Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Priebežné overovanie prípravy študentov na cvičeniach [WebTest]

Osičky

Dokumentácia k riadeniu

Vedúci tímu: Ing. Branislav Steinmüller

Členovia tímu: Bc. Silvia Macejková**,** Bc. Lukáš Csóka, Bc. Roman Pikna, Bc. Martin Dekan, Bc. Michal Farkaš, Bc. Pavel Sluka

Školský rok: 2015/2016

Obsah

Obsah 2
1 Úvod 4
2 Slovník pojmov 5
3 Role členov tímu 6
3.1 Role 6
3.2 Autorstvo kapitol v dokumentácií k riadeniu 7
4 Opis manažérskych činností 9
4.1 Manažment plánovania 9
4.2 Manažment tvorby dokumentácie 9
4.3 Manažment podpory vývoja a integrácie 9
4.4 Manažment kvality 9
4.5 Manažment plánovania a rozsahu projektu 10
4.6 Manažment komunikácie 10
4.6.1 Formálna komunikácia 10
4.7 Manažment rizík 12
5 Sumarizácia šprintov 14
5.1 1. Šprint – Bulbasaur 14
5.2 2. Šprint - Ivysaur 14
5.3 3. Šprint - Venusaur 15
5.4 4. Šprint – Charmander 15
5.5 5. Šprint - Charmeleon 15
6 Používané metodiky 16
6.1 Metodika písania dokumentácie 16

6.2 Metodika dokumentovania úloh 16
6.3 Metodika tvorby metodík 16
6.4 Metodika tvorby testovacích prípadov 16
6.5 Metodika ohlasovania chýb v systéme Jira 16
6.6 Metodika vytvárania úloh 16
6.7 Metodika verziovania a udržiavania zdrojových kódov 17
7 Globálna retrospektíva 18

1 Úvod

Práca v tíme nesie svoje úskalia a naučiť sa pracovať v tíme je zložitou úlohou pre všetkých členov tímu. Rozhodli sme sa pri vývoji riadiť agilnou technikou Scrum. Dĺžka šprintov bola stanovená na dva týždne. Na začiatku sme si rozdelili roly a zodpovednosti. V tomto dokumente sú opísané zodpovednosti a roly každého člena tímu a spôsob a rozsah pôsobenia na projekte. Dokumentácia taktiež sumarizuje priebeh doterajších šprintov, popisuje použité metodiky a obsahuje globálnu retrospektívu za uplynulé šprinty.

2 Slovník pojmov

Pojem	Vysvetlenie
Scrum	Agilná metóda vývoja, pri ktorej sa kladie dôraz na tímovú prácu
git	nástroj, pre správu kódu

3 Role členov tímu

3.1 Role

Role a zodpovednosti z nich vyplývajúce sme si v rámci tímu rozdelili nasledovne:

Silvia Macejková: Manažér komunikácie, manažér kvality, vedúci tímu

- Podnecovanie komunikácie v rámci tímu
- Správa zdrojového kódu – git master
- Sledovanie termínov a dohľad nad splenením plánov a úloh
- Spájanie vetiev v nástroji Git

Lukáš Csóka: Manažér rozvrhu a plánovania

- Plánovanie budúcej činnosti teamu
- Jira master

Michal Farkaš: Manažér vývoja AlefTNG, Zodpovedný za integráciu, Hlavný architekt

- Hlavný vývojár a správca AlefTNG

Martin Dekan: Manažér dokumentovania

- Dokumentácia k riadeniu
- Dokumentácia k dielu
- Zodpovednosť za webovú prezentáciu tímu

Roman Pikna: Manažér vývoja prototypu

- Nasadzovanie na serverovú časť
- Príprava novej verzie aplikácie

Pavel Sluka: Správa zápisov zo stretnutí, manažment vývoja AlefTNG

- Zápis stretnutí

3.2 Autorstvo kapitol v dokumentácií k riadeniu

Práca na jednotlivých kapitolách v dokumentácií k riadeniu bola nasledovná:

Martin Dekan

- Úvod
- Slovník pojmov
- Opis manažérskych činností
- Role členov tímu
- Sumarizácia šprintov
- Metodika tvorby dokumentácie
- Metodika tvorby metodík
- Metodika tvorby testovacích prípadov
- Metodika ohlasovanie chýb
- Metodika dokumentovania úloh
- Globálna retrospektíva

Silvia Macejková

- Metodika verziovania a udržiavania zdrojových kódov

Lukáš Csóka

- Metodika vytvárania úloh

Roman Pikna

- Metodika pre code review

Pavel Sluka

- Metodika písania zápisnice
- Zápisy zo stretnutí

Michal Farkaš

- Metodika a štandardy pre tvorbu kódu

Práca na jednotlivých kapitolách v dokumentácií k dielu bola nasledovná:

Martin Dekan

- Úvod
- Slovník pojmov
- Globálne ciele pre zimný semester
- Celkový pohľad – AlefTNG

Roman Pikna

- Moduly prototypu
- Dátový model prototypu

Silvia Macejková

- Celkový pohľad časť prototyp

Lukáš Csóka

- Moduly prototypu a AlefTNG

Pavel Sluka

- Moduly AlefTNG
- Opis aktuálneho stavu
- Dátový model AlefTNG

Michal Farkaš

- Moduly AlefTNG
- Opis aktuálneho stavu

4 Opis manažérskych činností

4.1 Manažment plánovania

Manažér plánovania zodpovedá za plánovanie úloh, ktoré sú nutné pre dokončenie jednotlivých šprintov. Informuje členov tímu o úlohách, ktoré je nutné vykonať a dohliada na vyťaženosť členov tímu. Vyriešenie úloh konzultuje s členmi tímu. Priorita úlohy záleží od viacerých faktorov. Najvyššiu prioritu má úloha, ktorá blokuje vykonávanie iných úloh. Nasledujú úlohy, ktoré sú nutné pre fungovanie tímu a produktu.

Plánovanie sa neustále mení získavaním nových informácií, udalosťami v škole a inými nepredvídateľnými udalosťami. Plánovanie sa priebežne komunikuje s ostatnými členmi tímu a kontroluje sa stav a zisťujúca potenciálne problémy.

4.2 Manažment tvorby dokumentácie

V tímovom projekte je vedených niekoľko druhov dokumentácie. Úlohou tohto manažéra je tvorba šablón jednotlivých typov dokumentov. Definovanie pravidiel a metodík pre tvorbu dokumentácie a jej správa. Dokumenty sú spísané jednotlivými členmi tímu a manažér tvorby dokumentácie dohliada na dodržiavanie metodík a pravidiel pre tvorbu dokumentácie a jej včasné zapracovanie.

V našom projekte sa vyskytujú dva hlavné druhy dokumentácie. Prvá je projektová dokumentácia a druhá sú zápisy zo stretnutí. Forma je rozdielna kvôli určeniu. Zápisy zo stretnutí sú viac orientované na tím. Dokumentácia je viac formálnym dokumentom pre všeobecné publikum.

4.3 Manažment podpory vývoja a integrácie

Definuje konfiguráciu, spôsob správy kódu, aktuálnosť a spájanie kódov. Pomocou dohodnutých pomocných prostriedkov, pomáha pri vývoji, nasadení a testovaní a dokumentovaní kódov.

Medzi jeho hlavné zodpovednosti patrí aj nasadzovanie nových verzií aplikácii na produkčný server, a samotný bezproblémový beh aplikácie. Udržiava aktuálnosť kódov podľa stanovaných metodík.

4.4 Manažment kvality

Manažér kvality vytvára metodiky tvorby testov, spracovanie chýb a tvorbu úloh a metód overenia správnosti riešenia. Zároveň by mal merať kvalitu kódu a iných metrík súvisiacich s projektom. Kvalita sa neodvíja len od počtu nájdených chýb, ale aj od metrík ako čistota kódu, prehľadnosť a čitateľnosť kódu. Na zabezpečenie kvality využívame niekoľko druhov testovania. Manažér kvality analyzuje výsledky automatických testov.

