

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Webový komunitný systém otázok a odpovedí

Dokumentácia k riadeniu projektu

Vedúci tímu: Ing. Ivan Srba

Členovia tímu: Bc. Rastislav Dobšovič, Bc. Marek Grznár, Bc. Jozef Harinek,
Bc. Samuel Molnár, Bc. Peter Páleník, Bc. Dušan Poizl, Bc. Pavol Zbell

Akademický rok: 2013/2014

Obsah

1 Úvod.....	1
2 Ponuka.....	2
2.1 Predstavenie členov tímu.....	2
2.2 Motivácia.....	3
3 Úlohy členov tímu.....	5
3.1 Manažérske úlohy.....	5
3.2 Podiel práce.....	5
3.3 Autori častí dokumentácie.....	7
4 Manažment kvality.....	8
4.1 Zvyšovanie kvality zdrojového kódu.....	8
4.2 Kontrola splnenia požiadaviek.....	13
5 Manažment monitorovania.....	21
6 Manažment rizík.....	21
6.1 Identifikované riziká.....	21
7 Manažment plánovania.....	25
7.1 Popis úlohy manažéra plánovania.....	25
7.2 Metodika pre vkladanie novej úlohy v nástroji Redmine.....	25
8 Manažment podpory vývoja a integrácie.....	27
8.1 Úvod k metodikám pre podporu vývoja a integrácie.....	28
8.2 Použité skratky a značky.....	28
8.3 Postupy.....	28
9 Manažment komunikácie a ľudských zdrojov.....	35
9.1 Úvod k metodike.....	35
9.2 Prehľad postupov pri riešení komunikácie.....	35
9.3 Jednotlivé komunikačné nástroje.....	36
10 Manažment tvorby dokumentácie.....	40
10.1 Metodiky pre dokumentáciu.....	40
11 Preberacie protokoly.....	42
A)Záznamy zo stretnutí.....	43
1 Zápis č. 1 zo stretnutia tímu č. 13.....	43
2 Zápis č. 2 zo stretnutia tímu č. 13.....	47

3 Zápis č. 3 zo stretnutia tímu č. 13.....	51
4 Zápis č. 4 zo stretnutia tímu č. 13.....	56
5 Zápis č. 5 zo stretnutia tímu č. 13.....	60
6 Zápis č. 6 zo stretnutia tímu č. 13.....	65
B)Záznamy z neformálnych stretnutí.....	69
1 Zápis č. 1 z neformálneho stretnutia tímu č. 13.....	69
2 Zápis č. 2 z neformálneho stretnutia tímu č. 13.....	71

1 Úvod

Tento dokument slúži ako dokumentácia k riadeniu projektu s názvom Webový komunitný systém otázok a odpovedí k predmetu Tímový projekt, ktorý je vypracovávaný na Fakulte informatiky a informačných technológií na Slovenskej univerzite v Bratislave.

Nami vytváraný webový systém je systém, ktorého úlohou je umožniť používateľom klásť otázky a odpovedanie na položené otázky. Z odpovedí si vyberie osoba, ktorá otázku položila, tú najvhodnejšiu a označí ju.

Druhá kapitola obsahuje ponuku tímu na nám pridelenú tému, ktorá obsahuje predstavenie členov tímu a motiváciu.

Tretia kapitola obsahuje jednotlivé záznamy zo stretnutí pre každý uskutočnený šprint.

Štvrtá kapitola sa venuje manažérskym rolám členov tímu a ich zodpovednostiam. Obsahuje aj tabuľku s autormi jednotlivých častí dokumentácie.

2 Ponuka

Pre pridelenie témy Webový komunitný systém otázok a odpovedí sme vypracovali ponuku.

2.1 Predstavenie členov tímu

Bc. Rastislav Dobšovič

- bakalársky projekt Optické rozpoznávanie notových záznamov, C++, OpenCV
- pracovná skúsenosť na pozícii databázového vývojára pre Orange Slovensko
- Oracle Database, PL/SQL, MySQL
- zaujíma sa o procesy návrhu softvéru, UML, návrhové vzory a hráva na klavíri

Bc. Marek Grznár

- bakalársky projekt Adaptívna spätná väzba vo webovom systéme, Ruby on Rails, Alef
- pracovná skúsenosť s technológiami ASP.NET, C#, SVN
- JavaScript, MySQL, git, Linux
- člen PeWe, zaujíma sa o vizualizáciu dát vo webových aplikáciách a rád hráva futbal

Bc. Jozef Harinek

- bakalársky projekt Automatická extrakcia kľúčových slov z výučbových dokumentov, Ruby on Rails, výskumný článok na konferenciu DocEng, príspevok na IIT.SRC
- letná stáž v Nemecku C++, Qt, SVN, skúsenosti s MySQL, SQLite, git, Linux
- zaujíma sa o NLP, webové aplikácie a ľahké ontológie
- člen výskumného semináru a PeWe, baví ho lezenie po skalách

Bc. Pavol Zbell

- bakalársky projekt Vyhľadávanie v zdrojových kódach so zohľadnením kontextu, Java, Eclipse APIs, Elasticsearch, PerConIK, pokročilé OOP v Jave a návrhové vzory všeobecne
- autor projektu Otvorené Súdy¹, sťahovanie a spracovanie dát, správa databázy, dizajn
- pokročilé skúsenosti s Twitter Bootstrap, Resque, Redis, git, Linux
- člen výskumného semináru, PeWe a aktívny freeride lyžiar

Bc. Samuel Molnár

- bakalársky projekt Navigácia pomocou oblakov dôležitých slov, Ruby on Rails, Annota, Elasticsearch, backbone.js, CoffeeScript, výskumný článok na SMAP a IIT.SRC
- autor projektu Otvorené Súdy¹, indikátory činnosti sudcov, Elasticsearch, správa servera
- presadzuje TDD, má intenzívnu skúsenosť s Rspec, Capybara, git, Linux
- člen výskumného semináru, PeWe a rád púšťa šarkany

¹ Otvorené Súdy – viac ako 1 milión dokumentov, viac ako 0,5 TB dát, viac ako tisíc návštev denne a hlavná cena v súťaži Reštart Slovensko. Ruby on Rails, PostgreSQL a tri-gramy, Elasticsearch, Resque, Redis. Twitter Bootstrap, SCSS, CoffeeScript, HTML 5. Spolupráca s Transparency International a open source v rámci siete GitHub.

Bc. Dušan Poizl

- bakalársky projekt Simulácia celulárnych automatov na GPU, C++, OpenCL, Qt
- skúsenosti s OpenGL, Gstreamer, git, Linux admin, iOS vývoj, MySQL, Ruby on Rails
- zaujíma sa o skladanie astronomických fotografií a učí sa po japonsky

Bc. Peter Páleník

- bakalársky projekt Cestná sieť ako komplexný graf, C#, MySQL
- pracuje s WPF, .NET technológiami a rád hráva futbal

2.2 Motivácia

Tému CQA sme si vybrali preto, že nás zaujíma web ako platforma pre tvorbu interaktívnych aplikácií. Myslíme si, že systém CQA je potrebný v rámci fakulty primárne preto, aby pomohol študentom v procese učenia sa nových technológií, riešením zadaní či prípravy na skúšky. Súčasne je tento systém potrebný aby inovatívnou formou nahradil už zastaralé riešenie v podobe obyčajného diskusného fóra. Pre študentov je výhodnejšie mať priame spojenie s uzavretou komunitou, ktorá rieši podobné problémy, v rámci fakultnej platformy ako na verejnom systéme CQA typicky StackOverflow.

Tím a námety pre danú tému:

- podľa vzoru zo StackOverflow by sa zaviedlo ohodnotenie prispievateľov, aby sa v neskorších fázach mohla určovať relevancia jednotlivých odpovedí,
- pokročilé kategorizovanie otázok, tagovanie a vyhľadávanie (fulltextové, fazetové a pod.),
- podpora pre kolaboratívne vypracovávanie otázok v reálnom čase,
- pre lepšiu názornosť umožniť pridávať do otázky médiá rôzneho formátu, hlavne podpora vkladania obrázkov, prípadne nástroj na modelovanie diagramov ako napr. UML,
- automatické formátovanie a zvýraznenie zdrojového kódu (chýba na StackOverflow),
- personalizovať obsah a odporúčať používateľom otázky, ktoré by ich mohli zaujímať,
- zavedenie prehľadnej štruktúry pre otázky a skupiny otázok, aby sa zachovali a boli ľahko dostupné aj pre ďalšie generácie študentov.

Tím a téma technologického hľadiska:

- väčšina členov tímu má reálnu skúsenosť s Ruby on Rails aplikáciami, ambíciu zdokonaľiť sa v jazyku Ruby a záujem o webové aplikácie z viacerých aspektov,
- ako tím nemáme problém postaviť riešenie na moderných technológiách, plne využiť potenciál NoSQL, vyhľadávacieho stroja Elasticsearch alebo podľa potreby nasadiť server side JavaScript či WebSockets riešenia,
- tím má v pláne klásť dôraz na TDD, čo je vo svete moderného vývoja webových aplikácií žiaduce a chce pritom využiť nástroj Capybara, prípadne Cucumber ako lepšiu možnosť definovania používateľských požiadaviek pri komunikácii so zákazníkom.

Tím a téma z výskumného hľadiska:

- väčšina členov tímu sú aktívni členovia PeWe,
- niektorým členom tímu sa podarilo na základe bakalárskych prác napísať výskumné články na IIT.SRC a medzinárodné vedecké konferencie,
- ďalší členovia tímu pracovali alebo pracujú na výskumných projektoch v rámci fakulty,
- ako tím máme ambíciu nájsť v téme výskumný problém, navrhnúť jeho riešenie a úspešne publikovať príspevok na IIT.SRC.

Skúsenosti tímu a predpoklady pre zvládnutie témy výrazne dopĺňajú autori úspešného projektu Otvorené Súdny, ktorí v produkčnom prostredí prevádzkujú Ruby on Rails aplikáciu postavenú na rovnakých alebo podobných technológiách, ktoré sa predpokladajú aj pri realizácii tejto témy.

Väčšina členov tímu má skúsenosť s Ruby on Rails, ambíciu zdokonaľiť sa v jazyku Ruby, záujem o webové aplikácie a potenciál, postaviť sa k téme aj z výskumného hľadiska, preto si ako tím myslíme, že splníme predpoklady na úspešné a nadštandardné vyriešenie danej témy.

3 Úlohy členov tímu

3.1 Manažérske úlohy

Manažérke úlohy boli pridelené na tímovom stretnutí podľa skúseností členov tímu.

Bc. Rastislav Dobšovič	Manažér plánovania
Bc. Marek Grznár	Manažér kvality
Bc. Jozef Harinek	Manažér komunikácie a ľudských zdrojov Vedúci tímu
Bc. Samuel Molnár	Manažér podpory vývoja
Bc. Peter Páleník	Manažér dokumentácie
Bc. Dušan Poizl	Manažér monitorovania
Bc. Pavol Zbell	Manažér rizík Zástupca vedúceho tímu a hlavný architekt

Tab. 1: Manažérske úlohy členov tímu

3.2 Podiel práce

3.2.1 Šprint 1 – Drone

Úloha	Zodpovedná osoba	Ohodnotenie User Story
Autorizácia	Samuel Molnár	3
Lokalizácia	Pavol Zbell	3
Prihlásenie	Samuel Molnár	5
Profil používateľa	Jozef Harinek, Marek Grznár	8
Zaznamenávanie udalostí	Pavol Zbell	5

Tab. 2: Podiel práce na prvom šprinte

3.2.2 Šprint 2 – Roach

Úloha	Zodpovedná osoba	Ohodnotenie User Story
Vloženie novej otázky	Rastislav Dobšovič	8
Zobrazenie otázky	Dušan Poizl	8
Zobrazenie nových otázok	Peter Páleník	5

Tab. 3: Podiel práce na druhom šprinte

3.2.3 Šprint 3 – Hydralisk

Úloha	Zodpovedná osoba	Ohodnotenie User Story
Výber najlepšej odpovede	Jozef Harinek	3
Zobrazenie odpovedí pri otázkach	Dušan Poizl, Pavol Zbell	3
Vloženie novej odpovede	Marek Grznár	8
Zobrazenie odpovedaných otázok	Peter Páleník	1

Tab. 4: Podiel práce na treťom šprinte

3.3 Autori častí dokumentácie

Kapitola	Autori
Úvod	Peter Páleník
Ponuka	Všetci členovia tímu
Záznamy zo stretnutí	Všetci členovia tímu
Úlohy členov tímu	Peter Páleník
Manažment kvality	Marek Grznár
Manažment monitorovania	Dušan Poizl
Manažment rizík	Pavol Zbell
Manažment plánovania	Rastislav Dobšovič
Manažment podpory vývoja a integrácie	Samuel Molnár
Manažment komunikácie a ľudských zdrojov	Jozef Harinek
Manažment tvorby dokumentácie	Peter Páleník

Tab. 5: Autori častí dokumentácie

4 Manažment kvality

Cieľom manažéra kvality v našom tíme je dosiahnuť aby projekt (jeho výsledky) dosiahol istú úroveň kvality vo viacerých oblastiach. Manažér kvality okrem toho dohliada na čo najoptimálnejšie riešenie pracovných postupov, ktorých úlohou je zvyšovať kvalitu v určitých oblastiach práce tímu na projekte. Kontrola splnenia požiadaviek a zvyšovanie kvality zdrojového kódu sú základné oblasti ktoré pokrýva manažment kvality.

