná	Vlado
1-124	Pozvanie používateľa do experimentu	Vyriešená	Vlado
1-125	Pozvanie používateľa do experimentu - špecifikácia	Akceptovaná	Miro
1-126	Export dát do CVS - Prepracovanie exportu	Akceptovaná	Peter
1-127	Profil používateľa - follower list je zobrazený dvakrát	Akceptovaná	Michal
1-129	Hodnotenie používateľov	Riešená	Dušan
1-131	Hodnotenie používateľov - špecifikácia	Akceptovaná	Miro
1-132	Pozvanie používateľa do experimentu - úprava správy experimentu	Vyriešená	Vlado

ID	Názov	Stav	Riešitelia
1-133	Pozvanie používateľa do experimentu - odoslanie pozvánky	Vyriešená	Vlado
1-134	Hodnotenie používateľov - úprava správy experimentu	Riešená	Dušan
1-135	Hodnotenie používateľov - systém hodnotenia	Riešená	Dušan
1-136	Hodnotenie používateľov - úprava detailu experimentu	Riešená	Dušan
1-137	Hodnotenie používateľov - zobrazovanie hodnotenia	Vyriešená	Dušan
1-138	Filtrovanie experimentov	Akceptovaná	Štefan
1-139	Filtrovanie používateľov	Akceptovaná	Štefan
1-142	Produkcia - premazávanie uploads adresára	Akceptovaná	Slavo

Tabuľka A.10: Stav úloh 4. šprintu

A.12 Zázpis č. 11 zo stretnutia tímu

Dátum stretnutia: 11.11.2014

Autor: Bc. Miroslav Šafárik

Overovateľ: Bc. Štefan Šmihla

A.12.1 Priebeh stretnutia

- na úvod stretnutia boli prejdené jednotlivé body z dokumentu pre zápisy
- padla pripomienka o pridaní tlačidla pre vyhľadávanie
 - bude riešené v 5. šprinte
- Števo informoval tím o úprave metodiky pre testovanie, konkrétne o údržbe testov - je potrebné pri každom zásahu do funkcionality skontrolovať aktuálne testy a zabezpečiť, aby ich zbehnutie skončilo úspešne - v prípade, že to nie je možné dosiahnuť, treba kontaktovať priamo Števa - bližšie info v metodike
- Slavo nasadil continuous integration - okrem iného sa pri pull requeste automaticky skontrolujú existujúce testy a ich prípadný pád je potrebné napraviť
- ďalej prebehla diskusia o prepisovnej dokumentácii do LaTeXu
 - padla dohoda, že každý zápis si prepíše svoj autor do LaTeXu - neplatí pre zápisy 0, 1, 2 a 3, ktoré už Slavo prepísal
 - rovnako sa vzájomne dohodnú autori príslušných retrospektív a zabezpečia ich prepis do LaTeXu
- padla pripomienka o možnosti vypnúť notifikácie v systéme - bude riešené v 5. šprinte
- prebehla diskusia o dĺžke načítania vybraných častí systému, najmä čo sa týka čakania na vykonanie akcií, pri ktorých sa posielajú maily - tieto trvajú dlho a bolo by vhodné, aby boli vykonávané asynchrónne
 - pravdepodobne bude riešené všeobecne prostredníctvom nejakého messaging queue
- ďalej prebehla diskusia o aktuálnom stave úloh z 5. šprintu
 - všetky úlohy sa stihli dokončiť v rámci šprintu
 - úloha pre hodnotenie participantov a zadávateľov experimentov bude v 5. šprinte prepracovaná tak, aby spĺňala predstavy vlastníka produktu uvedené na minulotýždňovom stretnutí

- prebehlo hlasovanie o zložitosti jednotlivých úloh pre 5. šprint - do úvahy sa bral najmä fakt, že sa jedna o týždňový šprint, preto boli vybrané do tohto šprintu len úlohy s malou zložitou
- rozdelili sme si jednotlivé úlohy

A.12.2 Prílohy

A.12.2.1 Príloha 1 – Stav úloh

ID	Názov	Stav	Pridelené
1-97	Vytvoriť retrospektívu za 3. šprint	Verified	Štefan, Michal
1-98	Editácia profilu - chýbajúce preklady pri nahrávaní nesprávnej profilovej fotografie	Fixed	Michal
1-100	Šablóna - vyhľadávanie / filtrovanie výsledkov	Verified	Peter, Michal
1-101	Šablóna - landing page	Verified	Peter, Michal
1-102	Šablóna - dashboard	Verified	Peter, Michal
1-103	Dynamické bočné menu	Verified	Peter, Michal
1-104	Datagrid pre zoznamy	Verified	Peter, Michal
1-105	Plátno biznis modelu	Verified	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír
1-106	Vyhľadávanie v systéme	Verified	Vladimír, Slavomír
1-107	Vyhľadávanie v systéme - špecifikácia	Verified	Dušan
1-108	Integrácia ElasticSearch	Verified	Slavomír
1-109	Spustenie experimentu	Verified	Miroslav
1-110	Spustenie experimentu - špecifikácia	Verified	Štefan
1-111	Integrácia 4. sprint	Verified	Slavomír
1-112	Filtrovanie výsledkov vyhľadávania	Verified	Štefan
1-113	Filtrovanie výsledkov vyhľadávania - špecifikácia.	Verified	Vladimír
1-114	Spustenie experimentu - príprava migrácie	Verified	Štefan
1-115	Spustenie experimentu - príprava cron úlohy	Verified	Miroslav
1-116	Spustenie experimentu - aktualizácia nastavení experimentu	Verified	Miroslav
1-117	Spustenie experimentu - aktualizácia notifikácií	Verified	Miroslav

ID	Názov	Stav	Pridelené
1-118	Spustenie experimentu - aktualizácia detailu experimentu	Verified	Miroslav
1-119	Dashboard - špecifikácia	Verified	Peter
1-122	Vyhľadávanie v systéme - vyhľadávanie experimentov	Verified	Slavomír
1-123	Vyhľadávanie v systéme - vyhľadávanie používateľov	Verified	Vladimír
1-124	Pozvanie používateľa do experimentu	Verified	Vladimír
1-125	Pozvanie používateľa do experimentu - špecifikácia	Verified	Miroslav
1-127	Profil používateľa - follower list je zobrazený dvakrát	Verified	Michal
1-129	Hodnotenie používateľov	Verified	Dušan
1-131	Hodnotenie používateľov - špecifikácia	Verified	Miroslavslav Šafárik
1-132	Pozvanie používateľa do experimentu - úprava správy experimentu	Verified	Vladimír
1-133	Pozvanie používateľa do experimentu - odoslanie pozvánky	Verified	Vladimír
1-134	Hodnotenie používateľov - úprava správy experimentu	Verified	Dušan
1-135	Hodnotenie používateľov - systém hodnotenia	Verified	Dušan
1-136	Hodnotenie používateľov - úprava detailu experimentu	Verified	Dušan
1-137	Hodnotenie používateľov - zobrazovanie hodnotenia	Verified	Dušan

ID	Názov	Stav	Pridelené
1-138	Filtrovanie používateľov.	Verified	Štefan
1-139	Filtrovanie experimentov.	Verified	Štefan

Tabuľka A.11: Stav úloh po 4. šprinte

A.12.2.2 Príloha 2 – Úlohy pre 5. šprint

ID	Názov	Zodpovedný	Pridelené
1-120	Vytvorenie a odovzdanie dokumentácie - inžinierske dielo - FINAL ZS	Dušan	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír
1-121	Vytvorenie a odovzdanie dokumentácie - riadenie - FINAL ZS	Dušan	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír
1-151	Vytvorenie textov pre Uvítaciu stránku (Landing Page)	Dušan	Peter, Dušan
1-167	Realizácia experimentu	Michal	Štefan
1-168	Inštrukcie pri spustení experimentu	Štefan	Michal
1-169	Pridávanie komentárov ku experimentu	Slavomír	Vladimír, Slavomír
1-170	Implementácia komentárov	Slavomír	Vladimír, Slavomír
1-171	Integrácia 5. šprint	-	Slavomír
1-172	Dashboard - implementácia	Michal	Peter
1-173	Špecifikácia - komentáre v experimente	-	Slavomír
1-174	Vypnutie/zapnutie notifikácií	Miroslav	Miroslav
1-178	Realizácia experimentu - vytvorenie špecifikácie	-	Štefan, Michal

Tabuľka A.12: Úlohy pre piaty šprint

A.13 Zápis č. 12 zo stretnutia tímu

Dátum stretnutia: 9.12.2014

Autor: Bc. Štefan Šmihla

Overovateľ: Bc. Peter Gašpar

A.13.1 Stav úloh 5. šprintu

ID	Názov	Stav	Pridelené
1-120	Vytvorenie a odovzdanie dokumentácie - inžinierske dielo - FINAL ZS	Riešená	Dušan, Peter, Vladimír, Michal, Miroslav, Slavomír, Štefan
1-121	Vytvorenie a odovzdanie dokumentácie - riadenie - FINAL ZS	Riešená	Dušan, Peter, Vladimír, Michal, Miroslav, Slavomír, Štefan
1-151	Vytvorenie textov pre Uvítaciu stránku (Landing Page)	Akceptovaná	Dušan, Peter
1-167	Realizácia experimentu	Akceptovaná	Štefan
1-168	Inštrukcie pri spustení experimentu	Akceptovaná	Michal
1-169	Pridávanie komentárov ku experimentu	Akceptovaná	Vladimír, Slavomír
1-170	Implementácia komentárov	Akceptovaná	Vladimír, Slavomír
1-171	Integrácia 5. šprint	Akceptovaná	Slavomír
1-172	Dashboard - implementácia	Akceptovaná	Peter
1-173	Špecifikácia - komentáre v experimente	Akceptovaná	Vladimír, Slavomír
1-174	Vypnutie / zapnutie notifikácií	Akceptovaná	Miroslav
1-178	Realizácia experimentu - vytvorenie špecifikácie	Akceptovaná	Štefan
1-180	Úprava komentárov	Akceptovaná	Slavomír