4.5 Manažment plánovania a rozsahu projektu

V tímovom projekte hrá kľúčovú rolu plánovanie a rozsah projektu. Tomu predchádza analýza požiadaviek. Pokračujeme konštruktívnou diskusiou členov tímu a vlastníka produktu. Preto sme stanovili ciele a vytvorili metodiky pre ich dosiahnutie. Dekompozíciou úloh sme dosiahli vhodnú granularitu pre postupný vývoj finálneho systému. Vyhodnocovaním plnenia úloh v jednotlivých šprintoch sme získali predstavu o vhodnom rozsahu projektu a ďalších šprintov. Kľúčovým pre kontrolu úloh bola v našom prípade Jira. Tento nástroj nám umožnil spätné sledovanie a umožnil približný náhľad na plánovaný šprint.

Manažment rozsahu krátkodobého plánu znamenal plánovanie najbližšieho šprintu. Plánovanie rozsahu bolo uskutočnené počas stretnutia za účasti čo najväčšieho množstva členov tímu. Pri tvorbe plánu sa dívame aj na riziká, ktoré sú s vývojom softvérového produktu úzko späté. Dôležitou časťou plánovania je aj odhad času potrebného pre vykonanie úloh. Používateľský príbeh môže vykonávať aj viac členov tímu, v takomto prípade sa vytvárajú podúlohy. Počas stretnutia sú identifikované používateľské príbehy zapísané do nástroja Jira. Použitá metodika je zapísaná v prílohe metodiky.

4.6 Manažment komunikácie

Komunikácia v tíme je základnou činnosťou, ktorá zabezpečuje fungovanie tímu. Zlyhaním komunikácie môže dôjsť k závažným problémom vo fungovaní tímu. Komunikácia pomáha spájať členov tímu v jeden celok a pomáha preklenúť rozdielnymi povahami členov. Pri komunikácií vznikajú nezhody a chyby, ktorým sa vďaka manažmentu komunikácie snažíme vyhnúť.

Dohoda členov tímu je základným kameňom tímu. Cieľom tímovej komunikácie je nájsť konsenzus. Problém nesúhlasného názoru jedného alebo viacerých členov tímu môže viesť k strate záujmu o projekt. Dohode členov tímu je umožnená diskusiou kde je potrebné hovoriť o problémoch a hľadaní riešení. Vždy je vhodné navrhnúť riešenie problému a pomáhať členovi, ktorý má problém.

Členovia tímu potrebujú prehľad o dianí, prehľad o úlohách a o plnení úloh. Preto boli zvolené komunikačné prostriedky. O týchto prostriedkoch sa hlasovalo a členovia tímu majú povinnosť ich aktívne využívať. Nástroje pre komunikáciu sa dajú rozdeliť do rovín a to formálnej komunikácie a neformálnej komunikácie.

4.6.1 Formálna komunikácia

Hipchat
Na začiatku projektu bol zvolený tento komunikačný kanál. Tento kanál mal poskytnúť prepojenie nami používaných nástrojov (jira, github). Nástoj poskytuje mobilnú aplikáciu a vytváranie miestností pre delenie informácií. Od tohto nástroja sme upustili kvôli problémovej integrácií s nástrojom Jira.

Slack

Bol zvolený ako nástupca Hipchat-u. Tento nástroj nám umožnil všetky vyššie spomenuté možnosti spolu s notifikáciami o dianí v Jire aj na Github-e. Tento nástroj je používaný ako hlavný aj pre priamu komunikáciu medzi členmi tímu aj pre verejné zdieľanie informácií alebo verejné otázky. Tento nástroj poskytuje možnosť upozornení na mobil aj na email. Umožňuje aj opravy komentárov. Používame jeho voľne dostupnú verziu. Tento nástroj je zároveň aj komunikačným kanálom s vedúcim nášho tímu.

Facebook skupina

Bola vytvorená ako primárny komunikačný kanál. Kvôli nevhodnému spôsobu notifikovania a primárnemu určeniu Facebooku sa od neho upustili aj na neformálnej úrovni.