4.1 Zvyšovanie kvality zdrojového kódu

Zvyšovanie kvality zdrojového kódu nepredstavuje žiadnu priamu pridanú biznis hodnotu výsledného produktu. Hlavnou hodnotou kvalitného zdrojového kódu je eliminovanie nákladov vynaložených na prácu členov tímu v pokročilej fázy vývoja projektu.

Kvalita zdrojového kódu úzko súvisí aj s jeho prehľadnosťou. Preto na dosiahnutie prehľadnosti zdrojového kódu je nutnosť zdefinovať pravidlá pre písanie zdrojového kódu. Jednotný formát zdrojového kódu určený pomocou týchto pravidiel pomáha zvýšiť efektivitu práce keďže odpadá náročnosť študovania zdrojového kódu od iného člena tímu, ktorý používa výrazne iné konvencie pri písaní kódu.

4.1.1 Úvod k metodike zvyšovania kvality zdrojového kódu

Účelom tejto metodiky je opis a zdefinovanie základných postupov pre jednotnú tvorbu a komentovanie zdrojových kódov. Metodika sa zameriava na konvencie písania zdrojového kódu, ktoré sú jeho tvorcovia povinní dodržiavať. Uvedená metodika sa zameriava na programovací jazyk ruby s využitím rámca (angl. frameworku) Ruby on Rails¹.

Metodika je určená nielen pre programátorov, ale aj pre manažéra kvality, ktorého úlohou je dohliadať na to či tvorba zdrojového kódu prebieha podľa pravidiel stanovených metodikou.

4.1.2 Zdefinovanie pojmov

Pojem	Vysvetlenie
meno	Meno autora (značky, kódu)
správa	Vyjadruje bližší opis značky (odôvodnenie)
snake_case	Konvencia alebo tiež štýl písania viacslovných názvov, kde každé slovo začína malým písmenom a slová sú navzájom oddelené _ (podtržníkom)
CamelCase	Konvencia alebo tiež štýl písania viacslovných názvov, kde každé slovo začína veľkým písmenom a jednotlivé slová nie sú navzájom oddelené.

Tab. 6: Použité pojmy s vysvetlením

¹ Rails: <http://rubyonrails.org/>

4.1.3 Tvorba zdrojového kódu v Ruby on Rails

Používaným jazykom pri písaní kódu ako aj komentárov je výhradne angličtina. Žiadne iné jazyky nie sú povolené.

Komentovanie a značkovanie zdrojového kódu

Komentáre

- V zdrojovom kóde nie sú povolené žiadne komentáre okrem dokumentačných komentárov. Píšme kód tak, aby bol jasný na prvý pohľad.
- Každá zakomentovaná časť zdrojového kódu musí mať pri sebe značku (viď nižšie) s odôvodnením.

Značky

- Značka sa do kódu píše rovnako ako komentár v tomto tvare {značka} ({meno}) {správa}
- Správa má byť stručná, ale jasná.
- V správe k značke sa nahrádzajte:
 - zmazať kód (angl. remove code) skratkou rm
 - presunúť kód (angl. move code) skratkou mv
- Povolené značky v kóde sú uvedené v Tab 7.

Značka	Význam
TODO	Potrebné vyriešiť
FIX	Potrebné opraviť

Tab. 7: Povolené značky v kóde

- Príklad :

```
# TODO (grznar) rm
...
<!-- TODO (grznar) mv-->
...
```

Zdrojový kód v jazyku ruby

Formátovanie zdrojového kódu

- Použiť kódovanie UTF-8.
- Na konci riadkov vždy vložiť `\n` znak.
- Na začiatku riadka použiť odsadenie 2 medzery.
- Nikdy nepoužívať (bodkočiarku) ; .
- Vkladať medzeru okolo operandov a (znaku rovná sa) =
príklad:

```
x = 1 + 2
```

- Žiadne vnútorné medzery pri () a [] .
- Vložiť voľný riadok medzi definíciami `def`, `class`, `module` a pod.
- Nedávať biely znak medzi funkciu a zoznam argumentov.
- Nenechať biele znaky na konci riadkov.

Syntax zdrojového kódu

- Prísny zákaz používať `for`, `then`, `and`, `or`, `not`.
- Pri definícii metód používať () iba ak má metóda nejaké argumenty.
- Zátvorky () použiť iba ak sprehládnia kód alebo si to vyžaduje syntax jazyka.
- Preferovať ternárny operátor oproti `if/then/else/end`

príklad:

```
#zle
if a < b
  then
    a
  else
```

```
        b
    #dobre
    a < b ? a : b
```

- Vhodne používať if/unless

príklad:

```
#zle
if !vyraz
#dobre
unless vyraz
```

- Nepoužívať unless spolu s else, pozitívny prípad bude vždy ako prvý.
- Používať radšej loop s break ako begin/end/until resp. begin/end/while.
- Vždy používať {} pre:
 - jednoriadkové bloky,
 - bloky, na ktoré nadväzuje volanie.

príklad:

```
{ "jednoriadkovy blok" }
#blok, na ktory nadvazuje volanie
{ ... }.join
```

- Nepoužívať return a self pokiaľ to nie je nutné.
- Nepoužívať operátory === a -> .
- Nepoužívať \$ (globálne) premenné.
- Používať proc namiesto Proc.new.

- Používať _ pre nepotrebné premenné
- príklad:

```
x = hash.map { |_, v| v + 1 }
```

- Používať Array.join a nie *
- Použiť radšej (1000..2000).include?(x) alebo x.between?(1000, 2000) namiesto x >= 1000 && x <= 2000

- Používať predikáty namiesto ==
príklad:

```
#zle
x == nil

#dobre
x.nil?
```

- Vyhybať sa používaniu vnorených podmienok pri riadení toku programu

Pomenovanie v zdrojovom kóde

- Pomenovať veci presne, ideálne jednoslovné
- Nepoužívať skratky dlhšie ako jedno písmeno (jednopísmenové skratky ale používať opatrne, odporúča sa použiť ich iba v blokoch). Napríklad: žiadne fld, ale celým slovom field alebo keď už skratka tak potom f
- Povolené viacpísmenové skratky sú:
 - args
 - params

Štýl	Použitie pre
snake_case	Symboly
snake_case	Metódy
snake_case	Premenné
CamelCase	Triedy
CamelCase	Moduly

Tab. 8: Rozdelenie štýlov pomenovania pre jednotlivé konštrukcie jazyka

- Predikáty nemajú prefix is, ale sufix ?

príklad:

```
#zle
user.is_single

#dobre
user.single?
```

Narábanie s výnimkami v zdrojovom kóde

- Používať `fail` namiesto `raise`, používať `raise` iba v prípade znovuvyhodenia zachytenej výnimky
- Používať `fail 'správa'` namiesto `fail RuntimeError, 'správa'`.
- Používať `fail NejakáVýnimka, 'správa'` namiesto `fail NejakáVýnimka.new('správa')`.
- Nepoužívať `return` v `ensure` bloku.
- Používať `def/rescue/end` namiesto `def/begin/rescue/end/end`.
- Nikdy nepotláčať výnimky.
- Zákaz používať výnimky pre riadenie toku programu.

Práca s množinami v zdrojovom kóde

- Používať `[]` a `{}` na vytvorenie poľa, resp. hash poľa namiesto `Array.new` a `Hash.new`.
- Používať priamo polia namiesto `%w()`, `%i()` a `pod..`
- Používať `first`, `second`, ..., `last` namiesto `[0]`, `[1]`, ..., `[-1]`.
- Používať `Set` namiesto `Array`, ak je to vhodné.
- Používať ako kľúče symboly namiesto reťazcov.
- Vždy pokiaľ je možné, použiť `{ one: 1 }` namiesto `{ :one => 1 }`.

Práca s reťazcami v zdrojovom kóde

- Používať `' '` pre reťazce namiesto `" "`, ak ide o reťazce, ktoré predstavujú jazykové preklady používať `" "`.
- Pre spájanie reťazcov preferujte `<<` miesto `+`.

4.2 Kontrola splnenia požiadaviek

V zmysle agilného vývoja kontrola splnenia požiadaviek prebieha už počas vývoja funkcionality. Teda samotné testy sú vytvorené ešte pred samotnou funkcionalitou. Každý člen tímu testuje časť, na ktorej pracuje.

4.2.1 Úvod k metodikám pre kontrolu splnenia požiadaviek

Účelom tejto metodiky je opis a zadefinovanie základných postupov pri testovaní softvéru. Metodika sa sústreďuje prevažne na viaceré typy testov od jednotkových (angl. Unit) až po akceptačné testy a ich využitie pre testovanie webových aplikácií. V predkladanej metodike sa na písanie testov využíva testovací rámec *Rspec*² v spolupráci s jeho nadstavbou *Capybara*³.

Metodika je inšpirovaná zaužívanými konvenciami pre formátovanie a písanie testov v *Rspec*. Je prevažne určená pre tímy využívajúce *Rspec* v spolupráci s rámcom pre vývoj webu *Ruby on Rails*⁴. Logická organizácia kapitol metodiky je inšpirovaná preferovaným postupom pri definovaní testov od akceptačných až po integračné a jednotkové testy.

4.2.2 Použité skratky a značky

DSL	Domain Specific Language <ul style="list-style-type: none">• doménovo špecifický jazyk
TDD	Test Driven Development <ul style="list-style-type: none">• prístup k tvorbe softvéru preferujúci písanie testov ako prvý krok implementácie softvéru
Rspec	Testovací rámec v jazyku Ruby, ktorý definuje intuitívne DSL pre tvorbu testov
Capybara	Nadstavba nad Rspec pre testovanie webových stránok
Example	Blok zdrojového kódu v Rspec, ktorý obsahuje samotný test
Feature	Funkcionalita, ktorá podporuje používateľský príbeh
Mock	Simulovaný objekt, ktorý imituje správanie reálneho objektu

4.2.1 Postupy

Nastavenie testovacieho prostredia pre Rspec

Nainštalujeme potrebné knižnice pomocou príkazu `bundle install`. Pre nastavenie testovacej databázy spustíme príkaz `rake db:test:load`, ktorý načíta aktuálnu schému databázy do testovacej databázy. Existujúce testy spustíme v rámci prostredia dekorovaného knižnicou *Bundler*⁵ pomocou príkazu `bundle exec rspec`.

2 Rspec: <https://github.com/rspec>

3 Capybara: <https://github.com/jnicklas/capybara>

4 Rails: <http://rubyonrails.org/>

5 Bundler: <https://github.com/bundler>

Prípava akceptačných testov pre scenáre používateľského príbehu

Všetky akceptačné testy pre používateľský príbeh (angl. User Story) sa umiestnené v adresári `spec/features`. Konkrétne testy pre jeden používateľský príbeh sú umiestnené v jednom súbore v adresári `features`.

