

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Prehliadka kódov v tímových projektoch

(CodeReview)

Dokumentácia riadenia

Tím: Lucky Seven

Vedúci projektu: Ing. Karol Rástočný

Kontakt: tp.1314.07@gmail.com

Ak. Rok : 2013/2014

Autori: Bc. Zuzana Grešlíková

Bc. Matej Chlebana

Bc. Tomáš Kepič

Bc. Patrik Oriskó

Bc. Patrik Samuhel

Bc. Michael Scholtz

Bc. Július Skrisa

OBSAH

1	ÚVOD	1-1
2	PREDSTAVENIE ČLENOV TÍMU	2-1
3	PONUKA TÍMU	3-1
3.1	PREHLIADKA KÓDOV V TÍMOVÝCH PROJEKTOCH	3-1
4	ÚLOHY ČLENOV TÍMU	4-1
4.1	PODIEL PRÁCE ČLENOV NA JEDNOTLIVÝCH ČASTIACH DOKUMENTÁCIE RIADENIA.....	4-1
4.2	PODIEL PRÁCE ČLENOV NA JEDNOTLIVÝCH ČASTIACH DOKUMENTÁCIE K INŽINIERSKEMU DIELU	4-2
5	MANAŽMENT KVALITY	5-1
5.1	SPŮSOB PÍSANIA ZDROJOVÉHO KÓDU V PROJEKTE	5-1
5.2	PREHLIADKY KÓDU.....	5-2
5.3	SLEDOVANIE KVALITY KÓDU	5-2
6	MANAŽMENT MONITOROVANIA	6-1
6.1	ŠPRINT - SEMIENKO.....	6-1
6.2	ŠPRINT 2 – KORIENOK.....	6-1
6.3	ŠPRINT 3 - STONKA.....	6-2
6.4	ŠPRINT 4 - VODIČKA.....	6-3
6.5	ŠPRINT 4 - SLNIEČKO.....	6-3
6.6	RÝCHLOSŤ VÝVOJA PROJEKTU.....	6-4
7	MANAŽMENT PODPORY VÝVOJA	7-1
7.1	KONFIGURÁCIA	7-1
7.2	VERZIOVANIE.....	7-1
7.3	INICIALIZÁCIA POUŽITÝCH TECHNOLOGIÍ	7-2
7.4	PODPORNÉ ČINNOSTI	7-2
8	MANAŽMENT ROZVRHU (PLÁNOVANIE)	8-1
8.1	ROZVRH ZIMNÉHO SEMESTRA	8-1
8.2	DŮLEŽITÉ TERMÍNY V ZIMNOM SEMESTRI	8-1
8.3	SYSTÉM TFS.....	8-2
9	MANAŽMENT KOMUNIKÁCIE A ĽUDSKÝCH ZDROJOV	9-1
9.1	KOMUNIKAČNÉ KANÁLY	9-1
9.2	KOLABORÁCIA ČLENOV TÍMU	9-2
10	MANAŽMENT TVORBY DOKUMENTÁCIE	10-1
11	MANAŽMENT RIZÍK	11-1
11.1	IDENTIFIKÁCIA A RIEŠENIE RIZÍK	11-1
11.2	RIZIKO – NEDOSTATOK SKÚSENOSTÍ S TECHNOLOGIOU MICROSOFT MVC	11-1
12	METODIKY	12-1
12.1	METODIKA IDENTIFIKÁCIE A RIEŠENIA RIZÍK	12-1
12.2	METODIKA TVORBY DOKUMENTÁCIE	12-7
12.3	METODIKA ZBERU A SPRACOVANIA POŽIADAVIEK	12-13
12.4	METODIKA VERZIOVANIA A PÍSANIA KOMENTÁROV K ODOVZDANIAM	12-19
12.5	METODIKA TESTOVANIA	12-23
12.6	KONVENCIE PÍSANIA ZDROJOVÉHO KÓDU V PROJEKTE.....	12-28
12.7	METODIKA ITERACIÍ PROJEKTU	12-34

Obsah

12.8	METODIKA LOGOVANIA.....	12-39
13	ROZVRH LETNÉHO SEMESTRA	13-1
13.1	HRUBÝ PLÁN SEMESTRA.....	13-1
13.2	DETAILNÝ OPIS PLÁNU NA PRVÉ TÝŽDNE LETNÉHO SEMESTRA.....	13-1
14	PRÍLOHY	14-1
	PRÍLOHA A	14-2
	PRÍLOHA B – PREBERACÍ PROTOKOL	14-32

1 ÚVOD

Dokumentácia vznikla na opis riadenia vývoja projektu názvom Prehliadka kódov v tímových projektoch, ktorý je vyvíjaný na Fakulte informatiky a informačných technológií STU v akademickom roku 2013/2014. V Úvodnej časti tejto dokumentácie sú stručne opísané jednotlivé kapitoly dokumentu. Kapitola 2 obsahuje základné informácie o členov a o ich znalostí. Kapitola 3 obsahuje ponuku tímu na získanie témy „Prehliadka kódov v tímových projektoch“.

Ďalšie kapitoly obsahujú opisy manažérskej činnosti členov tímu a príslušné metodiky k nim.

Na koniec dokumentu ako prílohy sú pridané zápisy zo stretnutiach a preberací protokol.

2 PREDSTAVENIE ČLENOV TÍMU

Všetky členovia tímu sú absolventi bakalárskeho štúdia na FIIT STU, odbor Informatika.

Bc. Július Skrisa

Skúsenosti:

- Tvorba webových stránok: *JS, PHP, CSS*
- Tvorba testovacích scenárov: integračné a akceptačné testy (*Internet banking, ZUNO*)
- *C#* s využitím *.NET framework* (*SIP proxy server, sieťová UDP aplikácia*)

Bakalárska práca zameraná na elektronický obchod a zabezpečenie.

Bc. Michael Scholtz

Skúsenosti:

- Webové projekty: *PHP (+frameworkCodeIgniter), MySQL, HTML, CSS, JS (jQuery)*
- Jazyky: *C, C#, Java, Lua* všetkyna pokročilej úrovni

Bakalárska práca zameraná na vývoj aplikácie na platformu Android využívajúcej princípy obohatenej reality.

Bc. Patrik Samuhel

Skúsenosti:

- *C, C# (.NET Framework)* všetky na pokročilej úrovni
- *MySQL, HTML, CSS, Java, XML, ASP.NET*

Bakalárska práca zameraná na vyhľadávanie podobností v textových dokumentoch.

Bc. Matej Chlebana

Skúsenosti:

- *C#, .NET, Silverlight, C, Java, HTML, Javascript* všetky na pokročilej úrovni
- *MsSQL, MySQL (ORMmapovače: NHibernate, ADO.NET, LINQ to SQL)*
- Pracovné skúsenosti: *.NET programátor* – vývoj intranetových a internetových *.NET* aplikácií

Bakalárska práca zameraná na vytvorenie testovacieho systému pre školy. (*.NET, Silverlight*).

Bc. Zuzana Grešlíková

Skúsenosti:

- *C, C#, Java, MySQL (ORMmapovače: Hibernate), HTML, XML*, jazyk symbolických inštrukcií
- Člen študentskej organizácie *YNET* a projektu *HowKnow*.
- Organizácia tímových aktivít.

Bakalárska práca zameraná na evolučné algoritmy a klastrovanie dát.

Bc. Tomáš Kepič

Skúsenosti:

- *C, C#* všetky na pokročilej úrovni
- *Adobe Photoshop, CSS, HTML, Java, MySQL, UML*

Bakalárska práca zameraná na skúmanie a predpovedanie údajov zaznamenaných postupne v čase.

Bc. Patrik Oriskó

Skúsenosti:

- *C, C#, Java, JavaScript, Lua, HTML, CSS* všetky na pokročilej úrovni
- Tvorba web aplikácií pracujúcich s databázovými prvkami
- Vývoj softvéru na základe návrhového vzoru „*Visitor design pattern*“

Bakalárska práca zameraná na generovanie dokumentácie zo zdrojových kódov pre jazyk *Lua*.

3 PONUKA TÍMU

3.1 PREHLIADKA KÓDOV V TÍMOVÝCH PROJEKTOCH

Prečo nás zaujal?

- Projekt využiteľný v praxi
- Spolupráca s firmou Gratex
- Testovaná následne používaný spolužiakmi v tímových projektoch
- Programovanie v C# a ďalších technológiach od Microsoftu
- Rovnaký výber užitočných predmetov (študenti SI)

Prečo náš tím?

- Máme skúsenosti s programovaním v C# a .NET, pri volení prostredia pre vypracovávanie školských zadaní sme si väčšinou vybrali technológiu .NET

Naše nápady pre projekt:

- Hodnotenie kódov od jednotlivých používateľov
- Tagovanie kódu
- Vyhľadávanie kódu podľa tagov (napr. vyhľadanie riešenia určitého problému)
- Vyznačenie časti zdrojového kódu vytvoreného inými používateľmi v jednom súbore
- Možnosť pridať vysvetlivky k jednotlivým blokom pri vložení kódu

4 ÚLOHY ČLENOV TÍMU

Meno	Rola
Grešlíková	Vedúca tímu
Chlebana	Manažment podpory vývoja
Kepič	Manažment rizík
Oriskó	Manažment tvorby dokumentácie
Samuhel	Manažment rozvrhu (plánovania)
Scholtz	Manažment kvality
Skrisa	Manažment monitorovania projektu

4.1 PODIEL PRÁCE ČLENOV NA JEDNOTLIVÝCH ČASTIACH DOKUMENTÁCIE RIADENIA

Číslo kapitoly	Názov kapitoly	Vypracoval
1	ÚVOD	Oriskó
2	PREDSTAVENIE ČLENOV TÍMU	Celý tím
3	PONUKA TÍMU	Celý tím
4	ÚLOHY ČLENOV TÍMU	Celý tím
5	MANAŽMENT KVALITY	Scholtz
6	MANAŽMENT MONITOROVANIA	Skrisa
7	MANAŽMENT PODPORY VÝVOJA	Chlebana
8	MANAŽMENT ROZVRHU (PLÁNOVANIE)	Samuhel
9	MANAŽMENT KOMUNIKÁCIE A ĽUDSKÝCH ZDROJOV	Grešlíková
10	MANAŽMENT TVORBY DOKUMENTÁCIE	Oriskó
11	MANAŽMENT RIZÍK	Kepič
12.1	METODIKA IDENTIFIKÁCIE A RIEŠENIA RIZÍK	Kepič
12.2	METODIKA TVORBY DOKUMENTÁCIE	Oriskó
12.3	METODIKA ZBERU A SPRACOVANIA POŽIADAVIEK	Grešlíková
12.4	METODIKA VERZIOVANIA A PÍSANIA KOMENTÁROV K ODOVZDANIAM	Chlebana
12.5	METODIKA TESTOVANIA	Skrisa
12.6	KONVENCIE PÍSANIA ZDROJOVÉHO KÓDU V PROJEKTE	Scholtz
12.7	METODIKA ITERÁCIÍ PROJEKTU	Samuhel
12.8	METODIKA LOGOVANIA	Skrisa
13	ROZVRH LETNĚHO SEMESTRA	Samuhel
14	PRÍLOHY	Celý tím