Jira

Systém Jira bol zvolený ako náš nástroj manažmentu úloh. Tento systém umožňuje vytváranie úloh, šprintov, určovanie zodpovedností, záznam odhadovaného času a záznam stráveného času. Aby sa predišlo nevedomosti o povinnostiach v tomto nástroji máme notifikácie v našim mailových schránkach. Jira je našim nástrojom pre záznam chýb a komunikáciu chýb s vývojárom príslušnej časti.

Tímové stretnutia

Stretnutia prebiehajú formálnou komunikáciou a za účasti čo najväčšieho množstva členov tímu. Hlavnou náplňou je prediskutovanie pridelených úloh, plánovanie šprintov, riešenie závažných otázok a diskusia k smerovaniu tímu.

Neformálna komunikácia

Patria sem komunikačné kanáli ako správy na Facebooku, mobilné správy a hovory. Táto časť spomína len komunikáciu, ktorej sa vyhýbame, alebo ju používame len výnimočne.

Teamviewer

V prípade, že je na úlohu pridelený väčší počet ľudí, ktorý sa nemôžu fyzicky stretnúť je tento nástroj dobrou pomôckou. Umožňuje vzdialené ovládanie počítača z iného počítača, alebo telefónu. Tento spôsob sa môže hodiť najmä pri pomoci s konfiguráciou prostredia a podobnými špecifickými úlohami. Ďalej je možné tento nástroj použiť aj pri párovom programovaní, ale hrozí konflikt pri ovládaní zdieľaného stroja a následné chyby vyplývajúce z konkurencie pri ovládaní. Tento nástroj preto pri párovom programovaní nie je odporúčaný.

Skype

V niektorých prípadoch sa môže hodiť telefonická komunikácia. Prípadom kedy sa táto komunikácia oplatí je chorý člen tímu, ktorý je členom párového programovania. Táto metóda sa používa len v krajných prípadoch kedy nie je dostupný žiadny iný člen tímu.

Osobné stretnutia

Predstavujú všetky stretnutia mimo formálnych tímových stretnutí. Vyznačujú sa prítomnosťou len časti tímu. Členovia sa môžu dohodnúť na neformálnom stretnutí aj mimo školy. Problémom, ale zostáva zdieľanie informácií, ktoré nezúčastnení členovia nezískajú. Predmetom diskusie by mali byť len informácie, ktoré nezúčastnený členovia nepotrebujú vedieť. Dôležitosť informácií je, ale relatívna a preto je vhodné robiť aj neformálne stretnutia s čo najväčšou účasťou.

4.7 Manažment rizík

Veľkým problémom vývoja je množstvo premenných, ktoré sa dynamicky menia. Tieto premenné vytvárajú potenciálne riziká, ktorým sa snažíme vyhnúť. Aby sme dopad a výskyt problémov minimalizovali sme zaviedli manažment rizík. Riziká, ktorým sa snažíme vyhnúť sú generické a špecifické.

Generické riziká môžu zasiahnuť ktorýkoľvek tím. Špecifické súvisia priamo s našim produktom alebo sa týkajú len jeho oblasti. Najčastejšie z nich sme identifikovali a umiestnili do nasledovnej tabuľky.

#	Popis	Riziko	Dopad	Pravdepodobnosť	Typ	Prevencia
1	Riziko zlého časového rozvrhu a plánovania	Nevhodné plánovanie, absencia plánovania	Stredný	Stredná	Generické	Agilný vývoj, Jira, diskusia
2	Riziko, duplicitnej práce, nevedomosť o tom, čo robia iný členovia tímu	Nadbytočná práca, nevykonané úlohy	Stredný	Stredná	Generické	Komunikácia, záznamy v Jire
3	Riziko vzniku chýb v programe, neprehľadný kód	Kvalita a udržateľnosť kódu	Stredný	Vysoká	Generické	Kontrola kódu, extrémne programovanie, code review
4	Riziko použitia zlej alebo nevhodnej architektúry, neviditeľnosť softvéru	Nevhodná architektúra	Vysoký	Stredná	Generické	Dôsledná analýza, diskusia s členmi tímu a vlastníkom produktu
5	Riziko výpadkov servera, preťaženie serveru, nízke SLA	Nevhodné testovanie	Stredný	Stredná	Generické	Unit testovanie, modulové testovanie, záťažové testovanie
6	Riziko nedorozumenia a nedodržania požiadaviek,	Nesplnenie špecifikácií	Vysoký	Nízka	Generické	Komunikácia s vlastníkom, prototyp pre overenie špecifikácie