Postup:

1. Vytvoriť súbor pre test používateľského príbehu alebo jeho feature.

Formát:

- `<názov feature>_spec.rb`

Príklad:

- `authentication_spec.rb`
- `change_password_spec.rb`

2. Do súboru `<názov feature>_spec.rb` pridať základnú definíciu testu pomocou kľúčového slova `describe`.
3. Textový opis pre `describe` zvoliť na základe názvu používateľského príbehu alebo jeho feature.

Formát:

- `describe '<Názov Feature>'`

Príklad:

- `describe 'Authentication'`
- `describe 'Change Password'`

Vytvorenie opisu akceptačného testu pre scenár používateľského príbehu

Kontext používateľa pri realizácii používateľského príbehu reprezentujeme v *Rspec* pomocou kľúčového slova `context`. Opis kontextu vyjadrujeme podmienčne pomocou spojky *when* alebo vyjadrením vstupu pomocou predložky *with*. Cieľ testu je reprezentovaný kľúčovým slovom `it` a opis cieľu je vyjadrený vždy v tretej osobe jednotného čísla.

Postup:

1. Pridať do bloku ohraničenom `describe` kontext používateľa pomocou kľúčového slova `context`.

Formát:

- `context 'when/with ...'`

Príklad:

- `context 'when logged in'`
- `context 'with service credentials'`

2. Do bloku `context` pridať cieľ testu pomocou kľúčového slova `it`.

Formát:

- `it '<sloveso v tretej osobe jednotného čísla> ...'`

Príklad:

- `it 'allows user to edit profile'`
- `it 'signs user up'`

V prípade akceptačných testov je `context` vždy vnorený maximálne dvakrát.

Vytvorenie akceptačného testu pre scenár používateľského príbehu

Pre interakciu s webovou stránkou používame rozhranie nástroja *Capybara*.

Pravidlá pre použitie rozhrania *Capybara*

- Pre navigáciu na konkrétne URL adresy vždy použiť metódu `visit` s URL adresou ako parametrom.

Príklad:

- `visit new_user_registration_path`
- Na simulovanie klikania tlačidiel a hyperliniek použiť metódy `click_button` a `click_link`. Jednotlivé elementy pre uvedené metódy identifikovať vždy len na základe textu, ktorý obsahujú.

- Na vyplňanie polí vo formulároch použiť vždy metódu `fill_in`. Ako identifikátor formulárového poľa je vždy použitá značka (angl. label) daného poľa.

Príklad:

- `fill_in 'user_name', with: 'Peter'`

- Obsah webovej stránky testovať vždy pomocou metódy `have_content`. Testovať vždy len textový obsah stránky.

Príklad:

- `expect(page).to have_content('Úspešne prihlásený.')`

- Úspešné nastavenie atribútov v odoslanom formulári testovať vždy len na vyplnených poliach znovu načítaného formulára pomocou metódy `have_field` alebo pomocou `have_content` na obsahu presmerovanej stránky.

Príklad:

- `expect(page).to have_field('user_name', with: 'Peter')`
- `expect(page).to have_content('Peter')`

Vytvorenie inštancií modelov pri testovaní

Pre tvorbu inštancií v modeloch používame knižnicu *FactoryGirl*⁶. Všetky definície inštancií pre *FactoryGirl* sú umiestnené v adresari `spec/factories`. Definície pre model sú uvedené individuálne v súbore s názvom modelu v množnom čísle.

Postup:

1. Vytvoriť súbor pre *FactoryGirl* definície pre daný model.
2. Vytvoriť definíciu pre model.
 - a) Nastaviť v definícii základné parametre pre inštanciu modelu.
 - b) Ak sa inštancia môže nachádzať vo viacerých stavoch, t.j. vo viacerých kombináciách jej atribútov, definovať viaceré kombinácie pomocou kľúčového slova `trait`.
 - c) Asociácie na iné modely vždy vyjadriť pomocou kľúčového slova `association`, pokiaľ sa jedná o `belongs_to` asociáciu.
 - d) Unikátne atribúty, ktoré musia byť odlišné pre všetky vytvorené inštalácie, nastaviť vždy

6 *FactoryGirl*: https://github.com/thoughtbot/factory_girl

pomocou kľúčového slova `sequence`.

e) Príklad:

```
factory :user do
  sequence(:login) { |n| "user_#{n}" }
  sequence(:email) { |n| "user_#{n}@example.com" }

  password          'password'
  password_confirmation 'password'

  association :school

  trait :as_ais do
    sequence(:login) { |n| "xuser_#{n}" }
    sequence(:ais_login) { |n| "xuser_#{n}" }
    sequence(:ais_uid) { |n| n }

    password          nil
    password_confirmation nil
  end
end
```

3. V testoch vytvoriť inštanciu modelu vždy pomocou metódy `create`. Iné hodnoty atribútov je potrebné špecifikovať pomocou asociatívneho poľa pre metódu `create`.

Formát: `create :<názov modelu>`

Príklad:

- `create :user`
- `create :user, login: 'samuel', password: 'password'`

Príprava testov pre model

Všetky testy pre modely sa nachádzajú v adresari `spec/models`. Testy pre model sú umiestnené v súbore s názvom modelu v jednotnom čísle.

Postup:

1. Vytvoriť súbor pre testy modelu.

Formát:

- `<názov modelu>_spec.rb`

Príklad:

- `user_spec.rb`

2. Do súboru `<názov modelu>_spec.rb` pridať základnú definíciu testu pomocou kľúčového slova `describe` a ako opis uviesť triedu modelu.

Príklad: `describe User`

Vytvorenie jednotkového testu pre validáciu modelu

Všetky testy pre validácie modelu sa nachádzajú na najvyššej úrovni v `describe` bloku daného modelu. Opis bloku `it` vždy obsahuje ako prvé slovo *requires* alebo *validates*. Ak validácia kontroluje viaceré typy hodnôt pre daný atribút, kontext testov pre danú validáciu odlišíme pomocou kľúčového slova `context`.

Príklad:

- `it 'requires password'`
- `context 'with AIS credentials' do`
`it 'does not require password'`

Vytvorenie jednotkového testu pre povolenia inštancie modelu

Všetky testy pre povolenia inštancie modelu sú umiestnené v `describe` bloku s názvom *Abilities* na najvyššej úrovni `describe` bloku daného modelu. Každý opis bloku `it` začína vždy slovom *allows* alebo *disallows*. Odlíšenie kontextu pre testy jedného povolenia realizujeme kľúčovým slovom `context`.

Príklad:

- `it 'disallows changing of name'`

Vytvorenie jednotkového testu pre metódu modelu

Všetky testy pre metódy modelu sa nachádzajú na najvyššej úrovni v `describe` bloku daného modelu. Test pre metódu vždy definujeme pomocou `describe` bloku. Opis `describe` bloku pre metódu vždy obsahuje len názov metódy prefixovaný znakom „#“ v prípade inštančnej metódy a znakom „.“ v prípade metódy na úrovni triedy.

Postup:

- Vytvoriť `describe` blok pre metódu.

Príklad:

- `describe '.find_by'`
- `describe '#activate'`
- Rozdeliť rôzne kontexty alebo typy parametrov metódy pomocou kľúčového slova `context`.
- Pridať opis kontextu v rovnakom formáte ako pri opise akceptačného testu používateľského príbehu.
- Vytvoriť `it` bloky s cieľom testu.

Vytvorenie integračného testu pre servisný objekt

Všetky testy pre servisné objekty sa nachádzajú v adresári `spec/services`. Testy pre metódy používajú rovnakú metodiku testovania ako v predchádzajúcej kapitole. Všetky objekty, ktoré spolu interagujú v rámci servisného objektu, majú imitované správanie pomocou syntaxe knižnice *Rspec-mocks*⁷. Mock objektu sa vždy vytvára kľúčovým slovom `double`. Metódy bez parametrov s konštantou návratovou hodnotou vždy uvedieme ako asociatívne pole pre kľúčové slovo `double`.

Postup:

- Vytvoriť `describe` blok pre metódu.
- Vytvoriť `mock` pre všetky objekty interagujúce v rámci metódy servisného objektu.

Príklad:

- `service = double(:service)`
- `user = double(:user, name: 'Peter')`
- Definovať vzájomné správanie objektov.

Príklad:

Nad objektom `service` očakávame zavolať metódu `authenticate!` s parametrami `login` a `password` a výsledok operácie tejto metódy očakávame pre metódu `update_attributes` objektu `user`.

⁷ Rspec-mocks: <https://github.com/rspec/rspec-mocks>

- `expect(service).to receive(:authenticate!).with('login', 'password').and_return(data)`
- `expect(user).to receive(:update_attributes).with(data)`
- Priradiť jednotlivé *mock* inštancie servisnému objektu.
- Zavolať testovanú metódu.

5 Manažment monitorovania

Úlohou manažéra monitorovania v našom tíme je sledovať ako postupujú práce na projekte. Sleduje ako sú zapisované hodiny v *Redmine* a progres jednotlivých úloh. Ak napríklad je zmenený progres bez toho aby boli zapísané hodiny upozorní na to daného člena tímu aby to doplnil. Porovnáva stav úloh s plánom a nakoľko sa projekt odchyľuje od naplánovanému postupu. Oneskorenia konzultuje s manažérom rizík. Sleduje mieru efektívnosti ohodnotenia jednotlivých úloh. Na to mu slúži graf vyhorenia a graf deviácie reportovaných hodín. Kontroluje taktiež či majú úlohy nastavený správny stav. Vyhodnocuje nakoľko bol tím efektívny v práci na projekte. Toto je podkladom pre plánovanie ďalších šprintov.

6 Manažment rizík

Manažérovi rizík v tíme pripadá úloha identifikácie, klasifikácie, navrhovania preventívnych a reaktívnych opatrení rizík, ktoré môžu ohroziť fungovanie tímu alebo samotný projekt. Manažér rizík podľa potreby (faktorov rizika, ako napr. pravdepodobnosti výskytu, odhadu rozsahu škôd) pravidelne a včas informuje ostatné zainteresované manažmenty.

Manažér rizík v tíme zastáva aj úlohu zástupcu vedúceho. Z tejto pozície pomáha vedúcemu tímu v jeho úlohách a pravidelne ho informuje o možných rizikách, ktoré môžu ohroziť splnenie termínov (v tomto prípade informuje aj manažéra plánovania) a radí ako im predísť. Spolu s vedúcim tímu dohliada na efektívnosť tímu pri riešení úloh a korektnosť medzil'udských aj pracovných vzťahov.

Pozícia hlavného architekta v tíme pripadá manažérovi rizík. V tomto prípade už pri architektonickom návrhu manažér rizík zvažuje dopady použitých technológií a riziká réžie vzhľadom na ich údržbu v rámci projektu. Z pohľadu hlavného architekta navrhuje, konzultuje a schvaľuje infraštruktúru informačného systému, použité technológie, knižnice a implementačné postupy (súvisiace s architektúrou, napr. rozhrania vlastných knižníc). Má na starosti databázovú schému na úrovni logického aj fyzického modelu. Konzultuje a schvaľuje všetky zmeny a úpravy v systéme na úrovni databázy.

6.1 Identifikované riziká

V rámci projektu boli identifikované a analyzované viaceré riziká, ktoré sú prehľadne zhrnuté v tabuľke Tab. 9. Klasifikácia alebo typ rizika nemusí byť nutne presná, keďže sa na niektoré riziká

dá pozrieť z viacerých uhlov pohľadu. Podobne rozsahy škôd môžu závisieť od viacerých faktorov. V prípade opatrení je uvedená pomlčka ak sa opatrenie nepodarilo identifikovať alebo je príliš všeobecné (napr. „Dôkladnejšie plánovať.“) alebo ak až príliš závisí od faktorov rizika a vyžaduje si hlbšiu analýzu, ktorá je nad rámec tohto dokumentu.