Tabuľka A.13: Stav úloh po 5. šprinte

A.13.2 Poznámky zo stretnutia

- Prebrali sme biznis panel. Porote sa nepáčil pojem *experiment*. Podnet na zmenu názvu, napr. *mikroúloha*.
- Prebrali sme úlohy z 5. šprintu.
 - Zostáva dokončiť retrospektívu.
- V januári nás čaká prezentácia pre prof. Bielikovú.
- Na stretnutí padli rôzne pripomienky k úlohám:
 - Máme viac používať dáta z *CrowdFlower*, mať to už v tej prezentácii.
 - Zistiť, či je možné získať skúsenosti používateľov cez *LinkedIn API*.
 - Treba spraviť dizajn pre podmienky úloh - možnosť vybrať z viacerých rozpätí veku používateľov.
 - Pri zobrazení zoznamu úloh stĺpec *Oprávnenosť* zaberá príliš veľa miesta na šírku.
 - Pridať pre zadávateľa možnosť mazať nevhodné komentáre.
 - Kurzor pri textových poliach prerobiť na textový kurzor (teraz symbol ruky).
 - Dať si pozor na prípadný ženský rod v textoch.
- Očakávanie vianočného šprintu.
 - Dušanovi zostalo prerobenie hodnotenia.
 - Dušan plánuje prerobiť systém prekladov.
 - Riešenie *TO DO* úloh.
 - Doriešenie rozpočtu pre používateľov a pre úlohy (aspoň vizuálne).

B Retrospektívy k šprintom

B.1 Retrospektíva k 1. šprintu

Autori: Dušan Cymorek, Peter Gašpar, Slavomír Šárik

Trvanie šprintu: 14.10. – 28.10

B.1.1 Zoznam úloh a stav ich riešenia

V tabuľke B.1 sa nachádza prehľad úloh, ktoré sme riešili počas prvého šprintu, stav ich splnenia a poznámky z nich vyplývajúce. Náplňou tohto šprintu bolo implementovanie základných funkcionalít systému.

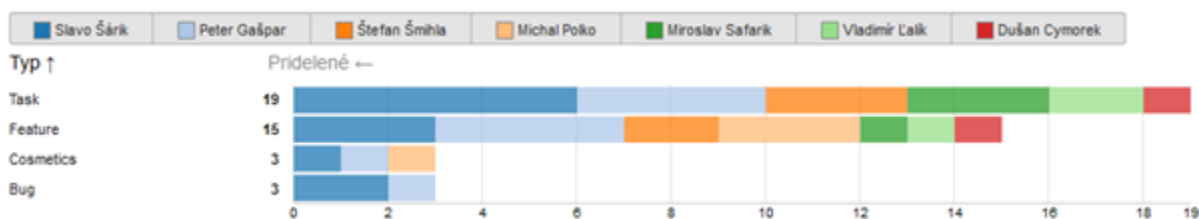
ID	Názov	Typ	Riešitelia	Záver
1-6	Jazykové nastavenia	Funkcionalita	Slavomír	Akceptovaná
1-7	Registrácia nového používateľa	Funkcionalita	Štefan, Slavomír	Akceptovaná
1-8	Registrácia nového používateľa - Vytvorenie modelu pre používateľa	Úloha	Štefan, Slavomír	Akceptovaná
1-9	Registrácia nového používateľa - Zasielanie mailov	Úloha	Štefan, Slavomír	Akceptovaná
1-10	Registrácia nového používateľa - Captcha	Úloha	Štefan	Akceptovaná
1-11	Registrácia nového používateľa - Aktivácia konta	Úloha	Slavomír	Akceptovaná
1-12	Pridávanie experimentov	Funkcionalita	Miroslav, Peter	Akceptovaná
1-13	Pridávanie experimentov - Vytvorenie modelu pre experiment	Úloha	Miroslav	Akceptovaná
1-14	Pridávanie experimentov - Názov, popis	Úloha	Miroslav	Akceptovaná
1-15	Pridávanie experimentov - Oprávnenia na prístup	Úloha	Miroslav	Akceptovaná
1-16	Pridávanie experimentov - Budget	Úloha	Peter	Akceptovaná
1-17	Pridávanie experimentov - Účastníci	Úloha	Peter	Akceptovaná
1-18	Pridávanie experimentov - Zverejnenie	Úloha	Peter	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-19	Autentifikácia používateľa (štandardne)	Funkcionality	Štefan, Slavomír	Akceptovaná
1-20	Autentifikácia používateľa (štandardne) - Obnovenie zabudnutého hesla	Úloha	Slavomír	Akceptovaná
1-21	Autentifikácia používateľa (štandardne) - Overenie prekročenia maximálneho počtu pokusov + dočasné zablokovanie konta	Úloha	Slavomír	Akceptovaná
1-22	Autentifikácia používateľa (štandardne) - Exspirácia tokenu (pri zabudnutí hesla)	Úloha	Slavomír	Akceptovaná
1-25	Detail profilu používateľa	Funkcionality	Dušan, Vladimír	Akceptovaná
1-26	Detail profilu používateľa - Zobrazenie info o používateľovi	Úloha	Dušan	Akceptovaná
1-27	Detail profilu používateľa - Profilový obrázok	Úloha	Dušan	Akceptovaná
1-28	Detail profilu používateľa - Priatelia (follow)	Úloha	Vladimír	Akceptovaná
1-29	Detail profilu používateľa - Poznámka o spoločnosti	Úloha	Vladimír	Akceptovaná
1-30	Šablóna - farby, základné prvky	Funkcionality	Michal, Peter	Akceptovaná
1-31	Šablóna - horný panel, bočný panel	Funkcionality	Michal, Peter	Akceptovaná
1-32	Šablóny - footer	Funkcionality	Michal, Peter	Akceptovaná
1-33	Vyriešiť automatický deploy do produkcie	Chyba	Slavomír	Akceptovaná
1-34	Aktualizácia stránky tímu	Kozmetické	Michal, Peter	Akceptovaná
1-36	[BUG] Preklady nefungujú v mailoch	Chyba	Slavomír	Akceptovaná
1-39	Aktualizácia plánu projektu na webe	Úloha	Peter	Akceptovaná
1-42	Aktualizácia dizajnu webu	Úloha	Peter	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-43	Úprava chybových stránok	Kozmetické	Slavomír	Akceptovaná

Tabuľka B.1: *Prehľad úloh*

Na nasledujúcom grafe (Obr. B.1) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. B.1: *Distribúcia jednotlivých typov úloh medzi autorov*

B.1.2 Časový prehľad

Tabuľka B.2 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	2	5,42
	Documentation	1	2,00
	Celkovo	3	7,42
Michal	Development	1	13,00
	Celkovo	1	13,00
Miroslav	Development	3	5,58
	Documentation	1	2,00
	Testing	1	2,00
	Celkovo	5	9,58
Peter	Development	4	9,37
	Documentation	1	0,18
	Testing	2	1,00
	Design	2	9,62
	Celkovo	9	20,17

Riešiteľ	Typ	Počet úloh	Počet hodín
Slavomír	Development	10	29,00
	Documentation	2	10,67
	Testing	1	0,50
	Maintenance	2	33,00
	Celkovo	15	73,17
Vladimír	Development	3	8,67
	Documentation	1	2,00
	Testing	1	1,67
	Celkovo	5	12,33
Štefan	Development	5	17,00
	Documentation	1	2,00
	Testing	2	3,00
	Celkovo	8	22,00

Tabuľka B.2: Časový prehľad úloh

B.1.3 Zhodnotenie

Pri retrospektíve k šprintu sme sa na stretnutí vyjadrili k jednotlivým úlohám a problémom, ktoré pri ich riešení nastali. Poznámky k jednotlivým úlohám sa nachádzajú v tabuľke B.3.

ID	Názov	Poznámka
1-6	Jazykové nastavenia	Dohodli sme sa, že položky v súbore s jazykmi budú z dôvodu prehľadnosti zoradené podľa abecedy. <u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> Vytvorenie metodiky na pridávanie a používanie prekladov, zjednotenie používania prekladov v kóde.
1-11	Registrácia nového používateľa - Aktivácia konta	Prehodnotili sme, že v nasledujúcich šprintoch bude potrebné vyriešiť automatické premazávanie nedokončených aktivácií konta. <u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> Vytvorenie automatickej úlohy na serveri na odstránenie starších už neplatných aktivácií.

ID	Názov	Poznámka
1-16	Pridávanie experimentov - Budget	<p>Pri nastavovaní budgetu sme zhodnotili, že je nezmyselné ho nastavovať pomocou políčka "Zvýšiť/Znížiť budget". Výška peňažného budgetu sa bude nastavovať priamo ako číselná hodnota.</p> <p><u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> V experimente zavedieme možnosť pridať aj iný ako finančný budget. Zmenu budgetu nebude možné vykonať po spustení experimentu.</p>
1-18	Pridávanie experimentov - Zverejnenie	<p>Prehodnotili sme, že súčasná klasifikácia ochrany experimentu z hľadiska prístupu (verejný, súkromný, na odkaz) nie je potrebná.</p> <p><u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> Z pôvodného návrhu ponecháme iba možnosť skryť experiment z výsledkov vyhľadávania.</p>
1-19	Autentifikácia používateľa (štandardne)	Diskutovali sme problém enormného vytvárania záznamov v databáze pri každom prihlásení. Riešenie bolo odložené na neskoršie fázy projektu.
1-27	Detail profilu používateľa - Profilový obrázok	Bol implementovaný cez Gravatar, rozhodli sme sa však, že umožníme používateľom nahrávať vlastný obrázok priamo na náš server. Úloha tak bola presunutá do druhého šprintu, kde súvisí s úlohou 1-51 Nastavenia profilu.

