

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava 4

Pinta.sk

Digital Sweatshop

Dokumentácia k riadeniu

Študijný program: Informačné systémy, Softvérové inžinierstvo

Ročník: 1.

Predmet: Tímový projekt I.

Vedúci projektu: Ing. Dušan Zeleník

Členovia tímu: Bc. Filip Bednárík

Bc. Róbert Černý

Bc. Marek Lenčes

Bc. Miroslav Molnár

Bc. Patrik Štrba

Bc. Martin Toma

Bc. Miroslav Vojtuš

Ak. rok: 2013/2014

Obsah

1	Úvod	4
1.1	Účel dokumentu	4
1.2	O projekte	4
1.3	Štruktúra dokumentu	4
2	Ponuka	5
2.1	Členovia tímu	5
2.1.1	Bc. Filip Bednárík	5
2.1.2	Bc. Róbert Černý	5
2.1.3	Bc. Marek Lenčes	5
2.1.4	Bc. Miroslav Molnár	5
2.1.5	Bc. Patrik Štrba	5
2.1.6	Bc. Martin Toma	6
2.1.7	Bc. Miroslav Vojtuš	6
2.2	Motivácia	6
2.3	Koncept riešenia	6
3	Záznamy zo stretnutí	7
3.1	Šablóna zápisnice	7
3.2	Zápis zo stretnutia č. 1	8
3.3	Zápis zo stretnutia č. 2	10
3.4	Zápis zo stretnutia č. 3	12
3.5	Zápis zo stretnutia č. 4	14
3.6	Zápis zo stretnutia č. 5	16
3.7	Zápis zo stretnutia č. 6	18
3.8	Zápis zo stretnutia č. 7	20
3.9	Zápis zo stretnutia č. 8	22
3.10	Zápis zo stretnutia č. 9	24
3.11	Zápis zo stretnutia č. 10	26
4	Úlohy členov tímu	28
4.1	Manažérske roly v tíme	28
4.2	Krátkodobé úlohy	28
4.3	Dlhodobé úlohy	29
4.4	Podiel členov na tvorbe dokumentácie k inžinierskemu dielu	30
4.5	Podiel členov pri tvorbe diela	30
5	Manažment rozvrhu a plánovania	30
5.1	Procesy	30
5.1.1	Tvorba a udržiavanie produktového backlogu	30
5.1.2	Tímové stretnutia a šprinty	31
5.1.3	Komunikácia počas šprintu	33
6	Manažment rizík	34

7	Manažment monitorovania projektu	39
7.1	Navrhovaný prístup	39
7.2	Aplikácia prístupu	40
7.2.1	Dokumentovanie práce	40
7.2.2	Vyhodnotenie stavu projektu	40
8	Manažment komunikácie a ľudských zdrojov	42
8.1	Metodika – Zaznamenávanie práce a sledovanie postupu	44
9	Manažment dokumentácie	46
9.1	Roly účastníkov	46
9.2	Projektová dokumentácia	46
9.3	Zápisnice	47
9.4	Dokumentácia zdrojového kódu	48
9.5	Procesy	48
9.5.1	Dokumentovanie dokončenej funkcionality	48
9.5.2	Revízia dokumentácie	48
9.5.3	Dokumentovanie stretnutí	49
10	Manažment kvality	50
10.1	Konvencie v jazyku Ruby a programovom rámci RoR	50
10.1.1	Všeobecné	50
10.1.2	Písanie názvov	50
10.1.3	Ruby konvencie	51
10.1.4	Migrácie	52
10.1.5	Model	52
10.1.6	Ovládač (ang. Controller)	52
10.1.7	Pohľad (ang. View)	53
10.2	Testovanie	53
10.2.1	Procesy	53
10.2.2	Konvencie pre testovanie	55
10.3	Spätná kontrola kvality	57
11	Manažment podpory vývoja	58
11.1	Použitie vetiev (branch) v tímovom repozitári	58
11.2	Potvrdenie funkcionality do tímového repozitára	59
11.3	Vytvorenie novej verzie	59
11.4	Vytvorenie novej funkcionality	59
11.5	Oprava chyby na vývojovej verzii aplikácie	60
11.6	Oprava chyby na produkčnej verzii aplikácie	60
11.7	Spolupráca na spoločnej funkcionalite	60
12	Plány	61
12.1	Zimný semester	61
12.2	Letný semester	61
12.2.1	Krátkodobý plán	61
12.2.2	Dlhodobý plán	62

1 Úvod

1.1 Účel dokumentu

Tento dokument je jednou z dvoch častí dokumentácie k vyvíjanému projektu, a obsahuje podrobnú dokumentáciu procesov riadenia pri vývoji projektu Pinta.sk vyvíjaného v rámci predmetu Tímový projekt na Fakulte informatiky a informačných technológií STU v Bratislave, so všeobecným názvom projektu Digital Sweatshop. Druhá časť dokumentácie s názvom Dokumentácia k inžinierskemu dielu, je dokumentáciou k vytváranému produktu.

1.2 O projekte

Projekt je akademickým dielom, a na jeho tvorbe sa podieľajú študenti spomenutej fakulty, konkrétne sedem členov tímu č. 11, ktorí svoj tím nazvali PintaGroup. Projekt Pinta.sk je originálnym a inovatívnym spôsobom ako si zarobiť peniaze za primeranú prácu aj z pohodlia domova, a zároveň ponúka možnosť rôznym firmám riešiť problémy a úlohy, pri riešení ktorých sa nezaobídu bez ľudskej pomoci. Ide o systém, presnejšie aplikáciu s webovým rozhraním pre zadávanie rôznych úloh za finančnú alebo inú odmenu. Vypracovanie týchto úloh za odmenu je podmienené zručnosťami a schopnosťami riešiteľa, a tiež jeho aktuálnou “pozíciou”, respektíve levelom na ktorom sa nachádza v rámci systému, podľa ktorého výšky sa mu sprístupňujú stále iné úlohy za vyššie odmeny.

Vytváranie projektu v rámci tímu je riadené Scrum metodikou, ide o agilný vývoj, teda o prinášanie novej funkcionality po každom šprinte, na ktoré je celkový vývoj aplikácie rozdelený. Tím sa počas šprintov pravidelne stretáva a konzultuje výsledky práce, svoj postup, a prípadné nedostatky v tíme.

1.3 Štruktúra dokumentu

Prvá kapitola dokumentu obsahuje ponuku vypracovaní tímom pre získanie projektu spomedzi ponúkaných projektov.

V druhej kapitole sa nachádzajú všetky doterajšie zápisnice zo stretnutí tímu podľa jednotlivých šprintov.

Tretia kapitola obsahuje zoznam krátkodobých a dlhodobých úloh členov v rámci tímu.

Ďalsie kapitoly obsahujú návody, postupy, a metodiky, ako používať rôzne nástroje, alebo ako postupovať pri vývoji produktu metodikou Scrum.

Predposledná kapitola obsahuje podrobný plán pre vypracovanie projektu na obidva semestre.

2 Ponuka

2.1 Členovia tímu

2.1.1 Bc. Filip Bednárík

Volám sa Filip, v tíme zastávam pozíciu hlavného softvérového architekta a popri tom sa venujem manažérstvu rizík. Okrem svojich programátorských povinností zastávam aj úlohu inšpektora pri prehliadkach zdrojového kódu. Vyštudoval som bakalára na FIITke a mám dlhoročné skúsenosti s prácou vo firmách na tvorbe rozličných webov. Vo voľnom čase sa venujem kamarátom, gitare a sem tam si niečo nakódim. Motto: *„Keď sa všetci chytíme vesla, tak hádam niekam doveslujeme“*.

2.1.2 Bc. Róbert Černý

Volám sa Robo, mám 22 rokov a v tíme zastávam úlohu manažéra plánovania a organizácie. Dozerám na správnosť priebehu stretnutí, usmerňujem komunikáciu a dozerám na dodržiavanie dohodnutých termínov pre plnenie potrebných úloh. Náš tím je agilný a moja metodika SCRUM-u je prispôbená našim podmienkam. Som taktiež absolventom bakalárskeho štúdia na FIIT STU v obore informatika. Mám praktické skúsenosti s vývojom webových aplikácií a som otvorený novým možnostiam a prístupom. Voľný čas trávim cestovaním, športovaním, počúvaním hudby a hľadaním zmyslu života. Motto: *„Shut up and squat“*.

2.1.3 Bc. Marek Lenčes

Moje meno je Marek a v tíme PintaGroup zohrávam úlohu manažéra dokumentácie. Kontrolujem jej stav, zabezpečujem jej konzistentnosť, bezchybnosť a som zodpovedný za jej finálne prezentovanie. Štúdium som absolvoval na FIIT STU a počas štúdia som sa priučil mnohým veciam. Medzi moje záľuby patrí navrhovanie 3D dizajnov interiérov, ale aj exteriérov budov, tiež modelovanie 3D grafiky vo všeobecnosti, oprava notebookov a kšeftovanie s elektronikou, príležitostne aj indoorové športy, napríklad biliard, bowling, či stolný tenis. V živote sa riadim heslom: *„Keď druhému jamu kopeš, sprav ju aspoň poriadne hlbokú.“*

2.1.4 Bc. Miroslav Molnár

Moje meno je Miroslav, v tíme zastávam úlohu monitorovania projektu. Baví ma kreatívna činnosť a často sa podieľam na hľadaní nových funkcionalít v systéme. Som absolvent FIT VUT a vo svojom voľnom čase sa venujem športu a záhradkárstvu. Motto: *„Život je krutý, život je pes a pes je najlepší priateľ človeka“*.

2.1.5 Bc. Patrik Štrba

Volám sa Patrik a mám 22 rokov. V tíme sa starám o spoločný tímový repozitár. Taktiež sa starám o proces nasadenia projektu na produkčný server. Som absolventom bakalárskeho štúdia na FIIT STU. Aktívne sa venujem start-up projektom a zaujímam sa o počítačovú bezpečnosť. Mám dlhoročné skúsenosti s vývojom webových aplikácií. Vo voľnom čase rád počúvam hudbu, učím nové zaujímavé veci alebo si rád pozriem dobrý film.

2.1.6 Bc. Martin Toma

Volám sa Martin mám 22 rokov a v tíme sa starám o to, aby sme spolu komunikovali čo možno najefektívnejšie. Zabýdam však skoro do každej oblasti nakoľko plním aj funkciu vedúceho projektu a chcem mať vo všetkom prehľad. Som absolventom bakalárskeho štúdia na FIIT STU, rád sa venujem najmä soft-skills oblasti a zaujímam sa o start-up scénu a agilné metódy. V súkromí si rád zájdem zašportovať, mám rád dobrú hudbu, chutné jedlo a pekné ženy. Mojim mottom je: „*Everything is possible with the right attitude!*“

2.1.7 Bc. Miroslav Vojtuš

Volám sa Miroslav, mám 23 rokov a v tíme sa starám o kvalitu produktu. Okrem tejto manažérskej role som pracant-vývojár alebo inak aj kóder. Vo voľnom čase sa rád venujem športom, hlavne streľbe z luku, alebo šermu. Rád pozerám fantastické filmy s nádychom histórie. Okrem toho ešte rád splavujem divoké rieky, a objavujem dobrodružstvá pri cestovaní svetom. Moje motto je: „*Just do it !!!*“

2.2 Motivácia

Základnou náplňou projektu je prostredníctvom webovej aplikácie poskytnúť istý druh rozhrania medzi dvoma skupinami ľudí, z ktorých jedna skupina potrebuje získať určitú formu spätnej väzby a druhá skupina ľudí si chce privyrobiť a v neposlednom rade aj trochu pomôcť. Využitie tohto rozhrania je tak veľké, že môžeme pokryť až veľmi širokú oblasť. Od akademickej pôdy pri rôznych formách úloh, od dotazníkov cez jednoduché úlohy až po zložitejšie overovanie hypotéz vrámci bakalárskych, diplomových či dizertačných prác. Ďalšou veľkou oblasťou je marketing. Nahradenie zastaraných marketingových techník a aktivít (telefónny marketing) za priame oslovenie cieľovej skupiny prostredníctvom nášho rozhrania. Možnosti využitia tohto rozhrania sú teda naozaj skoro neobmedzené.

2.3 Koncept riešenia

Základným cieľom projektu je vytvoriť aktívne používanú webovú či mobilnú aplikáciu, ktorej hlavným cieľom je sprostredkovať mikroprácu. Chceme dosiahnuť spokojnosť na oboch stranách (zadávateľ aj pracant). Zadávateľovi chceme ponúknuť možnosť získať kvalitnú spätnú väzbu tým že efektívne nájdeme správnych kandidátov na danú úlohu. Pracantovi chceme dať možnosť si privyrobiť a motivovať ho systémom reputácie a odmeňovania. Projekt má potenciál poskytnúť rozhranie využiteľné v rôznych oblastiach. Hlavným cieľom tohto projektu je však pokrytie akademickej a marketingovej oblasti. V budúcnosti sa však určite zameriame aj na iné oblasti.

3 Záznamy zo stretnutí

3.1 Šablóna zápisnice

Zápisnica č. 4. / (číslo zápisnice)

Téma Plánovanie 3. šprintu	Dátum 21.10.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Marek Lenčes	

Priebeh stretnutia

(Obsahuje informácie o priebehu stretnutia, najprv vyhodnotenie šprintu, čo sa spravilo, čo sa nestihlo, ďalej rozoberané veci na stretnutí, a nakoniec naplánovanie úloh do ďalšieho šprintu)

- Vyhodnotenie druhého šprintu
 - vyhodnotenie splnenia zadaných úloh
- Diskusia
 - čo robíme dobre, čo robíme zle, návrhy, pripomienky
- Naplánovanie ďalšieho šprintu
 - výber úloh z produktového backlogu a pridelenie členom tímu

Rozdelenie úloh

(Rozdelenie úloh z backlogu a pridelenie členom tímu bude usporiadane zobrazené v tabuľke, s krátkym popisom úlohy, menom člena, ktorému bola úloha pridelená, dátum kedy bola zadaná, kedy bude približne ukončená, a v akom stave sa momentálne nachádza. Príklad:)

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokladané ukončenie	Stav
3.1	Definovanie prvej user story a rozdelenie na podúlohy	Filip	14.10.2013	20.10.2013	Riešená
3.2	Analýza podporných nástrojov pre implementáciu v Ruby	Robo	14.10.2013	20.10.2013	Splnená

3.2 Zápis zo stretnutia č. 1

Zápisnica č. 1.