#	Riziko	Klasifikácia a typ	$P(r)$	Rozsah škôd	Spôsob identifikácie	Preventívne opatrenia	Reaktívne opatrenia
1	Zabezpečenia kvality a udržateľnosti softvéru	nákladov, interné, generické	stredná	stredný	Nestabilné distribúcie softvéru. Ťažkosti pri pridávaní novej funkcionality.	Dôkladne kontrolovať architektonické návrhy a rozhodnutia. Monitorovať kvalitu softvéru.	Zintenzívniť preventívne opatrenia.
2	Vytvorí sa iná funkcionality ako bola požadovaná zákazníkom.	splnenia požiadaviek, interné, generické	nízka	stredný	Zákazník nebude spokojný.	Dostatočne diskutovať funkcionality so zákazníkom.	Implementovať funkcionality znovu podľa požiadaviek v čo najkratšom čase.
3	Evidencia nekorektných vstupov od používateľov.	splnenia požiadaviek, externé, generické	stredná	stredný	Všimneme si nekorektné vstupy od používateľov.	-	Vytvoriť pozíciu pre správcu obsahu od používateľov.
4	Nepodarí sa získať dostatočnú základnú bázu používateľov.	nákladov, externé, špecifické	stredná	vysoký	Počet používateľov nebude korešpondovať s plánovaným počtom pre rozumné fungovanie projektu.	Dostatočne propagovať projekt v rámci fakulty, na sociálnych sieťach a iných médiách.	Prehodnotiť propagáciu a (podľa potreby opakovať) kroky uvedené v preventívnych opatreniach.
5	Používatelia nebudú vedieť pracovať s rozhraniami aplikácie (napr. rozhraním vyhľadávania).	splnenia požiadaviek, externé, špecifické	nižšia	stredný	Používatelia poskytnú negatívnu spätnú väzbu. Z logov sa ukáže, že používatelia danú funkcionality nepoužívajú.	Dôkladne realizovať UX stránku veci. Ponúknuť používateľom prezentáciu používania danej funkcionality.	Vyhľadať experta na UX, refaktorovať používateľské rozhrania a ponúknuť prezentáciu používania funkcionality.

6	Nastane nedostatok personálu pracujúcim na projekte, prípadne z projektu odídu dôležití ľudia.	splnenia požiadaviek, interné, generické	nízka	stredný	Nedodržia sa termíny závislé na pracovnej sile.	-	-
7	Aplikácia sa stane nedostupnou z dôvodu preťaženia serverov alebo inej internej chyby.	splnenia požiadaviek, interné, generické	vyššia	vysoký	Služba na monitorovanie chýb ohlásí kritický stav alebo sa pri načítaní stránky v prehliadači zobrazí chyba.	Dôkladne otestovať zaťaženie aplikačného serveru a používať TDD.	Podrobne vyšetriť príčinu nedostupnosti, refaktorovať a doplniť testy, prípadne začať škálovať jednotlivé komponenty.
8	Zníži sa zainteresovanosť a prehľad členov o dianí v projekte.	splnenia požiadaviek, interné, generické	vyššia	stredný	Zníži sa komunikácia v tíme, nestihnú sa úlohy z predchádzajúceho šprintu.	Individuálne dohliadnuť na plánovanie času jednotlivých členov tímu.	Analyzovať čas venovaný projektu, naplánovať menšie úlohy.

Tab. 9: Opis rizík

7 Manažment plánovania

7.1 Popis úlohy manažéra plánovania

Úloha manažéra plánovania v našom tíme pozostáva z niekoľkých povinností. Jeho prvoradou zodpovednosťou je vytváranie krátkodobého plánu na najbližší šprint, ktorý vzniká počas *scrum* stretnutia tímu. Na vzniku tohto plánu sa podieľa celý tím, avšak zodpovednosť nesie práve manažér plánovania (so súčinnosťou s manažérom rizík). Takisto vytvára odhad časov pre jednotlivé úlohy, pričom tieto časy konzultuje so zadávateľom a riešiteľom úlohy.

Po stretnutí je povinnosťou manažéra plánovania vytvoriť všetky identifikované úlohy aj v nástroji *Redmine*. Snaží sa dohliadať, aby členovia tímu pristupovali zodpovedne k práci v tomto nástroji a pravidelne aktualizovali status svojej činnosti. Pre konzistenciu a efektivitu práce v nástroji *Redmine* tiež vytvára a udržiava metodiky.

7.2 Metodika pre vkladanie novej úlohy v nástroji *Redmine*

Táto časť popisuje jednotlivé postupy pri vytváraní úloh v nástroji *Redmine* podľa ich typu.

7.2.1 Všeobecné pokyny pre vytváranie udalostí

Pri vytváraní udalostí sa riadime typom vytvárajúcej udalosti, avšak niektoré postupy platia pre všetky typy, alebo ich treba dopredu zdefinovať.

Tracker je vždy povinná položka. Môže nadobudnúť tieto hodnoty:

- *User Story*
 - platí pre všetky definované používateľské príbehy,
- *Task*
 - zadáva sa pre úlohy, ktoré vznikli delením používateľských príbehov,
- *Bug*
 - zadáva sa, pokiaľ treba opraviť menšiu chybu alebo vykonať úpravu v systéme,
- *Support*
 - platí pre úlohy, ktoré nemajú priamo implementačný charakter.

Pre *Subject* a *Description* sa viažu nasledujúce konvencie:

- písané v slovenčine,
- malými písmenami,
- bez diakritiky,
- skratky písať korektne (nie "fiit", ani "Fiit"; správne "FIIT"),
- dátumy písať vo formáte RRRRMMDD (napr. "stretnutie 20131113").

Status je vždy povinná položka a nastavuje sa na *new*.

Priority sa nastavuje na *normal*, pokiaľ manažér plánovania alebo vedúci tímu neurčí inak.

Assignee je povinná položka (pokiaľ si to typ udalosti nevyžaduje inak).

Target version sa určuje len v úlohách explicitne zviazaných s konkrétnym behom.

Due date je povinná položka (pokiaľ si to typ udalosti nevyžaduje inak).

Estimated time je povinná položka, ktorú je vždy potrebné konzultovať so zadávateľom a riešiteľom úlohy.

7.2.2 Plánovanie príbehov používateľa

Pre plánovaní šprintu sú jednotlivé príbehy používateľa rozdrobené na menšie úlohy – v nástroji *Redmine* dokumentujeme obe tieto úrovne.

Pre príbehy používateľa platí:

- ako *Tracker* je použité pole *User Story*,
- *Target version* je povinná položka,
- *Due date* sa neuvádza,
- *Estimated time* sa neuvádza.

Pre úlohy, ktoré prislúchajú konkrétnemu príbehu používateľa platí:

- ako *Tracker* je použité pole *Task*,
- *Target version* je povinná položka,
- pole *Parent task* je vždy povinná položka (nastavená ako *ID* príbehu používateľa, ktorej vytváraná úloha prislúcha),

- ak manažér plánovania neurčí inak, *Due date* sa nastaví na deň, kedy končí šprint, s ktorým je vytváraná úloha viazaná.

7.2.3 Udalosti pre stretnutia tímov

Po spoločnom stretnutí tímu je vždy vytvorená nová úloha, pre ktorú platí:

- ako *Tracker* je použité pole *Support*,
- *Subject* je v tvare "stretnutie RRRRMMDD" (napr. "stretnutie 20131113"),
- *Description* obsahuje informáciu o
 - dĺžke stretnutia,
 - účele stretnutia (pokiaľ nešlo o pravidelný *scrum*),
- *Due date* sa neuvádza,
- *Estimated time* sa neuvádza.

8 Manažment podpory vývoja a integrácie

Úlohou manažéra podpory vývoja a integrácie je príprava a údržba podporných prostriedkov pre projektový manažment a vývoj softvéru. V kontexte nášho tímu člen zodpovedný za uvedenú rolu spravuje a nastavuje systém pre projektový manažment *Redmine*⁸, webový server pre výsledný produkt a repozitáre uchovávané v službe *Github*⁹. Manažér má za úlohu sledovať dodržiavanie metodík pre prácu so systémom pre správu verzií zdrojového kódu *Git* a dodržiavanie metodík pre plánovanie v systéme *Redmine*. Manažér má taktiež na starosti bezchybný chod uvedených systémov a ich integráciu. Ďalšou dôležitou úlohou manažéra je nastavenie prostredia pre vývoj a testovanie produktu. Preto manažér udržiava verzie knižníc, kompatibilitu ich verzií a ich nastavenia pre rôzne prostredia aplikácie. Manažér je zodpovedný za nasadenie (angl. *deploy*) a bezchybný chod aplikácie na produkčnom serveri.

8 Redmine: <http://www.redmine.org/>

9 Github: <https://github.com/>

8.1 Úvod k metodikám pre podporu vývoja a integrácie

Cieľom tejto metodiky je definovanie základných pojmov a postupov pri manažovaní zdrojového kódu a verzií projektu. Metodika presne definuje prácu s nástrojmi na správu revízií zdrojového kódu, konvencie pre odovzdávanie a vetvenie zdrojového kódu a riešenie konfliktov. Metodika opisuje prácu s nástrojom Git¹⁰ a službou GitHub¹¹.

Metodika vychádza z klasických prístupov k manažmentu zdrojového kódu rovnako ako aj z prístupov používaných pri populárnych projektoch s otvoreným zdrojovým kódom. Jednotlivé postupy sú uvedené v chronologickom poradí v akom ich programátor pri vývoji spravidla použije, t.j. od inicializácie repozitára, cez odovzdávanie zdrojového kódu a pridávanie funkcionality, až po opravu chýb v softvéri.

8.2 Použité skratky a značky

RCS	System na správu revízií (angl. revision control system) zdrojového kódu a iných dokumentov a súborov projektu. ¹²
ITS	System na sledovanie úloh (angl. issue tracking system) v rámci projektu.
Git	Distribučovaný systém na správu revízií zdrojového kódu.
GitHub	Služba a sociálna sieť na zdieľanie a vývoj zdrojového kódu využívajúca Git.
Repozitár	Repozitár (angl. repository). Úložisko zdrojového kódu a iných dokumentov a súborov projektu v rámci systému na správu revízií. Rozlišujeme vzdialené (na serveri) a lokálne repozitáre.
Odobzdanie	Odobzdanie (angl. commit) je množina zmien v súboroch projektu v určitom čase podpísaná autorom.
Vetva	Vetva (angl. branch) je postupnosť odobzdaní. Vetvenie umožňuje spravovať súbory v jednej vetve repozitáru nezávisle od iných vetiev.
Feature	Funkcionalita, ktorá je podporená používateľským príbehom.

8.3 Postupy

8.3.1 Nastavenie nástroja Git a prepojenia so službou GitHub

Predpokladá sa nainštalovaný nástroj Git¹³ a vytvorený účet v sieti Github¹⁴. Používateľské meno v sieti GitHub sa odporúča zadať v tvare „[iniciála-mena][priezvisko]“ alebo „[meno][priezvisko]“ (s

10 Git: <http://git-scm.com>

11 GitHub: <http://github.com>

12 RCS – systém na správu revízií je tiež známy ako VCS – systém na správu verzií (angl. version control system) alebo SCMS – systém na manažment zdrojového kódu (angl. source code management system).

13 Inštalácia nástroja Git: <http://git-scm.com/book/en/Getting-Started-Installing-Git>

14 Vytvorenie účtu na sieti GitHub: <https://github.com/join>

malými písmenami), napr. „pzbell“ alebo „pavolzbell“.

Postup:

1. Nastaviť používateľské meno a e-mail v nástroji Git¹⁵. Používateľské meno musí byť v tvare „[Meno] [Priezvisko]“ (s diakritikou), napr. „Samuel Molnár“ alebo „Pavol Zbell“.

```
git config --global user.name "username"  
git config --global user.email "user@example.com"
```

2. Nastaviť rovnaký primárny e-mail účtu v sieti GitHub ako bol nastavený v nástroji Git.
3. Vygenerovať SSH kľúč a priradiť ho k účtu v sieti GitHub¹⁶.

```
ssh-keygen -t rsa -C "user@example.com"
```

Otvoriť vygenerovaný súbor `id_rsa.pub` a skopírovať obsah medzi kľúče SSH účtu v sieti GitHub a potvrdiť prídanie kľúča.

4. Pridať fotku k účtu v sieti GitHub pomocou služby Gravatar¹⁷.
5. Požiadat' správcu organizácie v sieti GitHub o členstvo (platí iba v prípade ak má tím vytvorenú organizáciu).
6. Informovať správcu vzdialených repozitárov o úspešnom nastavení a požiadať o pridelenie práv na odovzdávanie zdrojového kódu.