Tabuľka B.3: Poznámky k úlohám 1. šprintu

Ďalšie úlohy, ktoré nám vyplynuli pre nasledujúce šprinty:

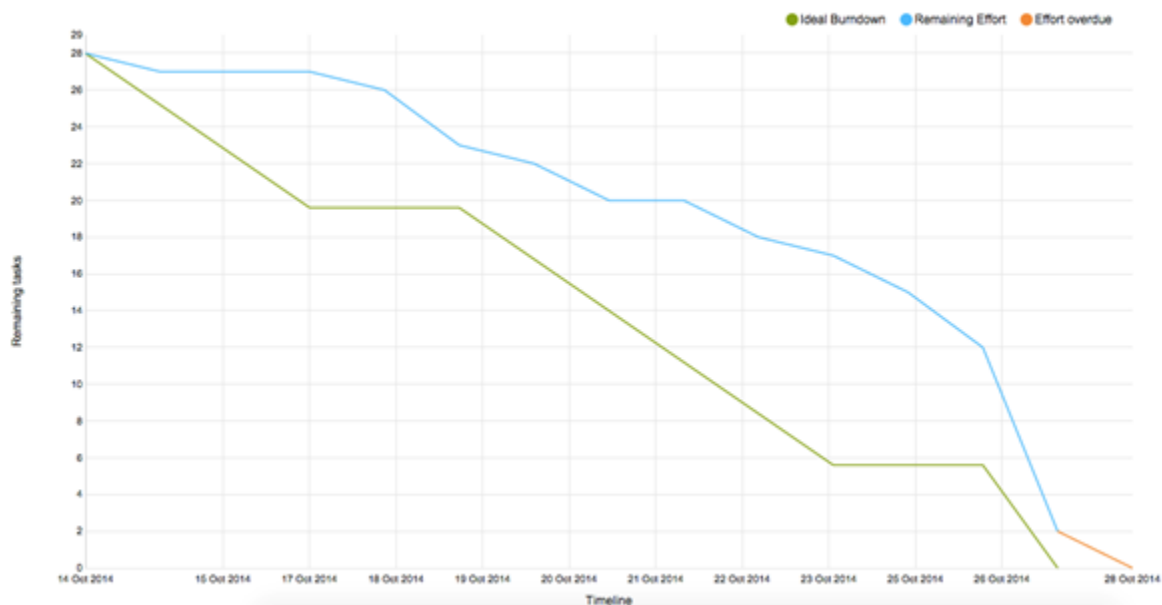
- Musíme si zdefinovať proces určený na prehliadky kódu.
- V nástroji YouTrack musíme pravidelne meniť stavy jednotlivých úloh. Niektoré z úloh sme v rámci zoznamovania sa s nástrojom označili ako vyriešené neskôr, ako sa nám ich skutočne poradilo vyriešiť. Taktiež nesmieme zabúdať predbežne si zaznamenávať čas trávený na jednotlivých úlohách.
- Počas prvého šprintu vytvárala úlohy v YouTracku osoba, ktorá bola stanovená ako vedúci stretnutia. Dohodli sme sa, že vo všetkých nasledujúcich šprintoch bude úlohy v YouTracku vždy vytvárať osoba zodpovedná za danú úlohu. Odbremeníme tím zát' až na vedúceho

stretnutia a zároveň bude zodpovedný za úlohu vždy informovaný (prostredníctvom upozornenia zo systému YouTrack) o stave jej plnenia.

- Zistili sme, že v mnohých prípadoch bola odhadovaná náročnosť úloh podhodnotená. V ďalších šprintoch už odhadujeme náročnosť aj s prihliadnutím na detaily. Zaviedli sme tiež používanie tzv. SCRUM pokeru, ktorý obsahuje štandardizované odhady úloh.
- Na základe prehľadu distribúcie úloh a ich časového prehľadu sme zistili, že je potrebné rovnomernejšie rozloženie záťaže medzi jednotlivých členov tímu.

B.1.4 Príloha 1: Burndown Chart

Na obrázku B.2 sa nachádza Burndown Chart za prvý šprint.

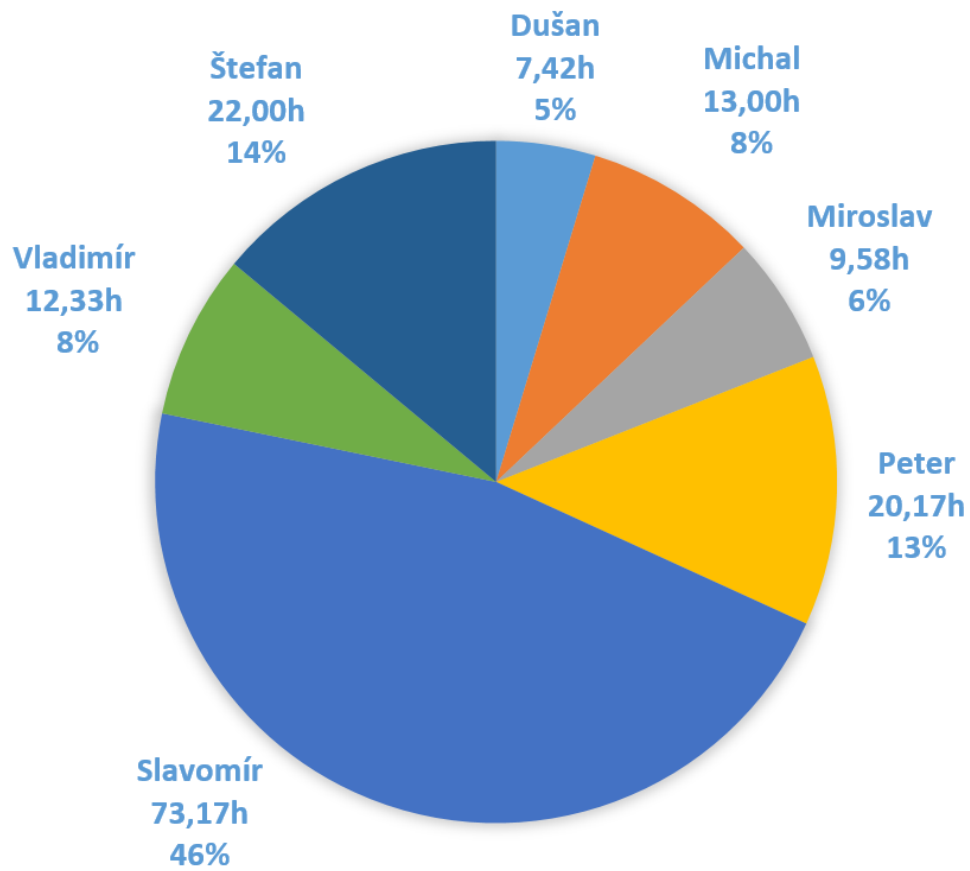


Obr. B.2: Burndown Chart za prvý šprint

B.1.5 Príloha 2: Distribúcia celkového času riešenia úloh medzi autorov

V grafe (Obrázok B.3) sa nachádza časové a následne percentuálne vyjadrenie podielu práce pre každého z autorov.

ČAS RIEŠENIA - CELKOVO



Obr. B.3: Distribúcia celkového času riešenia medzi autorov

B.2 Retrospektíva k 2. šprintu

Autori: Vladimír L'alk, Miroslav Šafárik

Trvanie šprintu: 28.10. – 11.11.

B.2.1 Zoznam úloh a stav ich riešenia

V tabuľke B.4 sa nachádza prehľad úloh, ktoré sme riešili počas druhého šprintu, stav ich splnenia a poznámky z nich vyplývajúce. Náplňou šprintu bolo implementovať rozširujúce funkcionality pre experiment (správa experimentu, detail experimentu) a nastavenia profilu používateľa. Okrem rozširujúcej funkcionality sa ako úplne nové implementovali notifikácie (ich vytvorenie v systéme a zobrazovanie).