4.2 PODIEL PRÁCE ČLENOV NA JEDNOTLIVÝCH ČASTIACH DOKUMENTÁCIE K INŽINIERSKEMU DIELU

Číslo kapitoly	Názov kapitoly	Vypracoval
1	ÚVOD	Skrisa
1.1	ŠTRUKTÚRA DOKUMENTU	Skrisa
1.2	ZADANIE PROJEKTU	Skrisa
2	STANOVENIE CIEĽOV	Samuhel
3	PRED PRVÝM ŠPRINTOM	Skrisa
4	SEMIENKO	
4.1	PRIHLÁSENIE POUŽÍVATEĽA AIS LOGINOM	Chlebana
4.2	PRIDANIE INFORMÁCIÍ O TÍME NA STRÁNKU PROJEKTU	Chlebana
4.3	URČENIE ADMINISTRÁTORA SYSTÉMU	Chlebana
4.4	PRIRADENIE POUŽÍVATEĽA K PROJEKTU	Grešlíková
4.5	ZOBRAZENIE ZOZNAMU PROJEKTOV POUŽÍVATEĽA	Grešlíková
4.6	NAČÍTANIE SÚBOROVEJ ŠTRUKTÚRY PROJEKTU DO STROMOVEJ REPREZENTÁCIE	Samuhel
4.7	NAČÍTANIE AST ŠTRUKTÚRY PROJEKTU DO STROMOVEJ REPREZENTÁCIE	Skrisa
4.8	PREHLIADANIE STROMOVEJ REPREZENTÁCIE	Scholtz
4.9	ZOBRAZENIE ZDROJOVÉHO KÓDU ZVOLENEJ ENTITY Z AST-RCS	Oriskó
4.10	ZMENA HESLA POUŽÍVATEĽA	Chlebana
4.11	REGISTRÁCIA EMAILU POUŽÍVATEĽA	Chlebana
4.12	ZVÝRAZNENIE SYNTAXE ZOBRAZENÉHO KÓDU	Oriskó
4.13	ZOBRAZENIE ZOZNAMU ODOVZDANÍ PROJEKTU	Kepič
5	KORIENOK	
5.1	ZOBRAZENIE ZOZNAMU PROJEKTOV POUŽÍVATEĽA Z NOVEJ VERZIE AST-RCS	Grešlíková
5.2	ROZBAĽOVATEĽNÁ STROMOVÁ ŠTRUKTÚRA	Scholtz
5.3	PRIRADENIE POUŽÍVATEĽA K PROJEKTU Z NOVEJ VERZIE AST-RCS	Grešlíková
5.4	ZOBRAZENIE STROMOVEJ ŠTRUKTÚRY VEDĽA ZOBRAZENIA ZDROJOVÉHO KÓDU	Scholtz
5.5	ZOBRAZENIE GRAFU ODOVZDANÍ PROJEKTU	Skrisa
5.6	ZOBRAZENIE ZOZNAMU ZMIEN VYKONANÝCH VO ZVOLENOM ODOVZDANÍ	Samuhel
5.7	ZOBRAZENIE ZDROJOVÉHO KÓDU ZMENENEJ ENTITY V ODOVZDANÍ SO ZVÝRAZNENÝMI ZMENAMI	Chlebana
5.8	NAČÍTANIE HISTÓRIE SÚBOROVEJ ENTITY Z AST-RCS	Grešlíková
5.9	NAČÍTANIE HISTÓRIE ENTITY ZDROJOVÉHO KÓDU Z AST-RCS	Kepič
5.10	ZOBRAZENIE HISTÓRIE ENTITY VO FORME TABUĽKY	Kepič
5.11	POROVNANIE ZDROJOVÉHO KÓDU DVOCH ZVOLENÝCH VERZIÍ ENTITY Z AST-RCS	Oriskó
5.12	NAČÍTANIE SÚBOROVEJ ŠTRUKTÚRY PROJEKTU DO STROMOVEJ REPREZENTÁCIE PRE NOVÚ VERZIU AST-RCS	Samuhel
5.13	NAČÍTANIE AST ŠTRUKTÚRY PROJEKTU DO STROMOVEJ REPREZENTÁCIE PRE NOVÚ VERZIU AST-RCS	Skrisa

4 Úlohy členov tímu

5.14	ZOBRAZENIE ZDROJOVÉHO KÓDU ZVOLENEJ ENTITY Z NOVEJ VERZIE AST-RCS	Oriskó
5.15	ZOBRAZENIE ZOZNAMU ODOVZDANÍ PROJEKTU V NOVEJ VERZII AST-RCS	Kepič
5.16	ÚSPEŠNÉ PRIHLÁSENIE S NEPLATNÝM HESLOM	Chlebana
6	STONKA	
6.1	VYTVORENIE VIZUÁLU STRÁNKY	Samuhel
6.2	VYTVORENIE KONTEXTOVÉHO MENU PRE POLOŽKY	Scholtz
6.3	KONTROLA PRÍSTUPU K DÁTAM	Chlebana
6.4	FILTRÁCIA HISTÓRIE ODOVZDANÍ	Kepič
6.5	STRÁNKOVANIE HISTÓRIE ODOVZDANÍ	Skrisa
6.6	PRIRADENIE PROJEKTOV K POUŽÍVATEĽOVI	Grešlíková
6.7	PRIRADENIE POUŽÍVATEĽOV K PROJEKTU	Grešlíková
6.8	URČENIE ADMINISTRÁTOROV SYSTÉMU	Grešlíková Chlebana
6.9	DEFINOVANIE AKTÍVNYCH ODKAZOV V KONTEXTOVOM MENU	Oriskó
6.10	ÚPLNÉ ZOBRAZENIE ČÍSLOVANIA RIADKOV	Scholtz
6.11	VIZUALIZÁCIA ROZBALENIA UZLA STROMU	Scholtz
6.12	PRÁZDNE RIADKY NA ZAČIATKU A KONCI KÓDU	Oriskó
7	CELKOVÝ POHĽAD NA PROJEKT	
7.1	TVORBA SOFTVÉROVÉHO PROJEKTU	Scholtz
7.2	TVORBA DOKUMENTÁCIE	Scholtz
8	VODIČKA	
8.1	ZOBRAZENIE ZDROJOVÉHO KÓDU SKRÝVA DOKUMENTAČNÉ TAGY	Oriskó
8.2	CHÝBAJÚCI OBRÁZOK KONTEXTOVÉHO MENU V STROME	Scholtz
8.3	PRI ZOBRAZENÍ POROVNANIA SÚBORU BEZ HISTÓRIE NASTANE CHYBA	Samuhel
8.4	PÍSMO V ZOBRAZENÍ CHANGESETOV SA PREKRÝVA	Skrisa
8.5	OBNOVENIE SEDENIA PRE ÚDAJE PROJEKTU	Samuhel
8.6	SÚBEŽNÉ SCOLLOVANIE V POROVNANÍ ZDROJOVÝCH KÓDOV	Samuhel
8.7	ZVÝRAZNENIE ENTITY V ZDROJOVOM KÓDE	Oriskó
8.8	ZBALENIE A ROZBALENIE CELÉHO STROMU	Scholtz
8.9	INFRAŠTRUKTÚRA PRE ZAZNAMENÁVANIE UDALOSTÍ	Skrisa
8.10	ODDELENIE RCS PROJEKTOV OD LOGICKÝCH PROJEKTOV	Chlebana
8.11	SPRÁVA LOGICKÉHO PROJEKTU	Kepič
8.12	SPRÁVA POŽIADAVIEK O ZAČLENENIE RCS PROJEKTU K LOGICKÉMU PROJEKTU	Scholtz
8.13	SPRÁVA POUŽÍVATEĽOV LOGICKÉHO PROJEKTU	Grešlíková
8.14	SPRÁVA ALIASOV POUŽÍVATEĽA	Oriskó, Chlebana
8.15	NAČÍTANIE ZOZNAMU ZNAČIEK PRIRADENÝCH PREHĽADANEJ SÚBOROVEJ ENTITE	Kepič
8.16	ZOBRAZENIE PROJEKTOV - NASTAVIŤ PEVNÚ ŠÍRKU ZOZNAMOV	Grešlíková
8.17	PROJEKTY SA ADMINISTRÁTOROVI NAČITAJÚ AŽ KEĎ SA PRIHLÁSI DRUHÝKRÁT	Grešlíková
9	SLNIEČKO	
9.1	KONTROLA PROJEKTOVÝCH PRÁV	Chlebana