Téma	Dátum	Čas	Miesto
Úvodné stretnutie	30.9.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Molnár		

Priebeh stretnutia

- Podrobné rozobratie témy projektu
 - Oboznámenie sa s problémovým prostredím
 - Popis štruktúry aplikácie
 - Popis podobných aplikácií a ich nedostatky
 - Možnosti využitia aplikácie v našom okolí
 - Definovanie potencionálnych používateľov a klientov
- Diskusia o projekte
 - Výber podporných nástrojov pre realizáciu projektu
 - Identifikácia prvých základných features aplikácie
 - Dohoda na dĺžke trvania šprintov
 - Dohoda na čase a mieste stretnutí
 - Rozdelenie jednotlivých rolí v tíme
 - Prediskutovanie vybraných podporných nástrojov - Rails, Github, Jira
 - Výber nástroja pre code review
 - Diskusia o problémoch pri použití code review nástrojov
- Naplánovanie akcií do prvého šprintu
 - Analýza existujúcich riešení
 - Identifikácia čo najväčšieho množstva features aplikácie
 - Vytvorenie prezentačnej stránky tímu
 - Dohodnutie sa na použití podporných nástrojov a ich inštalácia
 - Vytvorenie šablóny zápisnice
 - Vytvorenie Lean Canvas
 - Naštudovať základné konvencie jazyka Ruby a frameworku Rails
 - Dohodnúť sa na verzii týchto nástrojov, ktorá sa bude používať
 - Naštudovať konvencie používania verziovacieho systému GitHub
 - Konfigurácia podporných nástrojov pre naše potreby - Jira, Jenkins

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Definovanie dôležitých vlastností produktu, vymenovanie features, doporučiť technológie použité v projekte.	Filip, Robo, Martin, Patrik, Miro V, Miro M, Marek	30.9.2013	13.10.2013	Riešená
0.2	Inštalácia a konfigurácia podporných nástrojov	Martin, Patrik	30.9.2013	13.10.2013	Riešená
0.3	Vytvorenie prezentačnej stránky	Miro M., Marek, Robo	30.9.2013	13.10.2013	Riešená
0.4	Vytvorenie šablóny zápisnice	Marek	30.9.2013	13.10.2013	Riešená
0.5	Analýza existujúcich riešení	Marek	30.9.2013	13.10.2013	Riešená
0.6	Vytvorenie Lean Canvas	Spoločná	30.9.2013	13.10.2013	Riešená
0.7	Vymyslenie názvu aplikácie a tímu	Spoločná	30.9.2013	13.10.2013	Riešená
0.8	Štúdium Ruby on Rails, Github	Spoločná	30.9.2013	13.10.2013	Riešená

3.3 Zápis zo stretnutia č. 2

Zápisnica č. 2.

Téma Plánovanie 1. šprintu	Dátum 14.10.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Marek Lenčes	

Priebeh stretnutia

- Vyhodnotenie úvodnej prípravy
 - Vyhodnotenie splnenia zadaných úloh členmi tímu
 - Diskusia o problémoch pri úlohách
 - Prehodnotenie trvania dĺžky šprintov - 1-týždňové
 - Zhodnotenie vhodnosti vytvorenej prezentačnej stránky
- Diskusia o projekte
 - Prediskutovanie vybraných podporných nástrojov - Rails, Github, Jira
 - Diskusia o návrhoch názvu aplikácie - zhodli sme sa na Taskerix
 - Prvotné návrhy štruktúry a dizajnu aplikácie
 - Návrhy pre zlepšenie a rozšírenie vytvárania úloh používateľom
 - Diskusia o potrebe definovania produktového backlogu aplikácie
- Naplánovanie prvého šprintu
 - Potreba definovať úlohy v rámci prvého user story
 - Návrh rozhrania aplikácie
 - Vytvoriť ďalšie features pri vytváraní úlohy
 - Vytvoriť formuláre pre registráciu a prihlásenie
 - Vyriešiť autorizáciu a autentifikáciu pri prihlasovaní
 - Vytvorenie uploadera na obrázky
 - Formulár na zadanie podrobností vytváranej úlohy, štruktúra úlohy, nápoveda k úlohe, definovanie kategórií úloh

Stav úloh z úvodného stretnutia

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
0.1	Definovanie dôležitých vlastností produktu, vymenovanie features, doporučiť technológie použité v projekte.	Filip, Robo, Martin, Patrik, Miro V, Miro M, Marek	30.9.2013	13.10.2013	Splnená
0.2	Inštalácia a konfigurácia podporných nástrojov	Martin, Patrik	30.9.2013	13.10.2013	Splnená
0.3	Vytvorenie prezentačnej stránky	Miro M., Marek, Robo	30.9.2013	13.10.2013	Splnená
0.4	Vytvorenie šablóny zápisnice	Marek	30.9.2013	13.10.2013	Splnená
0.5	Analýza existujúcich riešení	Marek	30.9.2013	13.10.2013	Splnená
0.6	Vytvorenie Lean Canvas	Spoločná	30.9.2013	13.10.2013	Splnená
0.7	Vymyslenie názvu aplikácie a tímu	Spoločná	30.9.2013	13.10.2013	Splnená
0.8	Štúdium Ruby on Rails, Github	Spoločná	30.9.2013	13.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Definovanie úloh prvej User Story	Filip	14.10.2013	20.10.2013	Riešená
1.2	Definovanie úloh prvej User Story + návrh rozhrania aplikácie	Robo	14.10.2013	20.10.2013	Riešená
1.3	User model, autorizácia, autentifikácia používateľa	Martin, Patrik	14.10.2013	20.10.2013	Riešená
1.4	Uploader na obrázky a súbory	Miro V.	14.10.2013	20.10.2013	Riešená
1.5	Formulár na zadanie podrobností vytváratej úlohy, štruktúra úlohy, nápoveda k úlohe, definovanie kategórií úloh	Marek, Miro M.	14.10.2013	20.10.2013	Riešená
1.6	Vytvorenie rozhrania pre registráciu a prihlásenie	Martin, Patrik	14.10.2013	20.10.2013	Riešená
1.7	Vytvorenie rozhrania pre vytvorenie úlohy používateľom	Robo, Filip	14.10.2013	20.10.2013	Riešená

3.4 Zápis zo stretnutia č. 3

Zápisnica č. 3.

Téma	Dátum	Čas	Miesto
Plánovanie 2. šprintu	21.10.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Miroslav Vojtuš		

Priebeh stretnutia

- Vyhodnotenie prvého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie low fidelity návrhu dizajnu aplikácie
 - Ukážka funkcionality riešených features

- Diskusia o projekte
 - Identifikácia jednotlivých tém pre metodiky
 - Rozdelenie metodík medzi členov tímu
 - Metodika Code Review - Bc. Filip Bednárík
 - Metodika Scrum fungovania - Bc. Róbert Černý
 - Metodika Jira - Bc. Martin Toma
 - Metodika dokumentácie - Bc. Marek Lenčes
 - Metodika Git verziovacieho systému - Bc. Patrik Štrba
 - Metodika podpory vývoja - Bc. Miroslav Molnár
 - Metodika menných konvencií pre Rails - Bc. Miroslav Vojtuš
 - Diskusia o potrebe definovania produktového backlogu

- Naplánovanie ďalšieho šprintu
 - Potreba zjednotiť všetky vetvy na GitHub-e
 - Ďalšie hlbšie štúdium Rails
 - Dokončenie detailov jednotlivých funkcionalít

Stav úloh z prvého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
1.1	Definovanie úloh prvej User Story	Filip	14.10.2013	20.10.2013	Splnená
1.2	Definovanie úloh prvej User Story + návrh rozhrania aplikácie	Robo	14.10.2013	20.10.2013	Splnená
1.3	User model, autorizácia, autentifikácia používateľa	Martin, Patrik	14.10.2013	20.10.2013	Splnená
1.4	Uploader na obrázky a súbory	Miro V.	14.10.2013	20.10.2013	Splnená
1.5	Formulár na zadanie podrobností vytvárateľ úlohy, štruktúra úlohy, nápoeda k úlohe, definovanie kategórií úloh	Marek, Miro M.	14.10.2013	20.10.2013	Splnená
1.6	Vytvorenie rozhrania pre registráciu a prihlásenie	Martin, Patrik	14.10.2013	20.10.2013	Splnená
1.7	Vytvorenie rozhrania pre vytvorenie úlohy používateľom	Robo, Filip	14.10.2013	20.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Štúdium Rails	Spoločná	21.10.2013	27.10.2013	Riešená
2.2	Zjednotenie všetkých funkcionálov do jednej vetvy na GitHubu	Spoločná	21.10.2013	27.10.2013	Riešená
2.3	Dokončenie detailov na každej vetve	Spoločná	21.10.2013	27.10.2013	Riešená

3.5 Zápis zo stretnutia č. 4

Zápisnica č. 4.

Téma Plánovanie 3. šprintu	Dátum 28.10.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Marek Lenčes	

Priebeh stretnutia

- Vyhodnotenie druhého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
- Diskusia o projekte
 - Doplnenie produktového backlogu
 - Diskusia o zmene git klienta z GitHub-u na Bitbucket
 - Dekompozícia User Stories na úlohy
 - Dohoda na potrebe pravidiel - vypracovaním metodík
- Naplánovanie ďalšieho šprintu
 - Zjednotiť produktový backlog
 - Treba napísať všetky definované metodiky
 - Je potrebné vytvoriť tímový kalendár kvôli stretnutiam
 - Vyriešiť spustenie produkčného serveru pre hostovanie aplikácie
 - Treba zrevidovať všetky doterajšie zápisnice

Stav úloh z druhého šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
2.1	Štúdium Rails	Spoločná	21.10.2013	27.10.2013	Prebieha
2.2	Zjednotenie všetkých funkcionalít do jednej vetvy na GitHubu	Spoločná	21.10.2013	27.10.2013	Splnená
2.3	Dokončenie detailov na každej vetve	Spoločná	21.10.2013	27.10.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Usporiadanie produktového backlogu	Martin, Patrik	28.10.2013	03.11.2013	Riešená
3.2	Spísanie metodík	Spoločná	28.10.2013	03.11.2013	Riešená
3.3	Vytvorenie tímového kalendáru	Robo	28.10.2013	03.11.2013	Riešená
3.4	Spustenie produkčného servera	Patrik	28.10.2013	03.11.2013	Riešená
3.5	Revízia doterajších zápisníc	Marek	28.10.2013	03.11.2013	Riešená
3.6	Vytvorenie formulára pre vyplnenie profilu	Miro M.	28.10.2013	03.11.2013	Riešená

3.6 Zápis zo stretnutia č. 5

Zápisnica č. 5.

Téma	Dátum	Čas	Miesto
Plánovanie 4. šprintu	04.11.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Róbert Černý		

Priebeh stretnutia

- Vyhodnotenie tretieho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Dokončenie produktového backlogu
 - Migrácia na Bitbucket kvôli Jire
 - Rozpracované metodiky
- Diskusia o projekte
 - Riešenie názvu domény
 - Revízia integrácie projektu
 - Demo na tagovanie obrázkov
 - Testovanie Geo Lokácie
 - Prehodnotenie spôsobu komunikácie členov
 - Vytváranie sprint backlogu
- Naplánovanie ďalšieho šprintu
 - Pridanie novej funkcionality do aplikácie
 - Výpočet času, ktorý strávil pracovník plnením úloh
 - Vytvorenie filtrov pre pracovníkov
 - Vytvorenie kreditového systému pre pracovníkov
 - UI webovej aplikácie
 - Vytvorenie profilovej stránky pracovníka
 - registrácia pracovníkov cez sociálne siete
 - Migrácia na Bitbucket
 - Vytvorenie achievement systemu
 - Pridanie referral odkazov pre pracovníkov

Stav úloh z tretieho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
3.1	Usporiadanie produktového backlogu	Martin, Patrik	28.10.2013	03.11.2013	Splnená
3.2	Spísanie metodík	Spoločná	28.10.2013	03.11.2013	Riešená
3.3	Vytvorenie tímového kalendáru	Robo	28.10.2013	03.11.2013	Splnená
3.4	Spustenie produkčného servera	Patrik	28.10.2013	03.11.2013	Splnená
3.5	Revízia doterajších zápisníc	Marek	28.10.2013	03.11.2013	Splnená
3.6	Vytvorenie formulára pre vyplnenie profilu	Miro M.	28.10.2013	03.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Vytvorenie filtrov a kreditového systému	Filip	04.11.2013	10.11.2013	Riešená
4.2	Vytvorenie UI dizajnu aplikácie	Robo	04.11.2013	10.11.2013	Riešená
4.3	Vytvorenie referral odkazov	Marek	04.11.2013	10.11.2013	Riešená
4.4	Vytvorenie profilovej stránky	Miro M.	04.11.2013	10.11.2013	Riešená
4.5	Registrácie cez soc.siete + Bitbucket	Patrik	04.11.2013	10.11.2013	Riešená
4.6	Registrácie cez soc.siete + vytvorenie subtaskov	Martin	04.11.2013	10.11.2013	Riešená
4.7	Vytvorenie achievment systému	Miro V.	04.11.2013	10.11.2013	Riešená

3.7 Zápis zo stretnutia č. 6

Zápisnica č. 6.

Téma Plánovanie 5. šprintu	Dátum 11.11.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Filip Bednárík	

Priebeh stretnutia

- Vyhodnotenie štvrtého šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
- Diskusia o projekte
 - Riešenie achievement badges
 - Identifikovanie atribútov profilu
 - Skupiny používateľov
- Naplánovanie ďalšieho šprintu
 - Integrácia s LinkedIn a Facebook - údaje do profilu
 - Dokončenie dokumentácie
 - Vymyslieť achievementy
 - Integrovanie s ďalšími sociálnymi sieťami - Google Plus, Twitter

Stav úloh zo štvrtého sprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
4.1	Vytvorenie filtrov a kreditového systému	Filip	04.11.2013	10.11.2013	Splnená
4.2	Vytvorenie UI dizajnu aplikácie	Robo	04.11.2013	10.11.2013	Splnená
4.3	Vytvorenie referral odkazov	Marek	04.11.2013	10.11.2013	Riešená
4.4	Vytvorenie profilovej stránky	Miro M.	04.11.2013	10.11.2013	Splnená
4.5	Registrácie cez soc.siete + Bitbucket	Patrik	04.11.2013	10.11.2013	Splnená
4.6	Registrácie cez soc.siete + vytvorenie subtaskov	Martin	04.11.2013	10.11.2013	Splnená
4.7	Vytvorenie achievment systému	Miro V.	04.11.2013	10.11.2013	Riešená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	Vytváranie dokumentácie	Spoločná	11.11.2013	17.11.2013	Riešená
5.2	Vytvorenie UI dizajnu aplikácie	Robo	11.11.2013	17.11.2013	Riešená
5.3	Integrácia s Google účtom + referral odkazy dokončiť	Marek	11.11.2013	17.11.2013	Riešená
5.4	Integrácia s Twitter účtom	Miro M.	11.11.2013	17.11.2013	Riešená
5.5	Facebook lajky a komentáre, zmena profilu používateľa	Patrik, Martin	11.11.2013	17.11.2013	Riešená
5.6	Vymyslieť, implementovať achievmenty - merit gem	Miro V.	11.11.2013	17.11.2013	Riešená

3.8 Zápis zo stretnutia č. 7

Zápisnica č. 7.