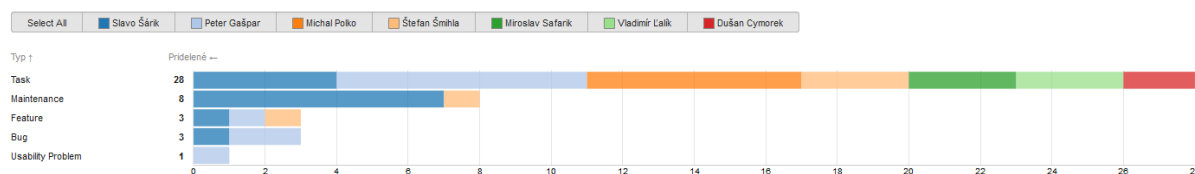
ID	Názov	Typ	Riešitelia	Záver
1-27	Profilový obrázok	Úloha	Michal	Akceptovaná
1-44	Správa experimentu	Funkcionality	Peter, Slavo	Akceptovaná
1-45	Šablóna - stránka experimentu	Úloha	Peter, Michal	Akceptovaná
1-46	Notifikácie	Úloha	Vlado, Miro	Akceptovaná
1-47	Šablóna - profil používateľa/spoločnosti	Úloha	Peter, Michal	Akceptovaná
1-48	Šablóna - registrácia/prihlásenie	Úloha	Peter, Michal	Akceptovaná
1-49	Detail experimentu	Funkcionality	Štefan	Akceptovaná
1-50	Pridanie neaktivovaného používateľa do experimentu	Chyba	Peter	Akceptovaná
1-51	Nastavenia profilu	Funkcionality	Dušan, Mišo	Riešená
1-52	Úprava prekladov	Úloha	Dušan	Akceptovaná
1-54	Upgrade Youtracku	Údržba	Slavo	Akceptovaná
1-55	Pridať roly tímu na hlavnú stránku	Problém s použititeľnosťou	Peter	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-56	Vytvoriť retrospektívu za 1. šprint	Úloha	Peter, Dušan, Štefan, Michal, Vlado, Miro, Slavo	Akceptovaná
1-59	Prepojenie nástrojov pre správu kódu s projektom	Údržba	Slavo	Akceptovaná
1-60	Nastavenie profilu - upload obrázka	Úloha	Michal	Akceptovaná
1-61	Nastavenie profilu - príprava migrácie	Úloha	Štefan	Akceptovaná
1-62	Nastavenie profilu - príprava view + napojenie na controller	Úloha	Dušan	Riešená
1-63	Vytvorenie udalosti v systéme	Úloha	Vlado	Akceptovaná
1-64	Nastavenie monitorovania servera a aplikácie	Údržba	Slavo	Akceptovaná
1-65	Správa experimentu - Zobrazenie používateľského rozhrania	Úloha	Slavo	Akceptovaná
1-66	Správa experimentu - Nastavenia detailov experimentu	Úloha	Slavo	Akceptovaná
1-67	Správa experimentu - Podmienky pre vstup používateľa do experimentu	Úloha	Slavo	Akceptovaná
1-68	Detail experimentu - zobrazenie informácií	Úloha	Štefan	Akceptovaná
1-69	Detail experimentu - prihlásenie do experimentu	Úloha	Štefan	Akceptovaná
1-71	Po vyžiadaní obnovy hesla sa podarí prihlásiť aj neaktívanému používateľovi	Chyba	Peter, Slavo	Akceptovaná
1-72	Vytváranie zdieľaných migrácií	Údržba	Slavo	Akceptovaná
1-73	Zobrazenie notifikácií	Úloha	Miro	Akceptovaná
1-74	Rekonfigurácia smtp	Údržba	Slavo	Akceptovaná
1-75	Správa experimentu - Značky	Úloha	Peter	Akceptovaná
1-76	Správa experimentu - Budget	Úloha	Peter	Akceptovaná
1-77	Správa experimentu - Správa participantov	Úloha	Peter	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-79	Príprava unit testov pre model v šprinte 2	Údržba	Štefan	Akceptovaná
1-80	Nastavenie servera na produkciu a staging	Údržba	Slavo	Akceptovaná
1-81	Integrácia systému - 2. šprint	Údržba	Slavo	Akceptovaná

Tabuľka B.4: Prehľad úloh

Na nasledujúcom grafe (obr. B.4) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. B.4: Distribúcia jednotlivých typov úloh medzi autorov

B.2.2 Časový prehľad

Tabuľka B.5 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	1	11,00
	Documentation	3	16,97
	Celkovo	4	27,97
Michal	Development	3	8,81
	Design	2	5,82
	Celkovo	5	16,63
Miroslav	Development	1	16,50
	Testing	1	1,00
	Documentation	1	3,00
	Celkovo	3	20,50

Riešiteľ	Typ	Počet úloh	Počet hodín
Peter	Development	7	24,05
	Testing	3	2,90
	Documentation	2	3,29
	Design	1	0,50
	Celkovo	13	30,24
Slavomír	Development	7	27,08
	Testing	1	0,50
	Documentation	2	13,50
	Maintenance	6	9,10
	Celkovo	16	50,18
Vladimír	Development	1	9,67
	Testing	1	4,29
	Documentation	2	4,39
	Celkovo	4	18,35
Štefan	Development	3	13,00
	Testing	1	3,00
	Documentation	1	3,00
	Celkovo	5	19,00

Tabuľka B.5: Časový prehľad úloh

B.2.3 Zhodnotenie

Počas hodnotenia šprintu č. 2 sme sa na stretnutí vyjadrili k jednotlivým úlohám a problémom s nimi spojenými. Poznámky k úlohám sa nachádzajú v tabuľke B.6.

ID	Názov	Poznámka
1-44	Správa experimentu	Dohodli sme, že je potrebné zmeniť systém pridávania podmienok pre experiment.
1-46	Notifikácie	Po zmenách vykonaných vo vytváraní experimentu sme sa dohodli, že notifikácia o vytvorení experimentu sa odošle až po jeho zverejnení a nie okamžite po vytvorení.
1-49	Detail experimentu	Dohodli sme sa na zmene názvu stĺpca „Stav zverejnenia“ na „Stav“.

ID	Názov	Poznámka
1-51	Nastavenie profilu	Zistili sme, že úlohu sa nepodarilo splniť v tomto šprinte. Dohodli sme sa, že skúsenosti používateľ a budú ako voľný text + automatické dopĺňanie textu. Zostáva dokončiť a doladiť fakturačnú a dodaciu adresu, doplniť pridávanie skúseností používateľ a.
1-52	Úprava prekladov	Prebehla úprava prekladov, pričom je vytvorená metodika na implementovanie prekladov, ktorou sa musia riadiť všetci členovia tímu.

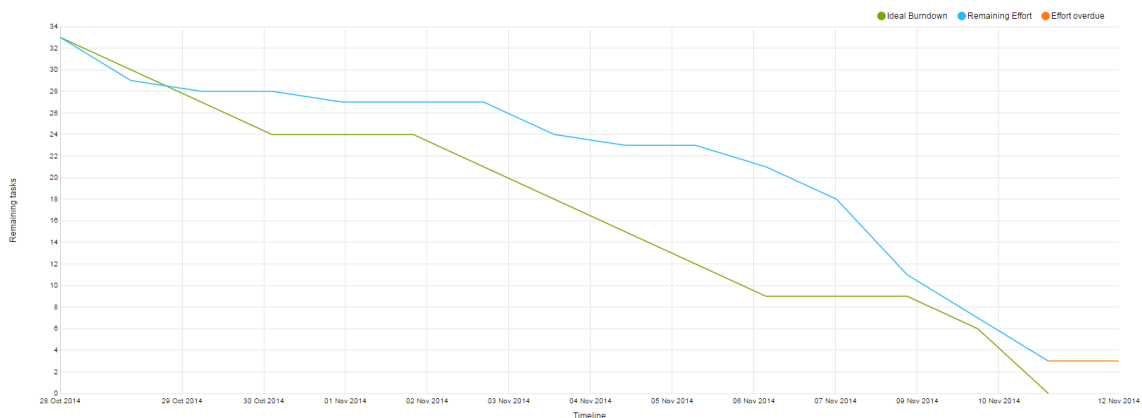
Tabuľka B.6: *Poznámky k úlohám 2. šprintu*

Ďalšie úlohy, ktoré vyplynuli pre nasledujúce šprinty:

- Každý člen tímu sa musí riadiť metodikami, ktoré sme si zdefinovali.
- Zmeniť oznamovací text pri obnovení hesla, aby sme zamedzili zisťovaniu emailových adries, ktoré sú v systéme zaregistrované
- Zvážiť možnosť použitia aplikácie na monitorovanie serveru, ktorú vytvára tím č. 10.
- V nasledujúcom šprinte refaktorovať doteraz implementované funkcionality, pričom je potrebné sa riadiť vytvorenými metodikami.
- Upraviť dokumentáciu tak aby bola konzistentná s implementovanou funkcionalitou.
- Na návrh prof. Bielikovej je potrebné preskúmať konkurenčnú službu CrowdFlower spolu s ich API.
- Realizovať odporúčanie experimentov nielen pomocou prirad'ovania značiek, ale aj na základe informácií o predchádzajúcich experimentov používateľ a.

B.2.4 Príloha 1: Burndown Chart

Na obrázku B.5 sa nachádza Burndown Chart za druhý šprint.

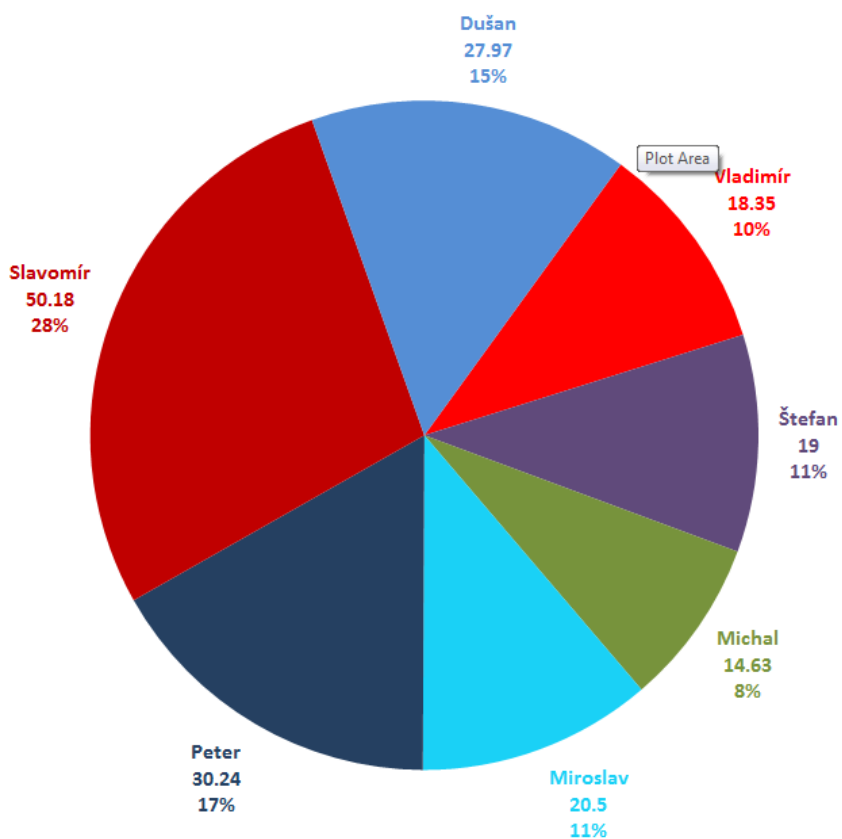


Obr. B.5: Burndown Chart za druhý šprint

B.2.5 Príloha 2: Distribúcia celkového času riešenia úloh medzi autorov

V grafe (obr. B.6) sa nachádza časové a následne percentuálne vyjadrenie podielu práce pre každého z autorov.