4 Úlohy členov tímu

9.2	ZMAZANIE STARÝCH PROJEKTOV	Chlebana
9.3	VYTVORENIE A NAŠTÝLOVANIE SYSTÉMOVÉHO ADMINISTRÁČNÉHO MENU	Samuhel
9.4	VÝŠKA RIADKOV PRI ZVÝRAZŇOVANÍ KÓDU NIE JE URČENÁ SPRÁVNE	Kepič
9.5	DOPLNENIE INFORMÁCIÍ O POŽIADAVKE NA ZAČLENENIE REPOZITÁRA DO PROJEKTU	Scholtz
9.6	ZOBRAZENIE KÓDU PO KLIKnutí NA MENO SÚBOROVEJ ALEBO KÓDOVEJ ENTITY V STROME	Scholtz
9.7	ZVÝRAZNENIE ENTITY V KÓDE ŽLTÝM OBDĹŽNIKOM	Oriskó
9.8	ZLÚČENIE MENNÝCH PRIESTOROV V AST STROME	Skrisa
9.9	ZOBRAZENIE TABUĽKY HISTÓRIE ENTITY VEDĽA STROMU	Oriskó
9.10	PREMENOVANIE POLOŽIEK COMPARE, HISTORY A GRAPH V KONTEXTOVOM MENU V STROME	Oriskó
9.11	KONTEXTOVÉ MENU V AST STROME A HISTÓRII	Scholtz
9.12	VÝBER IBA DVOCH VERZIÍ ENTÍT NA POROVNANIE	Oriskó
9.13	ZACHOVANIE HLAVNÉHO POHĽADU PO ZMENE REPOZITÁRA	Chlebana
9.14	SYSTÉMOVÁ SPRÁVA PROJEKTOV A REPOZITÁROV	Grešlíková
9.15	ŠTATISTIKY ZNAČIEK SÚ VŽDY POČÍTANÉ PRE REPOZITÁR WHEREISMYCODE	Samuhel
9.16	SPRÁVA ALIASOV POUŽÍVATEĽA	Oriskó
9.17	ZÁKLADNÉ ŠTATISTIKY NAD PROJEKTOM	Samuhel
9.18	USER S ADMINISTRÁTORSKÝMI PRÁVAMI SI MÔŽE ZRUŠIŤ ADMINISTRÁTORSKÉ PRÁVA	Grešlíková
9.19	STRÁNKOVANIE PRI NAČÍTANÍ AST PROJEKTOV DO DATABÁZY	Samuhel
9.20	FILTRÁCIA HISTÓRIE ODOVZDANÍ - DOPRACOVANIE	Kepič
10	CELKOVÝ POHĽAD NA PROJEKT PO PIATOM ŠPRINTE	Grešlíková Skrisa
11	STANOVENIE GLOBÁLNYCH CIEĽOV PROJEKTU NA LETNÝ SEMESTER	Scholtz
12	KVETINÁČIK	
12.1	SPREHĽADNENIE SPRÁVY POŽIADAVIEK O PRIRADENIE REPOZITÁRA K PROJEKTU	Scholtz
12.2	ZJEDNOTENIE ZOBRAZENIA ŠTRUKTÚRY REPOZITÁRA SO ZOBRAZENÍM SÚBOROV V ODOVZDANÍ	Chlebana
12.3	SPRÁVA PRÁV POUŽÍVATEĽOV PRIRADENÝCH K PROJEKTU	Grešlíková
12.4	ODDELENIE ŠTATISTÍK OD ZOZNAMU POUŽÍVATEĽOV PROJEKTU	Samuhel
12.5	INFORMOVANIE O NEXITUJÚCICH ŠTATISTIKÁCH PRE ZVOLENÝ FILTER	Samuhel
12.6	ZOBRAZENIE INFORMÁCIE O PREBIEHAJÚCOM NAČÍTANÍ KÓDU	Scholtz
12.7	UMOŽNIŤ PRECHOD NA ZADANÚ URL	Chlebana
12.8	NAČÍTANIE VŠETKÝCH TYPOV SÚBOROV	Samuhel
12.9	ZOBRAZENIE INFORMÁCIE O PREBIEHAJÚCOM NAČÍTANÍ ŠTATISTÍK	Samuhel
12.10	ZAČLENENIE ODOSLANIA POŽIADAVKY O PRIPOJENIE REPOZITÁRA POD JEDNOTNÉ ROZHRAŇENIE SPRÁVY PROJEKTU	Scholtz
12.11	FULLTEXTOVÉ VYHĽADÁVANIE NAD MENAMI SÚBOROVÝCH A KÓDOVÝCH ENTÍT	Oriskó
12.12	ZOBRAZENIE ALIASOV PRI ŠTATISTIKÁCH	Samuhel

4 Úlohy členov tímu

12.13	ZOBRAZENIE ZOZNAMU ZNAČIEK	Kepič
12.14	FILTROVANIE ZOZNAMU ZNAČIEK PODĽA ICH TYPU	Kepič
12.15	PRIRADENIE PROFILU POUŽÍVATEĽSKÝCH ZNAČIEK K PROJEKTU	Samuhel
12.16	PRESUNUTIE NA MIESTO V KÓDE, KTORÉMU JE PRIRADENÁ VYBRANÁ ZNAČKA	Oriskó
12.17	INFORMOVANIE O PREBIEHAJÚCOM NAČÍTANÍ PRI HISTÓRII	Chlebana
12.18	RESET HESLA	Grešlíková
12.19	INFORMOVANIE O PREBIEHAJÚCOM NAČÍTANÍ ŠTRUKTÚRY PROJEKTU	Scholtz
12.20	ZDIEĽANIE POMOCOU ODKAZU NA ZDROJOVÝ KÓD ENTITY	Skrisa
13	PRVÝ LÍSTOK	
13.1	PO PRIHLÁSENÍ DO SYSTÉMU JE ZOBRAZENÝ JAVASCRIPT	Chlebana
13.2	ZOBRAZENIE ZMIEN V KÓDE V ZOBRAZENÍ HISTÓRIE	Chlebana
13.3	ÚPRAVA DIZAJNU SPRÁVY PRÁV POUŽÍVATEĽOV PRIRADENÝCH K PROJEKTU	Grešlíková
13.4	PRIDANIE ODKAZOV NA ZNAČKY ZO ZOZNAMU ZNAČIEK	Kepič
13.5	ZOZNAM SPRÁV	Kepič
13.6	SPRÁVA NOTIFIKÁCIÍ	Grešlíková
13.7	VYTVORENIE A ZMAZANIE NOTIFIKÁCIE O VYTVORENÍ ZNAČKY	Scholtz
13.8	REFACTORING ŠTATISTÍK A STROMOVEJ ŠTRUKTÚRY	Samuhel
13.9	REFACTORING HISTÓRIE A ZOBRAZENIA ZDROJOVÝCH KÓDOV	Oriskó
13.10	ZVÝŠENIE PREHLADNOSTI ROZHRANIA	Samuhel
13.11	ÚPRAVA URL ODKAZOV NA SÚBORY	Scholtz
13.12	VYTVÁRANIE ZNAČIEK	Skrisa
13.13	SYNCHRÓNNE SCROLLOVANIE PRI ZOBRAZENÍ ZMIEN V SÚBORE	Chlebana
14	DRUHÝ LÍSTOK	
14.1	ROZSIAHLE CODEREVIEW	Chlebana
14.2	ZOBRAZENIE KONKRÉTNEJ NOTIFICATION MESSAGE V PARCIÁLNO M VIEW	Kepič
14.3	ZOZNAM SPRÁV MÁ BYŤ GLOBÁLNY	Kepič
14.4	ZMENA ODKAZU NA ZNAČKU V ZOZNAM E ZNAČIEK	Kepič
14.5	V ZNAČKÁCH ZOBRAZOVAŤ MENO AUTORA PODĽA ZOZNAMU ALIASOV (NIE LOGIN)	Kepič
14.6	PO FILTROVANÍ ZOZNAMU ZNAČIEK OSTÁVA V DROPDOWN LISTE ZOBRAZENÁ ROVNAKÁ MOŽNOSŤ	Kepič
14.7	KONTROLA TYPOV ATRIBÚTOV PRIDÁVANÝCH INFORMAČNÝCH ZNAČIEK PODĽA PROFILU ZNAČKY	Skrisa
14.8	PODPORA PRE ZOBRAZENIE ZNAČIEK UKOTVENÝCH POMOCOU SELEKTORA TYPU TEXTUALCONTEXTSELECTOR	Skrisa
14.9	PRI ZOBRAZENÍ REQUESTOV SA ZOBRAZÍ TABUĽKA AJ KEĎ NIE JE ŽIADEN REQUEST	Scholtz
14.10	ZMENIŤ FORMÁT INFORMÁCIE O PREBIEHAJÚCOM NAČÍTANÍ ŠTATISTÍK	Scholtz
14.11	NASTAVENIE PROFILU ZNAČIEK SA VŽDY INICIALIZUJE NA "BASEPROFILE"	Samuhel
14.12	SPOJIŤ ŠTATISTIKY AJ PRE MENÁ S RÔZNOU VEĽKOSŤOU PÍSM A	Samuhel
14.13	OKIENKO CONTEXT MENU – POSITION	Samuhel

4 Úlohy členov tímu

14.14	LINKY NA ZVÝRAZNIŤ (ODDELIŤ OD TEXTU) PODČIARKNUTÍM ČIARKOVANOU (BODKOVANOU) ČIAROU	Samuhel
14.15	PRI ZOBRAZENÍ ZMENENÝCH SÚBOROV V ODOVZDANÍ 2541 DÔJDE K CHYBE	Chlebana
14.16	ZMENA DIZAJNU DOMOVSKÉJ STRÁNKY PROJEKTOV	Oriskó
14.17	ZOBRAZENIE INFORMÁCIÍ O POČTE VÝSLEDKOV VYHĽADÁVANIA V SÚBOROCH	Oriskó
14.18	SPUSTENIE VYHĽADÁVANIA PRI STLAČENÍ KLÁVESY ENTER	Oriskó
14.19	NAHRADENIE STRÁNKY UNDER CONSTRUCTION DOMOVSKOU STRÁNKOU PROJEKTOV	Oriskó
14.20	VYMENIŤ RADIO BUTTONY V TABUĽKE ALIASOV ZA DELETE TLAČIDLÁ	Oriskó
14.21	MENU MÁ PRIVYSOKÝ Z-INDEX - NASTAVIŤ V SÚLADE S OSTATNÝMI PRVKAMI	Samuhel
14.22	PREROBIŤ ODKAZ NA SPRÁVU ALIASOV, TAK ABY BOL INTUITÍVNY	Oriskó
14.23	TLAČIDLO "ADD NEW NOTIFICATION" V ADMINISTRÁCIÍ NOTIFIKÁCIÍ NAFORMÁTOVAŤ AKO TLAČIDLO	Grešlíková
14.24	ZRUŠENIE TABUĽKY V TABUĽKE PRE SPRÁVU POUŽÍVATEĽOV PROJEKTU A PRE SPRÁVU REPOZITÁROV	Grešlíková
14.25	ODSTRÁNENIE ČIARKOVANÉHO RIADKU V TABUĽKE ADMINISTRÁTOROV	Grešlíková
14.26	PRIAME ODOSIELANIE ZMIEN V SPRÁVE POUŽÍVATEĽOV BEZ POUŽITIA SUBMIT TLAČIDLA	Grešlíková
14.27	V TABUĽKE PRE SPRÁVU POUŽÍVATEĽOV PROJEKTU A PRE SPRÁVU PROJEKTOV NAHRADIŤ TLAČIDLO SELECT PRIAMYM VÝBEROM PROJEKTU	Grešlíková
14.28	TLAČIDLÁ PRE ÚPRAVU, ZMAZANIE A ZAPNUTIE EMAILOV V SPRÁVE NOTIFIKÁCIÍ	Grešlíková
14.29	DATEPICKER V ŠTATISTIKÁCH	Samuhel
14.30	ZMENA DIZAJNU TLAČIDIEL	Samuhel
14.31	ZOBRAZENIE IKONIEK DOMOV A NASTAVENIA VPRAVO VEDĽA IKONY ODHLÁSENIA	Samuhel
14.32	SPRACOVANIE NOTIFIKÁCIE O VYTVORENÍ/ZMAZANÍ ZNAČKY	Scholtz
14.34	[BUG] LOADTAGS.TAGSFROMFILE() - CAST EXCEPTION	Kepič
15	TRETÍ LÍSTOK	
15.1	NA ZAČIATKU ZDROJOVÉHO SÚBORU JE ZOBRAZENÝ RIADOK NAVYŠE	Oriskó
15.2	VŠETKY PODSTRÁNKY MAJÚ TITLE "WELCOME TO CODEREVIEW WEB"	Oriskó
15.3	ZMENIŤ JS ZVÝRAZŇUJÚCI KÓD	Oriskó
15.4	DIALÓGY PRE PRIDANIE ZNAČKY A ZOBRAZENIE NOTIFIKÁCIE ZJEDNOTIŤ S POUŽITÍM DIALOGU Z JQUERY	Oriskó
15.5	ZJEDNOTENIE VÝŠKY STROMU A ZDROJOVÉHO KÓDU	Scholtz
15.6	ZMAZANIE ZNAČIEK SO SELEKTOROM INÝM AKO LINESELECTOR	Scholtz
15.7	OPRAVENIE SYNCHRÓNNEHO SCROLLOVANIA	Scholtz
15.8	ZMAZANIE CR PROJEKTU	Chlebana
15.9	MASTER ALIAS POUŽÍVATEĽOV	Chlebana
15.10	FAREBNÉ ZVÝRAZNENIE AUTORA ODOVZDANIA V HISTÓRII A FILTROVANIE HISTÓRIE PODĽA AUTOROV	Skrisa