Téma Plánovanie 6. šprintu	Dátum 18.11.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Patrik Štrba	

Priebeh stretnutia

- Vyhodnotenie piateho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Prezentácia metodík
 - Diskusia a pripomienky k metodikám
 - Potreba rebrandingu pre nasadenie produkčnej verzie
 - Diskusia k dokumentácii, a častiam potrebným pre vypracovanie
- Naplánovanie ďalšieho šprintu
 - Potreba zmeniť vytváranie úloh na viackrokové formuláre
 - Spraviť rebranding produktu - zmena názvu, loga
 - Prerobenie tímového webu popri rebrandingu
 - Zmeniť organizáciu údajov v profile
 - Možnosť komentovania a lajkovania Youtube videa
 - Použitie ďalších informácií z LinkedIn profilu
 - Lokalizácia aplikácie do viacerých jazykov

Stav úloh z piateho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
5.1	Vytváranie dokumentácie	Spoločná	11.11.2013	17.11.2013	Riešená
5.2	Vytvorenie UI dizajnu aplikácie	Robo	11.11.2013	17.11.2013	Splnená
5.3	Integrácia s Google účtom + referral odkazy dokončiť	Marek	11.11.2013	17.11.2013	Splnená
5.4	Integrácia s Twitter účtom	Miro M.	11.11.2013	17.11.2013	Riešená
5.5	Facebook lajky a komentáre, zmena profilu používateľa	Patrik, Martin	11.11.2013	17.11.2013	Splnená
5.6	Vymyslieť, implementovať achievements - merit gem	Miro V.	11.11.2013	17.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Vytvorenie viackrokových formulárov pre zadanie úlohy	Filip	18.11.2013	24.11.2013	Riešená
6.2	Home, práva na stránky	Robo	18.11.2013	25.11.2013	Riešená
6.3	Spojenie dokumentácie	Marek	18.11.2013	24.11.2013	Riešená
6.4	Youtube videá - like, komentár, možnosť pozretia celého videa	Miro M.	18.11.2013	24.11.2013	Riešená
6.5	Údaje z LinkedIn - vzdelanie, práca, zručnosti	Patrik	18.11.2013	24.11.2013	Riešená
6.6	Rebranding - web tímu, dizajn	Marek, Martin	18.11.2013	24.11.2013	Riešená
6.7	Zmena organizácie profilu, dizajn profilu	Miro V.	18.11.2013	24.11.2013	Riešená

3.9 Zápis zo stretnutia č. 8

Zápisnica č. 8.

Téma Plánovanie 7. šprintu	Dátum 25.11.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Martin Toma	

Priebeh stretnutia

- Vyhodnotenie šiesteho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Diskusia k podstatným častiam aplikácie
 - Ochrana osobných údajov používateľov v systéme
 - Definovanie nových typov úloh v systéme, a ich reálne použitie
 - Iteratívna komunikácia zadávateľa s pracovníkom o úlohe
 - Zameranie sa na vytvorenie funkčného scenára
 - Vytvorenie úloh pre použitie v primárnom prostredí - FIIT STU
 - Obmedzenie vytváraných úloh na dotazníky
 - Diskusia k dokumentácii, čo chýba, čo treba dorobiť
 - Rozdelenie používateľov do skupín v rámci systému
- Naplánovanie ďalšieho šprintu
 - Vytvorenie prezentačných materiálov aplikácie - plagát
 - Vytvorenie landing page pre aplikáciu
 - Lokalizácia do viacerých jazykov

Stav úloh zo šiesteho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
6.1	Vytvorenie viackrokových formulárov pre zadanie úlohy	Filip	18.11.2013	24.11.2013	Splnená
6.2	Home, práva na stránky	Robo	18.11.2013	25.11.2013	Splnená
6.3	Spojenie dokumentácie	Marek	18.11.2013	24.11.2013	Splnená
6.4	Youtube videá - like, komentár, možnosť pozretia celého videa	Miro M.	18.11.2013	24.11.2013	Riešená
6.5	Údaje z LinkedIn - vzdelanie, práca, zručnosti	Patrik	18.11.2013	24.11.2013	Splnená
6.6	Rebranding - web tímu, dizajn	Marek, Martin	18.11.2013	24.11.2013	Splnená
6.7	Zmena organizácie profilu, dizajn profilu	Miro V.	18.11.2013	24.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Logovanie cez routa, id úloh/id pracovník tag	Filip	25.11.2013	01.12.2013	Riešená
7.2	Vytvorenie high chart grafov pre zobrazenie výsledkov úlohy	Robo	25.11.2013	01.12.2013	Riešená
7.3	Vytvorenie landing page obrázkov a tímového posteru	Marek	25.11.2013	01.12.2013	Riešená
7.4	Preklad lokalizačných súborov	Miro M.	25.11.2013	01.12.2013	Riešená
7.5	Vytvorenie landing page	Patrik, Martin	25.11.2013	01.12.2013	Riešená
7.6	Odmeny za vyplnenie profilu, vytvorenie reorganizovaného profilu	Miro V.	25.11.2013	01.12.2013	Riešená

3.10 Zápis zo stretnutia č. 9

Zápisnica č. 9.

Téma Plánovanie 8. šprintu	Dátum 02.12.2013	Čas 18:00	Miesto Jobsovo softvérové štúdio
Prítomní členovia		Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš	
Vedúci stretnutia		Ing. Dušan Zeleník	
Zapisovateľ		Bc. Miroslav Molnár	

Priebeh stretnutia

- Vyhodnotenie siedmeho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Diskusia k podstatným častiam aplikácie
 - Potrebné vytvoriť logovanie
 - Identifikovanie ďalších typov úloh
 - Špeciálne domény aplikácie, namiesto viacerých typov úloh
 - Refactoring kódu
 - Vytvorenie Pie grafov
- Naplánovanie ďalšieho šprintu
 - Tvorba dokumentácie, dokumentovanie dorobených funkcionalít
 - Kontrola dokumentácie na chyby, nekonzistentnosť
 - Vytvorenie testov pre funkcionality
 - Cucumber test framework

Stav úloh zo siedmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
7.1	Logovanie cez routa, id úloh/id pracovník tag	Filip	25.11.2013	01.12.2013	Splnená
7.2	Vytvorenie high chart grafov pre zobrazenie výsledkov úlohy	Robo	25.11.2013	01.12.2013	Splnená
7.3	Vytvorenie landing page obrázkov a tímového posteru	Marek	25.11.2013	01.12.2013	Splnená
7.4	Preklad lokalizačných súborov	Miro M.	25.11.2013	01.12.2013	Splnená
7.5	Vytvorenie landing page	Patrik, Martin	25.11.2013	01.12.2013	Splnená
7.6	Odmeny za vyplnenie profilu, vytvorenie reorganizovaného profilu	Miro V.	25.11.2013	01.12.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
8.1	Refactoring kódu	Filip	02.11.2013	08.11.2013	Riešená
8.2	Vytvorenie ďalších high chartov	Robo	02.11.2013	08.11.2013	Riešená
8.3	Tvorba dokumentácie, kontrola chýb, nekonzistencií, definovanie obsahu	Marek	02.11.2013	08.11.2013	Riešená
8.4	Upravenie zobrazenie badges, vymyslieť nové badges	Miro M.	02.11.2013	08.11.2013	Riešená
8.5	Fazetta filter pre filtrovanie úloh, výsledkov, atď.	Patrik	02.11.2013	08.11.2013	Riešená
8.6	Integrácia cucumber test frameworku, grafy výsledkov	Martin	02.11.2013	08.11.2013	Riešená
8.7	Vytvorenie novej verzie profilu	Miro V.	02.11.2013	08.11.2013	Riešená

3.11 Zápis zo stretnutia č. 10

Zápisnica č. 10.

Téma	Dátum	Čas	Miesto
Záverečné stretnutie	09.12.2013	18:00	Jobsovo softvérové štúdio
Prítomní členovia	Bc. Filip Bednárík Bc. Róbert Černý Bc. Marek Lenčes Bc. Miroslav Molnár Bc. Patrik Štrba Bc. Martin Toma Bc. Miroslav Vojtuš		
Vedúci stretnutia	Ing. Dušan Zeleník		
Zapisovateľ	Bc. Marek Lenčes		

Priebeh stretnutia

- Vyhodnotenie ôsmeho šprintu
 - Vyhodnotenie splnenia zadaných úloh
 - Diskusia o problémoch pri šprinte
 - Zhodnotenie doterajšej práce
 - Pripomienky k zlepšeniu fungovania tímu
 - Predvedenie splnených úloh a zhodnotenie úplnosti
- Diskusia o projekte
 - Dokončenie dokumentácie do odovzdania

Stav úloh z ôsmeho šprintu

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
8.1	Refactoring kódu	Filip	02.11.2013	08.11.2013	Splnená
8.2	Vytvorenie ďalších high chartov	Robo	02.11.2013	08.11.2013	Splnená
8.3	Tvorba dokumentácie, kontrola chýb, nekonzistencií, definovanie obsahu	Marek	02.11.2013	08.11.2013	Splnená
8.4	Upravenie zobrazenie badges, vymyslieť nové badges	Miro M.	02.11.2013	08.11.2013	Splnená
8.5	Fazetta filter pre filtrovanie úloh, výsledkov, atď.	Patrik	02.11.2013	08.11.2013	Splnená
8.6	Integrácia cucumber test frameworku, grafy výsledkov	Martin	02.11.2013	08.11.2013	Splnená
8.7	Vytvorenie novej verzie profilu	Miro V.	02.11.2013	08.11.2013	Splnená

Rozdelenie úloh

ID	Popis úlohy	Pridelené členovi	Dátum zadania	Predpokl. ukončenie	Stav
9.1	Dokončenie dokumentácie	Spoločná	09.12.2013	12.12.2013	Riešená

4 Úlohy členů týmu

4.1 Manažerské role v týmu

Manažerské role v týmu jsme si rozdělili při prvním šprintu, každý člen týmu má při tvorbě projektu hlavní manažerskou zodpovědnost za fungování dané části vývoje produktu.

Meno	Zodpovednosť
Bc. Martin Toma	Manažér komunikácie a ľudských zdrojov (vedúci tímu)
Bc. Filip Bednárik	Manažér rizík
Bc. Róbert Černý	Manažér rozvrhu a plánovania
Bc. Marek Lenčes	Manažér dokumentovania
Bc. Miroslav Molnár	Manažér monitorovania
Bc. Miroslav Vojtuš	Manažér kvality
Bc. Patrik Štrba	Manažment podpory vývoja

4.2 Krátkodobé úlohy

Meno	Úloha
Bc. Martin Toma	Manažér komunikácie a ľudských zdrojov (vedúci tímu)
Bc. Filip Bednárik	Manažér rizík
Bc. Róbert Černý	Manažér rozvrhu a plánovania
Bc. Marek Lenčes	Manažér dokumentovania
Bc. Miroslav Molnár	Manažér monitorovania
Bc. Miroslav Vojtuš	Manažér kvality
Bc. Patrik Štrba	Manažment podpory vývoja

4.3 Dlhodobé úlohy

Meno	Zodpovednosť
Bc. Martin Toma	Zodpovedá za riadenie a usmerňovanie komunikácie v tíme. Rovnako má za úlohu sprostredkovať komunikáciu medzi zadávateľom projektu a tímom. Okrem toho dohliada nad nástrojom JIRA do ktorého po každom stretnutí zapíše a prideli jednotlivé úlohy. Má prehľad o celom systéme a rieši vysoko úrovňové problémy vrámci celej produktovej domény.
Bc. Filip Bednárik	Má úlohu hlavného inšpektora pri prehliadkach zdrojového kódu. Je softvérovým architektom a stará sa o integritu softvérového riešenia. Tiež má za úlohu identifikovať včas riziká a s pomocou tímu korektne na tieto riziká reagovať. Popri tom písať zápisnice a podieľať sa na identifikovaní a implementácii nových funkcií softvéru.
Bc. Róbert Černý	Má za úlohu dozerat na správnosť priebehu tímového stretnutia. Kontroluje termíny a dozerá na ich dodržiavanie. Navrhne dátumy a synchronizuje ich so zdieľaným tímovým kalendárom. Dozerá na plnenie krátkodobého a dlhodobého plánu.
Bc. Marek Lenčes	Má za úlohu spracovávať výslednú podobu dokumentácie, skontrolovať nekonzistentné označovanie rovnakých pojmov, opraviť gramatické chyby, spracovať výsledné formátovanie dokumentu, a zabezpečiť, aby obsahoval všetky náležitosti potrebné pri jeho odovzdaní. Tiež má za úlohu spracovávať zápisnice zo stretnutí, kontrolovať gramatické chyby, zabezpečiť pravdivosť údajov v zápisnici, a aktualizovať tímový web pravidelným pridávaním zápisníc a aktuálnych verzií obidvoch častí dokumentácie na stránku tímového webu.
Bc. Miroslav Molnár	Zabezpečenie monitorovania aktuálneho stavu projektu voči dlhodobému plánu projektu. Informovať členov tímu o stave projektu. Navrhnuť optimálne riešenia pri zmene plánu z hľadiska nazbieraných informácií z predchádzajúceho monitorovania.
Bc. Miroslav Vojtuš	Zabezpečiť kvalitu produktu podľa požiadaviek. Robí preventívne testy funkčných častí systému. Rozhoduje o zavedení a hĺbke automatizovaného testovaní.
Bc. Patrik Štrba	Správa spoločného repozitára a procesu nasadenia projektu na produkčný server. Kontrola pull-requestov a zabezpečovanie korektnosti dát v repozitári. Vytváranie master vetvy pre produkčný server a vytváranie nových verzií produktu.

4.4 Podiel členov na tvorbe dokumentácie k inžinierskemu dielu

Meno	Podiel
Bc. Martin Toma	13 %
Bc. Filip Bednárík	13 %
Bc. Róbert Černý	13 %
Bc. Marek Lenčes	22 %
Bc. Miroslav Molnár	13 %
Bc. Patrik Štrba	13 %
Bc. Miroslav Vojtuš	13 %

4.5 Podiel členov pri tvorbe diela

Meno	Podiel
Bc. Martin Toma	14,3 %
Bc. Filip Bednárík	14,3 %
Bc. Róbert Černý	14,3 %
Bc. Marek Lenčes	14,3 %
Bc. Miroslav Molnár	14,3 %
Bc. Patrik Štrba	14,3 %
Bc. Miroslav Vojtuš	14,3 %

5 Manažment rozvrhu a plánovania

Autor: Bc. Róbert Černý

V tejto kapitole sú opísané procesy, ktoré prebiehajú počas plánovania. Náš tím si vybral agilnú metódu vývoja. Používame teda *Scrum*, čo je agilný, prírastkový a iteratívny projektový manažment.

5.1 Procesy

5.1.1 Tvorba a udržiavanie produktového backlogu

Vstup: žiadny alebo nevyhovujúci produktový backlog

Výstup: produktový backlog so všetkými náležitosťami

Produktový backlog vytvára produktový vlastník na základe priorit a požiadaviek zákazníkov. Obsahuje používateľské príbehy, ktorá sa zväčša dajú rozdeliť do jednoduchších podúloh. Forma zápisu týchto príbehov je obsiahnutá v šablóne nižšie. Backlog sa nachádza v systéme *Atlassian JIRA*®, ktorý je vhodný na jeho reprezentáciu.

	Popis	Príklad
ID	Identifikátor, začíname od 0001	0212
Názov	Maximálne jedna veta, stručná, jasná, vyjadrujúca podstatu	Autentifikácia používateľa
Dôležitosť	Určuje produktový vlastník a je to číselná hodnota, čím vyššia, tým vyššia priorita	110
Odhad prácnosti	Slúži na porovnanie s inými príbehmi, ide o odhad, určuje ho spoločne celý tím <i>hodnota = počet ľudí x počet hodín</i>	10
Demonštrovanie	Definujeme, ako je možné daný príbeh prezentovať	Registrujeme sa, autentifikujeme sa pomocou odkazu z e-mailu, prihlásime sa, vstúpime na domovskú stránku
Poznámky	Doplňujúce informácie	Použijeme vhodný GEM

Odhad prácnosti je atribút, ktorý sa odhaduje až pri rozdeľovaní úloh medzi členmi pri tvorbe backlogu šprintu. Počas vývoja sa totižto odhad členov tímu spresňuje, čo zefektívni proces plánovania a dodržiavanie termínov.