	ktoré určil vlastník.					
7	Riziko nespokojnosti používateľov s novým systémom	Prosby o nepoužívanie systému	Stredný	Nízka	Špecifické	Pozitívna prezentácia systému, motivácia pre používanie, vhodný marketing
8	Riziko nedodania prototypu včas na použitie cvičeniam	Nutná úprava zmeny hodnotenia predmetu	Vysoký	Nízka	Špecifické	Kontrola úloh a jeden skúšobný týždeň na cvičeniach
9	Riziko problémov s programovaním v Ruby on Rails	Nedostatočná znalosť Ruby on Rails	Stredný	Vysoká	Špecifické	Stretnutia a prezentácie frameworku a jazyka Ruby on Rails, extrémne programovanie
10	Riziko nemožnosti testovania veľkým množstvom študentov	Projekt tesa nebude súčasťou žiadneho predmetu	Vysoký	Nízka	Špecifické	Stretnutia s vyučujúcimi vhodných predmetov

5 Sumarizácia šprintov

V tejto časti sa vyhodnotí plnenie úloh v danom šprinte a plnenie úloh jednotlivými členmi tímu.

5.1 1. Šprint – Bulbasaur

V prvom šprinte sme riešili základné funkcie systému pre overovanie vedomostí študentov. Rozbehali sme prostredia. Zostavili sme základné architektúry pre prototyp aj pre AlefTNG. Určili sme ďalšie smerovanie projektu a dátové modely. Prototyp mal od tohto momentu možnosť prihlásenie pomocou AIS LDAP, študenti dostali možnosť zobraziť si testy s otázkami, mohli odpovedať na otvorené otázky, cvičiaci mohol zobraziť aktuálne testy, poskytnúť prístupové heslo študentom a administrátor mohol pridávať otázky. V systéme AlefTNG boli pridané nasledovné funkcionality: študent si môže zobraziť test po zadaní hesla(alfanumerického kódu), študent mohol vidieť pridelené otázky, cvičiaci môže ukončiť test na cvičení (heslo sa stáva neplatným) a tým pádom študenti, ktorý to nestihli už nemôžu odoslať svoj test. Administrátor môže vkladať, meniť odstraňovať cvičenia. Špecifikácie boli overované na predmete operačné systémy, kde bol tento prototyp prvý krát použitý v ostrej prevádzke.

V tomto šprinte sme identifikovali problém s vyriešením cvičnej úlohy, ktorá bola presunutá do ďalšieho šprintu. Tieto úlohy sa ukázali ako zložité pre členov tímu, ktorí nemali skúsenosti s frameworkom Ruby on Rails.

Množstvo úloh bolo riešených aj mimo šprintu a riešili sa otázky, ktoré sa týkali fungovania tímu, dokumentácie a webovej stránky. Objavili sa aj prvé problémy a zmenil sa aj prístup ku komunikácií, vedeniu projektu.

5.2 2. Šprint - Ivysaur

Tento šprint znamenal posun hlavne v systéme AlefTNG. Študent v tomto systéme mohol už odpovedať na otvorené otázky, ktorá má v zadaní obrázok dokonca mohol odpovedať aj na výberové otázky (multichoice). Po odoslaní testovej odpovede už nebolo možné druhý krát písať absolvovaný test a študent mohol nahrádzať cvičenie. Bol vylepšený systém pre pridávanie cvičení, učiteľ a administrátor už nemohli písať testy.

Z pohľadu prototypu mohol cvičiaci zobraziť anonymizované odpovede študentov. Odpovede študentov sú zoskupené podľa odpovedí a je zobrazený ich skutočný počet. Zbieranie štatistiky naberá na dôležitosti a zvyšuje sa aj obľúbenosť prototypu medzi študentmi. Taktiež boli pridané aj otázky s viacerými možnosťami (multichoice).