4 Úlohy členov tímu

15.11	ÚPRAVA DIZAJNU ADD TAG	Samuhel
15.12	PRI ZMENE VEĽKOSTI OKNA SA POKAZÍ DIZAJN ŠTATISTÍK	Samuhel
15.13	FILTROVANIE ZOZNAMU ZNAČIEK PODĽA TYPOV A AUTOROV	Kepič
15.14	AUTOMATICKÁ AKTUALIZÁCIA POČTU NEPRIJATÝCH NOTIFIKÁCIÍ	Kepič
15.15	ZOBRAZENIE "LOADERU" PRI NAČÍTANÍ ZOZNAMU ZNAČIEK	Kepič
15.16	FILTER CR A AST PROJEKTOV V SPRÁVE REPOZITÁROV A FILTER PROJEKTOV A POUŽÍVATEĽOV V SPRÁVE POUŽÍVATEĽOV PROJEKTU	Grešlíková
15.17	ZOBRAZIŤ MENÁ POUŽÍVATEĽOV V SPRÁVE PRÁV POUŽÍVATEĽOV PROJEKTU	Grešlíková
15.18	USPORIADANIE ZOZNAMOV PODĽA ABECEDY	Grešlíková
15.19	ZMENA DIZAJNU DOMOVSKÉJ STRÁNKY PROJEKTOV	Oriskó
15.20	NAHRADENIE STRÁNKY UNDER CONSTRUCTION DOMOVSKOU STRÁNKOU PROJEKTOV	Oriskó
15.21	CONTEXT MENU V STROME JE PREKRYTÉ KÓDOM	Samuhel
15.22	OŠETRENIE PRIDANIA PRÁZDNEHO ALIASU (PRÁZDNY STRING)	Oriskó
15.23	[BUG] ZNAČKY SA NEZOBRAZUJÚ KOREKTNE	Kepič
16	ŠTVORLÍSTOK	
16.1	NOTIFIKÁCIA SA PRI DOMOVSKÉJ STRÁNKE PROJEKTU ZOBRAZILA AKO HLAVNÁ STRÁNKA (NIE DIALÓG)	Samuhel
16.2	PO PRECHODE NA ADRESU VYGENEROVANÚ NOTIFIKÁCIU JE ODMIETNUTÝ PRÍSTUP	Scholtz
16.3	PRI ZOBRAZENÍ ZMIEN VO ZVOLENÝCH ODOVZDANIACH SÚBORU SA ZOBRAZÍ AJ JEDEN VYBRATÝ SÚBOR	Oriskó
16.4	ZOBRAZENIE AUTORA POMOCOU DISPLAY NAME PRI ZNAČKÁCH	Kepič
16.5	UPRAVIŤ ZOBRAZENIE ALL NOTIFICATION MESSAGES - ZARADENIE, SHOW MORE	Kepič
16.6	ZLE PORADIE NOTIFIKÁCII VO VYSKAKOVACOM "MENU" A AJ KEĎ SA ĎALEJ KLIKNE NA "SEE ALL". "SEE ALL" TREBA PREPÍSAŤ NA "SHOW ALL"	Samuhel
16.7	ZOBRAZENIE <i>DISPLAY NAME</i> PRI SPRÁVE POUŽÍVATEĽOV	Grešlíková
17	CELKOVÝ POHĽAD	Celý tím
18	ČO SME NESTIHLI	Skrisa, Scholtz
19	ČO SME SA NAUČILI	Oriskó
20	PRÍLOHA – POUŽÍVATEĽSKÁ PRÍRUČKA	Samuhel
21	PRÍLOHA – SYSTEMOVÁ PRÍRUČKA	Scholtz

5 MANAŽMENT KVALITY

Pozíciu manažéra kvality v tíme zastáva člen Bc. Michael Scholtz. Jeho úlohou bola hlavne kontrola kvality samotného projektu a kódu, ktorý je produkovaný členmi tímu.

5.1 SPÔSOB PÍSANIA ZDROJOVÉHO KÓDU V PROJEKTE

Jednou z úloh bolo zabezpečiť, aby kód napísaný všetkými členmi tímu bol jednotne písaný, formátovaný a aby jeho jednotný formát, ktorý je braný aj z vizuálneho hľadiska, zabezpečil ľahšiu čitateľnosť a pochopiteľnosť.

Ako prvé sa rozhodovalo o samotnom jazyku v rámci zdrojových kódov. V tomto bol tím nerozhodný. Polovica členov chcela používať čisto angličtinu, iní chceli pripustiť používanie slovenského jazyka aj v rámci kódu v názvoch tried, metód a premenných. Nakoniec bolo rozhodnuté, že samotný kód bude písaný čisto v anglickom jazyku, pričom komentáre k jednotlivým častiam kódu budú písané v slovenskom jazyku bez diakritiky.

Následne bol poskytnutý pre všetkých členov dokument "C# CodingConventions"¹, ktorý dopodrobna zhŕňa spôsob formátovania zdrojového kódu napísaného v programovacom jazyku C#. Tento dokument pojednáva o tom, ako tvoriť názvy premenných, tried aj metód, akým spôsobom majú byť odsadzované jednotlivé úrovne kódu, alebo akým spôsobom má byť formátovaný blok kódu.

Tieto pravidlá boli neskôr zhrnuté v samostatnom dokumente "Konvencie písania zdrojového kódu v projekte", ktorý pojednáva o najdôležitejších pravidlách týkajúcich sa nielen jazyka C#, ale hovorí aj o tom, akým spôsobom má byť písaný kód v jazyku CSS a akým spôsobom majú byť písané súbory CSHTML. Okrem toho tento dokument definuje aj samotnú štruktúru projektu a rozdelenie jednotlivých zdrojových súborov do podadresárov, čím je zaručená vyššia konzistencia v rámci projektu.

Nasledujúca ukážka (obr. č. 1) zobrazuje niekoľko pravidiel aplikovaných na kus kódu:

```
// Ukazkový komentár pre ukazkový kód.  
if (userId == 0)  
{  
    return true;  
}  
else  
{  
    return false;  
}
```

Obr. č. 1 Ukážka správne naformátovaného kódu.

¹ C# CodingConventions. Dostupné online: <<http://msdn.microsoft.com/en-us/library/vstudio/ff926074.aspx>>. 18.11.2013

5.2 PREHLIADKY KÓDU

Členovia tímu majú možnosť svoj kód navzájom hodnotiť a vyjadrovať sa k nemu. Na to slúži nástroj PerConIK. PerConIK je jednoduchý nástroj, vďaka ktorému môže každý člen tímu hodnotiť kód, resp. sa k nemu vyjadrovať, komentovať ho, značkovať ho. Takýmto spôsobom môžu členovia efektívnejšie kolaborovať pri návrhu zmien a zlepšení v rámci kódu.

Pre všetkých členov bol poskytnutý v zdieľanom adresáry inštalačný súbor ako aj návod, ktorý jednoducho po krokoch vysvetľuje, akým spôsobom je potrebné tento nástroj nastaviť, aby bol plne funkčný. Nastavenia treba vykonať ako v samotnom nástroji, tak aj priamo v programovacom prostredí Microsoft VisualStudio 2012. Ten následne dokáže podporovať priame vkladanie značiek.

Samotné značky začínajú dvomi lomítkami ("//") a teda sú vo forme komentára. Existuje niekoľko druhov značiek, pričom každá značka má iný význam. Tieto značky obsahujú aj parametre, ktoré bližšie špecifikujú samotnú vlastnosť značky, napr. Priority, ktorá určuje prioritu vykonania akcie danej značky.

5.3 SLEDOVANIE KVALITY KÓDU

V neposlednom rade bola sledovaná aj kvalita kódu členov tímu a bolo navrhnutých niekoľko vylepšení. Tieto vylepšenia boli zaznamenané aj formou značiek v kóde pomocou nástroja PerConIK.

Okrem toho prebehlo niekoľko diskusií, kedy sa riešili problémy technického rázu a navrhovalo sa, akým spôsobom je čo najlepšie a najefektívnejšie vyriešiť existujúci problém.