5.1.2 Tímové stretnutia a šprinty

Stretnutia prebiehajú opakovane, konkrétny deň v týždni. Jednotlivé šprinty začínajú od ukončenia stretnutia a trvajú až do začatia nasledujúceho stretnutia. Nakoľko nie je možné šprintovať neustále, každý si vyberie ľubovoľný deň v týždni, kedy odдыхuje. Počas šprintu sa predpokladá, že všetci členovia chápu priradené úlohy.

Stretnutia prebiehajú v miestnosti s pripojením na internet a tabuľou pre jednoduchšiu spoluprácu pri vizualizácii. Trvanie tímového stretnutia sa zbytočne nepredlžuje a trvá v priemere hodinu. V tomto časovom intervale prebiehajú postupne podľa poradia tieto procesy:

Revízia šprintu a jeho výsledky	5.1.2.A
Príprava a plánovanie šprintu	5.1.2.B

5.1.2.A Revízia šprintu a jeho výsledky

Vstup: backlog ukončeného šprintu

Výstup: aktualizovaný produktový backlog, úspešnosť šprintu, vyhodnotenie

Týmto procesom začína každé stretnutie a slúži na vyhodnotenie šprintu. Každý člen sa v priebehu pár minút vyjadrí ako sa mu v šprinte darilo. Zhodnotí svoje výsledky a predvedie demo na ktorom preukáže svoju prácu. Vedúci tímu označí úlohu za hotovú ak predvedenie spĺňa požiadavky úlohy. Na konci revízie sa nedokončené úlohy vrátia do produktového backlogu a v prípade potreby sa upraví ich dôležitosť alebo sa zaradia do nasledujúceho šprintu.

Ďalšou súčasťou tohto procesu je prehliadka kódov, ktorá má informatívnu, ako aj kontrolnú funkciu. Pomocou metodiky na prehliadku kódov je možné informovať programátorov o nových algoritmoch či zmenách v implementácií. Metodika písania zdrojových kódov zrýchli tento proces, nakoľko bude mať zdrojový kód uniformnú štruktúru.

Vedúci tímu má na konci priestor pre vznesenie postrehov a problémov ktoré sa vyskytli počas šprintu. Je potrebné aby navrhol riešenie a tím sa začne riadiť podľa týchto pravidiel.

5.1.2.B Príprava a plánovanie šprintu

Prvým krokom v tomto procese je definovať cieľ šprintu. Príkladom môže byť: „dokončenie 3 používateľských príbehov“. Vedúci tímu vyberie člena, ktorý bude zapisovať informácie o stretnutí a šprinte do zápisnice. Zverejnenie zápisnice na tímovom webe je povinnosťou manažéra dokumentácie, ktorý robí poslednú revíziu a opravu zápisnice. Pred rozchodom tímu sa dohodne termín ďalšieho stretnutia. Priestor tu majú hlavne členovia, ktorí pracujú v skupine. Všeobecne však platí, že hlavné stretnutie sa koná presne o týždeň, kedy končí šprint.

Tvorba backlogu šprintu	5.1.2.B.a
Priradenie úloh členom	5.1.2.B.b
Priradenie úloh členom	5.1.2.B.c

5.1.2.B.a Tvorba backlogu šprintu

Vstup: produktový backlog je v poriadku

Výstup: nový backlog nasledujúceho šprintu

V tomto procese sa vytvára backlog pre nastávajúci šprint. Pred začiatkom je však potrebné aby bol produktový backlog aktuálny. Nové úlohy vyberá produktový vlastník, ktorý zároveň komunikuje s tímom o tomto výbere. Členovia tímu spoločne odhadujú, koľko úloh bude možné v jednom šprinte splniť. Tento odhad je subjektívny a najdôležitejšou súčasťou tohto procesu je vzájomná komunikácia. Po výbere úloh je potrebné určiť ich priradenie.

5.1.2.B.b Priradenie úloh členom

Vstup: backlog nasledujúceho šprintu

Výstup: úlohy sú priradené, každý vie na čom pracuje

Jednotlivé úlohy sa v systéme JIRA priradia v priebehu prvého dňa šprintu. Vedúci tímu postupne priradí tieto úlohy priamo na stretnutí. Priradenie sa uchová v zápisnici. Úlohy pridelujeme podľa žiadostí členov. Inak ich pridelí vedúci tímu podľa vlastného uváženia.

5.1.2.B.c Rozdelenie úloh na podúlohy

Vstup: úlohy priradené konkrétnemu členovi (skupine)

Výstup: podúlohy jednotlivých úloh

Tento proces je vhodné použiť v prípade, že priradená úloha je zložitejšia a je možné ju logicky rozdeliť na jednoduchšie časti. V prípade že konkrétnu úlohu majú priradenú viacerí členovia, je vhodné aby si navzájom priradili podúlohy podľa svojho uváženia. Tento proces prebehne priamo na tímovom stretnutí. Výsledkom je priradenie elementárnejších úloh jednotlivým členom tímu, ktoré sú zapísané do zápisnice.

Vedúci tímu sa snaží čo najskôr aktualizovať systém JIRA a nastaviť aktuálnu tabuľu šprintu.

5.1.3 Komunikácia počas šprintu

Počas šprintu sú všetky relevantné informácie o priebehu a stave uvedené v systéme JIRA. Toto je prvé miesto ktoré treba navštíviť v prípade nejasností. Tabuľka šprintu je prístupná pre všetkých a je vždy aktuálna.

Pri objavení novej chyby v zdrojových kódach sa jej reprezentácia pridá do systému JIRA ako nový problém. Samotné priradenie zabezpečuje vedúci tímu.

V prípade implementačných nejasností sa člen tímu primárne obracia na zadávateľa danej úlohy. Ak na úlohe spolupracuje viac členov, komunikácia prebieha priamo medzi nimi.

Komunikačné kanály podľa priority:

- JIRA (komentovanie, priradovanie úloh, tabuľka šprintu)
- Google Hangouts (možnosť konferenčných hovorov)
- Google Groups (vytváranie príspevkov podľa tém)
- Google Calendar (termíny PintaGroup)
- mobilný telefón alebo iný osobný kontakt

Pri problémoch s dodržaním termínov je nutné upozorniť vedúceho tímu, ktorý zasiahne do priebehu šprintu a jeho zmenu zverejní v systéme JIRA, pričom sa odošle upozornenie o zmene formou e-mailu. O tejto zmene upozorní aj produktového vlastníka.

6 Manažment rizík

Autor: Bc. Filip Bednárík

Riziká rozdeľujeme na rôzne typy. Podľa základného delenia na generické a špeci-
fické. Generické riziká sú podobné pri všetkých softvérových projektoch. Vybrali
sme najčastejšie a popísali sme ako tieto generické riziká vplývajú priamo na náš
projekt a ako chceme znížiť tieto riziká na minimum.

Riziko redundantnej práce

Typ rizika: Generické, Organizačné, Zdroje

Pravdepodobnosť: Vysoká

Popis rizika: Pri používaní nových technológií sa často stretávame s nepreskúma-
nými problémami, na ktoré je potrebné hľadať riešenia, ktoré môžu byť časovo
náročné. Je neefektívne ak sa na tom istom probléme zaseknú dvaja vývojári.

Škody a dôsledky rizika: Oneskorenie termínov, Demotivácia a frustrácia zamest-
nancov, Vyššie náklady na vývoj.

Riešenie rizika: Pre zníženie rizika pochybenia plánujeme organizovať pravidelné
stretnutia tímu s prezentáciami problémov, ktoré sa vyskytnú pri vývoji alebo ná-
vrhu aplikácie a tiež o potencionálne vhodných technológiách vhodných pre tento
projekt a existujúcich modulov, tzv. „gemov“.

Nerealistické rozvrhy a rozpočty

Typ rizika: Generické, Manažment projektu, Plánovanie

Pravdepodobnosť: Stredná

Popis rizika: Plány a rozvrhy môžu byť nerealistické, rozpočty môžu byť prehnané
alebo nedostatočné.

Škody a dôsledky rizika: Príliš tesný plán môže spôsobiť: Nesplnenie termínov, De-
motivácia a frustrácia zamestnancov. Naopak príliš voľný plán: Pomalý vývoj,
vyššie náklady, nespokojnosť zákazníkov pri pomalom riešení problémov s apliká-
ciou. Prehnaný rozpočet: Nezískanie investora. Príliš nízky rozpočet: Problémy s
dodatocnými požiadavkami o zvýšenie, nespokojnosť zamestnancov.

Riešenie rizika: Pri plánovaní počítame s rôznymi možnosťami. Vývoj bude agilný
a inkrementálny, a využijeme systém pre správu úloh, do ktorého pravidelne čle-
novia tímu budú vkladať podrobné odhady. Chceme robiť prototypy aplikácie.

Zlý návrh aplikácie.

Typ rizika: Generické, Technické, Kvalita

Pravdepodobnosť: Stredná

Popis rizika: Aplikácia môže mať zlú architektúru, jednotlivé časti zle oddelené
alebo neoddelené vôbec.

Škody a dôsledky rizika: Nekonzistentnosť aplikácie, Redundancia kódu, Opakova-
nie chýb a z toho vyplývajúce vyššie náklady na vývoj.

Riešenie rizika: Jednotlivé časti aplikácie budú vo forme modulov a niektoré mo-
duly budú použiteľné aj osobitne. Modulárnosť nám zabezpečí možnosť využitia
rovnakých modulov v rôznych častiach aplikácie. Na návrhu aplikácie sa bude
podieľať skúsený vývojár.

Vypracovanie nevyhovujúcej funkcionality.

Typ rizika: Generické, Manažment projektu, Komunikácia

Pravdepodobnosť: Vysoká

Popis rizika: Používateľom sa funkcionality nemusí páčiť, môže byť zbytočná alebo v aplikácii dokonca prekážať.

Škody a dôsledky rizika: Úbytok počtu používateľov a teda nižší zisk, Zbytočne vynaložené zdroje na vývoj nepotrebných funkcií, oddialenie vývoja potrebnej funkcionality.

Riešenie rizika: Pred vytváraním novej funkcionality vypracujeme prieskum, či by takáto funkcionality vyhovovala. Máme tiež záujem o overené technológie a funkcionality. Aplikáciu, vzhľadom na to, že beží na našom hardvéri, môžeme aktualizovať a opravovať priebežne a často. Aplikácia bude navrhnutá tak aby bola používateľsky prívetivá a nepotrebovala množstvo dokumentácie, aj keď pomocníka do aplikácie zahrnieme. Používatelia budú mať možnosť odoslať spätnú väzbu priamo v aplikácii.

Vytvorenie nevyhovujúceho používateľského rozhrania.

Typ rizika: Generické, Technické, Zložitosť a rozhrania

Pravdepodobnosť: Stredná

Popis rizika: Pri návrhu aplikácie sa nepočíta s prostredím, kde sa aplikácia bude používať.

Škody a dôsledky rizika: Úbytok používateľov a teda nižšie zisky.

Riešenie rizika: Základom kvalitného používateľského rozhrania je správna charakteristika používateľa. Používateľské rozhranie preto budeme musieť prispôbiť tak, aby bolo vhodné pre technicky nezdatného používateľa ale zároveň zachovať prvky ako skratky pre expertov. Pre všetky spôsoby použitia našej aplikácie preto vypracujeme scenáre, na základe ktorých prispôbíme aj používateľské rozhranie. Vývoj rozhrania bude prebiehať v šablónach a teda v budúcnosti si bude môcť používateľ vybrať používateľské rozhranie, ktoré mu bude vyhovovať.

Neustále zmeny požiadaviek.

Typ rizika: Generické, Externé, Zákazník

Pravdepodobnosť: Nízka

Popis rizika: Zdrojom požiadaviek bude spätná väzba od používateľov vo forme prieskumov ale, zo samotnej aplikácie pomocou formulára ale aj logmi o používaní aplikácie.

Škody a dôsledky rizika: Zvýšené náklady na vývoj, väčšia šanca na vypracovanie nevyhovujúcej funkcionality.

Riešenie rizika: Na základe týchto informácií vytvoríme požiadavky na zmeny aplikácie, ktoré budeme postupne zahŕňať do jednotlivých inkrementov aplikácie. Jednotlivé verzie aplikácie budú vydávané často a nepravidelne podľa požiadaviek. Jednotlivé funkcie však budú pred vydaním otestované, v prípade veľkej funkcionality sprístupníme aj beta verziu produktu.

Nedostatky výkonu v reálnom čase.

Typ rizika: Generické, Technické, Výkon a spoľahlivosť

Pravdepodobnosť: Vysoká

Popis rizika: Je možné, že zvyšujúcim množstvom používateľov nebude stačiť výkon serverov. Niektorí používatelia môžu mať problém s pomalým pripojením alebo s veľkosťou obrázkov a skriptov.

Škody a dôsledky rizika: Spomalenie aplikácie, nespokojnosť používateľov, odrezanie počtu používateľov od funkčnosti. Spomalenie alebo nedostupnosť serveru. Úbytok používateľov a straty v ziskoch.

Riešenie rizika: Pravidelné benchmarky aplikácie a záťažové testy serverov. Kvalitné logovanie akcií používateľa v našej aplikácii nám umožní lepšie odhaliť problémy s jednotlivými modulmi a tieto problémy rýchlo odstrániť. Naša aplikácia bude mať rôzne nastavenia a režimy, v ktorých môže fungovať, čím chceme zabezpečiť pohodlné používanie pre rôzne typy používateľov.

Nedostatok zadávateľov úloh.

Typ rizika: Špecifické, Externé, Zadávatel

Pravdepodobnosť: Vysoká

Popis rizika: Môže sa stať, že nebude dostatočný záujem zo strany zadávateľov úloh, alebo zadávateľa nebudú potrebovať použiť náš systém na zadanie úlohy, pretože nebudú mať potrebu riešiť svoju úlohu externým spôsobom.

Škody a dôsledky rizika: Nedostatok úloh. Nespokojnosť používateľov. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov a straty v ziskoch.

Riešenie rizika: Pre používanie aplikácie musíme zadávateľov dostatočne motivovať. Podstatou je dostatočne oboznámiť potencionálnych zadávateľov s funkcionalitou a princípom zadávania úloh. Dôležité je tiež zdôraznenie výhod, ktoré zadávateľ získa, ak úlohu do systému zadá. Podstatná časť je teda vytvorenie reklamy a prezentovanie aplikácie rôznymi spôsobmi, a zdôraznenie jej predností.

Nedostatok záujmu pracovníkov.

Typ rizika: Špecifické, Externé, Zákazník

Pravdepodobnosť: Stredná

Popis rizika: Ak nastane situácia, že zadávateľov aj úloh bude dostatok, ale nebude dostatok pracovníkov na riešenie úloh, nastáva problém. Pracovníkom sa odmeny môžu zdať príliš nízke, za ktoré sa im neoplatí danú úlohu vypracovať.

Škody a dôsledky rizika: Odškodnenie zadávateľov. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov(zadávateľov) a straty v ziskoch.

Riešenie rizika: Pre čo najväčšie motivovanie pracovníka je potrebné udržať pracovníka v napätí, a v túžbe po vyššom zisku. Preto je riešením vytvorenie odmeňovacieho systému, v ktorom bude pracovník postupne napredovať na vyššie pozície, čím bude mať možnosť vypracovávať úlohy za väčšie odmeny. Riešením je tiež vytvorenie zaujímavých funkcionalít, ktoré spestria pracovníkovi používanie aplikácie, a umožnia mu získať odmenu aj za elementárne činnosti. napr. získanie nového člena, prihlásenie cez sociálnu sieť, a iné.