Vyriešenie cvičnej úlohy bolo presunuté do ďalšieho šprintu. Táto úloha bol uzavretá pre všetkých, ktorý nepracujú s Ruby on Rails. Znalosť Ruby on Rails si vyžaduje čas a preto prebieha jeho štúdium. Venovali sme sa aj témam ako sú metodiky, testovanie a webová stránka, ktorá dostane nový dizajn. Ustálil sa komunikačný kanál a zaviedli sa nové metodiky.
Boli presunuté aj niektoré ďalšie úlohy, niektoré z nich kvôli nedostatočným informáciám, alebo prílišnej náročnosti šprintu. V prototype vnikli námietky voči niektorým úlohám. Takou úlohou je „Systém zobrazí

doplňujúce informácie o sebe“. Preto bola presunutá do ďalšieho šprintu. Ďalším špeciálnym prípadom je úloha „Študent dostane otázku podľa zadaného kľúča“ si vyžaduje lepšiu znalosť odporúčacích algorytmov AlefTNG a tým pádom aj frameworku Ruby on Rails. Ostatné úlohy sa nestihli spraviť kvôli veľkej náročnosti používateľských príbehov.

5.3 3. Šprint - Venusaur

V prototype boli ošetrené špeciálne prípady ako cvičiaci a administrátor, ktorý už po novom nemôžu písať test a bola pridaná funkčnosť pre prihlásenie bez použitia LDAP.

AlefTNG získal možnosť importu termínov cvičení a otázok vo formáte csv. Prototyp a AlefTNG sa týmto stávajú rovnocennými. Študenti dostávajú otázky podľa zadaného kľúča. Otázky, ktoré dostáva sú už riadené naším odporúčacím algoritmom. Do databázy je zapísaná každá odpoveď a ak existuje aj správna odpoveď je správnosť študentovej odpovede automaticky overená.

Z pohľadu cvičiaceho nastala len jedna zmena. Cvičiaci si môže zobraziť zoznam všetkých svojich cvičení v prehľadnej tabuľke.

5.4 4. Šprint – Charmander

V prototype pribudla už len možnosť zobrazenia doplňujúcich informácií o sebe. Prototyp je od tohto cenným zdrojom údajov a štatistík.

V AlefTNG dostal náš odporúčací algoritmus možnosť aby všetci študenti dostali zhodnú otázky alebo otázky v prípadoch kedy, že je otázka označená ako špeciálna. Vyučujúci môže kontrolovať dochádzku študentov priamo cez systém. Bola pridaná aj základná štatistika odpovedí po ukončení testu. Týmto krokom sa AlefTNG a prototyp stávajú veľmi podobnými systémami. Zároveň majú oba systémy podobný systém nahrávania otázok a cvičení a prenos údajov medzi nimi nepredstavuje žiadny problém.

Boli preberané aj témy ďalšieho testovania študentov. Vybraný predmet bude známy na konci piateho šprintu.

5.5 5. Šprint - Charmeleon

Tento šprint nám slúžil na zavedenie novej bázy znalostí na štýl wikipedie. Získali sme skúsenosti z manažmentov a zaviedli sme nové opatrenia pre spracúvanie úloh. Vymenili sme pôvodnú stránka za stránku s novým dizajnom.

Na našom serveri aktuálne bežia štyri webové aplikácie – AlefTNG, prototyp, webová stránka tímu a báza znalostí. Počas nasadzovania sme sa stretali s problémami s používateľskými účtami. Vďaka veľkému úsiliu, pomoci nášho vedúceho a porcií stráveného času sme sa dostali cez problémy. Zvyšný čas sme využili pre kompletizáciu našej dokumentácie.

6 Používané metodiky

Využívanie metodík je jedným zo stavebných kameňov úspešného a opakovateľného projektu. Ak sa používajú normované úkony je možné člena, ktorý aktuálne nemôže pracovať na projekte nahradiť iným členom pretože poznám postupy, ktorými dosiahnuť želaný výsledok. V tejto kapitole sú spísané metodiky, ktoré sme doteraz využili.
Všetky použité metodiky je možné nájsť v časti metodiky.