8.3.2 Inicializácia vzdialeného repozitára

Najskôr sa vytvorí nový lokálny repozitár a následne sa vykoná prvé odovzdanie do vzdialeného repozitára v rámci služby GitHub. Názov repozitára musí byť malými písmenami a ako oddeľovač sa smie použiť iba krátka pomlčka, napr. „project-site“.

Postup:

1. Vytvoriť vzdialený repozitár projektu službou GitHub¹⁸. Pri tvorbe repozitáru cez rozhranie služby nepridať automaticky žiadne súbory ani nevykonať žiadne odovzdanie. Výstupom tohto procesu sú aktívne HTTPS a SSH adresy vzdialeného repozitáru, napr.:

```
https://github.com/username/project.git  
git@github.com:username/project.git
```

2. Vytvoriť lokálny repozitár projektu v novom adresári:

```
mkdir project
```

15 Nastavenie používateľského mena a e-mailu: <https://help.github.com/articles/set-up-git#platform-all>

16 Vygenerovanie a priradenie SSH kľúča: <https://help.github.com/articles/generating-ssh-keys#platform-all>

17 Gravatar: <http://gravatar.com>

18 Vytvorenie vzdialeného repozitáru službou GitHub: <https://github.com/new>


```
cd project
git init
```

3. Vytvoriť súbor `.gitignore`¹⁹. Obsah súboru musí aspoň základne pokrývať potreby projektu vzhľadom na použitý programovací jazyk, knižnice alebo aplikačné rámce.
4. Vytvoriť súbor `README.md`, ktorý obsahuje text `TODO`.
5. Vytvoriť súbor `LICENSE.md`, ktorý obsahuje text `TODO`.
6. Vykonať prvé odovzdanie so správou „Initial commit“:

```
git add .
git commit -m "Initial commit"
```

7. Pridať referenciu na vzdialený repozitár projektu.

```
git remote add origin git@github.com:username/project.git
```

8. Odovzdať vykonané zmeny do vzdialeného repozitára projektu.

```
git push origin master
```

9. V sieti GitHub prideliť práva členom tímu na odovzdávanie zdrojového kódu a podľa potreby informovať členov tímu o vzniku repozitára.

8.3.3 Inicializácia lokálneho repozitára

Inicializácia lokálneho repozitára na základe existujúceho vzdialeného repozitára sa vykonáva príkazom:

```
git clone git@github.com:username/project.git
```

8.3.4 Pravidlá pre odovzdávanie zdrojového kódu

Je žiadúce aby odovzdania mali jemnú granularitu zmien a zároveň aby zmeny v odovzdaní medzi sebou logicky súviseli.

Pre správu odovzdania (angl. commit message) platí:

- píše sa v angličtine,
- prvé písmeno je vždy veľké,
- na konci správy nie je bodka,
- na delenie správy na menšie časti sa používa čiarka (správa nesmie obsahovať žiadnu inú

19 Súbor `.gitignore`: <http://git-scm.com/docs/gitignore>

interpunkciu),

- má maximálne 50 znakov,
- skratky sa píše korektne (nepísať „xml“, ani „Xml“ ale správne „XML“).

V kontexte jazyka Ruby pre správu odovzdania ďalej platí:

- názvy modulov a tried sa píše iba v tvare „XmlParser“ alebo „XML Parser“,
- neuvádza sa menný priestor modulov alebo tried (nepísať „Core::XmlParser“).

V kontexte ITS pre správu odovzdania ďalej platí:

- má tvar „[akcia-nad-stavom-úlohy] #[identifikátor-úlohy] [zvyšok-správy]“,
- akcia nad stavom úlohy v ITS a je buď Fix, Implement, Resolve alebo Close,
- referencia na úlohu v ITS sa uvádza v tvare „#[identifikátor-úlohy]“.

Príklady správ odovzdania:

```
Disable user registration via email
Refactor user model, fix user helpers
Add XmlParser, XmlNode and XmlHelpers
Update XML library, add XML Parser
Fix #31 correct URL to documents
```

8.3.5 Pravidlá pre vetvenie zdrojového kódu

Pre názov vetvy platí:

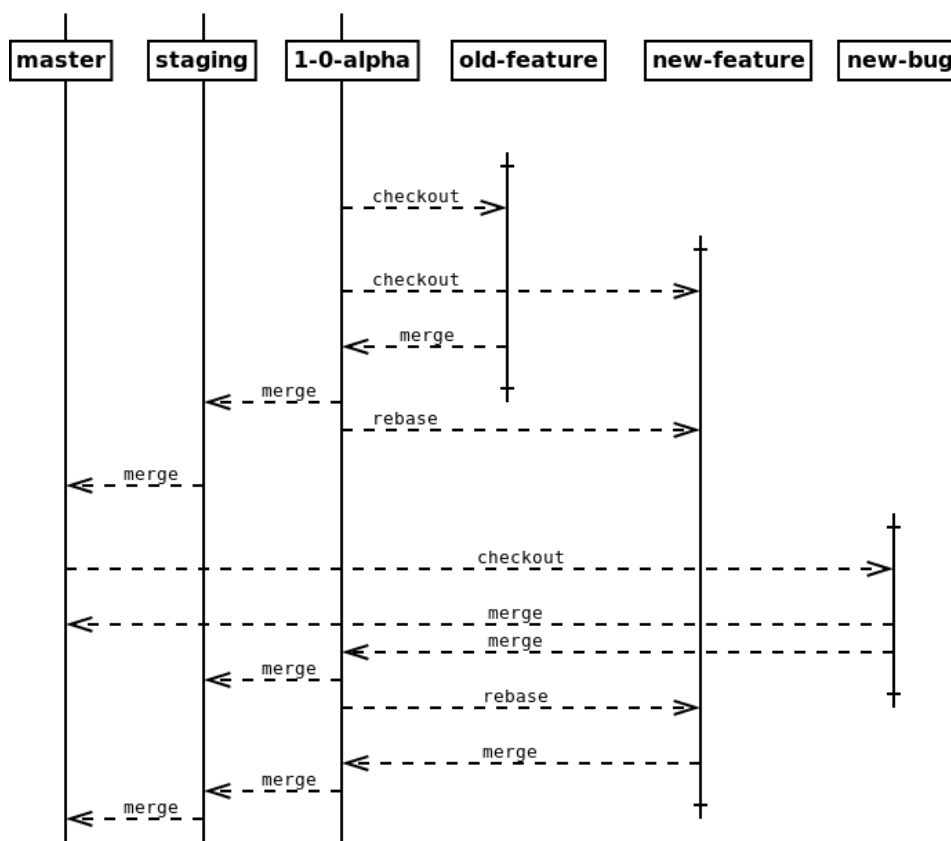
- píše sa v angličtine,
- obsahuje iba malé písmená,
- používa krátku pomlčku ako oddeľovač slov (neobsahuje žiadnu inú interpunkciu),
- je stručný a presný.

V rámci hlavného repozitára projektu, v ktorom je aplikácia dodávaná zákazníčkovi. Rozlišujeme tieto vetvy:

- Produkčná vetva (angl. production branch), je distribuovaná zákazníčkovi a obsahuje funkcionality schválenú zákazníčkovi. Názov vetvy je „master“.
- Testovacia vetva (angl. staging branch), obsahuje novú a otestovanú funkcionality pripravenú na schválenie zákazníčkovi. Vetva sa spája s produkčnou vetvou a do nej sa spája

aktuálna vývojová vetva. Názov vetvy je „staging“.

- Vývojové vetvy (angl. development branch), reflektujú aktuálny stav jednotlivých verzií projektu. Do týchto vetiev sa spájajú vetvy, ktoré implementujú funkcionality a opravujú chyby. Názvy vetiev majú tvar „[major]-[minor]-[qualifier]“, kde jednotlivé premenné vychádzajú z označenia vyvíjanej verzie, napr. „1-0-alpha“.
- Vetvy pre novú funkcionality (angl. feature branch), v každej sa implementuje práve jedna nová funkcionality podporená používateľským príbehom. Názvy vetiev majú sufix „-feature“, napr. „user-authorization-feature“.
- Vetvy pre opravy chýb (angl. bug fix branch), v každej sa implementuje oprava práve jednej chyby nahlásenej v systéme ITS. Názvy vetiev majú sufix „-bug“, napr. „login-bypass-bug“.
- Ostatné vetvy sú spravidla tie, v ktorých prebieha migrácia na nové verzie použitých technológií alebo refaktorovanie zdrojového kódu.



Obr. 1: Vývoj vetiev repozitáru v čase.

Na diagrame vyobrazenom na Obr. 1. je načrtnutý vývoj vetiev repozitáru s projektom v čase.

8.3.6 Pravidlá pre riešenie konfliktov

Ak nastane konflikt v paralelne editovaných súboroch pri spájaní vetiev, tak platí, že:

- pred nasledujúcim odovzdaním dotknutých súborov musia byť všetky konflikty vyriešené,
- sa riešením konfliktov nesmie narušiť existujúca funkcionálnosť softvéru.

8.3.7 Pridanie novej funkcionality podporenej používateľským príbehom

Pridanie novej funkcionality predpokladá existenciu zodpovedajúcej úlohy v ITS.

Postup:

1. Vytvoriť novú lokálnu vetvu s funkcionálnosťou.

```
git checkout -b new-feature
```

2. Ak už bola vetva s funkcionálnosťou nahraná do vzdialeného repozitára je potrebné vykonať aktualizovanie lokálneho zdrojového kódu:

```
git pull --rebase origin new-feature
```

3. Ak pribudli odovzдания do príslušnej vývojovej vetvy je potrebné aktualizovať vetvu s funkcionálnosťou o tieto odovzдания:

```
git rebase 1-0-alpha
```

4. Implementovať novú funkcionálnosť, po častiach lokálne odovzdávať zdrojový kód a podľa potreby sa vracáť ku krokom číslo 2 a 3.

5. Lokálne skontrolovať odovzdaný zdrojový kód, skontrolovať a spustiť testy.

6. Nahrať zdrojový kód vetvy s funkcionálnosťou do vzdialeného repozitára.

```
git push origin new-feature
```

7. Požiadať poverenú osobu o prehliadku zdrojového kódu.

- Ak poverená osoba implementáciu akceptuje, tak sa pokračuje krokom číslo 8,
- inak poverená osoba určí čo treba opraviť a pokračuje sa krokom číslo 2.

8. Spojiť vetvu s príslušnou vývojovou vetvou.

```
git checkout 1-0-alpha  
git merge new-feature
```

9. Vyriešiť konflikty, ak nastali.
10. Nahrať zdrojový kód príslušnej vývojovej vetvy do vzdialeného repozitára.

```
git push origin 1-0-alpha
```

11. Požiadat' poverenú osobu o spojenie vetvy s testovacou vetvou.

8.3.8 Riešenie kritických úloh pre produkčnú vetvu

Riešenie rýchlej opravy funkcionality alebo iného vážneho problému v produkčnej vetve smie vykonávať iba poverená osoba pričom sa predpokladá existencia zodpovedajúcej úlohy v ITS.

Postup:

1. Vytvoriť novú lokálnu vetvu s opravou z produkčnej vetvy.

```
git checkout master
git pull origin master
git branch new-bug
```

2. Implementovať opravu a po častiach lokálne odovzdávať zdrojový kód.
3. Lokálne skontrolovať odovzdaný zdrojový kód, skontrolovať a spustiť testy.
4. Nahrať zdrojový kód vetvy s opravou do vzdialeného repozitára.

```
git push origin new-bug
```

5. Spojiť vetvu s produkčnou vetvou a príslušnou vývojovou vetvou.

```
git checkout master
git merge new-bug

git checkout 1-0-alpha
git merge new-bug
```

6. Vyriešiť konflikty, ak nastali (predpokladajú sa konflikty s príslušnou vývojovou vetvou).
7. Nahrať zdrojový kód produkčnej vetvy a príslušnej vývojovej vetvy do vzdialeného repozitára.

```
git push origin master
git push origin 1-0-alpha
```

8. Požiadat' poverenú osobu o nasadenie produkčnej vetvy do prevádzky.

9 Manažment komunikácie a ľudských zdrojov

Úlohou manažéra komunikácie v našom tíme je iniciovať a viesť komunikáciu v rámci tímu. Je potrebné komunikovať s produktovým vlastníkom, byť schopný posunúť na neho komunikáciu ktorá mu patrí, zároveň ho však príliš nezaťažovať. Manažér komunikácie riadi komunikáciu v tíme, aby bol každý dostatočne informovaný o všetkých dôležitých skutočnostiach.