Neschopnosť validovať splnenie úloh.

Typ rizika: Špecifické, Technické, Kvalita

Pravdepodobnosť: Vysoká

Popis rizika: V systéme poskytujeme viacero typov úloh, tieto typy je možné aj

rozšířit o další typy. Při každé úloze je potřebné validovat, či pracovník splnil základné požiadavky pre vypracovanie úlohy. Ak nevieme skontrolovať a verifikovať údaje zadané pracovníkom, predstavuje to vážne riziko.

Škody a dôsledky rizika: Nespokojnosť zadávateľov. Vypracované úlohy nevyhovujú podmienkam vypracovania. Odmena za úlohu bola vyplatená neprávom, práca nebola dostatočná. Narastajúca nedôveryhodnosť aplikácie. Neschopnosť ďalej prevádzkovať aplikáciu. Úbytok používateľov(zadávateľov) a straty v ziskoch.

Riešenie rizika: Na to, aby sme vedeli úlohy správne validovať, musíme mať detailne dekomponovaný každý typ úlohy, a zachytiť podstatu pri vypracovávaní úlohy. Je potrebné, aby sme vytvorili dostačujúce algoritmy, ktoré budú spoľahlivo validovať základné splnenie požiadaviek pri vypracovaní úloh pracovníkmi.

Nekonzistentnosť pri zadávaní úloh.

Typ rizika: Špecifické, Technické, Kvalita

Pravdepodobnosť: Stredná

Popis rizika: Každá úloha má pri vytváraní svoje špecifické rozhranie, podľa svojho typu. Ak sa zadávateľ pokúsi vložiť nekompatibilné údaje do systému, nastane problém, a aplikácia sa môže správať neočakávane.

Škody a dôsledky rizika: Nespokojnosť zadávateľov. Narušená integrita aplikácie. Narastajúca nedôveryhodnosť aplikácie. Úbytok používateľov(zadávateľov) a straty v ziskoch.

Riešenie rizika: Pre vyriešenie problému nekonzistentnosti údajov je potrebné presne špecifikovať formát údajov prijateľných aplikáciou. Dôležité je tiež ošetrovanie všetkých možných chybových vstupov, ktoré môžu pri zadávaní úlohy nastať. Je potrebné identifikovať všetky prípady, v ktorých môže nastať narušenie integrity aplikácie a tieto prípady ošetriť.

Zhrnutie rizík

Názov rizika	Typ rizika	Pravdepodobnosť	Stručné riešenie
Redundantná práca	Generické	Vysoká	Stretnutia vývojárov
Nerealistické rozvrhy a rozpočty	Generické	Stredná	Agilný vývoj, Nástroj pre projektové plánovanie, Beta verzie
Zlý návrh aplikácie	Generické	Stredná	Modulárnosť, Skúsenejší architekt
Nevyhovujúca funkcionálnosť	Generické	Vysoká	Prieskumy a spätná väzba z aplikácie, Časté aktualizácie, Rozdelenie aplikácie.
Nevyhovujúce UI	Generické	Stredná	UI na základe scenárov použitia
Zmeny požiadaviek	Generické	Nízka	Inkrementálny vývoj, Testovanie, Beta testy
Pochybenie zdroju dát	Špecifické	Stredná	Viac zdrojov a prekrývanie, Komunikácia
Nedostatky výkonu	Generické	Vysoká	Modulárnosť aplikácie, Benchmarky, Rôzne režimy aplikácie, Zátťažové testy
Nedostatok zadávateľov úloh	Špecifické	Vysoká	Motivácia zadávateľov, Reklama, Prezentácia
Nedostatok záujmu pracovníkov	Špecifické	Stredná	Motivácia pracovníkov, Odmeny, Sociálne siete
Neschopnosť validovať splnenie úloh	Špecifické	Vysoká	Dekompozícia, Kvalitné algoritmy
Nekonzistentnosť pri zadávaní úloh	Špecifické	Stredná	Špecifikácia formátov, Ošetrovanie chýb

7 Manažment monitorovania projektu

Autor: Bc. Miroslav Molnár

Monitorovanie projektu predstavuje neustále kontrolovanie aktuálneho stavu projektu voči plánovanému cieľu. Pre presnejšie monitorovanie sa využíva projektový plán stanovený na začiatku projektu. Faktory vplývajúce na monitorovanie sú čas, náklady, kvalita a riziká. Hlavná charakteristika týchto faktorov je, že sa navzájom ovplyvňujú. Ak sa kvalita zvýši, predĺži sa spotrebovaný čas a náklady na projekte. Naopak sa znížia možné riziká nekvalitného návrhu. Z toho dôvodu je rýchlosť vývoja dynamická. Monitorovanie nám prináša informácie o aktuálnom stave a tým umožňuje prispôbiť postup vývoja k úspešnému dosiahnutiu cieľa projektu. Zodpovednosť za monitorovanie projektu má manažér monitorovania spolu s vedúcim tímom. Pre získanie kvalitných a aktuálnych výsledkov monitorovania je potrebné, aby všetci členovia tímu dodržiavali základné postupy pri tvorbe vývoja a vypracovávaní úloh popísané bližšie v metodike „Metodika – Zaznamenávanie práce a sledovanie postupu“.

7.1 Navrhovaný prístup

Kontrola aktuálneho stavu projektu

Kontrola stavu projektu počas jeho vývoja a overovanie dokončenia a aktuálneho stavu úloh.

stav úloh - vyhodnotenie aktuálneho výsledku úlohy a zistenie jej správnosti s požiadavkami zadanými pri jej vytvorení.

spotreba času - kontrola spotrebovaného času na úkoly u jednotlivcov ale aj u celého tímu.

úprava harmonogramu - riadenie zmien spôsobených neočakávanými problémami počas vývoja.

projektový plán - pri výskyte problémov treba upraviť a prispôbiť k úspešnému dosiahnutiu cieľa projektu. Problémy vplývajúce na zmenu projektového plánu:

- výskyt problému, ktorý nebol detekovaný pri tvorbe projektového plánu a teraz ho treba do neho zaradiť,
- zmena zamerania člena tímu na inú oblasť. Schopnosti člena tímu sa viac hodia na inú problematiku, alebo sa člen tímu špecializoval na inú problematiku. Takéto preorganizovanie zdrojov je veľmi nebezpečné, ale ak je dobre premyslené, prinesie efektívne a kvalitné výsledky pri riešení úloh. Manažér monitorovania je často prvý, ktorý má dostatok informácií na to, aby vedel odporučiť vedúcemu tímu správnu preorganizáciu zodpovednosti členov.

Typy monitorovania

Monitorovanie projektu delíme podľa stavu dokončenia projektu na:

- priebežné - monitorovanie prebieha počas vývoja projektu. Monitoruje sa aktuálny stav projektu v porovnaní s projektovým plánom. Priebežné monitorovanie prebieha neustále a vykonáva ho manažér monitorovania s hlavným vedúcim tímom. Hlavné monitorovanie činností členov tímu a stavu projektu prebieha na stretnutiach tímu po každom šprinte. Na týchto stretnutiach

sa kontrolujú výsledky jednotlivcov, ale aj celého tímu. Rozdeľujú sa ďalšie úlohy z plánu. Ak vznikli nejaké problémy, celý tím ich zhodnotí a spoločne sa dohodnú na možnostiach ich riešenia. Problémy sú následne zahrnuté do celkového plánu projektu,

- záverečné - zahŕňa zhodnotenie priebehu vývoja projektu. Monitorovanie zhodnotí prácu jednotlivcov a celého tímu. Detegujú sa chybné postupy vývoja. Problémy, ktoré vznikli behom vývoja sú prehodnotené a hľadajú sa príčiny ich vzniku. Nad aktuálnym stavom projektu sa urobí kompletný monitoring. Prejde sa zoznam úloh, ktoré neboli uskutočnené a prehodnotí sa ich význam. Naplánuje sa nový šprint do ktorého sa nevyriešené úlohy zahrnú,
- následné - po nasadení projektu monitorovanie pokračuje. Aktuálny stav projektu sa porovnáva s novým plánom zhotoveným pred nasadením projektu. Monitorovacie techniky sa nemenia. Mení sa dynamickejšie plán, ktorý sa tvorí z prevažne úloh typu „chyba“, alebo rozšírenia funkcionality projektu. Monitorovanie tejto časti je zložitejšie. Dôvodom je vysoká priorita chýb, ktoré musia byť riešené hneď po ich detekcii v projekte. Vysokú priority úloh typu „chyba“ vytvára fakt, že systém je používaný používateľmi, zatial čo pri vývoji s projektom pracovali iba členovia tímu.

7.2 Aplikácia prístupu

Pre monitorovanie práce jednotlivcov a vyhodnocovanie práce celého tímu je navrhnutý prístup, ktorý sa zaviedol aj do praxe. Pre rozdeľovanie, pridelenie a monitorovanie stavu pridelených úloh členom tímu sme použili systém JIRA.

7.2.1 Dokumentovanie práce

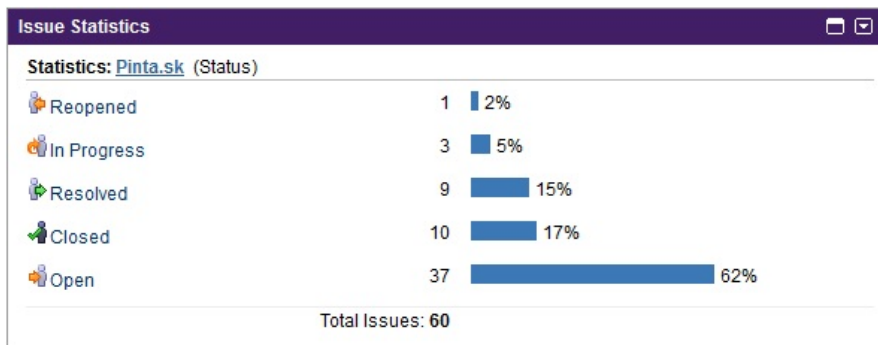
Každý člen tímu je povinný zdokumentovať v systéme JIRA prácu, ktorú vykonal na úlohe. Členovia tímu sú na základe funkcionality ktorú im JIRA prináša schopní monitorovať sami seba. Dokumentácia je potrebná pri odhalovaní zdroja chýb, modifikácii programu. Je treba zachytiť všetky kľúčové fakty, ktoré budú v budúcnosti zúročené pri modifikácii projektu. Dokumentácia je veľmi dobrým zdrojom monitorovania práce členov tímu. Pri väčšom množstve dokumentov je možné k úlohe pripojiť súbory, odkazy a komentáre. Komentáre k úlohe môžu pridávať aj iní členovia tímu. Všetky tieto informácie evidujú nielen aktuálny stav úlohy, ale aj jej priebeh vypracovania.

7.2.2 Vyhodnotenie stavu projektu

Stav projektu je možné vyhodnocovať aj priebežne z aktuálneho stavu zaznamenanom v systéme JIRA. Jednotlivé informácie pomáhajú nielen manažérovi monitorovania a hlavnému vedúcemu tímu, ale aj všetkým ostatným členom tímu získať aktuálny stav projektu, ale aj stav ich samých v rámci tímu. Najrýchlejšie zobrazenie a pochopenie veľkého množstva údajov je grafickým spôsobom. Pomocou grafického zobrazenia sa môže zobrazit veľké množstvo dát. Systém JIRA umožní členom tímu zobraziť viaceré informácie pomocou grafu.

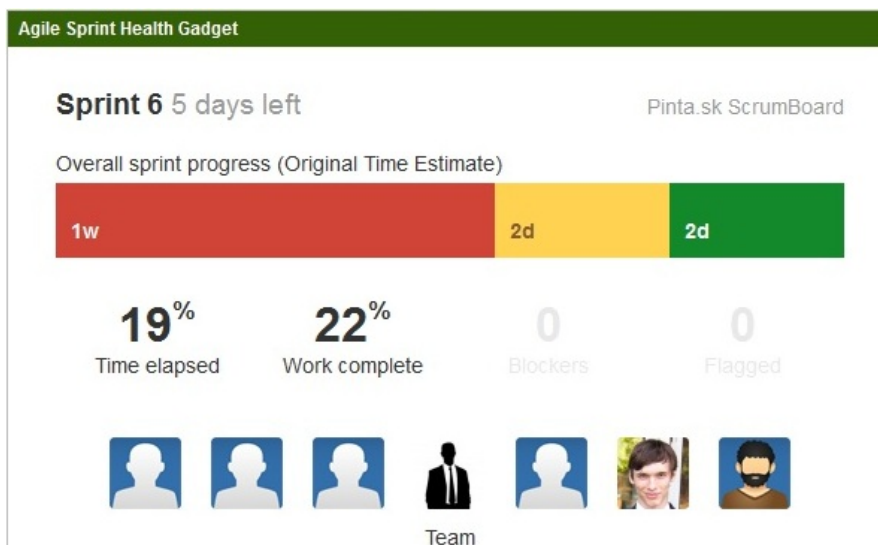
Pre zobrazenie štatistiky vytvorených úloh slúži graf zobrazený na Obrázku 1. Graf triedi úlohy do skupín podľa ich aktuálneho stavu. Manažér monitorovania z grafu deteguje aktuálny stav úloh a ich množstvo v projekte. Z grafu manažér

monitorovania neodhaduje čas potrebný na spravenie všetkých úloh, lebo úlohy majú rôznu dĺžku trvania nezohľadnenú v grafe.



Obr. 1: Stav úloh v systéme.

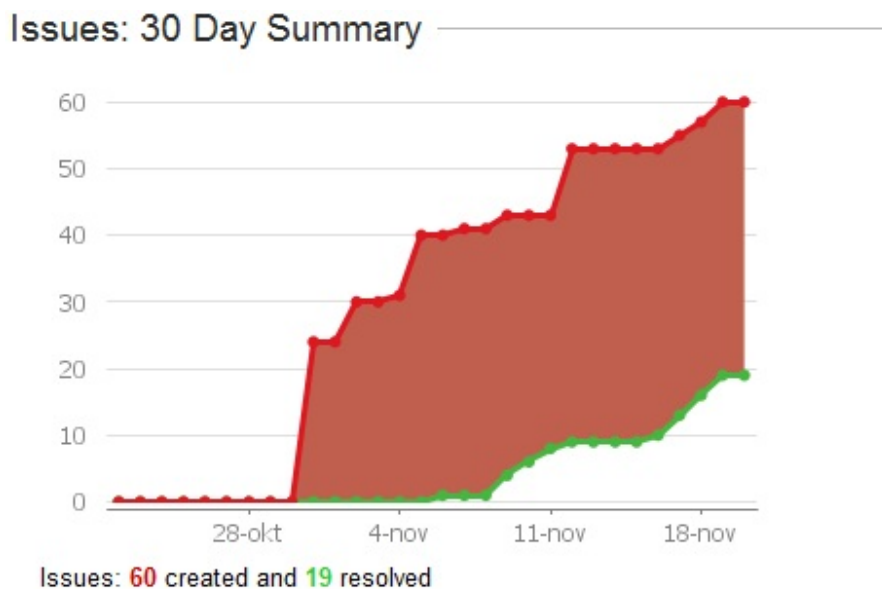
Stav úloh je možné monitorovať aj v rôznych časových intervaloch. V našom projekte sa tím stretáva každý týždeň a preto máme nastavený sedem dňový šprint. Dole nižšie je zobrazený graficky aktuálny stav úloh rozdelených v jednom šprinte. Vidíme koľko z úloh je spravených a koľko sa práve robí a nakoľko sa ešte nezačalo robiť. Manažér monitorovania má takto aktuálny stav úloh kedykoľvek potrebuje. Bližšie Obrázok 2.