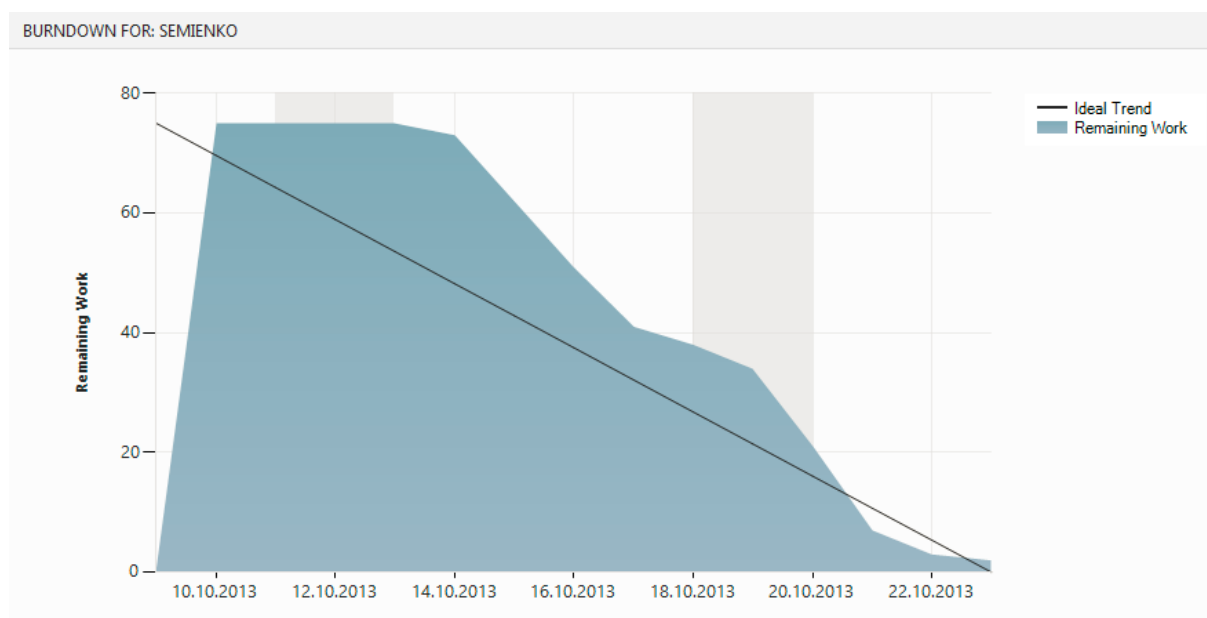
6 MANAŽMENT MONITOROVANIA

6.1 ŠPRINT - SEMIENKO

Celkom príbehov: 13

Dokončených príbehov: 12

V prvej polovici šprintu sa členovia tímu oboznamovali s prostredím a s frameworkom ASP.NET MVC 4. Táto skutočnosť sa odrazila na grafe v prvých 4 dňoch kedy bola vyprodukovaná minimálna časť. Po oboznámení sa z prostredím bola práca rozbehnuté v správnom tempe tak aby vykompenzovala začiatkové zdržanie. Ďalším elementom, ktorý zbrzdil vývoj príbehov je fakt, že jednému z členov tímu sa vyskytla porucha na zariadení, ktoré nevyhnutne potreboval. Napriek týmto kritickým elementom bol šprint ukončený iba pár hodín po termíne.



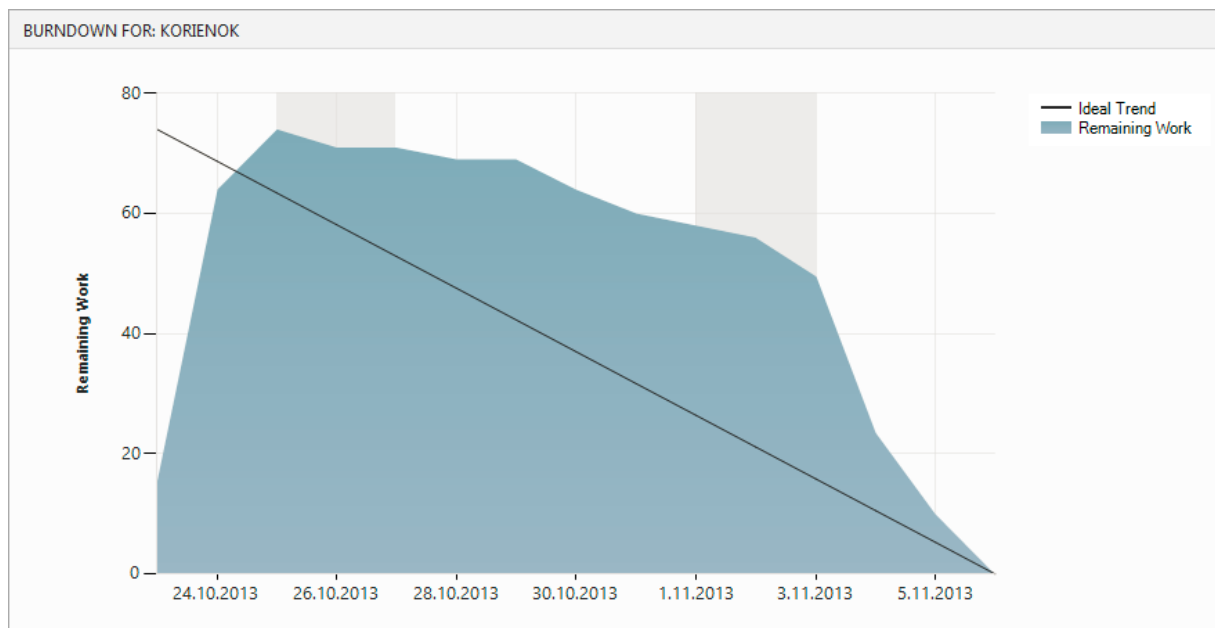
Obr. č. 2Burndownchart šprintu semienko

6.2 ŠPRINT 2 – KORIENOK

Celkom príbehov: 16

Dokončených príbehov: 16

V prvom týždni druhého šprintu sa vykonávali analýzy vhodných knižníc pre niektoré príbehy ako aj úprava existujúcich príbehov na novú verziu AST-RCS systému. Členovia tímu sa venovali týmto úpravám a v niektorých prípadoch bola podcenená potrebná práca na vykonanie týchto úprav. Z týchto dôvodov vyzerá graf v prvom týždni bez väčšieho progresu. Avšak práca potrebná na ukončenie zvyšných príbehov bola vynahradená v druhom týždni. Šprint bol úspešne dokončený a všetky špecifikované príbehy boli implementované.



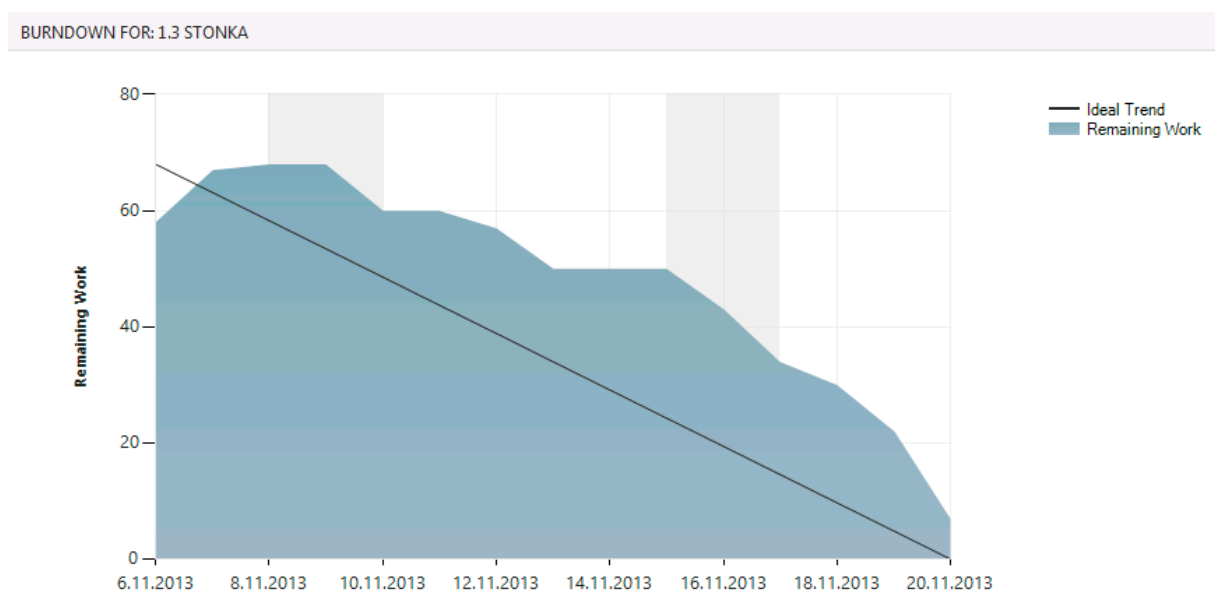
Obr. č. 3 Burndownchart šprintu korienok

6.3 ŠPRINT 3 - STONKA

Celkom príbehov: 15

Dokončených príbehov: 13

V prvej polovici šprintu mali všetci členovia tímu zápočtové písomky, ktoré im nedovolili riešiť a tvoriť príbehy tímového projektu. V druhom týždni sa členovia už plne venovali tímovému projektu. Avšak mimo implementácie príbehov mali členovia tímu prácu s dokončovaním a úpravou metodík, projektovej dokumentácie a dokumentácie riadenia. V dôsledku zaneprázdnenosti všetkých členov, v priebehu celých dvoch týždňoch šprintu, nebolo možné splniť všetky úlohy. Avšak na dokončenie zostalo len malé percento práce (7/68).



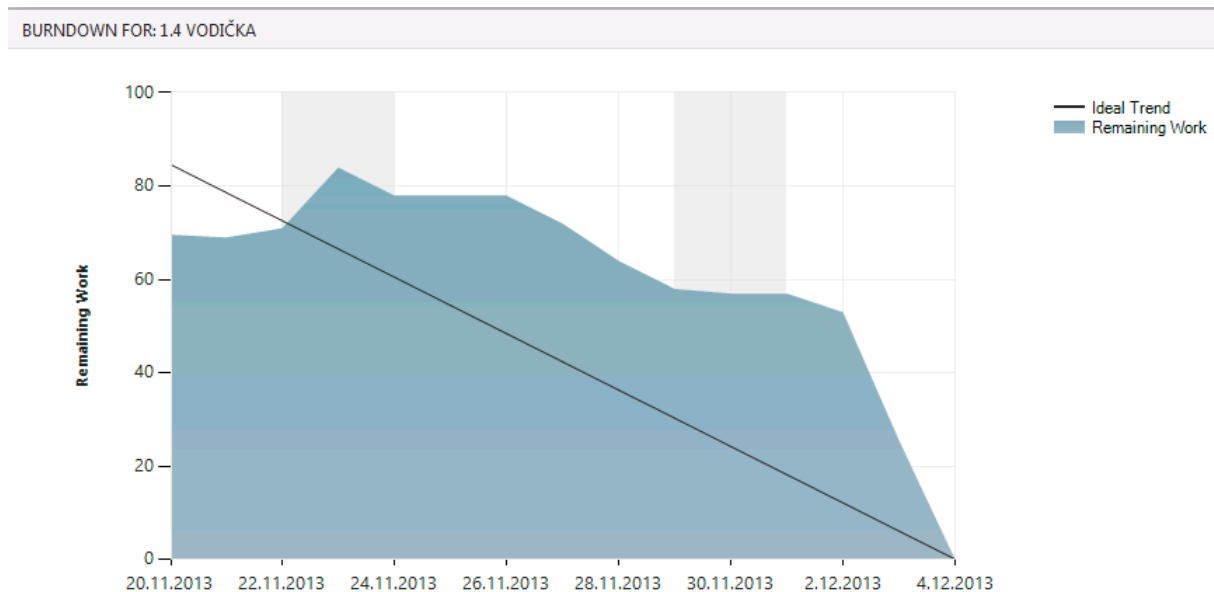
Obr. č. 4 Burndownchart šprintu Stonka

6.4 ŠPRINT 4 - VODIČKA

Celkom príbehov: 17

Dokončených príbehov: 17

Zápočtové písomky prebiehali aj počas tohto šprintu. Taktiež iné školské zadania zaneprázdňovali členov tímu. Z tohto dôvodu bola väčšina práce dokončená až tesne pred koncom šprintu. Nakoniec sa však podarilo dokončiť všetky príbehy podľa plánu.