V našom tíme je manažér komunikácie aj vedúci tímu. Jeho úlohou je viesť stretnutia ak sa ocitnú v mŕtvom bode. Ďalšou dôležitou úlohou je tiež stmelovať tím a snažiť sa o jeho čo najväčšiu efektivitu. Zodpovednosťou vedúceho tímu je odovzdať produktovému vlastníkovi hotový produkt po každom šprinte, resp. ho upovedomiť, ak nastane situácia, že sa práca naplánovaná v danom šprinte nestíha. Vedúci tímu sa musí postarať o to, aby bolo všetko v tíme v poriadku čo sa týka medziľudských vzťahov aj práce.

9.1 Úvod k metodike

Táto metodika má za cieľ opísať a zadefinovať postupy pre komunikáciu v tíme. Vysvetľuje ako sa má v tíme správne komunikovať, akým kanálom (nástrojom) je potrebné riešiť jednotlivé typy komunikácie a ako postupovať pri používaní jednotlivých komunikačných kanálov. Metodika je určená všetkým členom tímu.

Na komunikáciu v tíme sa používajú maily (Gmail), chat klient (Google Hangouts), telefonická komunikácia a nástroj Redmine.

9.2 Prehľad postupov pri riešení komunikácie

Činnosť/nástroj	Mail	Chat	Telefón	Redmine
Komunikácia s celým tímom naraz	x			
Manažovanie tímu	x			
Reportovanie činnosti	x			
Nastavenie nástrojov	(x)			x
Opakované činnosti	(x)			x
Oboznámenie o novej metodike	x			
Pomoc pri probléme s implementáciou	x	x	x	
Upresnenie si	x	x		

podrobností o úlohe				
Urgentný problém	(x)		x	
Žiadosť o revíziu úlohy (kódu)				x
Žiadosť o pomoc pri úlohe				x
Komunikácia k úlohe				x
Vytvorenie novej metodiky				x

9.3 Jednotlivé komunikačné nástroje

9.3.1 Mailová komunikácia

Prostredníctvom mailov sa rieši komunikácia, pri ktorej je potrebné, aby zostala zachovaná pre ďalšiu potrebu, komunikácia s členmi tímu ktorí sú offline, alebo komunikácia s celým tímom naraz.

Typy problémov ktoré sa riešia v našom tíme mailami:

- Komunikácia s celým tímom naraz.
- Manažovanie tímu.
- Reportovanie činnosti tímu.
- Nastavovanie nástrojov – ak je to opakovane používané všetkými členmi tímu, dať do wiki (viď komunikácia v Redmine).
- Činnosti ktoré je potrebné vykonať viackrát – ak je to používané všetkými členmi tímu, dať do wiki (viď komunikácia v Redmine).
- Oboznámenie ostatných členov o novom obsahu vo wiki.

Nástroj na komunikáciu

- Na mailovú komunikáciu vrámci tímu používať výhradne Google skupinu tp-1314-13@googlegroups.com.
 - Tu je možná výnimka ak sa nekomunikuje s celým tímom – potom sa použijú mailové

adresy jednotlivcov.

- Na komunikáciu s vedúcim používať jeho mailovú adresu `srba@fiit.stuba.sk`

Predmet emailu

- Predmet emailu musí vždy začínať reťazcom [tp] .
- V predmete je stručný opis toho o čom sa v maily hovorí – zopár slov z ktorých je jasný cieľ emailu.
- V prípade že je to s úrne, je potrebné indikovať urgentnosť v predmete – zväžiť telefonickú komunikáciu.

Zahrnutie príjemcov

- Do príjemcov zaradiť všetkých, ktorých sa mail priamo dotýka.
- Pri upovedomení inej osoby o komunikácii je potrebné pripojiť daného príjemcu do kópie.
- Nepoužívať skrytú kópiu.
 - Ak je potreba dať vedieť inej osobe interne, daný mail je potrebné preposlať.

Obsah správy

- V správe stručne a výstižne popísať to, čo je jej predmetom.
- Do jednej správy nemiešať viacero rôznych predmetov.
 - Ak nastane taká situácia, je potrebné napísať viacero správ – 1správa = 1 problém.
 - Pri oznamoch prostredníctvom mailu je ich možné zahrnúť viacero do 1 správy.

9.3.2 Chat komunikácia

Prostredníctvom chatu sa riešia problémy ktoré vzniknú napr. pri implementácií a sú malého rozsahu. Táto komunikácia musí byť iba taká, ktorá neovplyvňuje celý tím a nemusí o nej nikto iný vedieť. Na chat komunikáciu v tíme sa používa *Google chat*.

Typy situácií riešených pomocou chatu:

- Riešenie malých problémov (pri implementácií – v prípade že je potrebná konzultácia).
- Dohodnutie podrobností s jednotlivými členmi tímu (o stretnutí, o vopred spísaných požiadavkách na user story).

- Upresnenie podrobností o vopred dohodnutých Používateľských príbehoch (angl. User story), úlohách, atď.

Google Hangout

Pri konzultovaní je jednou z možností konzultácie komunikácia cez *Google Hangout*. Túto formu komunikácie je vhodné využiť najmä pri učení sa novej technológie s ktorou už iní členovia tímu prišli do styku (komunikovať s jedným z nich).

Konzultácia prebieha dvoma spôsobmi:

- Iba hlasová komunikácia.
- Komunikácia spojená so zdieľaním obrazovky.
 - Táto komunikácia prebieha na spôsob párového programovania, kde jeden z účastníkov programuje a druhý mu komentuje kód.

9.3.3 Telefonická komunikácia

Telefonický kontakt na všetkých členov tímu je prístupný v Redmine. Telefonicky sa riešia urgentné problémy, núdzové situácie, na ktoré je potrebná okamžitá spätná väzba. Telefonicky dostupný by mal byť počas dňa každý člen tímu, ak by nastala potreba vyriešiť súrnu situáciu.

9.3.4 Redmine komunikácia

V systéme Redmine sa komunikuje ohľadom stavu a priradenia úloh. Celá komunikácia vzťahujúca sa k jednotlivým úlohám je riešená v tomto systéme. Pre rýchlejšiu spätnú väzbu je odporúčané nastaviť si systém Redmine tak, že o každej zmene v systéme od iného používateľa príde notifikácia na email (v nastaveniach Redmine). Pre viac informácií ako pracovať s úlohami v Redmine viď Metodiku k manažmentu plánovania.

Typy situácií riešených v Redmine:

- Žiadosť o revíziu úlohy (kódu).
- Žiadosť o doplnenie časti úlohy.
- Komunikácia k úlohe.
- Komunikovanie postupov práce ostatným členom tímu.

Žiadosť o revíziu úlohy (kódu)

Po dokončení úlohy je pri niektorých činnostiach potrebná kontrola. Na komunikovanie tohto stavu úlohy sa použije Redmine.

Postup notifikácie člena tímu o žiadosti o revíziu úlohy (kódu):

- Otvoriť úlohu v Redmine.
- Zvoliť *aktualizovať úlohu*.
- *Zodpovednú osobu* nastaviť na toho člena tímu ktorý má na starosti revíziu danej úlohy.
- Do komentára k úlohe pridať poznámky k riešeniu úlohy a napísať požiadavku na revíziu.

Komunikácia k úlohe

Celá komunikácia k riešeniu úlohy musí prebiehať v rámci úlohy v Redmine. Je to potrebné kvôli zachovaniu histórie a prehľadnosti jednotlivých úloh. Taktiež ak by bolo v budúcnosti potrebné riešiť problém podobný tomu ako v aktuálnej úlohe, riešenie bude jednoducho spätne dohľadateľné.

Komunikácia prebieha nasledovne:

- Otvoriť príslušnú úlohu v Redmine a zvoliť *aktualizovať úlohu*.
- Komunikáciu k úlohe napísať do komentáru.
- V prípade že je potrebná pomoc od iného člena tímu, zmeniť zodpovednú osobu na daného člena a do komentára napísať popis problému.
 - ak nie je vopred jasné od koho je potrebná pomoc, zodpovedná osoba sa nemení – žiada sa celý tím.

Komunikácia prostredníctvom wiki

Vo *wiki* na Redmine je komunikácia s tímom prostredníctvom metodík a postupov. Ak chce niekto oznámiť tímu postup riešenia nejakej činnosti, využije na to tento priestor.

Do metodík patrí:

- používanie podporných nástrojov (IDE, verziovanie softvéru, Redmine , ...),
- spôsob písania dokumentov,
- pomenovanie súborov,
- používanie Gitu, pravidiel pri programovaní,
- každá činnosť, ktorú musí tím vykonávať jednotne.

Pri pridávaní novej metodiky je potrebné postupovať nasledovne:

- - ak je to doplnenie už existujúcej metodiky:

- Do príslušnej metodiky doplním svoju časť.
- Je potrebné dodržať formátovanie ako je v celej *wiki* (pre viac informácií ako formátovať viď Metodiku pre používanie Redmine).
- Upovedomiť členov tímu o novej metodike mailom.
- ak je potrebné vytvoriť novú metodiku:
 - V časti *wiki* v Redmine vytvoriť nový vstup pre metodiku.
 - Metodiku napísať štruktúrovane, v bodoch.
 - Písať jasné a stručné body.
 - Upovedomiť členov tímu o novej metodike mailom.

10 Manažment tvorby dokumentácie

Manažér dokumentácie v našom tíme sa stará celistvosť dokumentácií. Prepája jednotlivé časti dokumentácií vytvorené inými členmi tímu. Ďalšou úlohou, ktorú manažér dokumentácie vykonáva, je kontrola a zapracovávanie zápisníc zo stretnutí.

Prepájanie jednotlivých častí prebieha pomocou funkcií vstavaných v LibreOffice, kde sa navzájom porovnávajú dokumenty a vyberajú sa potrebné časti. Aby bolo nutné len minimum opráv, je potrebné, vytvárať dokument podľa dohodnutých pravidiel. Tým sa minimalizujú odlišnosti. Pre tento dôvod boli spísané nasledovné metodiky.

10.1 Metodiky pre dokumentáciu

Hlavnou úlohou tejto metodiky je zadefinovanie základných pravidiel pri vytváraní dokumentácie riadenia, projektovej dokumentácie, zápisníc z formálnych a neformálnych stretnutí.

Pre jednoduchosť a konzistenciu jednotlivých častí dokumentácií boli vytvorené šablóny. Metodika je určená pre všetkých členov tímu, keďže sa každý podieľa na tvorbe dokumentácie.

10.1.1 Zápisnice

Je potrebné, aby všetky osoby podieľajúce sa na dokumentácií a zápisniciach mali jednotný textový editor. My sme si zvolili LibreOffice. Preto je potrebná jeho inštalácia. Jednotný štýl je nastavený priamo v šablóne pre vytváranie zápisníc.

Zápisnice sa nepíšu vždy nanovo ale dopĺňajú sa do šablóny. Podľa šablóny je nutné uviesť v úvodnej strane každej zápisnice:

- Meno autora (formát: titul meno priezvisko titul)
- Dátum vytvorenia zápisnice (formát: DD.MM.RRRR)
- Čas stretnutia (formát HH:MM – HH:MM)
- Položka miestnosť je predvyplnená, keďže sa nepredpokladá, že by sa mohla meniť
- Vedúci stretnutia je osoba, ktorá písala zápisnicu týždeň pred tým.
- V položke prítomní sa uvádzajú prítomné osoby na stretnutí (formát: titul meno priezvisko titul)

Na novú stranu sa píše samotný priebeh stretnutia. Opisujú sa tu jednotlivé body riešené na stretnutí. Formát:

- A
 - a

Na novej strane sa nachádza stav úloh z predchádzajúceho stretnutia. Stav úloh je zapísaný v tabuľke, ktorá obsahuje nasledovné stĺpce:

- ID – Obsahuje identifikačné číslo úlohy
- Opis – Stručný opis úlohy (Veľké začiatkové písmeno)
- Zodpovedná osoba (Priezvisko)
- Dátum zadania – Dátum, kedy bola úloha vytvorená(formát: DD.MM.RRRR)
- Predpokladané ukončenie – Predpokladaný dátum ukončenia(formát: DD.MM.RRRR)
- Stav – Aktuálny stav úlohy (môže mať stavy: nová, riešená, splnená)

Na ďalšej strane sa nachádza tabuľka s novými úlohami do nasledujúceho stretnutia, ktorá sa vyplní rovnako ako predchádzajúca tabuľka.