ČAS RIEŠENIA - CELKOVO



Obr. B.6: Distribúcia celkového času riešenia medzi autorov

B.3 Retrospektíva k 3. šprintu

Autori: Michal Polko, Štefan Šmihla

Trvanie šprintu: 11.11. – 18.11.

B.3.1 Zoznam úloh a stav ich riešenia

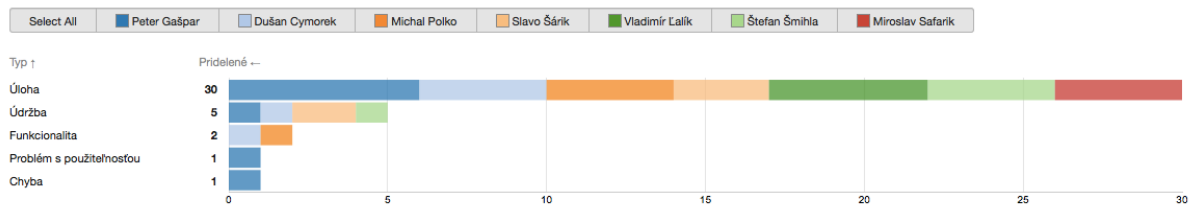
V Tabuľke B.7 sa nachádza prehľad úloh, ktoré sme riešili počas tretieho šprintu, stav ich splnenia a poznámky z nich vyplývajúce. Náplňou šprintu bol refactoring existujúcej funkcionality a opravovanie chýb.

ID	Názov	Typ	Riešitelia	Záver
1-51	Nastavenia profilu	Funkcionality	Dušan, Michal	Akceptovaná
1-57	Vytvorenie a odovzdanie dokumentácie – inžinierske dielo	Úloha	všetci	Akceptovaná
1-58	Vytvorenie a odovzdanie dokumentácie – riadenie	Úloha	všetci	Akceptovaná
1-62	Nastavenia profilu – príprava view + napojenie na controller	Úloha	Dušan	Akceptovaná
1-82	Integrácia systému – 3. šprint	Údržba	Slavo	Akceptovaná
1-83	Refactoring – pridávanie experimentu	Úloha	Peter	Akceptovaná
1-84	Chyba pri vyplnení registračného formulára	Bug	Peter	Akceptovaná
1-85	Preformulovanie textov	Problém s použiteľnosťou	Peter	Akceptovaná
1-87	Refactoring – experiment	Údržba	Štefan, Slavo	Akceptovaná
1-88	Refactoring – participanti	Úloha	Peter	Akceptovaná
1-89	Refactoring – dizajn	Úloha	Peter, Michal	Akceptovaná
1-90	Refactoring – preklady	Maintenance	Peter, Dušan	Akceptovaná
1-91	Priebežná príprava unit testov	Úloha	Štefan	Akceptovaná
1-92	Refactoring – follow	Úloha	Vlado	Akceptovaná
1-93	Refactoring – notifikácie	Úloha	Miro, Vlado	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-95	Vytvoriť retrospektívu za 2. šprint	Úloha	všetci	Akceptovaná

Tabuľka B.7: Prehľad úloh

Na nasledujúcom grafe (Obrázok B.7) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. B.7: Distribúcia jednotlivých typov úloh medzi autorov

B.3.2 Časový prehľad

Tabuľka B.8 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	1	9,77
	Documentation	2	7,13
	Maintentance	1	0,85
	Celkovo	4	19,75
Michal	Documentation	2	4,03
	Design	1	2,13
	Celkovo	3	6,17
Miroslav	Development	1	1,25
	Documentation	2	6,75
	Celkovo	3	8
Peter	Development	4	8,82
	Documentation	2	8,88
	Design	1	1,97
	Celkovo	7	19,67

Riešiteľ	Typ	Počet úloh	Počet hodín
Slavomír	Development	1	9,58
	Documentation	3	7,62
	Maintenance	1	1,5
	Celkovo	5	18,7
Vladimír	Development	2	4,58
	Documentation	5	5,33
	Celkovo	7	9,92
Štefan	Development	4	5
	Documentation	2	1,5
	Celkovo	6	6,5

Tabuľka B.8: Časový prehľad úloh

B.3.3 Zhodnotenie

Počas hodnotenia šprintu č. B.9 sme sa na stretnutí vyjadrili k jednotlivým úlohám a problémom s nimi spojenými. Poznámky k úlohám sa nachádzajú v tabuľke B.9.

ID	Názov	Poznámka
1-51	Dokončenie nastavení profilu	Úloha bola úspešne dokončená z predchádzajúceho šprintu. Bude však potrebné upraviť priečinok pre používateľské fotografie na produkcii.
1-57 & 1-58	Dokumentácia	Dohodli sme sa na používaní $\text{L}^{\text{T}}\text{E}^{\text{X}}$ pre písanie dokumentácie.
1-83	Refactoring – pridávanie experimentu	Dohodli sme sa, že pridávanie experimentu bude riešené cez modálne okno, nie Ajax.
1-87	Refactoring – experiment	Pridávanie používateľa do experimentu, pri požadovanom vzdelaní sa zatiaľ rieši podmienkou zhody.
1-89	Refactoring – dizajn	Na stretnutí prebehli krátke diskusie k návrhom na zlepšenie GUI.
1-90	Refactoring – preklady	Texty na stránke boli preformulované a smajlíky boli odstránené.
1-91	Priebežná príprava unit testov	Testy sú v súčasnosti funkčné a takmer kompletne pokrývajú registráciu, prihlásenie a reset hesla. Ostatné funkcionality sú pokryté len čiastočne.

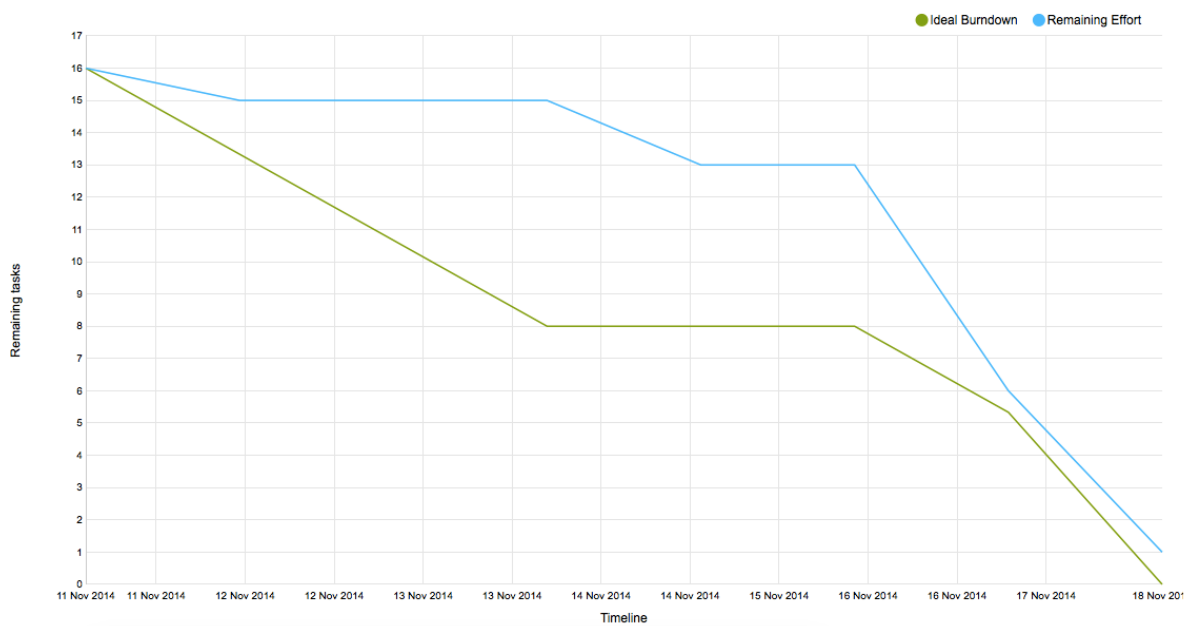
ID	Názov	Poznámka
----	-------	----------

Tabuľka B.9: Poznámky k úlohám

Ďalšie úlohy, ktoré vyplynuli pre nasledujúce šprinty:

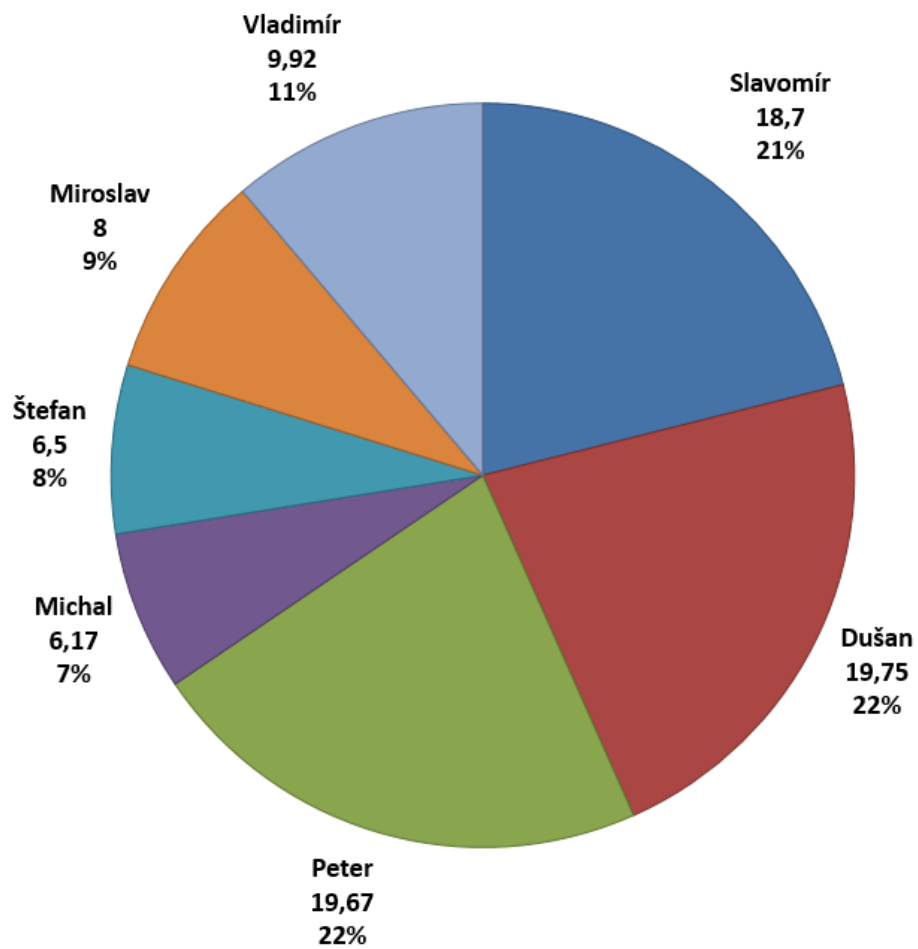
- Pozrieť nastavenia *YouTracku* týkajúce sa *Burndown Chart* - v súčasnom stave sa graf mení len ak sa úloha označí za *Fixed*. Nereflektuje počet strávených hodín pri práci na úlohách, čo vyúsťuje vo veľkej odchýlke voči ideálnemu stavu.
- Nasadiť *continuous integration* na server pre zvýšenie kvality kódu.
- Prejsť ponuku team *collaboration* portálov, nasadiť u nás a presunúť tam metodiky a návody z tímového Google Drive.

B.3.4 Príloha 1: Burndown Chart



Obr. B.8: *Burndown Chart* za tretí šprint

B.3.5 Príloha 2: Distribúcia celkového času riešenia úloh medzi autorov



Obr. B.9: Distribúcia celkového času riešenia medzi autorov

B.4 Retrospektíva k 4. šprintu

Autori: Dušan Cymorek, Peter Gašpar, Slavomír Šárik

Trvanie šprintu: 18.11. – 2.12.

B.4.1 Zoznam úloh a stav ich riešenia

V tabuľke B.10 sa nachádza prehľad úloh, ktoré sme riešili počas štvrtého šprintu, stav ich splnenia a poznámky z nich vyplývajúce. Náplňou tohto šprintu bolo implementovanie základných funkcionalít systému.

ID	Názov	Typ	Riešitelia	Záver
1-97	Vytvoriť retrospektívu za 3. šprint	Úloha	Štefan, Michal	Akceptovaná
1-98	Editácia profilu - chýbajúce preklady pri nahrávaní nesprávnej profilovej fotografie	Chyba	Michal	Akceptovaná
1-99	Editácia profilu - pridať texty o maximálnej veľkosti profilovej fotografie a type súboru.	Úloha	Michal	Akceptovaná
1-100	Šablóna - vyhľadávanie / filtrovanie výsledkov	Úloha	Peter, Michal	Akceptovaná
1-101	Šablóna - landing page	Úloha	Peter, Michal	Akceptovaná
1-102	Šablóna - dashboard	Úloha	Peter, Michal	Akceptovaná
1-103	Dynamické bočné menu	Úloha	Peter, Michal	Akceptovaná
1-104	Datagrid pre zoznamy	Úloha	Peter, Michal	Akceptovaná
1-105	Plátno biznis modelu	Úloha	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír	Akceptovaná
1-106	Vyhľadávanie v systéme	Úloha	Vladimír, Slavomír	Akceptovaná

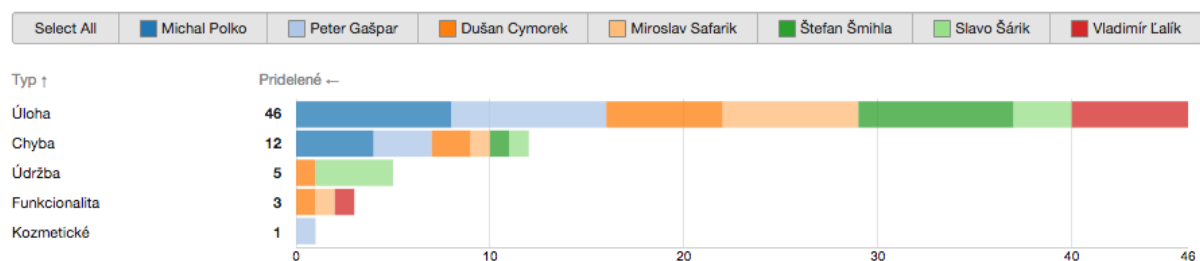
ID	Názov	Typ	Riešitelia	Záver
1-107	Vyhľadávanie v systéme - špecifikácia	Úloha	Dušan	Akceptovaná
1-108	Integrácia ElasticSearch	Údržba	Slavomír	Akceptovaná
1-109	Spustenie experimentu	Funkcionálna	Miroslav	Akceptovaná
1-110	Spustenie experimentu - špecifikácia	Úloha	Štefan	Akceptovaná
1-111	Integrácia 4. sprint	Údržba	Slavomír	Akceptovaná
1-112	Filtrovanie výsledkov vyhľadávania	Úloha	Štefan	Akceptovaná
1-113	Filtrovanie výsledkov vyhľadávania - špecifikácia.	Úloha	Vladimír	Akceptovaná
1-114	Spustenie experimentu - príprava migrácie	Úloha	Štefan	Akceptovaná
1-115	Spustenie experimentu - príprava cron úlohy	Úloha	Miroslav	Akceptovaná
1-116	Spustenie experimentu - aktualizácia nastavení experimentu	Úloha	Miroslav	Akceptovaná
1-117	Spustenie experimentu - aktualizácia notifikácií	Úloha	Miroslav	Akceptovaná
1-118	Spustenie experimentu - aktualizácia detailu experimentu	Úloha	Miroslav	Akceptovaná
1-119	Dashboard - špecifikácia	Úloha	Peter	Akceptovaná
1-122	Vyhľadávanie v systéme - vyhľadávanie experimentov	Úloha	Slavomír	Akceptovaná
1-123	Vyhľadávanie v systéme - vyhľadávanie používateľov	Úloha	Vladimír	Akceptovaná
1-124	Pozvanie používateľa do experimentu	Funkcionálna	Vladimír	Akceptovaná
1-125	Pozvanie používateľa do experimentu - špecifikácia	Úloha	Miroslav	Akceptovaná
1-126	Export dát do CVS - Prepracovanie exportu	Úloha	Peter	Akceptovaná
1-127	Profil používateľa - follower list je zobrazený dvakrát	Chyba	Michal	Akceptovaná
1-128	Profil používateľa - v názve firmy sa zobrazuje AAA a v zozname skillov TODO	Chyba	Dušan	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-129	Hodnotenie používateľov	Funkcionalita	Dušan	Akceptovaná
1-131	Hodnotenie používateľov - špecifikácia	Úloha	Miroslav	Akceptovaná
1-132	Pozvanie používateľa do experimentu - úprava správy experimentu	Úloha	Vladimír	Akceptovaná
1-133	Pozvanie používateľa do experimentu - odoslanie pozvánky	Úloha	Vladimír	Akceptovaná
1-134	Hodnotenie používateľov - úprava správy experimentu	Úloha	Dušan	Akceptovaná
1-135	Hodnotenie používateľov - systém hodnotenia	Úloha	Dušan	Akceptovaná
1-136	Hodnotenie používateľov - úprava detailu experimentu	Úloha	Dušan	Akceptovaná
1-137	Hodnotenie používateľov - zobrazovanie hodnotenia	Úloha	Dušan	Akceptovaná
1-138	Filtrovanie používateľov.	Úloha	Štefan	Akceptovaná
1-139	Filtrovanie experimentov.	Úloha	Štefan	Akceptovaná
1-142	Produkcia - premazávanie uploads adresára	Chyba	Slavomír	Akceptovaná
1-144	Notifikácie - znefunkčnenie stránky po zmazení notifikovaného experimentu	Chyba	Miroslav	Akceptovaná
1-147	Experiment - nesprávne profilové fotografie	Chyba	Michal	Akceptovaná
1-148	Oprava validácie používateľa pri prihlasovaní	Údržba	Dušan	Akceptovaná
1-150	Chyby pri schvaľovaní participantov	Chyba	Peter	Akceptovaná
1-155	Vytvorenie skriptu na vytvorenie dummy databázy	Úloha	Štefan	Akceptovaná
1-157	Landing Page - Nezobrazovanie správ	Chyba	Michal	Akceptovaná
1-160	Integrácia MediaWiki	Údržba	Slavomír	Akceptovaná
1-161	Stránkovanie - prepojiť stránkovanie s Bootstrap rámcom	Kozmetické	Peter	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-162	Používateľ - možnosť editácie profilu iného používateľa	Chyba	Dušan	Akceptovaná
1-163	Continuous integration	Údržba	Slavomír	Akceptovaná
1-165	Oprava existujúcich unit testov	Chyba	Štefan	Akceptovaná

Tabuľka B.10: Prehľad úloh

Na nasledujúcom grafe (Obr. B.10) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. B.10: Distribúcia jednotlivých typov úloh medzi autorov

B.4.2 Časový prehľad

Tabuľka B.11 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	4	6,22
	Documentation	2	7,20
	Testing	1	1,80
	Maintenance	3	1,28
	Celkovo	10	16,50
Michal	Development	6	9,12
	Documentation	1	1,57
	Design	2	4,73
	Celkovo	9	15,42

Riešiteľ	Typ	Počet úloh	Počet hodín
Miroslav	Development	4	8,42
	Documentation	3	6,42
	Testing	3	4,00
	Maintenance	1	0,83
	Celkovo	11	19,67
Peter	Development	7	25,33
	Documentation	1	0,92
	Testing	1	0,50
	Design	2	2,70
	Celkovo	11	29,45
Slavomír	Development	1	18,42
	Documentation	1	1,08
	Testing	2	4,22
	Maintenance	6	16,58
	Celkovo	10	40,30
Vladimír	Development	2	10,33
	Documentation	1	3,60
	Testing	1	0,42
	Celkovo	4	14,35
Štefan	Development	5	24,00
	Documentation	3	8,00
	Celkovo	8	32,00

Tabuľka B.11: Časový prehľad úloh

B.4.3 Zhodnotenie

Pri retrospektíve k šprintu sme sa na stretnutí vyjadrili k jednotlivým úlohám a problémom, ktoré pri ich riešení nastali. Poznámky k jednotlivým úlohám sa nachádzajú v tabuľke B.12.