Obr. č. 5 Burndownchart šprintu Vodička

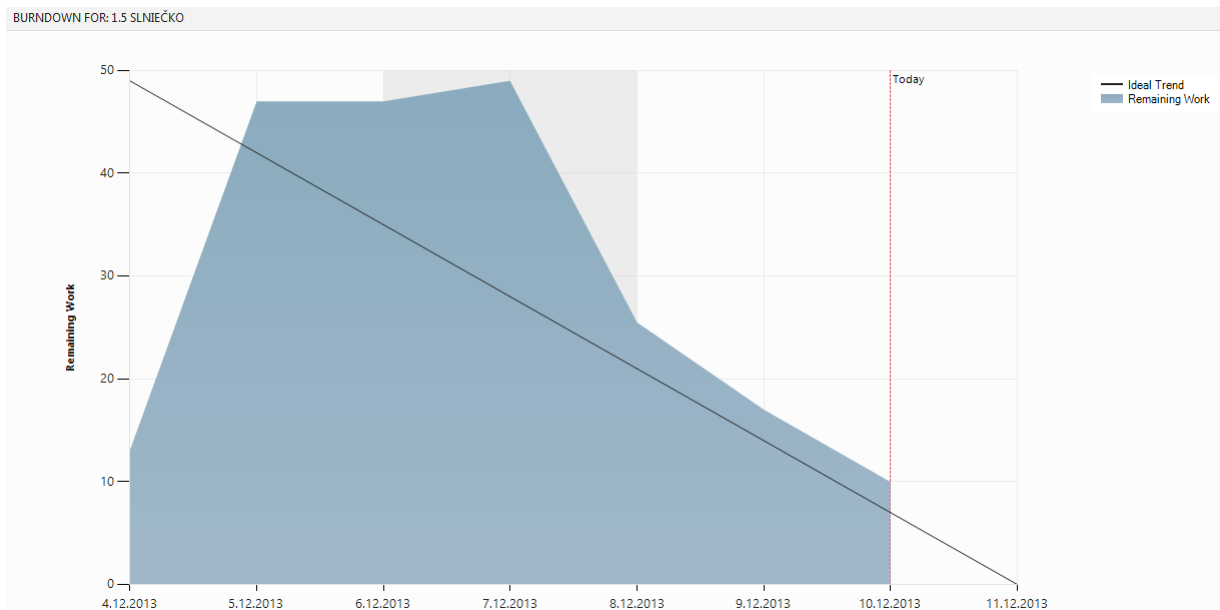
6.5 ŠPRINT 4 - SLNIEČKO

Celkom príbehov: 21

Priebežne dokončených príbehov: 15

Posledný šprint zimného semestra sa venoval hlavne opravám chýb, ktoré boli zistené na systéme a iným menším úloh. Tento šprint trval iba jeden týždeň avšak tím pracuje na menších úlohách. Keďže burndownchart je iba týždňový sú viditeľnejšie detailnejšie zmeny. Prvé dni šprintu graf narastá z dôvodu pridávania odhadovanej doby práce členmi tímu. V druhej časti týždňa členovia tímu zapracovali na svojich úlohách a graf sa dostal približne na ideálny progres.

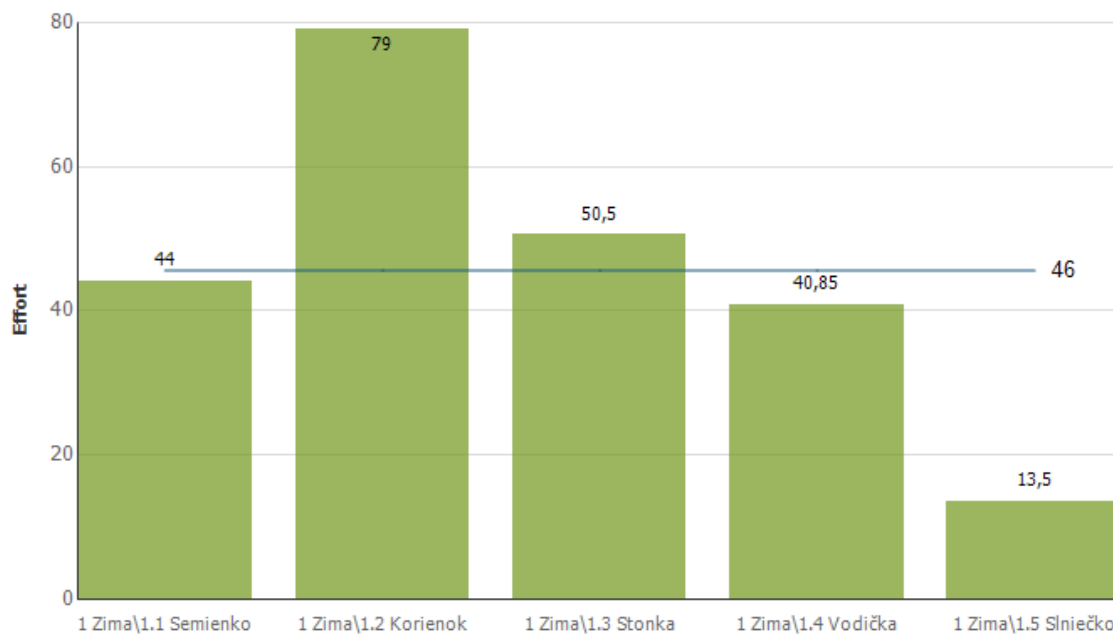
6 Manažment monitorovania



Obr. č. 6 Burndownchart šprintu Slniečko

6.6 RÝCHLOSŤ VÝVOJA PROJEKTU

V tejto kapitole je uvedená rýchlosť vývoja projektu v jednotlivých šprintoch podľa „Effort“. „Effort“ body označujú obtiažnosť dokončenia jednotlivých úloh. Obtiažnosť určujú programátori na stretnutí podľa svojho uváženia. Sčítané obtiažnosti v jednotlivých šprintoch sú viditeľné v nasledujúcom obrázku.



Obr. č. 7 Graf znázorňujúci obtiažnosť všetkých úloh v danom šprinte.

6 Manažment monitorovania

Obťažnosť úloh počas zimného semestra klesala vďaka učeniu sa pracovať v danom prostredí ako aj tým, že v druhej fáze semestra bolo pridelovaných menej úloh pre členov kvôli práci na iných predmetoch.

7 MANAŽMENT PODPORY VÝVOJA

Pozíciu manažéra podpory vývoja zastáva člen Bc. Matej Chlebana. Mal niekoľko povinností, ktoré by sa dali rozdeliť do niekoľkých častí.

7.1 KONFIGURÁCIA

Jedna z prvých povinností bola nainštalovať a nakonfigurovať server, ktorý bude slúžiť na uloženie webovej stránky tímu a hlavne ako úložisko pre produkt. Pridelený server bol pripravený pre operačný systém Fedora, ale z dôvodu použitých technológií sme nainštalovali 64 bitový operačný systém Windows Server 2012 R2. Ako webový server sme použili Internet InformationServices 8 (IIS 8).

Náš produkt aktívne využíva relačnú databázu. Na tento účel sme potrebovali nainštalovať databázový server. Náš produkt využíva Microsoft technológie a z dôvodu kompatibility sme zvolili relačný databázový server Microsoft SQL Server 2012 verziu Business Intelligence. Každý člen tímu má na svojej lokálnej stanici nainštalovanú vlastnú inštanciu databázového servera pre účely vývoja. Oddelením pracovných staníc od serverovej databázy sme docielili nezávislosť dostupnosti servera pri vývoji, napríklad pri práci bez internetového pripojenia. Server používa vlastnú databázu, v ktorej sa ukladajú len dáta z nasadenej verzie.

Pre jednoduché nasadzovanie verzií nášho produktu sme doinštalovali Web Deploy verzie 3.5. Nakonfigurovaný bol tak, aby pri každom vytvorení novej verzie nasadil najnovšiu verziu na IIS.

Pri centralizovanom vývoji produktu môže vzniknúť taká situácia, že sa na lokálnych vývojových staniciach vývojárov aplikácia správa inak ako na serveri. Pre tento účel sme nainštalovali nástroj na vzdialený debugging aplikácie nasadenej na serveri - RemoteDebugging.

Na lokálnych staniciach všetkých členov tímu sme nainštalovali vývojové prostredie VisualStudio 2012, ktoré má integrovanú podporu verziovaného systému Team Foundation Server (TFS).

7.2 VERZIOVANIE

Pri verziovaní nášho produktu sme si museli stanoviť podmienky pre odovzdávanie nových verzií. Nové verzie produktu je nutné odovzdávať vždy po ukončení úlohy a po pridaní nových súborov do projektu. Jednou z hlavných podmienok je, aby každá nová verzia bola skompilovateľná a spustiteľná. V prípade, že z nejakého dôvodu nebude nová verzia skompilovateľná, Web Deploy sme nastavili tak, aby túto verziu na server nenasadil.