10.1.2 Projektová dokumentácia

Pri písaní dokumentácie platia nasledovné pravidlá

- Povinné uvádzanie referencií na knižnice, projekty. Píšu sa kurzívou a s číslom na odkaz. Pod čiarou bude „názov: http://...“.
- Nepoužívať krátku pomlčku ale dlhú (v libreoffice 2x krátka pomlčka).

- Kód sa v texte píše fontom Liberation Mono veľkosť 11.
- Zo šablóny, do ktorej sa dopisuje, sa nič nevymazáva.
- Používať prednastavené štýly.
- Nemeniť štýly.
- Nepoužívať anglické výrazy. V prípade, že je anglická verzia nutná, dať do zátvorky (angl.).

Každý člen tímu musí dokumentovať používateľský príbeh(angl. User Story) , ktorý vyvíjal. Nevytvára sa nový dokument, ale vychádza sa zo šablóny. Každá osoba spisuje svoju časť na miesto na to vyhradené v rámci šablóny. Takto prepísaná šablóna sa uloží ako nový dokument, pričom názov má tvar: tp-1314-13-projektova-dokumentacia-X.odt (X je ID dokumentovaného používateľského príbehu).

Každý používateľský príbeh (angl. User Story) má nasledovné body:

- Názov (h3) –Je vo formáte „Šprint – Názov šprintu“.
- Úloha (h2) – Stručná charakteristika úlohy s rozsahom 2 – 5 viet.
- Návrh (h2) – Návrh riešenia danej problematiky. Doplnený môže byť o náčrty ako bude vyzerat implementácia.
- Implementácia (h2) – Spôsob implementácie riešenia problematiky. Môže obsahovať časti kódu napísané podľa pokynov.

11 Preberacie protokoly

A) Záznamy zo stretnutí

1 Zápis č. 1 zo stretnutia tímu č. 13

Autor: Bc. Pavol Zbell

Dátum: 9. 10. 2013

Čas stretnutia: 8:30 – 10:50

Miestnosť: Jobsovo softvérové štúdio

Vedúci stretnutia: Bc. Jozef Harinek

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Poznámka: na nultom stretnutí 2. 10. 2013 sa priblížila téma, termíny, rozdelili sa roly a úlohy členov v tíme, definovali sa základné interné procesy a metodiky tímu, vybrali sa podporné nástroje a ďalej sa diskutovala téma.
- Pomenovanie User Stories, naplnenie backlogu a diskusia
 - 1. šprint: logovanie udalostí v systéme, autorizácia, základný profil používateľa, lokalizácia zodpovedaných otázok.
 - 2. šprint: topic model, pridanie otázky, zobrazenie otázky, zobrazenie zoznamu nových a zodpovedaných otázok.
 - 3. šprint: vkladanie nových odpovedí, hlasovanie za otázky a odpovede, vkladanie komentárov k otázke a odpovediam, výber najlepšej odpovede.
 - 4. šprint: vyhľadávanie v otázkach, preferencie a práva.
 - 5. šprint: refactoring.
- Ostatné User Stories pre letný semester: dashboard, administrácia, social stuff, achievements, design.
- Doplnenie a diskusia o metodikách a úlohách pre členov do najbližšieho stretnutia.

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
446	Pripraviť A3 prezentáciu tímu	Zbell	2. 10. 2013	9. 10. 2013	Splnená
450	Pripraviť vzor pre zápisnicu	Páleník	2. 10. 2013	7. 10. 2013	Splnená
447	Získať prístup k virtuálnemu serveru	Molnár	2. 10. 2013	9. 10. 2013	Splnená
457	Názvy šprintov	Harinek	3. 10. 2013	9. 10. 2013	Splnená
459	Názov projektu	Harinek	3. 10. 2013	6. 10. 2013	Splnená
456	Názov tímu	Harinek	3. 10. 2013	4. 10. 2013	Splnená
458	Kostra webovej prezentácie tímu	Zbell	4. 10. 2013	13. 10. 2013	Riešená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
499	Profil používateľa	Zbell	9. 10. 2013	16. 10. 2013	Nová
483	Autorizácia	Molnár	9. 10. 2013	16. 10. 2013	Nová
496	Prihlásenie	Molnár	9. 10. 2013	16. 10. 2013	Nová
491	Logovanie	Zbell	9. 10. 2013	16. 10. 2013	Nová
487	Lokalizácia	Zbell	9. 10. 2013	16. 10. 2013	Nová
477	Vygenerovať Rails aplikáciu	Zbell	8. 10. 2013	9. 10. 2013	Riešená
461	Texty pre webovú prezentáciu tímu	Dobšovič	4. 10. 2013	13. 10. 2013	Riešená
480	Zpracovať user stories do dokumentácie	Páleník	9. 10. 2013	13. 10. 2013	Nová
445	Pripraviť základy vybraných metódik	Zbell	8. 10. 2013	16. 10. 2013	Riešená

2 Zázpis č. 2 zo stretnutia tímu č. 13

Autor: Bc. Jozef Harinek

Dátum: 16. 10. 2013

Čas stretnutia: 9:00 – 10:50

Miestnosť: Jobsovo softvérové štúdio

Vedúci stretnutia: Bc. Pavol Zbell

Prítomní:
Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Rekapitulácia stavu úloh a rozdelenie úloh do ďalšieho týždňa
 - logovanie už funguje
 - lokalizácia sa doplní
 - autorizácia – funguje
 - prihlasovanie pomocou mena, hesla, prípadne ais login, teraz funguje iba ten
 - zmazanie účtu sme zrušili – v prípade potreby sa vyrieši s adminom
 - doplniť editáciu a zobrazenie profilu
- Dostupný spôsob prihlasovania riešiť cez config
- Dohodnutie si workflow pre code review
 - task v Redmine sa nastaví na 90 % done (ak už je celá práca hotová, inak nie) a Assignee sa nastaví ten kto má urobiť code review. Po code review sa assignee prepne naspäť na autora úlohy a keď sa spracujú pripomienky, môže sa nastaviť 100 %
- Modul pre Redmine na Burn down charts – nájsť a otestovať
- Diskusia o modeli pre používateľa
 - login + nickname, alebo len jedno z nich?
 - Došli sme k záveru že oboje, kvôli možnosti anonymizácie používateľov
- Diskusia o integrácií so sociálnymi sieťami
 - zatiaľ iba linky na profil

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
499	Profil používateľa	Zbell	9. 10. 2013	16. 10. 2013	Riešená
483	Autorizácia	Molnár	9.10.2013	16.10.2013	Riešená
496	Prihlásenie	Molnár	9.10.2013	16.10.2013	Splnená
491	Logovanie	Zbell	9.10.2013	16.10.2013	Splnená
487	Lokalizácia	Zbell	9.10.2013	16.10.2013	Riešená
477	Vygenerovať Rails aplikáciu	Zbell	8.10.2013	9.10.2013	Splnená
461	Texty pre webovú prezentáciu tímu	Dobšovič	4.10.2013	13.10.2013	Riešená
480	Zpracovať user stories do dokumentácie	Páleník	9.10.2013	13.10.2013	Splnená
445	Pripraviť základy vybraných metodík	Zbell	8.10.2013	16.10.2013	Riešená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
500	Zobrazenie profilu používateľa	Grznár	16. 10. 2013	23. 10. 2013	Nová
501	Editácia profilu používateľa	Harinek	16. 10. 2013	23. 10. 2013	Nová
516	Rozbehat lokalne Redmine a dokázat vytvorit burn down chart	Poizl	16. 10. 2013	20. 10. 2013	Nová
517	Vytvorit template pre neformalnu zapisnicu	Páleník	16.10.2013	20.10.2013	Nová

3 Zázpis č. 3 zo stretnutia tímu č. 13

Autor: Bc. Rastislav Dobšovič

Dátum: 23. 10. 2013

Čas stretnutia: 8:30 – 10:50

Miestnosť: Jobsovo softvérové štúdio

Vedúci stretnutia: Bc. Jozef Harinek

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Rekapitulácia stavu support tasks z minulého týždňa a identifikovanie nových
 - server beží
 - web funguje
 - vypracovať prihlášku na TP Cup
- Rekapitulácia stavu user stories a zhodnotenie 1. šprintu
 - logovanie – funguje
 - lokalizácia – potrebné doplniť metodiku, inak funguje
 - prihlásenie – zakázať možnosti „registrovať“ a „zabudli ste heslo“ na stránke, inak funguje
 - zobrazenie profilu – vytvorili sme náčrt dizajnu – je potrebné ho implementovať
 - editácia profilu - presunuté do 2. šprintu; doplniť použitie tabs
- Zhodnotenie 1. šprintu
 - dôraz klásť aj na manažment rizík
 - dokumentarista má vykonávať kontrolu zápisníc
 - v zápisnici majú byť uvedené user stories a pod nimi hierarchicky tasks
- Identifikovanie user stories pre 2. šprint
 - vloženie novej otázky
 - zobrazenie otázky
 - zobrazenie nových a zodpovedaných otázok
- Každá user story rozdelená na rovnaké tasks:
 - Dia
 - migrácia
 - model

- testy
- controller
- view

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
499	Profil používateľa	Zbell	9. 10. 2013	16. 10. 2013	Riešená
483	Autorizácia	Molnár	9.10.2013	16.10.2013	Riešená
487	Lokalizácia	Zbell	9.10.2013	16.10.2013	Splnená
461	Texty pre webovú prezentáciu tímu	Dobšovič	4.10.2013	13.10.2013	Splnená
445	Pripraviť základy vybraných metodík	Zbell	8.10.2013	16.10.2013	Splnená
516	Rozbehať lokálne Redmine a dokázať vytvoriť burn down chart	Poizl	16. 10. 2013	20. 10. 2013	Splnená
517	Vytvoriť šablónu pre neformálnu zpisnicu	Páleník	16.10.2013	20.10.2013	Splnená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
534	Zdokumentovať 1. sprint	Páleník	23.10.2013	6.11.2013	Nová
535	Vytvoriť template k inžinierskej dokumentácii	Páleník	23.10.2013	6.11.2013	Nová
537	Nahodiť charty na redmine	Molnár	23.10.2013	6.11.2013	Nová
563	Vypracovať dve zapisnice neformálnych stretnutí	Páleník	23.10.2013	6.11.2013	Nová
538	Deployment	Molnár	23.10.2013	6.11.2013	Nová
540	Vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
543	Dňa pre vloženie novej otázky	Dobšovič	23.10.2013	28.10.2013	Nová
544	Migrácia pre vloženie novej otázky	Dobšovič	23.10.2013	28.10.2013	Nová
545	Model pre vloženie novej otázky	Dobšovič	23.10.2013	30.10.2013	Nová
546	Testy pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
547	Controller pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
548	View pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
541	Zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
549	Dňa pre zobrazenie otázky	Poizl	23.10.2013	28.10.2013	Nová
550	Migrácia pre zobrazenie otázky	Poizl	23.10.2013	28.10.2013	Nová
551	Model pre vloženie novej otázky	Poizl	23.10.2013	30.10.2013	Nová
552	Testy pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
553	Controller pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
554	View pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
542	Zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová
555	Dňa pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	28.10.2013	Nová
556	Migrácia pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	28.10.2013	Nová
557	Model pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	30.10.2013	Nová
558	Testy pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová
559	Controller pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová
560	View pre zobrazenie nových otázok	Páleník	23.10.2013	6.11.2013	Nová

4 Zázpis č. 4 zo stretnutia tímu č. 13

Autor: Bc. Peter Páleník
Dátum: 30. 10. 2013
Čas stretnutia: 8:30 – 10:00
Miestnosť: Jobsovo softvérové štúdio

Vedúci stretnutia: Bc. Rastislav Dobšovič

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Rekapitulácia stavu support tasks z minulého týždňa
 - dokončenie tasku deployment
 - prihláška na TP-Cup je hotová a odoslaná
- Rekapitulácia stavu user stories z minulého týždňa
 - zistilo sa, že všetky user stories majú rovnaký model
 - diagram v nástorji Dia pre aktuálne user stories je hotový
 - migrácie aj modely pre aktuálne user stories sú hotové
- Diskusia k otázkam
 - potreba viac stavov pre odpoveď?
 - boolean, integer alebo string?
- Počítat' do budúcnosti s potrebou automatického refreshovania
- Zmena user story zo zobrazenie nových a zodpovedaných otázok na zobrazenie nových otázok
- Diskusia o manažérskych pozíciách
 - čo sa pre daný manažment očakáva?
 - čo sa skutočne vykonáva?