6.1 Metodika písania dokumentácie

Dokumentáciou sa rozumie každý text napísaný počas tímového projektu. Dokumentáciu tvoria všetci členovia tímu a preto je nutné ju štandardizovať. Kvôli dokumentácií je vhodné používať vzorové ukážky a určiť si spôsoby akými dokumentácie písať. Na vytvorenie dokumentácie používame Microsoft Office a jeho online verziu, ktorá je zdieľaná cez OneDrive.

6.2 Metodika dokumentovania úloh

Metodika je spätá s tvorbou dokumentácie k dielu. Túto metodiku používa každý kto dokumentuje používateľské príbehy. Jej úlohou je rovnaké zjednotiť spôsob akým je úloha opísaná. Táto úloha použitá pri dokumentovaní každého používateľského príbehu, ktorý bol dokončený.

6.3 Metodika tvorby metodík

Bola vytvorená v nadväznosti na predchádzajúcu metodiku. Určila aké sú povinné a voliteľné prvky. Podľa nej naši členovia tímu vytvárali ďalšie metodiky. Táto metodika zároveň slúžila ako vzor, v ktorom bolo možné zmeniť obsah a vytvoriť tak novú metodiku.

6.4 Metodika tvorby testovacích prípadov

Táto metodika doposiaľ nebola využitá. Jej hromadné využitie sa plánuje na najbližší šprint.

6.5 Metodika ohlasovania chýb v systéme Jira

Táto metodika je už aktuálne v platnosti. Jej využitie je vždy pri vzniku chyby v prototype alebo v AlefTNG. Chyby, ktoré sú ohlásené pomocou tejto metodiky poskytnú vývojárovi väčšinu potrebných informácií. Jej použitie môže zabrániť aj chybám kedy je testovaná stará verzia. A pomáha vývojárovi odhaliť pôvod chýb v čo najkratšom čase.

6.6 Metodika vytvárania úloh

Metodika, ktorú sme používali skôr než bola spísaná je základným kameňom nášho tímu. Používajú ju všetci členovia tímu pri každom stretnutí, každej práci so systémom Jira. Zahŕňa nielen vytváranie úloh, ale celý životný cyklus úlohy od vytvorenia po uzatvorenie a zápis stráveného času.

6.7 Metodika verziovania a udržiavania zdrojových kódov

Keďže využívame nástroj git, ktorý nepoznajú všetci členovia tak treba poznať konvencie a vyhnúť sa niektorých operáciám. Preto je tu postup, ktorý vyžaduje základné znalosti pojmov a príkazov. Vďaka tejto metodike vzniká menej konfliktov pri vývoji. Metodika opisuje nielen pridávanie prírastkov, ale aj vytváranie nových vetiev.

7 Globálna retrospektíva

Novinky na uvedenie	S čím prestať
Programovanie všetkých členov v Ruby on Rails	Vývoj prototypu
Rozkladať úlohy do šprintov rovnomerne	Neproduktívna debata
Tvorba technickej dokumentácie	Práce na poslednú chvíľu
Vylepšenie dizajnu	
Kontrola čitateľnosti kódu v Ruby on Rails	
Správne rozvrhnutie funkcionality medzi klienta a server	
Automatické testovanie	
Automatické nasadzovanie	
Tvorba inštalačnej príručky	
Tvorba vlastnej wiki	
Konzistentnejšia a podrobnejšia dokumentácia	
Poučenie sa z iných systémov a prototypu	
Dodržiavanie konvencií	
Zavádzanie užitočných metodík	
Zverejňovanie dokumentov v internej wiki	

Novinky na uvedenie

Počas prvých dvoch šprintov sme zistili, že vývoj aplikácie AlefTNG je zložitý pre členov tímu, ktorí nemajú skúsenosti s Ruby on Rails. Preto sme vývoj dočasne nechali na našich skúsenejších Ruby programátorov a rozhodli sa využiť neznalosť tohto frameworku v náš prospech. Keďže tento jazyk by mal byť ľahko čitateľný a samodokumentujúci tak sme sa rozhodli zaviesť kontrolu kódu pomocou čítania kódu. Vďaka týmto krokom sa zoznámime aj so systémom aj s frameworkom Ruby on Rails.