Obr. 2: Stav úloh počas jedného šprintu.

Je však dôležité aby všetci členovia tímu dodržiavali pravidlá práce so systémom JIRA, ktoré sú bližšie popísané v „Metodika – Zaznamenávanie práce a sledovanie postupu“.

Pre monitorovanie celkovej správy úloh nám slúži graf na Obrázku 3, ktorý popisuje aktuálny stav úloh v systéme. Na tomto grafe je jasne vidieť novovzniknuté úlohy. Na začiatku sa pre lepšiu prehľadnosť a orientáciu v monitorovaní aktuálneho stavu zvolia iba hlavné úlohy. Tie sa následne delia na podúlohy.



Obr. 3: Stav úloh v projekte

8 Manažment komunikácie a ľudských zdrojov

Autor: Bc. Martin Toma

V tejto kapitole popisujeme procesy spojené s riadením komunikácie a s manažmentom ľudských zdrojov. Vedúci tímu má teda za úlohu pracovať ako spojka medzi produktovým vlastníkom (zadávatelom projektu) a zvyškom tímu. Jeho úlohou je rozdeľovať úlohy a agilne reagovať na vzniknuté problémy či požiadavky z oboch strán.

Hlavným pracovným nástrojom pri týchto procesoch je JIRA, softvér na evidenciu a sledovanie úloh v projektoch (v našom prípade vo verzii 5.1) od spoločnosti Atlassian. Okrem sledovania úloh ponúka tento nástroj možnosti ako sledovať postup práce na konkrétnom šprinte a zároveň sa dá využiť ako miesto na komunikáciu špecificky určenú pre konkrétnu pridelenú úlohu.

Vedúci tímu sleduje priebeh aktuálneho šprintu, odhadovaný zostávajúci čas prác a iné metriky, ktoré mu umožňujú vytvoriť si obraz o fungovaní tímu. Okrem toho, spostredkováva komunikáciu medzi členmi tímu a snaží sa aby bol celý tím vrátane zadávateľa projektu v obraze. Vedúci tímu je zároveň vedúcim projektu v kontexte nástroja JIRA. Jeho úlohy súvisiace s týmto nástrojom sú:

- Udržiavanie a aktualizácia projektových metadát, verzií a komponentov,
- správa, evidovanie a sledovanie všetkých úloh, primárne však tých ktoré nepatria pod konkrétny komponent alebo neboli správne priradené,
- náprava chybné špecifikovaných atribútov a zabezpečenie konzistencie v rámci projektu.

Na to aby mohol vedúci tímu túto prácu vykonávať je potrebná aktivita aj ostatných členov tímu. Čo sa týka nástroja JIRA, medzi ich úlohy patrí primárne:

- Participácia na identifikovaní všetkých chýb v zdrojovom kóde a ich zaevidovanie v systéme,

- participácia na identifikovaní nových funkcionalít a ich evidencia v systéme,
- správa, evidovanie a sledovanie všetkých úloh ktoré patria pod môj komponent,
- aktualizácia stavu (aj zostávajúceho času) mne pridelených úloh v systéme (špeciálne pri úlohách na aktuálny šprint (Metodika SCRUM)).

Vedúci tímu má povinnosť byť dostupný na komunikačných kanáloch ktoré tím najčastejšie používa. V našom prípade ide o Google Hangouts, e-mailovú komunikáciu a nástroj JIRA. Vedúci tímu má prehľad o tom, kto čo robil, robí, a v budúcnosti plánuje robiť. Má mať prehľad o tom, n koho je dobré sa obrátiť v prípade špecifickejších otázok. Vedúci tímu zároveň plní úlohu SCRUM master-a. Nie je však klasickou autoritou. Nevykonáva rozhodnutia samostatne ale len po dohode väčšiny tímu. Jeho úlohou je jednotlivým členom tímu ponúknuť podporu a motivovať ich k lepším výkonom.

Vedúci tímu by sa mal zúčastniť na všetkých tímových formálnych aj neformálnych stretnutiach. Každé dôležité rozhodnutie ohľadom akéhokoľvek procesu spojeného s projektom by malo byť prediskutované s celým tímom, minimálne však s vedúcim tímom ktorý ďalej informuje zainteresované osoby.

Všetky vyššie spomenuté zodpovednosti a úlohy sa v súčasnosti vedúci tímu snaží zabezpečiť avšak vzhľadom na náročnosť a prvú skúsenosť s danou pozíciou nie sú všetky procesy ideálne. Dôležité však je, že rovnako ako všetky procesy, tak aj proces riadenia sa kontinuálne zlepšuje a tím sa zvyšuje aj celková kvalita tímu a jeho výsledkov.

Vedúci tímu má v našom prípade na starosti aj prezentáciu tímu na verejnosti. Má za úlohu pracovať na značke tímu a budovať jeho identitu. Mal by byť rovnako ako zadávateľ projektu v kontakte s externým prostredím. Prenáša farchy na seba a tým ponúka ostatným členom tímu viac možností na nerušenú a produktívnu prácu na projekte. Samozrejme aj on sám sa zúčastňuje vývojových aktivít. Jeho prínos však nespočíva len v počte implementovaných funkcionalít. Aj keď je v prípade tohto tímového projektu funkcia vedúceho tímu obmedzenejšia ako v reálnom prostredí, stále však plní dôležitú úlohu.

Z hľadiska riadenia ľudských zdrojov je vedúci tímu zodpovedný za to, aby v prípade preťaženia jedného člena tímu dokázal zabezpečiť rovnováhu v tíme. Po dohode so zadávateľom projektu konzultuje zásahy do aktuálneho šprintu. Či už ide o pridanie alebo naopak odstránenie niektorých úloh. Musí však zabezpečiť to, aby sa ku každému členovi tímu pristupovalo rovnako. Nesmie sa stať že niektorí členovia tímu neustále nestíhajú dokončiť dohodnutú prácu. V takomto nepríjemnom prípade sa opäť po porade so zadávateľom projektu rozhodnú ako budú postupovať.

Pri týchto úlohách je hlavným pracovným nástrojom vedúceho tímu už spomínaný nástroj JIRA. Medzi jeho možnosti patrí už spomínané zaznamenávanie práce. Generujú sa tzv. „burndown“ grafy, ktoré ukazujú reálny vývoj prác na projekte. Na to, aby tieto grafy mali výpovednú hodnotu je nutné, aby každý člen tímu zaznamenával každú, aj nedokončenú prácu na jemu pridelenú úlohu. Proces zaznamenania práce v nástroji JIRA je popísaný podrobne v nasledujúcej podkapitole.

8.1 Metodika – Zaznamenávanie práce a sledovanie postupu

Ciel: Dokončil som časť práce na mne pridenej úlohe. Chcem do systému založovať ako dlho som na nej pracoval z dôvodu sledovania postupu implementácie.

Vstup: Mne pridelená issue v nástroji JIRA, v stave vykonávania (ang. „In progress“).]

Výstup: Uzavretá issue so zaznamenaným časom práce alebo aktualizovaná issue o hotovú časť. Aktuálny graf postupu (ang. „Burn-down chart“).

Na logovanie práce budeme používať funkcionality „log work“. Nebudeme teda používať postup pri ktorom JIRA automaticky počíta čas od zmeny statusu issue z prebiehajúcej na hotovú.

Na stránke danej issue v JIRE cez záložku „More actions“ → „Log Work“. Zobrazí sa formulár zobrazený na Obrázku 4. Ten vyplníme podľa Tabuľky 1.

Log Work

Time Spent * (eg. 3w 4d 12h) ?
An estimate of how much time you have spent working.

Date Started * 📅

Remaining Estimate Adjust automatically
the estimate will be reduced by the amount of work done, but never below 0.

Use existing estimate of 1 day

Set to (eg. 3w 4d 12h)

Reduce by (eg. 3w 4d 12h)

Work Description

📄 ? 🗑️ Viewable by All Users

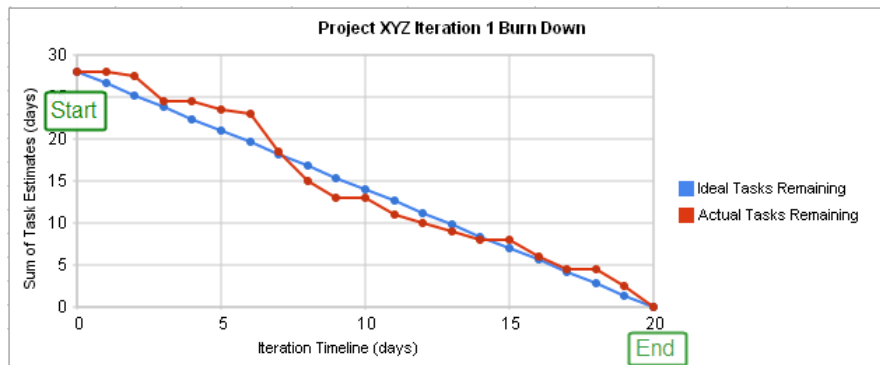
Log Cancel

Obr. 4: Log Work

Správnym a pravidelným zaznamenávaním práce použitím tohto dohodnutého postupu sme schopní získať reálny obraz o priebehu šprintu. Na základe týchto dát sa generuje spomínaný „burndown“ graf, ktorého príklad je zobrazený na Obrázku 5. V tomto prípade ide o veľmi dobrý a pomerne priamočiary graf skúseného SCRUM tímu. V našom prípade však vďaka neúplnosti dát a nie úplne zvládnutej SCRUM metodike vyzerajú tieto grafy značne horšie. V najbližších šprintoch však už dbáme na správne postupy a hlavne zodpovedné zaznamenávanie práce čo značne zlepši použiteľnosť týchto dát.

Tab. 1: Vyplnenie Log Work-u

Atribút	Ako vyplniť
Time spent (uplynutý čas)	Čo najpresnejšie zadefinovaný čas strávený pri práci na danej issue. Ide o celkový čas vrátane analýzy problému, hľadania riešenia, a implementácie. Príklad: „4h 30m“.
Date started (dátum začatia práce)	Dátum a čas kedy sme začali na danej issue pracovať (Môže sa odlišovať od času a dátumu pridelenia úlohy).
Remaining estimate (Odhad ešte potrebného času)	Odhadovaný zostávajúci čas. Vždy treba vyplniť. Na základe týchto údajov sa kontroluje stav šprintu. Skreslené odhady skresľujú celý obraz o šprinte.
Work description (Popis vykonanej práce)	Komentár k vykonanej práci. Súvisí priamo s konkrétnym issue, ktoré logujeme. V prípade, že úloha ešte nebola dokončená, obsahuje popis toho, čo už bolo implementované a čo je ešte potrebné dokončiť.



Obr. 5: Burndown graf

V prípade, že budú dáta správne tak už nič nebráni tímu v rýchlejšom napredovaní a vývoji, napríklad aj v oblasti kvality a presnosti odhadovania časovej náročnosti úloh. Nakoľko to je jeden z hlavných dôvodov problémov, objavených použitím tohto grafu. Vedúci tímu je teda zodpovedný za to aby sa výsledky a zistenia aplikovali tak, aby zlepšili proces akým tím funguje v nasledujúcej iterácii.

Vedúci tímu však samozrejme nie je jediný kto spomenuté veci sleduje. V prípade „burndown“ grafu spolupracuje primárne s manažérom plánovania. S manažérom rizík, ktorý je súbežne aj jeho zástupcom, spolupracuje skoro pri všetkých procesoch. Samozrejma je jeho spolupráca so zadávateľom projektu. Vedúci tímu má teda za úlohu sledovať všetky procesy pričom dbá na to, aby boli všetci členovia tímu spokojní, aby mali všetky potrebné informácie, a aby celý tím fungoval efektívne a aby dosahoval vytýčené ciele.

9 Manažment dokumentácie

Autor: Bc. Marek Lenčes

9.1 Roly účastníkov

Rola	Povinnosti
Produktový vlastník	<ul style="list-style-type: none"> • čítanie a kontrola dokumentácie • návrhy na zlepšenie dokumentácie • hodnotenie kvality dokumentácie
Manažér dokumentácie	<ul style="list-style-type: none"> • kontrola konzistentnosti a úplnosti dokumentácie a zápisníc • kontrola gramatickej správnosti • úpravy formátovania dokumentov
Všetci členovia	<ul style="list-style-type: none"> • vytváranie projektovej dokumentácie • vytváranie zápisníc • komentovanie zdrojových kódov • snaha o gramatickú jednoznačnosť a bezchybnosť

9.2 Projektová dokumentácia

Dokumentovanie vytvárajúcej aplikácie bude prebiehať priebežne v rámci každého šprintu. Každá nová funkcionálna v podobe konkrétnej feature v rámci user story bude zdokumentovaná členom tímu, ktorý mal túto funkcionálnu v danom šprinte na starosti. Každý člen tímu je povinný pravidelne aktualizovať dokumentáciu, keď vytvorí novú feature, ktorá bude funkčná a bude zlúčená s hlavnou vetvou v git repozitári.

Projektová dokumentácia sa bude nachádzať v spoločnom adresári v repozitári Google Drive, kde budú mať k nej všetci členovia tímu prístup na úpravy. Na Google Drive sa budú nachádzať nasledovné dokumenty:

- Projektová dokumentácia - verzia na úpravy,
- Projektová dokumentácia - publikovateľná, finálna verzia, opravená a správne naformátovaná,

- Zápisnice zo stretnutí - verzia na úpravy,
- Zápisnice zo stretnutí - publikovateľná, finálna verzia, opravená a správne naformátovaná
- Všetky metodiky, ktorými sa členovia tímu riadia
- UML diagramy vo formáte *.xml* dostupné na úpravy každým členom tímu
- UML diagramy vo formáte *.xml* vo finálnej verzii, a vo formáte *.jpg* vo finálnej verzii

Manažér dokumentácie je zodpovedný za publikovanie finálnych verzií jednotlivých dokumentov, jeho úlohami sú:

- oprava gramatických chýb
- úprava štruktúry a vzhľadu (L^AT_EX)
- jednotnosť v používaní pojmov
- konzistencia jazyka, prípadne definovanie neznámych pojmov
- korektné začlenenie obrázkov a tabuliek
- kontrola správnosti zápisníc
- finalizácia zápisníc
- kontrola komentárov v zdrojovom kóde

Pri vytváraní návrhu architektúry v UML (diagramy - dátový model) sme sa dohodli na modelovacom nástroji *Visual Paradigm*. Diagramy budú vytvárané a upravované všetkými členmi tímu, budú ukladané do repozitára Google Drive vo formáte *.xml*, ktorý je možné upravovať importovaním do modelovacieho nástroja. Finálna verzia diagramov bude uložená do osobitného priečinka, a okrem formátu *.xml* bude v ňom dostupná aj inštancia diagramu vo forme obrázku vo formáte *.jpg*.