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
534	Zdokumentovať 1. sprint	Páleník	23.10.2013	6.11.2013	Riešená
535	Vytvoriť template k inžinierskej dokumentácii	Páleník	23.10.2013	6.11.2013	Riešená
537	Nahodiť charty na redmine	Molnár	23.10.2013	6.11.2013	Splnená
563	Vypracovať dve zapisnice neformálnych stretnutí	Páleník	23.10.2013	6.11.2013	Splnená
538	Deployment	Molnár	23.10.2013	6.11.2013	Splnená
540	Vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Riešená
543	Dňa pre vloženie novej otázky	Dobšovič	23.10.2013	28.10.2013	Splnená
544	Migrácia pre vloženie novej otázky	Dobšovič	23.10.2013	28.10.2013	Splnená
545	Model pre vloženie novej otázky	Dobšovič	23.10.2013	30.10.2013	Splnená
541	Zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Riešená
549	Dňa pre zobrazenie otázky	Poizl	23.10.2013	28.10.2013	Splnená
550	Migrácia pre zobrazenie otázky	Poizl	23.10.2013	28.10.2013	Splnená
551	Model pre vloženie novej otázky	Poizl	23.10.2013	30.10.2013	Splnená
542	Zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Riešená
555	Dňa pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	28.10.2013	Splnená
556	Migrácia pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	28.10.2013	Splnená
557	Model pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	30.10.2013	Splnená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
546	Testy pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
547	Controller pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
548	View pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Nová
552	Testy pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
553	Controller pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
554	View pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
558	Testy pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová
559	Controller pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová
560	View pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Nová

5 Zázpis č. 5 zo stretnutia tímu č. 13

Autor: Bc. Samuel Molnár
Dátum: 06. 11. 2013
Čas stretnutia: 8:30 – 10:50
Miestnosť: Jobsovo softvérové štúdio

Vedúci stretnutia: Bc. Peter Páleník

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Rekapitulácia stavu support tasks z minulého týždňa
 - dokončenie dokumentácie k user stories prihlásenie a lokalizácia.
 - rozpracovaná dokumentácia pre user story logovanie, editácia profilu a zobrazenie profilu.
 - analyzované možnosti git hook pre analýzu súborov a textu commit message.
- Rekapitulácia stavu user stories z minulého týždňa
 - profil používateľa je hotový
 - editácia profilu používateľa je dokončená až na absenciu validácií v modeli a ich testov
 - vloženie novej otázky, zobrazenie nových otázok a zobrazenie novej otázky majú pripravený controller a návrh view
- Diskusia k architektúre modelov
 - označenia odpovedí budú rozšíriteľné pomocou definovaných značiek v modeli AnswerLabel
 - asociácie v modeloch sa uvádzajú v skrátenej formáte, bez prefixu asociovaného modelu
- Diskusia k git branching a workflow metodikám
 - práca v jednotlivých vetvách sa spája pomocou merge do spoločnej alpha vetvy a vývojári robí rebase voči vetve alpha
 - do vetvy staging a master sa spájajú konkrétne hotové user stories
- Naplánovať pravidelné týždenné stretnutia za účelom podpory pair programming

Retrospektíva

Start

- dokončiť úlohy s predstihom, aby mohli čím skôr prejsť fázou *Code Review*
- viac osobných stretnutí za účelom aplikácie techniky *pair programming*
- aplikovať upravený git workflow
- zlepšiť komunikáciu na úrovni tím – product owner

Keep

- neformálne stretnutia
- testovanie
- code review

Stop

- porušovanie metodík

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
546	Testy pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Riešená
547	Controller pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Riešená
548	View pre vloženie novej otázky	Dobšovič	23.10.2013	6.11.2013	Riešená
552	Testy pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Nová
553	Controller pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Riešená
554	View pre zobrazenie otázky	Poizl	23.10.2013	6.11.2013	Riešená
558	Testy pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Riešená
559	Controller pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Riešená
560	View pre zobrazenie nových a zodpovedaných otázok	Páleník	23.10.2013	6.11.2013	Riešená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
593	Zobrazenie odpovedanych otazok	Páleník	6.11.2013	20.11.2013	Nová
592	Vyber najlepszej odpovede	Harinek	6.11.2013	20.11.2013	Nová
591	Zobrazenie odpovedi pri zobrazeni otazky	Poizl	6.11.2013	20.11.2013	Nová
589	View pre vlozenie novej odpovede	Grznár	6.11.2013	20.11.2013	Nová
588	Controller pre otazky	Grznár	6.11.2013	20.11.2013	Nová
587	Testy pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
586	Model pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
585	Migracia pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
584	Dia pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
583	Vlozenie novej odpovede	Grznár	6.11.2013	20.11.2013	Nová

6 Zázpis č. 6 zo stretnutia tímu č. 13

Autor: Bc. Marek Grznár

Dátum: 13. 11. 2013

Čas stretnutia: 8:30 – 10:50

Miestnosť: 3.08 zasadačka UISI

Vedúci stretnutia: Bc. Samuel Molnár

Prítomní:
Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Rekapitulácia stavu support tasks z minulého týždňa
 - rozpracovaná dokumentácia pre jednotlivé user stories
 - implementované git hook
 - vypracovaná prezentácia tímu pre SCRUM panel
- Rekapitulácia stavu user stories (buggov) z minulého týždňa
 - implementované tagovanie
 - oprava buggov
 - pridanie novej otázky je hotové
 - implementované indexovanie zobrazených otázok
- Ukážka s vysvetlením CanCan testov
- Diskusia k user stories, ktoré neboli akceptované product ownerom
 - pre pridanie novej otázky sa indentifikovali buggy, ktoré si vyžadujú opravu
 - úprava view pre zobrazenie otázky
- Diskusia ku code review
 - ukážka ako by kód nemal vyzerat' a ako by ho bolo vhodné opraviť
 - HTML kód treba mať na 100% ak sa dá

Stav úloh z predchádzajúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
593	Zobrazenie odpovedaných otázok	Páleník	6.11.2013	20.11.2013	Nová
592	Vyber najlepšej odpovede	Harinek	6.11.2013	20.11.2013	Nová
591	Zobrazenie odpovedi pri zobrazení otázky	Poizl	6.11.2013	20.11.2013	Nová
589	View pre vloženie novej odpovede	Grznár	6.11.2013	20.11.2013	Nová
588	Controller pre otázky	Grznár	6.11.2013	20.11.2013	Nová
587	Testy pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
586	Model pre odpovede	Grznár	6.11.2013	20.11.2013	Nová
585	Migrácia pre odpovede	Grznár	6.11.2013	20.11.2013	Riešená
584	Dia pre odpovede	Grznár	6.11.2013	20.11.2013	Riešená
583	Vloženie novej odpovede	Grznár	6.11.2013	20.11.2013	Riešená

Nové úlohy do nasledujúceho stretnutia

ID	Opis	Zodpovedná osoba	Dátum zadania	Predpokladané ukončenie	Stav
600	Spraviť helper pre name usera		13.11.2013	20.11.2013	Nová
601	Opraviť testy	Molnár	13.11.2013	13.11.2013	Nová
602	Nastylovať hornu listu	Zbell	13.11.2013	20.11.2013	Nová
603	Nastaviť na parameter veci, ako registrovať, zabudli ste heslo...	Molnár	13.11.2013	20.11.2013	Nová
604	Upraviť placeholder v textboxe pre pridanie tagu v pridanií novej otázky	Molnár	13.11.2013	20.11.2013	Nová
605	Preložiť hlasku pri pokuse uložiť neexistujúcu kategóriu pri pridani novej otázky	Molnár	13.11.2013	20.11.2013	Nová
606	Nastylovať pridanie kategórie pri pridani novej otázky	Zbell	13.11.2013	20.11.2013	Nová
607	V ponuke tagov dať nový tag slabou zelenou (pri pridani novej otázky)	Zbell	13.11.2013	20.11.2013	Nová
608	Normalizácia pridávani tagov	Molnár	13.11.2013	20.11.2013	Nová
609	Zapamätanie si tagov pri chybe pri reloadede	Molnár	13.11.2013	20.11.2013	Nová
611	Pri zmene tabov v nastaveniach mení URL	Harinek	13.11.2013	4.12.2013	Nová
612	Doplniť retrospektívu do zapisnice zo stretnutia 20131106	Molnár	13.11.2013	17.11.2013	Nová
613	Zdokumentovať svoju časť vývoja v doterajších sprintoch	Dobšovič	13.11.2013	17.11.2013	Nová
614	Zdokumentovať svoju časť vývoja v doterajších sprintoch	Páleník	13.11.2013	17.11.2013	Nová
615	Zdokumentovať svoju časť vývoja v doterajších sprintoch	Poizl	13.11.2013	17.11.2013	Nová
616	Zdokumentovať svoju manažerskú rolu	Poizl	13.11.2013	17.11.2013	Nová
617	Zdokumentovať svoju manažerskú rolu	Harinek	13.11.2013	17.11.2013	Nová
618	Zdokumentovať svoju manažerskú rolu	Grznár	13.11.2013	17.11.2013	Nová
619	Zdokumentovať svoju manažerskú rolu	Zbell	13.11.2013	17.11.2013	Nová
620	Zdokumentovať svoju manažerskú rolu	Páleník	13.11.2013	17.11.2013	Nová
621	Zdokumentovať svoju manažerskú rolu	Dobšovič	13.11.2013	17.11.2013	Nová
622	Zdokumentovať svoju manažerskú rolu	Molnár	13.11.2013	17.11.2013	Nová

B) Záznamy z neformálnych stretnutí

1 Zápis č. 1 z neformálneho stretnutia tímu č. 13

Autor: Bc. Peter Páleník

Dátum: 8 10. 2013

Čas stretnutia: 9:00 – 11:00

Miestnosť: 3.08

Vedúci stretnutia: Bc. Pavol Zbell

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Zhodnotenie poznatkov členov tímu s používaním nástroja git
- Uvedenie členov tímu do práce s nástrojom git
 - Stanovenie pravidiel pre používanie nástroja git
 - Ukážka preberania a vkladania dát na git
 - Stanovenie metodík na písanie *commit message*
 - Ukážka písania a mazania *commit message*

2 Zázpis č. 2 z neformálneho stretnutia tímu č. 13

Autor: Bc. Peter Páleník

Dátum: 15. 10. 2013

Čas stretnutia: 9:00 – 11:00

Miestnosť: 3.08

Vedúci stretnutia: Bc. Samuel Molnár

Prítomní: Ing. Ivan Srba
Bc. Rastislav Dobšovič
Bc. Marek Grznár
Bc. Jozef Harinek
Bc. Samuel Molnár
Bc. Peter Páleník
Bc. Dušan Poizl
Bc. Pavol Zbell

Priebeh stretnutia

- Základné inštrukcie k programovaniu v Ruby
- Uvedenie členov tímu do problematiky testovania a vysvetlenie jej dôležitosti
- Prezentácia k Ruby a k testovaniu
 - Stanovenie základných pravidiel pri tvorbe testov
 - Priblíženie práce so Service Objects a Capybara
 - Ukážka tvorby testov priamo v editore vim s komentovaním kódu
- *Diskusia k ukázkovým testom*