Každé odovzdanie novej verzie musí obsahovať komentár, ktorý písal programátor. Skladá sa zo stručného opisu logických zmien, ktoré nastali oproti poslednej verzií. Štruktúra komentára sa skladá z kľúčového slova napísaného anglickým názvom, popisujúceho typ zmeny, ktorá nastala. Za kľúčovým slovom nasledujú odrážky so štruktúrovaným popisom zmien, ktoré sa píšú v slovenskom jazyku bez diakritiky, každá na novom riadku. Kľúčové slová, ich význam a štruktúra popisu je nasledovná:

Štruktúra komentára	Popis
Add: - pridaná entita 1 - pridaná entita 2 ..	Stručný popis každej pridanej entity (názov + čo robí), oddelených odrážkami, každá na novom riadku.
Change: - upravená entita 1 - upravená entita 2..	Stručný popis každej zmenenej entity (názov + čo robí), oddelených odrážkami, každá na novom riadku.
Delete: - vymazaná entita 1 - vymazaná entita 2..	Názov vymazanej entity + dôvod prečo bola vymazaná, oddelených odrážkami, každá na novom riadku.
Fix: - opravená entita 1 - opravená entita 2..	Názov opravenej entity + popis chyby, ktorá bola opravená, oddelená odrážkami, každá na novom riadku.
Test: - pridaný / zmenený / opravený - test	Typ: „pridaný/zmenený/opravený“ + stručný opis čo testuje. Každý test oddelený odrážkou, každý na novom riadku.
Codereview - otagovaný kód	Popis codereview – pridané tagy / opravené tagy + popis

7.3 INICIALIZÁCIA POUŽITÝCH TECHNOLOGIÍ

Ako hlavnou použitou technológiou pre vývoj sme použili ASP.NET MVC 4 framework. Ako syntax písania kódu ASP.NET web stránok sme vybrali Razor, hlavne kvôli prehľadnosti a jednoduchosti zdrojového kódu.

Pre jednoduchšiu prácu s databázou sme použili objektovo relačný mapovačLinqToSQL.

7.4 PODPORNÉ ČINNOSTI

Manažér podpory vývoja mal okrem spomenutých činností, na starosti inicializáciu projektu, konfiguráciu TFS servera ako iné činnosti, bez ktorých by nebolo možné začať samotný vývoj. Na začiatku bolo potrebné prideliť členom tímu práva a roly v systéme TFS.

8 MANAŽMENT ROZVRHU (PLÁNOVANIE)

Pre správnu organizáciu práce v tíme je potrebné vytvoriť plán postupu práce. Tento plán slúži na zorganizovanie a zosúladenie práce v tíme aby boli dodržané všetky zadané termíny. Pri nesprávnom návrh plánu by mohli vzniknúť nedorozumenia, ktoré spôsobia nedodržanie dôležitých termínov a následne celkové zlyhanie vývoja

V našom tímovom projekte je potrebné správne naplánovať postup práce aby boli dodržané termíny odovzdania a výsledný produkt bol použiteľný v praxi.

8.1 ROZVRH ZIMNÉHO SEMESTRA

8.1.1 Prvý šprint – Semienko **9.10.2013 – 23.10.2013**

Zoznámenie sa s prostredím. Analýza technológie ASP.NET MVC 4. Zoznámenie s databázou AST-RCS. Vytvorenie webovej stránky tímu. Administratíva používateľských účtov. Načítanie projektov z databázy AST-RCS a zobrazenie súborovej štruktúry a obsahu súborov.

8.1.2 Druhý šprint – Koriенок **23.10.2013 – 6.11.2013**

Dopĺňanie určených komponentov systému. Úprava vzhľadu stránky. Doplnenie metód pre zobrazenie zmien v projektoch.

8.1.3 Tretí šprint - Stonka **6.11.2013 – 20.11.2013**

Zosúladenie zobrazenia používateľského rozhrania. Zjednotenia navigácie v projekte. Správa zabezpečenia. Prepojenie jednotlivých komponentov.

8.1.4 Štvrtý šprint - Vodička **20.11.2013 – 4.12.2013**

Logovanie činnosti. Oddelenie logiky projektov na fyzické a logické projekty. Úpravy rozhraní. Implementácia správy značiek v projekte.

8.1.5 Piaty šprint - Slniečko **4.12.2013 – 11.12.2013**

Ukončenie práce v zimnom semestri. Doladenie chýb. Testovanie použiteľnosti

8.2 DÔLEŽITÉ TERMÍNY V ZIMNOM SEMESTRI

- 25.9.2013 – odovzdanie kompetencií tímu
- 9. týždeň – stretnutie tímu 20.11.2013 – odovzdanie prvých verzií dokumentácií
 - dokumentácia k produktu – prvé tri šprinty, big Picture
 - dokumentácia k riadeniu
- 2.12.2013 – SCRUM panel - prezentácia

8 Manažment rozvrhu plánovanie

- 12.12.2013 – finálne odovzdanie v zimnom semestri
 - dokumentácia k produktu – prvých päť šprintov
 - dokumentácia k riadeniu a používateľská prezentácia prototypu
 - opis prototypu

8.3 SYSTÉM TFS

Pre podporu nielen plánovania v tíme slúži systém TFS. Systém využívame pre manažment prác v tíme. Celkový časový harmonogram zimného semestra je v systéme zaevidovaný do jednotlivých šprintov:

	Iterations	Start Date	End Date
	WhereIsMyCode		
<input type="checkbox"/>	Zima	9.10.2013	12.12.2013
<input checked="" type="checkbox"/>	Semienko	9.10.2013	23.10.2013
<input checked="" type="checkbox"/>	Korienok	23.10.2013	6.11.2013
<input checked="" type="checkbox"/>	Stonka	6.11.2013	20.11.2013
<input checked="" type="checkbox"/>	Vodička	20.11.2013	4.12.2013
<input checked="" type="checkbox"/>	Slniečko	4.12.2013	11.12.2013
<input type="checkbox"/>	Leto	17.2.2014	26.5.2014

Systém zobrazuje priebeh pomocou prehľadného burndown chartu. Podľa krivky môžeme monitorovať rýchlosť práce na požiadavkách a prípadne na nezrovnalosti reagovať.

Systém TFS prehľadne zobrazuje rozdelenie práce jednotlivých členov v tíme. V prípade dodržania plánu a zrealizovania všetkých požiadaviek určených v jednotlivých šprintoch by mal byť systém včas dokončený.

9 MANAŽMENT KOMUNIKÁCIE A ĽUDSKÝCH ZDROJOV

Základom každého dobrého projektu je komunikácia, a to nielen so zákazníkom, ale najmä vo vnútri tímu. Pozíciu manažéra komunikácie a ľudských zdrojov zastáva člen Bc. Zuzana Grešíliková.

9.1 KOMUNIKAČNÉ KANÁLY

Náš tím využíva na komunikáciu niekoľko komunikačných kanálov. Patria medzi ne:

- Osobné stretnutia
- Email
- Google group (Webové fórum)
- Sociálne siete (Chat v rámci tímu, alebo jednotlivo medzi členmi tímu)
- Konferenčné hovory cez internet
- Telefonické rozhovory

9.1.1 Osobné stretnutia

Stretnutia prebiehajú na pravidelnej báze každý týždeň od 12:00 do 15:00. Stretnutia sa konajú v Jobsovom softvérovom štúdiu na fakulte FIIT STU. Na stretnutí členovia tímu diskutujú s vlastníkom produktu (pedagogický vedúci), riešia problémy, ktoré nastali pri riešení zadaných úloh, prezentujú dosiahnuté výsledky, diskutujú o stave projektu. Náplň stretnutia závisí najmä na tom, v akom štádiu sa nachádza aktuálny šprint. V prípade začiatku nového šprintu sa členovia tímu sústredia najmä na pridelenie úloh a diskusiu o daných úlohách. V prípade, že sa stretnutie uskutočňuje v priebehu šprintu, je zamerané najmä na riešenie vzniknutých problémov. Tieto stretnutia prebiehajú v uvoľnenej atmosfére a členovia tímu sa neboja prezentovať svoje názory.

Členovia tímu sa neoficiálne stretávajú aj mimo týchto oficiálnych stretnutí. Náplňou týchto stretnutí je najmä riešenie vzniknutých problémov, no i teambuildingové aktivity tímu.

9.1.2 Sociálne siete

Členovia tímu využívajú najmä komunikáciu cez sociálne siete, a to preto, že poskytuje najrýchlejšiu spätnú väzbu od ostatných členov tímu. Na komunikáciu sa využíva najmä skupinový chat na Facebook-u. Nevýhodou tejto komunikácie je však zlá prehľadnosť pri spätnom vyhľadávaní informácií.

Pre sprehľadnenie bola vytvorená uzatvorená skupina na Facebook-u, kde môžu členovia pridávať príspevky ohľadne aktuálne diskutovaných tém, a taktiež vytvárať hlasovanie o dôležitých rozhodnutiach tímu. Členovia tímu sú povinní informovať ostatných členov tímu o dôležitých skutočnostiach príspevkom v tejto skupine alebo príspevkom v nižšie uvedenej Google group.

9.1.3 Email

Členovia tímu majú vytvorené spoločné mailové konto. Komunikácia emailom prebieha len okrajovo a slúži najmä na komunikáciu pedagogického vedúceho s tímom.

9 Manažment komunikácie a ľudských zdrojov

9.1.4 Google Group

Členovia tímu majú spoločne vytvorenú Google group, ktorá je vytvorená ako webové fórum. Na začiatku každého šprintu sa v danej skupine otvárajú nové témy, ktoré sú zamerané na riešenie individuálnych problémov, najdôležitejšie informácie k aktuálnemu šprintu. Google group bola založená najmä na efektívnejšie hľadanie dôležitých informácií, ktoré nebolo možné dosiahnuť pri komunikácii cez sociálne siete. Členovia tímu môžu vytvárať témy vo fóre na základe vlastného uváženia.

9.1.5 Konferenčné hovory cez internet

Na komunikáciu konferenčnými hovormi sa využíva Skype. Využíva sa najmä v krízových situáciách, a to prípadoch, keď je jednoduchšie komunikovať ústne, ako písomne, no nie je možné osobné stretnutie členov tímu.

9.1.6 Telefonické rozhovory

Komunikácia cez telefón sa využíva najmä v krízových situáciách, keď nastane problém, ktorý je potrebné urgentne riešiť, a daný člen tímu nie je dostupný pomocou inej formy komunikácie.

9.2 KOLABORÁCIA ČLENOV TÍMU

Na podporu spolupráce v tíme využívame najmä tieto nástroje:

- Dropbox
- Skydrive
- Team Foundation Server (TFS)

9.2.1 Dropbox

Slúži, ako úložisko dokumentácií a iných dôležitých dokumentov, databázových skriptov, inštalračných súborov, manuálov a iných dôležitých dát tímu.

9.2.2 Skydrive

Využívaný ako podporný nástroj na spracovanie dokumentov v tíme. Slúži najmä na spoločnú tvorbu dokumentov v reálnom čase.