ID	Názov	Poznámka
1-106	Vyhľadávanie v systéme	Je potrebné pridať tlačidlo pre vyhľadávanie. Súčasný stav, kedy sa vyhľadávanie potvrdzuje stlačením tlačidla Enter, nemusí byť jasný pre všetkých používateľov. <u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> Pridanie tlačidla na odoslanie vyhľadávaného výrazu.
1-129	Hodnotenie používateľov	Zhodli sme sa na zmene spôsobu hodnotenia zo súčasných dvoch stavov (páči, nepáči) na viacstupňové hodnotenie. Okrem toho je potrebné myslieť na vhodné škálovanie hodnotenia. <u>Úlohy, ktoré vyplynuli pre ďalšie šprinty:</u> Úprava systému hodnotenia na viacstupňové hodnotenie.

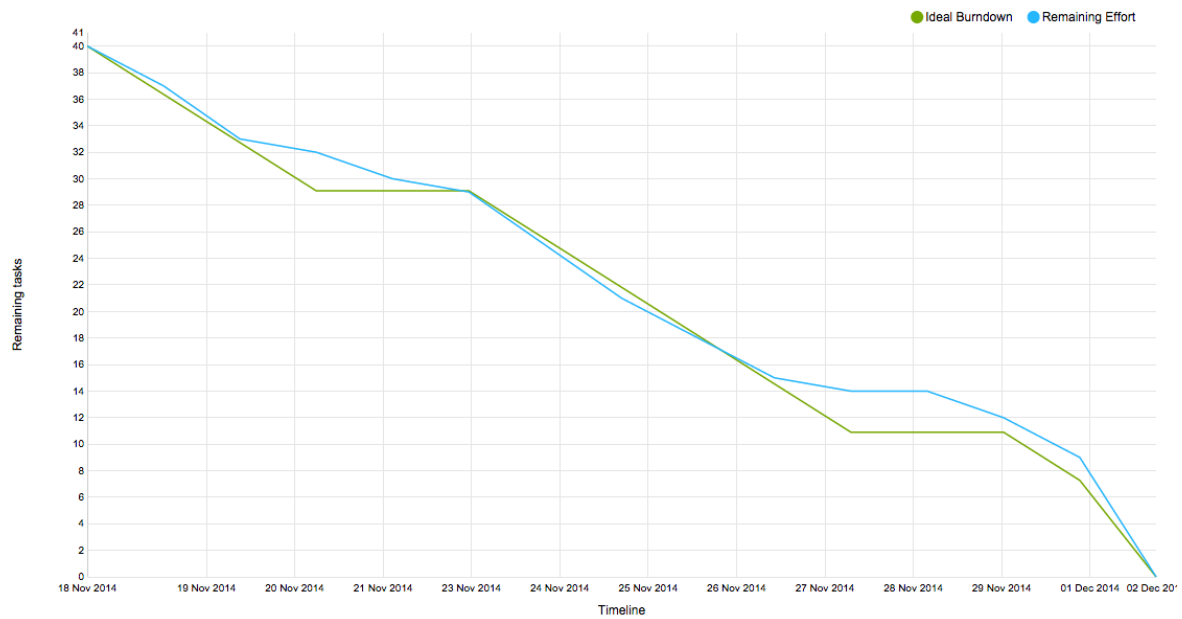
Tabuľka B.12: Poznámky k úlohám

Ďalšie úlohy, ktoré nám vyplynuli pre nasledujúce šprinty:

- Dohodli sme sa na používaní MediaWiki na špecifikácie a metodiky, aby sa na jednom mieste nachádzala vždy najnovšia verzia. Tento spôsob sa bude používať od letného semestra.
- V zdrojovom kóde sa nám nahromadili úlohy označené TO DO, dohodli sme sa, že budú riešené v ďalších šprintoch.
- Premyslieť možné spôsoby odmien pre participantov. Jedno z navrhovaných riešení bolo vytoriť v systéme vlastnú menu a obchod s rôznymi predmetmi.
- Je potrebné definovať životný cyklus experimentu.
- Pri každom zásahu do funkcionality skontrolovať aktuálne testy a zabezpečiť, aby ich zbehnutie skončilo úspešne - v prípade, že to nie je možné dosiahnuť, treba kontaktovať priamo Števa - bližšie info v metodike.
- Slavo nasadil continuous integration. Prípadné chyby, ktoré sa prejavajú pri testovaní je potrebné opraviť.

B.4.4 Príloha 1: Burndown Chart

Na obrázku B.11 sa nachádza Burndown Chart za štvrtý šprint.

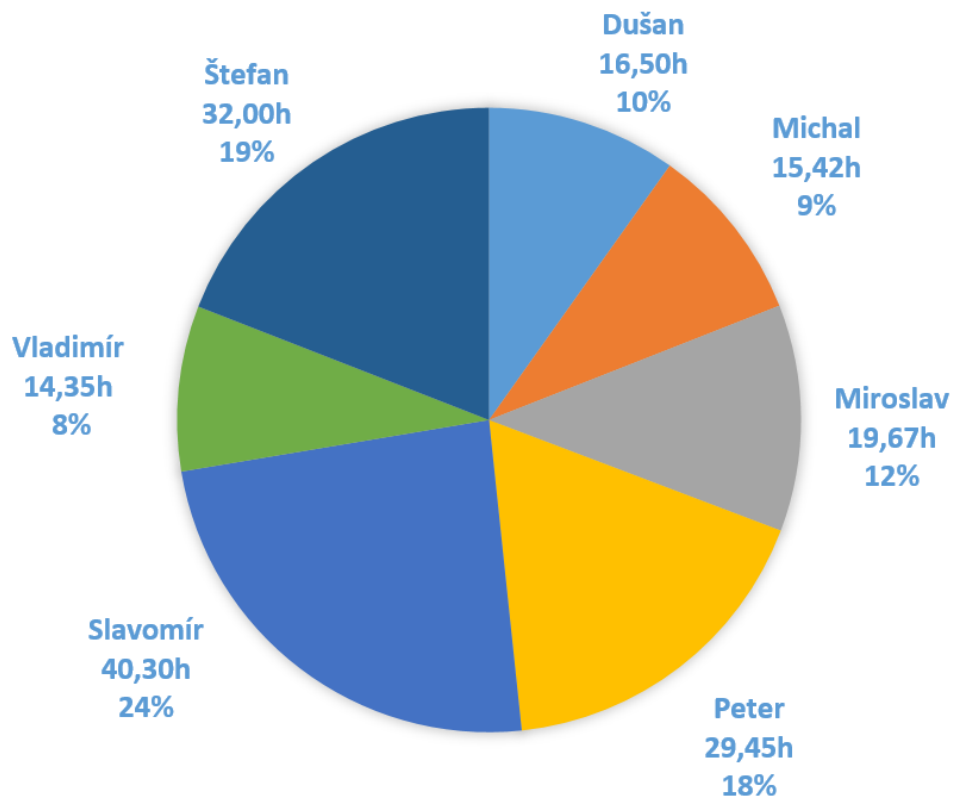


Obr. B.11: *Burndown Chart za štvrtý šprint*

B.4.5 Príloha 2: Distribúcia celkového času riešenia úloh medzi autorov

V grafe (Obrázok B.12) sa nachádza časové a následne percentuálne vyjadrenie podielu práce pre každého z autorov.

ČAS RIEŠENIA - CELKOVO



Obr. B.12: Distribúcia celkového času riešenia medzi autorov

B.5 Retrospektíva k 5. šprintu

Autori: Vladimír L'alík, Miroslav Šafárik

Trvanie šprintu: 2.12. – 9.12.

B.5.1 Zoznam úloh a stav ich riešenia

V tabuľke B.13 sa nachádza prehľad úloh, ktoré sme riešili počas štvrtého šprintu, stav ich splnenia a poznámky z nich vyplývajúce. Náplňou šprintu bol pridávanie novej funkcionality, dizajnových riešení a oprava chýb.