9.3 Zápisnice

Tvorba zápisníc bude prebiehať vždy na stretnutí v aktuálnom šprinte, a bude tvorená vždy iným členom tímu. Zápisnica má presne stanovenú štruktúru, ktorú treba dodržať, čo bude mať na starosti zapisovateľ. Zápisnica ešte bude po stretnutí zrevidovaná zapisovateľom, prípadne doplnená a odovzdaná bude do repozitára dokumentov, kde bude ešte raz zrevidovaná manažérom dokumentácie. Ten sa postará o opravu pravopisných chýb a kontrolu dodržania stanovenej štruktúry. Následne bude zápisnica uložená na prezentačnú stránku tímu, stane sa tak maximálne do konca posledného dňa príslušného šprintu.

9.4 Dokumentácia zdrojového kódu

Komentovanie zdrojového kódu nám uľahčí orientáciu v jednotlivých častiach zdrojového kódu, a umožní aj identifikovať, kto vytvoril daný kód. Preto sa bude na začiatku každého vytvoreného súboru uvádzať kto ho vytvoril, krátky názov, a čo daný kód reprezentuje, príklad:

```

1 =begin
2 Bc. Buduci Inzinier
3 Users Controller
4 obsahuje funkcie na spravu pouzivatelov
5 =end

```

Ďalší príklad, v HTML zdrojovom kóde

```

1 <!--
2 Bc. Buduci Inzinier
3 Task Create View
4 zobrazuje pouzivatelovi formular pre vytvorenie ulohy
5 -->

```

Komentáre v rámci zdrojového kódu sú nutné iba v prípade, že kód obsahuje nejasné jazykové konštrukcie, alebo zložitejšie mechanizmy, ktoré budú ľahšie pochopiteľné pre ostatných členov tímu pomocou komentárov.

9.5 Procesy

Číslo	Názov procesu
8.1	Dokumentovanie dokončenej funkcionality
8.2	Revízia dokumentácie
8.3	Dokumentovanie stretnutí

9.5.1 Dokumentovanie dokončenej funkcionality

Vstup: Nezdokumentovaná časť práve naprogramovanej funkcionality

Výstup: Vytvorená dokumentácia pre hotovú funkcionality

Postup:

1. Člen tímu otvorí príslušný program s dokumentáciou v repozitári Google Drive
2. Člen tímu zdokumentuje vytvorenú funkcionality
3. Člen tímu skontroluje správnosť a pravdivosť opísanej funkcionality

9.5.2 Revízia dokumentácie

Vstup: Vytvorená dokumentácia alebo jej časť

Výstup: Opravená dokumentácia

Postup:

1. Manažér dokumentácie otvorí príslušný dokument s dokumentáciou
2. Manažér dokumentácie skontroluje pridanú časť dokumentácie, opraví chyby, skontroluje konzistenciu

3. Manažér dokumentácie presunie skontrolovanú časť dokumentácie do finálnej verzie dokumentu

9.5.3 Dokumentovanie stretnutí

Vstup: Témy rozoberané na stretnutí tímu

Výstup: Zápisnica z aktuálneho stretnutia

Postup:

1. Člen tímu otvorí šablónu zápisnice
2. Člen tímu zapisuje všetky potrebné údaje do zápisnice
3. Ostatní členovia tímu na záver zápisnicu skontrolujú a potvrdia
4. Člen tímu uloží zápisnicu do repozitára Google Drive a na prezentačnú stránku tímu

10 Manažment kvality

Autor: Bc. Miroslav Vojtuš

10.1 Konvencie v jazyku Ruby a programovom rámci RoR

S kvalitou kódu určite súvisí aj čitateľnosť kódu. Pre zabezpečenie čitateľnosti máme vypracovanú metodiku konvencií pre jazyk Ruby a programový rámec Ruby on Rails. Zvýšením čitateľnosti zároveň znižujeme riziko odchodu člena tímu, keďže dobre čitateľný kód sa aj dobre absorbuje novým človekom. Taktiež pri dodržaní spoločných postupov a konvencií daných Ruby on Rails rámcom, zabezpečíme konzistenciu kódu, a tým zredukujeme duplicitný kód, čo opäť vedie k vyššej kvalite kódu.

V ďalších kapitolách sa nachádza upresnenie(ohraničenie) konvencií daných programovým rámcom RoR a jazykom Ruby. Opísané sú konvencie písania názvov, štandardných postupov a organizácie zdrojových kódov. Účelom je zabezpečiť kvalitu kódu a jeho čitateľnosť.

10.1.1 Všeobecné

- nikdy neodovzdávajte do zdieľaného úložiska zakomentovaný zdrojový kód. V prípade, že vykonanie kódu blokujete len dočasne, osamostatnite si ho lokálne,
- minimalizujte komplexnosť kódu dekompozíciou do samostatných funkčných blokov(napr. súkromné metódy),
- opisné komentáre používajte len na komplikované (ťažko pochopiteľné) časti kódu. Aj v takomto prípade preferujte dekompozíciu a plnovýznamové názvy premenných a metód,
- inštančné premenné používajte len kde je nevyhnutné (napr. na zdieľanie informácií s pohľadom)
- na prenos informácií medzi metódami používajte výhradne predávanie pomocou parametrov
- na tvorbu kódu uprednostnite dostupné generátory. Tie zabezpečia správne vytvorenie/umiestnenie súborov so zdrojovými kódmi. Nepoužité vygenerované súbory a časti kódu odstráňte,
- vlastné knižnice umiestňujte do priečinka '`<rails_root>/lib`'
- knižnice tretích strán uprednostňujte vo forme gem-ov pred manuálnym sťahovaním zdrojových súborov. Ak je to nutné, tieto umiestnite do priečinka '`<rails_root>/lib`'

10.1.2 Písanie názvov

- všetky názvy v zdrojových kódoch musia byť písané výhradne v anglickom jazyku
- musia byť krátke a dostatočne opisné na vyjadrenie účelu

- každá premenná musí spĺňať kritéria dané jazykom Ruby a zároveň formát daný pravidlami opísanými v nasledujúcej tabuľke:

oblasť	príklady	pravidlá zápisu
<i>inštančné, globálne premenné</i>	variable longer_variable	slová malými písmenami oddelené znakom '_' slová musia byť plnovýznamové regex: <code>^[a-z]+([a-z][_])*\$</code>
<i>lokálne premenné</i>	local_variable i	slová začínajúce malým písmenom oddelené znakom '_' regex: <code>^[a-z]+([a-zA-Z0-9][_])*\$</code>
<i>konštanty</i>	CONSTANT LONG_CONST	slová veľkými písmenami oddelené znakom '_' slová musia byť plnovýznamové regex: <code>^[A-Z]+([A-Z][_])*\$</code>
<i>triedy, moduly</i>	Module1 Controller::One	slová začínajúce veľkými písmenami slová musia byť plnovýznamové regex: <code>^[A-Z]+([a-z0-9]+([A-Z]::[A-Z])?)*\$</code>
<i>verejné metódy</i>	method_one method?	slová malými písmenami oddelené znakom '_' slová musia byť plnovýznamové regex: <code>^[a-z]+([a-z][_])*[!]?+\$</code>
<i>mená súborov</i>	model_one.rb controller_name.rb view.erb	slová malými písmenami oddelené znakom '_' zakončené platnou príponou regex: <code>^[a-z]+([a-z][_])*\.[a-z]+\$</code> <i>vinímkou sú migračné skripty, ktoré môžu obsahovať číslíce na začiatku názvu</i>

Obr. 6: Pravidlá formátovania

10.1.3 Ruby konvencie

Jazyk Ruby často umožňuje zápis niektorých typických úloh rôznymi prístupmi. V nasledujúcich odsekoch sú opísané preferované postupy, ktoré zabezpečia čitateľnosť a konzistenciu zdrojových kódov. Tieto je nutné dodržiavať, pokiaľ nie je nevyhnutné konať inak, v takom prípade treba mať primerané argumenty,

- nepoužívajte metódu `.nil?` - pokiaľ to nie je explicitne nutné. Na testovanie či bol objekt inicializovaný používajte `if object` nie `if !object.nil?` ani `if object != nil`.
- používajte `unless` - Na testovanie, či objekt nie je inicializovaný používajte `unless object` namiesto `if !object`,
- používajte jednoriadkové podmienené vykonanie:

```

1 if element.processable?
2   element.process
3   puts "element #{element} processed"
4 end
5
6 element.process if element.processable?

```

- `do end` - v prípade viacriadkových blokov

```

1 elements.each do |element|
2   element.process
3   puts "object #{element} processed"
4 end

```

- `{}` - v prípade jednoriadkových blokov

```
1 elements.each {|element| element.process}
```

- *zátvorky používajte len ak je to nutné* - zvyčajne len pri vnorenom volaní funkcií. Jednoduchý zoznam parametrov nemusí byť ohraničený zátvorkami:

```
1 return do_a_job(param, get_job(args, :console))
2 do_a_job param, :exec
```

10.1.4 Migrácie

- pre každý zásah do databázovej schémy píšete migračný kód
- na vytvorenie migrácie uprednostňujte dostupné generátory
- názvy generačných skriptov musia byť samo-opisné
- nikdy neupravujte staršie migrácie (neplatí pre lokálne, nezdieľané)
- pre dlhé migračné skripty píšete sprievodný text na výstupnú konzolu
 - pre výstup na konzolu používajte výhradne metódu *say*
 - neinformujte o každej zmene, len sumárne výpisy (napr. „*User → 30 rows has been removed*“, „*— added column to Users*“)

10.1.5 Model

- organizácia kódu
 - relačné príkazy
 - validačné príkazy
 - ďalšie skupiny príkazov
 - deklarácie verejných metód
- využívajú už existujúce validačné metódy (zdroj: dokumentácia RoR)
- pri každej zmene modelu je nutné upraviť testy (viac v časti testovanie)

10.1.6 Ovládač (ang. Controller)

- organizácia kódu
 - filtrovacie príkazy
 - verejné metódy prístupné cez adresu
 - ostatné verejné metódy
 - súkromné metódy (príkaz *private* oddeliť desiatimi znakmi ‘#’)
- z dôvodu ochrany pred SQL-Injection útokom pre dopyty na databázu používajte podmienky s oddelenými premennými

```
1 User.find(:conditions => ["name like ? and age >= ?",
2 @user.name, @user.age])
```

10.1.7 Pohľad (ang. View)

- z dôvodu prevencie pred XSS útokom, akékoľvek dáta prijaté od používateľa zobrazte cez metódu `h()`
- jednoduché reťazce nikdy nevypisujte cez `<%= "reťazec"%>`

10.2 Testovanie

Ďalším podstatným prvkom zabezpečenia kvality je testovanie. Čím lepšie je otestovaný kód, tým lepší je výstup a nižšia pravdepodobnosť výskytu chýb. Kód je možné testovať na viacerých úrovniach preto v tejto kapitole opisujem tri základne úrovne:

- jednotkové testovanie
 - najnižšia úroveň, kde sa testujú jednotlivé funkčné bloky podľa špecifikácie
 - základná forma testovania, ktorou sa predchádza kalkulačným a multivariantovým chybám
- funkčné testovanie
 - stredná úroveň kde sa testujú komplexnejšie funkcionálne prvky na úrovni ovládača
 - každá akcia ovládača by mala mať aspoň základné funkčné testy
- integračné testovanie
 - vyššia úroveň, kde sa testuje správanie aplikácie na úrovni používateľských prípadov, prípadne používateľských príbehov
 - časovo náročné definovanie i vykonanie týchto testov
 - najlepšie overí fungovanie systému naprieč jednotlivými inkrementami

V nasledujúcich kapitolách sú opísané metodické pokyny, ktoré musí dodržiavať každý člen tímu podieľajúci sa na tvorbe zdrojových kódov.

10.2.1 Procesy

V tejto kapitole sú definované procesy, ktoré je nutné dodržiavať pre zabezpečenie kvality kódu. V nasledujúcej tabuľke sú vymenované:

#	Názov postupu	Kapitola
1	Tvorba jednotkových testov	10.2.1.A
2	Tvorba funkčných testov	10.2.1.B
3	Tvorba integračných testov	10.2.1.C
4	Modifikovanie jednotkových testov	10.2.1.D
5	Modifikovanie funkčných testov	10.2.1.E
6	Modifikovanie integračných testov	10.2.1.F
7	Spustenie testov	10.2.1.G

10.2.1.A Tvorba jednotkových testov

Vstup: Neotestovaný komplexný funkčný kód

Výstup: Testovacia jednotka pre každú komplexnú funkcionalitu

Proces:

1. Vytvoriť príslušný súboru s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť prípravky pre testovanie
3. Vytvoriť podmienené jednotkové testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú komplexnú funkciu/metódu

10.2.1.B Tvorba funkčných testov

Vstup: Neotestovaná akcia ovládača

Výstup: Funkčný test pre akciu ovládača

Proces:

1. Vytvoriť príslušný súboru s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť funkčné testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú verejnú akciu ovládača

10.2.1.C Tvorba integračných testov

Vstup: Neotestovaný používateľský prípad

Výstup: Integračný test pre základné hranice používateľského prípadu

Proces:

1. Vytvoriť príslušný súboru s názvom a umiestnením podľa konvencií RoR
2. Vytvoriť integračné testy spĺňajúce formát definovaný v kapitole 10.2.2 pre každú komplexnú funkciu/metódu

10.2.1.D Modifikovanie jednotkových testov

Vstup: Zmenená funkcionalita s existujúcim jednotkovým testom

Výstup: Rozšírená testovacia jednotka

Proces:

1. Podľa zmeny funkcionality upraviť prípravky
2. V závislosti na zmene očakávaných výstupov upraviť podmienené testy, aby zahrnuli túto zmenu a zároveň zahrnuli nové krajné prípady ktoré vznikli

10.2.1.E Modifikovanie funkčných testov

Vstup: Zmenená logika správania akcie ovládača

Výstup: Rozšírený funkčný test

Proces:

1. V závislosti na zmene upraviť funkčný test tak, aby zahrňoval upravenú logiku.

10.2.1.F Modifikovanie integračných testov

Vstup: Zmenená logika používateľského prípadu

Výstup: Rozšírený integračný test

Proces:

1. V závislosti na zmene upraviť integračný test aby zahrňoval upravenú logiku.

10.2.1.G Spustenie testov

Vstup: Zdrojový kód s vytvorenými testovacími jednotkami a integračnými testami

Výstup: Zdrojový kód, ktorého kvalita je overená vykonanými testami

Proces:

1. Pred zdieľaním zdrojových kódov spustiť automatizované testy - v konzole spustiť príkaz *rake test* nad koreňovým priečinkom aplikácie.
2. Ak upravený kód neprejde testami, upraviť zdrojový kód alebo testy.