Úlohy v jednotlivých šprintoch sa líšia náročnosťou. Náročnosť na vedomosti a čas sa môže líšiť. Niekedy neodhadnutie náročnosti úlohy môže znamenať presun úlohy do ďalšieho šprintu. Preto je dôležité správne určiť náročnosť úloh a príbehy, ktoré majú vyššiu prioritu riešenia.

Aktuálny dizajn prvkov obrazovky sa môže meniť na základe požiadaviek, ktoré môžu prichádzať. Dizajn zobrazovania a využívania grafických prvkov môže byť zmenou vhodnou do letného semestra.

Správne určiť hrúbku klienta a serveru môže byť ťažšou úlohou než sa na prvý pohľad zdá. Akcie vykonávané na klientskom stroji musia byť vhodne vybrané a nemôžu ohroziť bezpečnosť aplikácie.

Automatické testovanie aplikácie by malo byť zavedené v dvoch krokoch. Prvým krokom sú testy v Ruby on Rails, ktoré sa vykonajú na uzavretom systéme a na inej databáze. Druhým krokom je testovanie používateľmi.

Ručné nasadzovanie na server zaberá veľa času. Tento čas sa dá využiť lepším spôsobom a preto sme rozhodli zaviesť automatické nasadzovanie pomocou skriptu.

Nástrojov pre vytvorenie stránky pre zber vedomostí je veľké množstvo, ale buď im chýbajú niektoré funkcionality alebo majú veľa funkcionalít a je ťažšie sa v nich orientovať. Rozhodnutie pre vlastnú wiki sa preto črtá ako najvhodnejšie riešenie. Aktuálne máme vypracovaný návrh vlastnej wiki postavenej na redakčnom systéme drupal. Testovacia prevádzka je plánovaná na začiatok letného semestra.

Postupným vývojom sa rozrastá aj dokumentácia a vznikajú stále nové poznatky, ktoré je dôležité zdokumentovať. S tým priamo súvisí aj tvorba technickej dokumentácie. Tvorba technickej dokumentácie sa líši pre každý jazyk. Dokumentáciu tvoria komentáre a generovaná technická dokumentácia. Nástroj pre generovanie technickej dokumentácie sa zvolí neskôr.
Ostatné systémy ku ktorým existuje spätná väzba nám poskytujú vhodnú pôdu pre získavanie inšpirácie a požiadaviek pre projekt. Poučenie sa na chybách nám pomáha posunúť sa vpred.
Keďže pracujeme s novým systémom a rozrastá sa počet metodík je nutné používať konvencie a metodiky, ktoré nám uľahčujú prácu a zabraňujú neskorším komplikáciám. Zavádzanie nových metodík je preto nutnosťou v každom tíme.
Aktuálny stav nám ukazuje neprehľadnosť zdieľania dokumentov v komunikačnej aplikácií slack. Táto splikácia nebola stavaná ako znalostná databáza. Preto je vhodné nové dokumenty zverejňovať v internej databáze znalostí.

S čím prestať

Keďže sa jednalo o prototyp tak jeho vývoj sa ukončuje. Práce na ňom neprestávajú keďže je zdrojom nových poznatkov a štatistík. Študenti a vyučujúci, ktorí tento systém využívajú nám môžu stále odovzdávať nové návrhy na zmenu a tak vylepšovať našu hlavnú aplikáciu.
Každé stretnutie so sebou nesie témy, ktoré sú mimo hlavnú preberanú tému. Je dôležité sa držať témy a neodbočovať. Šetrí sa čas členov tímu. Čas členov tímu je limitovaný ostatnými predmetmi a títo členovia tímu by nemali byť ukrátení o dôležité informácie.
Práca na poslednú chvíľu môže vzniknúť kvôli odkladaniu na neskôr kedy môžu prísť iné úlohy mimo predmetu tímový projekt. Tento problém sa dá vyriešiť skorším vypracovaním úloh. Najdôležitejšie je vypracovať úlohy, ktoré blokujú ostatných členov tímu.