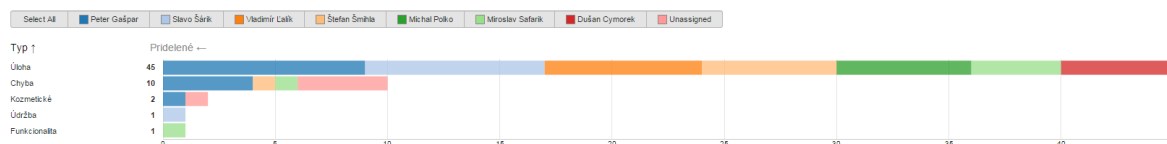
ID	Názov	Typ	Riešitelia	Záver
1-96	Oprava dizajnu nastavení profilu	Úloha	Peter	Vytvorená
1-120	Vytvorenie a odovzdanie dokumentácie - inžinierske dielo - FINAL ZS	Úloha	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír	Riešená
1-121	Vytvorenie a odovzdanie dokumentácie - riadenie - FINAL ZS	Úloha	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír	Riešená
1-143	Editácia profilu - Chyba pri dlhom texte "Ö mne"	Chyba	-	Vytvorená
1-145	Experiment management - zakázať vymazanie experimentu zo systému	Chyba	-	Vytvorená
1-146	Používateľ - upraviť account_status v modeli používateľa	Chyba	-	Vytvorená
1-149	Preklady - pridať podporu pre časové preklady	Kozmetické	Peter	Akceptovaná
1-151	Vytvorenie textov pre Uvítaciu stránku (Landing Page)	Úloha	Peter, Dušan	Akceptovaná
1-152	Notifikácie - chyba v zobrazovaní notifikácií	Chyba	Miroslav	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-154	Dashboard - nesprávny zoznam experimentov	Chyba	Peter	Akceptovaná
1-156	Vyhľadávanie - chyba pri vyhľadávaní špeciálnych znakov	Chyba	-	Vytvorená
1-158	Prihlasovanie/Obnova hesla - chyba po prihlásení	Chyba	Peter	Dokončené
1-159	Odhlásenie - Chyba pri odhlasovaní z dashboardu	Chyba	Peter	Dokončené
1-166	User profile - chybné zobrazenie účasti v experiment	Chyba	Peter	Dokončené
1-167	Realizácia experimentu	Úloha	Štefan	Akceptovaná
1-168	Inštrukcie pri spustení experimentu	Úloha	Michal	Akceptovaná
1-169	Pridávanie komentárov ku experimentu	Úloha	Vladimír, Slavomír	Akceptovaná
1-170	Implementácia komentárov	Úloha	Vladimír, Slavomír	Akceptovaná
1-171	Integrácia 5. šprint	Údržba	Slavomír	Akceptovaná
1-172	Dashboard - implementácia	Úloha	Peter	Akceptovaná
1-173	Špecifikácia - komentáre v experimente	Úloha	Vladimír, Slavomír	Akceptovaná
1-174	Vypnutie / zapnutie notifikácií	Funkcionálna	Miroslav	Akceptovaná
1-175	Profil - fotografia	Kozmetické	-	Vytvorená
1-176	Komentáre v experimentoch - upgrade + dizajn	Úloha	Peter, Michal	Riešená
1-177	Vytvorenie prezentácie manažmentu riadenia projektu	Úloha	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír	Vytvorená
1-178	Realizácia experimentu - vytvorenie špecifikácie	Úloha	Štefan	Akceptovaná
1-179	Aktualizácia exportu z YouTracku	Úloha	Peter	Akceptovaná
1-180	Úprava komentárov	Úloha	Slavomír	Akceptovaná

ID	Názov	Typ	Riešitelia	Záver
1-181	Vytvorenie prezentácie pre biznis plátno	Úloha	Peter, Dušan, Štefan, Michal, Vladimír, Miroslav, Slavomír	Akceptovaná
1-182	Chyba pri otvarani experimentu	Chyba	Štefan	Dokončené

Tabuľka B.13: *Prehľad úloh*

Na nasledujúcom grafe (Obrázok B.13) môžeme vidieť distribúciu jednotlivých typov úloh medzi autorov.



Obr. B.13: *Distribúcia jednotlivých typov úloh medzi autorov*

B.5.2 Časový prehľad

Tabuľka B.14 obsahuje prehľad úloh a potrebného času na ich realizáciu pre jednotlivých členov tímu.

Riešiteľ	Typ	Počet úloh	Počet hodín
Dušan	Development	1	2,43
	Documentation	1	3,19
	Maintenance	1	1,15
	Celkovo	3	6,77
Michal	Development	3	6,59
	Documentation	2	4,46
	Celkovo	5	11,05

Riešiteľ	Typ	Počet úloh	Počet hodín
Miroslav	Development	1	1,17
	Testing	1	0,50
	Documentation	2	8,83
	Maintenance	1	0,58
	Celkovo	5	11,08
Peter	Development	9	13,48
	Documentation	2	5,90
	Celkovo	11	19,38
Slavomír	Development	3	8,00
	Documentation	2	10,33
	Maintenance	1	2,67
	Celkovo	7	21,00
Vladimír	Development	2	7,02
	Testing	2	2,17
	Documentation	3	4,58
	Celkovo	7	13,77
Štefan	Development	2	13,50
	Documentation	2	6,50
	Celkovo	4	20,00

Tabuľka B.14: Časový prehľad úloh

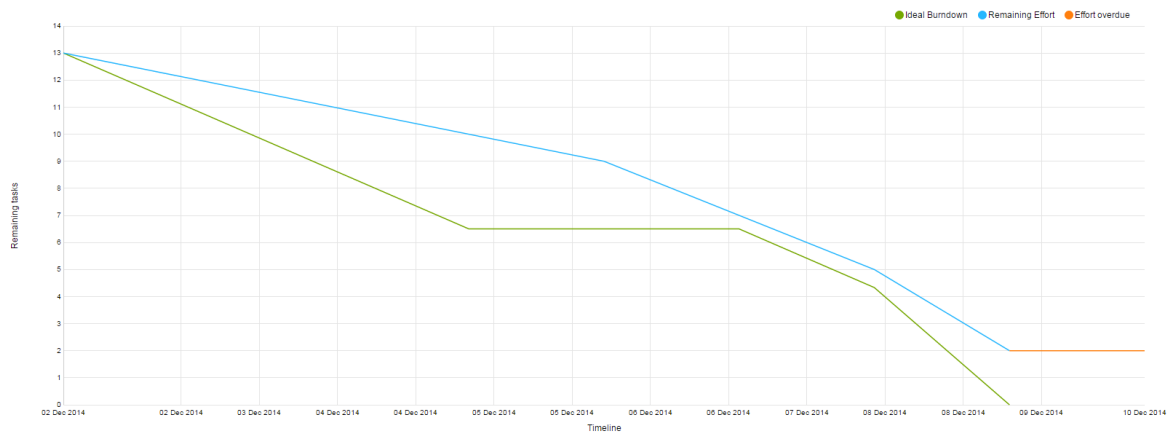
B.5.3 Zhodnotenie

Pri retrospektíve k šprintu sme sa na stretnutí vyjadrili k jednotlivým úlohám a problémom, ktoré pri ich riešení nastali. Ďalšie úlohy, ktoré nám vyplynuli z tohto šprintu:

- Úlohy, ktoré neboli dokončené v tomto šprinte, budú uzavreté počas skúškového obdobia vo vianočnom šprinte.
- Treba spraviť dizajn pre podmienky úloh.
- Pri zobrazení zoznamu úloh stĺpec „Oprávnenosť“ zaberá príliš veľa miesta na šírku.
- Pridať pre zadávateľa možnosť mazať nevhodné komentáre.
- Kurzor pri textových poliach prerobiť na textový kurzor, nie ukazovák.
- Dávať si pozor na prípadný ženský rod v textoch.

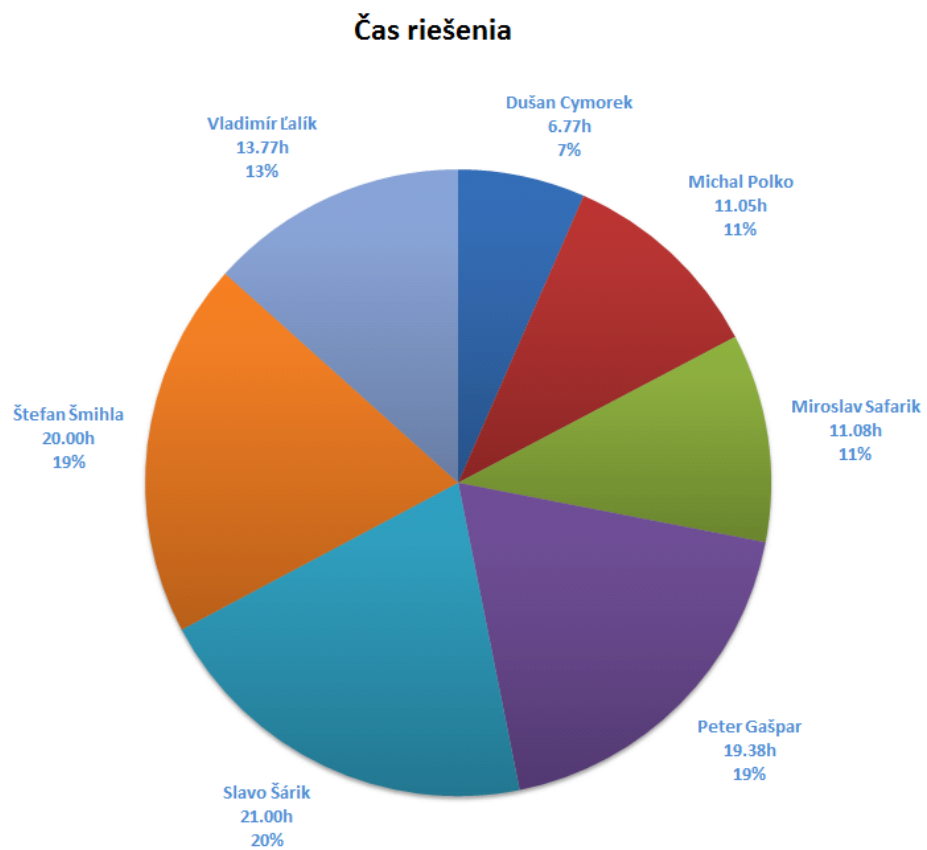
- Dušan plánuje prerobiť systém prekladov.
- Riešenie úloh, ktoré sú označené v kóde ako *TO DO*.
- Doriešenie rozpočtu pre používateľov a pre úlohy.

B.5.4 Príloha 1: Burndown Chart



Obr. B.14: *Burndown Chart za piaty šprint*

B.5.5 Príloha 2: Distribúcia celkového času riešenia úloh medzi autorov



Obr. B.15: Distribúcia celkového času riešenia medzi autorov