10.2.2 Konvencie pre testovanie

Nasledujúce kapitoly definujú interné konvencie pre jednotlivé typy testov. Spoločné ohraničenia:

- test musí mať plnovýznamové pomenovanie
- každý test testuje len jeden prípad
- čitateľnosť a samo-opisnosť má prednosť pred efektivitou
- využívajte generátory pre správne umiestnenie a pomenovanie testov
- na prerušenie testu nepoužívajte výnimky

10.2.2.A Jednotkové testy

Minimálne prípady pre testovanie:

- nový alebo upravený validačný skript
- nová metóda v modeli, ktorá má komplexnú funkcionálnosť
- nové ohraničenie modelového objektu

Ohraničenia:

- testovanie výhradne pomocou *ActiveSupport::TestCase*

```

1 class UserTest < ActiveSupport::TestCase
2   fixtures :users
3
4   test "user name can't be null" do
5     assert_not_nil users(:user).name
6   end
7 end

```

10.2.2.B Funkčné testy

Minimálne prípady pre testovanie:

- úspešnosť dopytu na akciu
- správne presmerovanie používateľa
- používateľ bol autentifikovaný
- správne nastavené objekty na výstupe
- používateľovi sa zobrazila požadovaná správa

Ohraničenia:

- testovanie výhradne pomocou *ActionController::TestCase*

```

1 class UsersTest < ActionController::TestCase
2   test "should create user" do
3     assert_difference('User.count') do
4       user :create, user: {name: 'some name'}
5     end
6
7     assert_redirected_to user_path(assigns(:user))
8   end
9 end

```

10.2.2.C Integrované testy

Minimálne prípady pre testovanie:

- dokončený používateľský prípad, ktorý nemá naplánované významne úpravy logiky vykonania

Ohraničenia:

- testovanie výhradne pomocou *ActionDispatch::IntegrationTest*

```

1 class UserFlowsTest < ActionDispatch::IntegrationTest
2
3   fixtures :users
4
5   test "login and browse site" do
6     https!
7     get "/sign_in"
8     assert_response :success
9
10    post_via_redirect "/sign_in",
11                      username: users(:user).username,
12                      password: users(:user).password
13    assert_equal '/tasks', path
14    assert_equal 'Welcome user!', flash[:notice]
15
16    https!(false)

```



```

17   get "/posts/all"
18   assert_response :success
19   assert_assigns(:tasks)
20 end
21 end

```

10.3 Spätná kontrola kvality

Kvalita kódu sa dá do určitej miery merať. Existujú metriky ktoré umožňujú sledovať kvalitatívne nedostatky a zistiť ich zdroj(či už ľudský alebo technický). Jedným z hlavných ukazovateľov je ukazovateľ pokrytia kódu testami (z ang. code coverage). Na získanie tohto ukazovateľa pre jednotlivé triedy a metódy v jazyku ruby používame gem (plugin/knižnica pre ruby programy) *MetricFu*. Tento gem kombinuje metriky omnoho viac metrík z viacerých zdrojov:

Názov gemu (zdroja)	Opis
Cane	Blokuje zostavenie projektu v prípade zistených nedostatkov v kóde. (parametrizovateľný)
Churn	Zaznamenáva históriu zmien jednotlivých funkčných blokov. Výstupom je frekvencia zmien.
Flog	Reportuje najzložitejší kód. Vytvára reporty, ktoré sú dobrým ukazovateľom zdrojov potenciálnych problémov.
Reek	Vyhľadáva pachy v kóde.
Saikuro	Analyzátor cyklomatickej zložitosti pre jazyk Ruby.
Roodi	Analyzuje kvalitu Objektovo Orientovaného Dizajnu.
CodeMetrics	Analyzuje základné metriky kódu, ako napríklad počet riadkov, metód, tried.
Rails best practices	Analyzuje kvalitu z pohľadu najlepších praktík v programovom rámci RoR.
SimpleCov	Reportuje mieru pokrytia kódu testami

11 Manažment podpory vývoja

Autor: Bc. Patrik Štrba

V prvých šprintoch sme si nastavili procesy, ktoré majú zabezpečovať kvalitu produktu. Medzi tieto procesy patrí dodávanie kódu, vytváranie verzií produktu alebo vytváranie produkčných verzií a ich nasadenie.

V procese dodávania kódu sme použili architektúru založenú na spoločnom tímovom repozitári, kde má každý člen tímu právo čítať zmeny, ktoré boli vykonané na danom repozitári. Každý člen v tíme má potom svoj vlastný repozitár, na ktorý už má právo zápisu. Architektúra je postavená na spoločnom repozitári a samostatných repozitároch pre každého člena, kde sa zmeny vykonávajú pomocou *pull-request*, kde každý člen samostatne pracuje na svojom repozitári. Do svojho repozitára si ukladá zmeny vykonané v systéme potrebné pre úspešné dokončenie úlohy pridelenej na stretnutí. Po dokončení sa všetky tieto zmeny sa zabalia do požiadavky (*pull-request*), ktorá je následne spracovávaná v procese schvaľovania takýchto požiadaviek. Týmto sme dokázali zabezpečiť prísun kvalitného kódu, kedy sa schvaľujú len kódy, ktoré sú čitateľné a nie je v nich na prvý pohľad vidno vznik chýb a podobne. Týmto zároveň vykonávame aj *code review*, na ktorom sa podieľajú všetci členovia tímu.

Ďalší proces, ktorý vznikol pre zvýšenie kvality a jednoduchšie nasadenie produktu, je produkčné nasadenie, kedy je dôležité dodávať zmeny rýchlo, avšak je potrebné produkt otestovať ešte pred nasadením do ostrej prevádzky. Preto sme použili balík nástrojov, ktoré nám umožňujú efektívnejšie a zároveň nám pridávajú možnosť jednoduchšej manipulácie pri vytváraní produkčných verzií. Zároveň nám umožňuje jednoducho prezerať vytvorené predchádzajúce produkčné verzie a v prípade vytvorenia nefunkčnej produkčnej verzie, nasadiť staršiu verziu, avšak bez chýb.

V každom zložitejšom softvérovom produkte sa vyskytujú chyby a preto musíme tento problém riešiť. Ideálne je udržiavať počet nájdených chýb na nule a preto je dôležité si tieto chyby evidovať. Na evidovanie chýb používame nástroj JIRA. Ten nám umožňuje jednoducho vytvárať a ďalej manipulovať s chybami, ktoré objavil jeden z členov alebo nám boli nahlásené cudzou osobou. Tieto chyby vieme pridelať osobám, ktoré mali na starosť funkcionality v ktorej sa chyba vyskytla a preto je z pohľadu efektívnosti najlepšie aby chybu riešil tvorca danej funkcionality. V prípade ak sa jedná o naliehavú chybu, je možné ju riešiť priamo členom, ktorý chybu našiel alebo bol o nej informovaný.

11.1 Použitie vetiev (branch) v tímovom repozitári

V tímovom repozitári sa nachádzajú vetvy:

- master
- develop

Master vetva obsahuje funkcionality, ktorá je otestovaná a pripravená na proces nasadenia do produkcie. Tento proces vykonáva administrátor repozitára.

Develop vetva obsahuje funkcionality na ktorej sa pracuje, avšak sa zároveň považuje za dokončenú. Do tejto vetvy nesmie byť pridaná funkcionality, ktorá obsahuje chyby alebo nie je dokončená.

11.2 Potvrdenie funkcionality do tímového repozitára

Vstup: Vytvorený pull-request

Výstup: Spracovaný pull-request

Proces:

1. O potvrdzovanie pull-requestov sa starajú správcovia repozitára. Každý pull-request je najskôr potreba analyzovať. Po analýze sa následne zmeny potvrdia, alebo zrušia a doplnia poznámkou, čo nie je vhodné a treba prerobiť.
2. Schvaľovanie prebieha formou označenia požiadavky za schválenú. Pre začlenenie požiadavky musia obaja správcovia repozitára udeliť schválenie aby mohla byť požiadavka schválená.
3. Ak je pull-request v poriadku, tak sa označí ako schválený a následne je zlúčený. V prípade ak sa jedná o sùrne opravy, tak je možné proces schvaľovania preskočiť a požiadavku schváliť priamo.

11.3 Vytvorenie novej verzie

Vstup: Vytvorené commity

Výstup: Vytvorenie značky (tag) označujúca daný commit za novú verziu

Proces:

1. Vytvorenie prebieha vždy po uzatvorení úseku dokončených úloh, kde už nie je známa žiadna chyba.
2. Číslo verzie sa bude deliť na 2 časti, pričom časti budú oddelené bodkou.
3. Prvé číslo udáva verziu produktu. Toto číslo sa zmení len pri dokončení všetkých zmien, ktoré sú naplánované do novej verzie.
4. Druhé číslo udáva verziu, ktorá zmení len pri dokončení všetkej funkcionality, ktorá bola naplánovaná na daný šprint.
5. Osoba poverená správou verzií je administrátor repozitára.

11.4 Vytvorenie novej funkcionality

Vstup: Pridelená úloha v systéme na evidenciu úloh

Výstup: Vytvorená vetva s požadovanou funkcionalitou

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „feature-“ názov popisujúci danú funkcionalitu
2. Vytvoriť novú funkcionalitu
3. Vytvoriť commit a napísať správu opisujúcu funkcionalitu (čo a ako bolo implementované)
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *develop*

11.5 Oprava chyby na vývojovej verzii aplikácie

Vstup: Objavená chyba a nahlásená v systéme pre správu chýb

Výstup: Vytvorená vetva s požadovaným názvom

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „bug-“ a číselný identifikátor chyby
2. Vytvoriť opravu pre danú chybu
3. Vytvoriť commit a napísať správu opisujúca chybu
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *develop*

11.6 Oprava chyby na produkčnej verzii aplikácie

Vstup: Objavená chyba a nahlásená v systéme pre správu chýb

Výstup: Vytvorená vetva s požadovaným názvom

Proces:

1. Vytvoriť novú vetvu a zvoliť názov s pridaním prefixu „hotfix-“ a číselný identifikátor chyby
2. Vytvoriť opravu pre danú chybu
3. Vytvoriť commit a napísať správu opisujúca chybu
4. Nahrať zmeny na vzdialený repozitár (push)
5. Vytvoriť požiadavku na pridanie zmien do tímového repozitára, konkrétne do vetvy *master*

11.7 Spolupráca na spoločnej funkcionalite

Vstup: Oprávnenie na repozitáre ostatných členov s ktorými je potrebná spolupráca

Nahrané zmeny na vzdialených repozitároch (push)

Pridané vzdialené repozitáre potrebných členov v lokálnom repozitári

Výstup: Možnosť spolupracovať na úlohe spoločne

Proces:

1. Načítať (fetch) si zmeny z repozitára z ktorého chcem pridať funkcionalitu
2. Zlúčiť (merge) lokálnu vetvu (local branch), na ktorej chcem pridať požadovanú funkcionalitu so vzdialenou vetvou (remote branch), kde sa nachádza požadovaná funkcionalita.
3. Nahrať zmeny na svoj repozitár (push)
4. Prvé tri kroky opakovať, pokiaľ nie je funkcionalita hotová.

12 Plány

12.1 Zimný semester

Tab. 2: Plán na zimný semester

Týždeň	Dátum začiatku	Dátum konca	Popis činnosti
1	Začiatok semestra	15.10.2012	<ul style="list-style-type: none"> • Tvorba webovej stránky • Dizajn plagátu • Analyzovanie domény Digital Sweats-hop a podobných riešení • Výber vhodných technológií a programovacích jazykov • Vytvorenie šablón pre dokumentáciu • Vytvorenie jednotnej šablóny pre zápisky
4-6	17.10.2013	31.10.2013	<ul style="list-style-type: none"> • Vytvorenie backlogu projektu • Rozbehanie dohodnutých technológií • Implementovať prihlasovanie používateľa a základný dátový model
6-8	31.10.2013	14.10.2013	<ul style="list-style-type: none"> • Vytvorenie profilu používateľa • Integrácia so sociálnymi sieťami • Vytvoriť UI pre aplikáciu • Analyzovať a vybrať vhodný názov pre našu aplikáciu
8-10	14.11.2013	28.11.2013	<ul style="list-style-type: none"> • Vytvorenie finálnych verzií metodík • Zaradenie vytvorených metodík do procesu vývoja • Vytvoriť a analyzovať kategórie úloh v aplikácií
10-12	28.11.2013	12.12.2013	<ul style="list-style-type: none"> • Doladovanie a refaktoring • Testovanie • Doladovanie dokumentácie • Prezentovanie stavu projektu na konci semestra

12.2 Letný semester

12.2.1 Krátkodobý plán

Počas skúškového obdobia sme sa ako tím dohodli na týchto povinnostiach, ktoré do začatia letného semestra splní každý náš člen. Výsledky sa budú prezentovať na prvom tímovom stretnutí v letnom semestri.

- Refaktorovanie a upratovanie projektu
- Ošetrovanie a doladovanie doteraz implementovaných funkcionalít
- Testovanie aplikácie
- Úprava grafických chýb
- Zjednotenie dizajnu (farby, tvary, navigácia, logá, motívy)
- Vytvorenie funkčnej verzie do vetvy Master
- Vo vetve develop na do začiatku semestra aktualizuje funkčná upravená verzia so všetkými zmenami, ktoré prebehli počas skúškového obdobia
- Migrácia členov na jednotnú databázovú technológiu (MySQL)
- Študovanie metodík
- Aspoň 3 stretnutia celého tímu prostredníctvom Google Hangouts
- Každý člen spíše 3 nové user story, ktoré mu v systéme chýbajú
- Každý člen spíše 3 user story, ktoré su podľa neho najmenej podstatné z doteraz implementovaných

12.2.2 Dlhodobý plán

Tab. 3: Plán na letný semester

Týždeň	Dátum začiatku	Dátum konca	Popis činnosti
1	20.02.2014	26.02.2014	<ul style="list-style-type: none"> • Retrospektíva, čo sa spravilo za skúškové obdobie
2-3	27.02.2014	12.03.2014	<ul style="list-style-type: none"> • Vylepšenie grafickej podoby aplikácie • Práca na dátovom modeli a jeho zapracovanie do aplikácie • Optimalizácia a zrýchľovanie aplikácie • Doplňenie rozhraní, pre externé služby • Testovanie aplikácie model
4-5	13.03.2014	25.03.2014	<ul style="list-style-type: none"> • Funkcionalita orientovaná na zadávateľa • Úprava rozhrania pre zadávanie nových úloh • Vytvorenie prehľadnej štatistiky výsledkov pre zadávateľov • Úprava admin panelu, úprava kompetencií rolí v systéme • Štatistické údaje pre administrátora, kontrola nad financiami
6-7	26.03.2014	09.04.2014	<ul style="list-style-type: none"> • Funkcionalita orientovaná na pracovníkov • Prehodnotenie štruktúry profilu pracovníka • Pridanie nových ocenení za dobre odvedenú prácu • Finančný model a odmeňovanie
8	10.04.2014		<ul style="list-style-type: none"> • Práca na plagáte IIT.SRC • Oprava chýb a testovanie • Vytvorenie kompletného prekladu aplikácie pre SK a ENG
9-10	17.04.2014	29.04.2014	<ul style="list-style-type: none"> • Príprava na TPCup • Testovanie aplikácie v reálnych podmienkach • Stretnutia s potenciálnymi klientmi a používateľmi • Dohoda o pokračovaní vo vývoji projektu po ukončení predmetu
11 - 12	30.04.2014	Koniec semestra	<ul style="list-style-type: none"> • Údržba, posledné úpravy zdrojového kódu • Dokončenie dokumentácie • Odovzdanie a prezentovanie výsledkov • Víťazstvo v súťaži TPCup

PRÍLOHY

A: Preberací protokol

Preberací protokol

Tímový projekt 2013/2014

Tím 11 - Pinta.sk

Predmet odovzdávania:

- Dokumentácia k riadeniu projektu - zimný semester
- Dokumentácia k inžinierskemu dielu - zimný semester

Pedagogický vedúci - Ing. Dušan Zeleník

Podpisom potvrdzuje prevzatie vyššie uvedených častí projektu a dokumentácie

dátum _____

podpis _____