9.2.3 Team Foundation Server (TFS)

Tento produkt sa využíva najmä na správu zdrojového kódu, zadávanie požiadaviek, riadenie projektu pri agilnom vývoji softvéru. Pokrýva celý proces vývoja softvéru. Táto platforma umožňuje efektívne riadenie vývoja softvérových produktov, počas celého ich životného cyklu.

V rámci tímu v TFS vytvárame backlog projektu. Tiež v ňom vytvárame jednotlivé šprinty (etapy vývoja softvéru). Monitorujeme a plánujeme vývoj, no tiež nám umožňuje vidieť štatistiky našej práce. Okrem iného TFS umožňuje aj prehliadanie histórie changesetov a zmien aj s menami ich autorov a dátumami zmien. TFS využívame aj na testovanie vyvíjaného softvéru.

10 MANAŽMENT TVORBY DOKUMENTÁCIE

Pozíciu manažéra tvorby dokumentácie v tíme zastáva člen Bc. Patrik Oriskó. Jeho úlohou bola určiť nástroj na vytvorenie technickej dokumentácie a dokumentácie riadenia. Zvolený bol nástroj *SkyDrive*, ktorá umožňuje prístup všetkých členov do dokumentu naraz. Členovia si môžu editovať dokumenty priamo vo webovom nástroji, ktorý *SkyDrive* ponúka, alebo majú možnosť editovať lokálnu kópiu dokumentu v novších verziách *MS Office* (od *MS Office 2010*) priamym nahrávaním do repozitára *SkyDrive* pri uložení súboru. Po definovaní nástroja pre písanie dokumentov, bola vytvorená dokumentácia k inžinierskemu dielu a dokumentácia riadenia a následne boli pridelené práva na editovanie pre každého člena v tíme.

Ďalšie čiastkové dokumenty boli zdieľané pomocou repozitára *Dropbox*. K zdieľanému priečinku v repozitári majú prístup všetky členovia tímu. V repozitári *Dropbox* sú prístupné dokumenty, ako šablóna pre zápisy zo stretnutia, zdokumentované metodiky, ktorými sa členovia tímu riadia a ďalšie manuály pre prácu na projekte. Priebežne sa pridávajú a overujú aj zápisy zo stretnutiach.

Úlohy manažéra dokumentácie sú nasledovné:

- vytvorenie štruktúry dokumentov,
- štylizovanie, korekcia, aktualizácia obsahu dokumentov,
- pridávanie potrebných čiastkových dokumentov,
- rozdelenie úloh pri tvorbe dokumentácií,
- vytvorenie preberacieho protokolu,
- finalizovanie verzií dokumentov,
- odovzdávanie dokumentov vedúcemu projektu.

11 MANAŽMENT RIZÍK

Hlavnou úlohou manažéra rizík je identifikovať a predchádzať možným rizikám, ktoré by mohli ohroziť vývoj produktu. Túto pozíciu v tíme zastáva člen Bc. Tomáš Kepič.

11.1 IDENTIFIKÁCIA A RIEŠENIE RIZÍK

Počas práce na projekte sa vyskytuje množstvo problémov. Niektoré jednoduchšie, niektoré sú však náročnejšie. Problémy, ktoré sú väčšinou vážneho charakteru a dokážu v značnej miere ohroziť vývoj projektu alebo správne fungovanie tímu, nazývame riziká.

Nato, aby sme predišli rizikám a vyhli sa tak rôznym stratám v tíme, či už časovým, hardvérovým alebo personálnym, musíme vedieť včas identifikovať riziko a predísť mu. Z tohto dôvodu je potrebné rizika zrozumiteľne charakterizovať, a to počas celého trvania projektu, nielen na začiatku.

V kapitole Metodika identifikácie a riešenia rizík sme opísali 7 rôznych rizík.

Opisy rizík pozostávali z týchto častí:

- Názov – dostatočne výstižný názov, pomocou ktorého vieme, o aké riziko sa jedná
- Spúšťač – situácia, po ktorej nastáva riziko
- Dôsledok - následky spôsobené neskorým odhalením a neriešením rizika
- Pravdepodobnosť nastania – odhadujeme, prípadne vieme určiť podľa doterajších skúseností
- Vyhnutie - najdôležitejšia časť manažmentu rizík, minimalizuje straty
- Riešenie – ak riziko nakoniec nastane, ponúkame postup ako ho čo najefektívnejšie vyriešiť

11.2 RIZIKO – NEDOSTATOK SKÚSENOSTÍ S TECHNOLOGIOU MICROSOFT MVC

Toto bolo jedno z prvých rizík, ktoré sme identifikovali. Väčšina členov tímu nemala skúsenosti s technológiou Microsoft MVC. Preto bolo potrebné sa túto technológiu čo najrýchlejšie naučiť. Toto riziko môžeme opísať nasledovne:

Názov: Nedostatok skúsenosti s technológiou Microsoft MVC

Spúšťač: povinná technológia pri projekte, s ktorou členovia tímu nemajú dostatočné skúsenosti

Dôsledok: problémy pri programovaní navrhnutých riešení až odchod člena z tímu

Pravdepodobnosť nastania: stredná

Vyhnutie: študovanie tutoriálov a manuálov, znalosť nadväzujúcich technológií

Riešenie:

1. Na začiatku projektu pridelovanie jednoduchších úloh – zoznámenie sa s technológiou
2. Postupné pridelovanie zložitejších úloh – rozširovanie vedomostí
3. Pozorovanie narastajúcich vedomostí a správny odhad tomu zodpovedajúcich úloh

12 METODIKY

12.1 METODIKA IDENTIFIKÁCIE A RIEŠENIA RIZÍK

Obsahom tejto metodiky je identifikovanie rizík v tíme a poskytnutie ich riešenia. Na riešenie určitých špecifických problémov pri práci existujú iné metodiky tímu, ktoré poskytujú podrobný popis špecifickej oblasti. Táto metodika je zameraná na riešenie rizík, ktoré by v značnej miere ohrozili prácu na projekte a jeho zodpovedajúcu kvalitu. Metodika poskytuje riešenia pre otázky ohľadom hardvéru, problému s časom, vylúčenia alebo odchodu člena tímu. V prípade, že nenájdeme postup, ktorý by pokrýval riešenie nášho problému, uvádzame v práci aj tabuľku so zodpovednosťami členov tímu, na ktorých je potrebné sa potom obrátiť.

12.1.1 Dedikácia

Touto metodikou sa riadi členovia tímu číslo 7 s názvom Lucky Seven, v školskom roku 2013/2014 pri vypracovaní tímového projektu, ktorí sa dostali počas práce do kritickej situácie, ktorá by mohla ohroziť kvalitu výsledného produktu a jednotlivých odovzdaných častí.

12.1.2 Zoznam súvisiacich metodík

- Metodika iterácií projektu

12.1.3 Pojmy a skratky

Pojem	Vysvetlenie
AST-RCS	Počítačový server slúžiaci na získavanie údajov o projektoch
Commit	Uloženie aktuálnej rozpracovanej verzie projektu na server
Človekohodina	Práca, ktorú poskytne človek za jednu hodinu
Productbacklog	Obsahuje možnosti produktu, ktoré sa vypracujú pomocou úloh (taskov)
Šprint	Čas vymedzený na plnenie úloh
Task	Najmenšia jednotka pri konkretizácii úlohy Úloha, ktorú má vypracovať určitý člen tímu
Teambuilding	Neformálne stretnutie pre upevnenie vzťahov v tíme
Wi-Fi	Bezdrôtová sieť slúžiaca na pripojenie k internetu Nemusi však poskytovať stabilné pripojenie
Productowner	Vlastník produktu Zadávatel' požiadaviek na funkcionality produktu
Supervisor	Pedagogický vedúci tímu
Webmaster	Osoba zodpovedná za beh a tvorbu webovej stránky

Tab. č. 1: Tabuľka pojmov

12.1.4 Súhrn rol a zodpovedností účastníkov

Rola	Zodpovednosť
Supervisor	Dozor nad celým tímom
Manažér rizík	Pomáha predísť kritickým situáciám Rieši kritické situácie
Manažér rozvrhu (plánovania)	Poskytuje časové plány, čím redukuje časové riziká
Člen tímu	Podieľa sa na tímových riešeniach kritických situácií

Tab. č. 2: Roly a zodpovednosti účastníkov

12.1.5 Identifikácia rizík

V tejto časti sa zaoberáme identifikáciou rizík, ktoré nie sú triviálnymi problémami popísanými v iných metodikách, ale vážne riziká ohrozujúce vývoj a kvalitu výsledného produktu. Pri každom riziku vytvoríme štruktúrovaný opis a riešenie kritickej situácie, ktorá vznikne pri jeho nastaní.

V metodike sú identifikované nasledujúce riziká:

- Nefunkčný pracovný počítač príslušníka tímu
- Problém so serverom
- Zlý odhad času práce – počas a pred odovzdaním šprintu
- Nevyriešená úloha (*task*) v šprinte – prípad pri alebo po odovzdaní šprintu
- Neplnenie si povinnosti člena tímu
- Predčasný odchod člena tímu
- Nastane udalosť, na ktorú neexistuje metodika

12.1.6 Opisy jednotlivých rizík

Nefunkčný pracovný počítač príslušníka tímu

Spúšťač: člen tímu nemá funkčný počítač - výpadok počítača člena tímu je dlhodobejší a nebude trvať iba pár dní, prípadne nastane v kritickej situácii ako je tesne pred odovzdaním šprintu

Dôsledok: nemožnosť splnenia zadaných úloh v šprinte

Pravdepodobnosť nastania: nízka

Vyhnutie: každý musí byť zodpovedný a brať ohľad na to, že jeho počítač je dôležitý pre prácu, a preto by mal s ním narábať opatrne aby sa manuálne nepoškodil

Riešenie:

1. Pre vyriešenie situácie je potrebné čo najrýchlejšie zohnať náhradný počítač. Možnosťou je opýtať sa iných členov tímu, priateľov a podobne
2. Ak sa nenájde dostatočne výkonný počítač pre prácu, je potrebné nahlásiť tento problém tímu
3. Riešime 2 typy problémov, ak nemáme náhradný počítač:
 - Dlhodobejší výpadok, oznámený v predstihu pred odovzdaním
 - V tíme sa vedie diskusia k úlohám

- Ak sú úlohy poškodeného člena tímu dôležité a nadväzujú na iné úlohy v aktuálnom šprinte, je potrebné ich rozdelenie medzi ostatných členov
- Ak úlohy nenadväzujú na iné, prípadne môžu byť vykonané neskôr, tak budú presunuté do ďalšieho šprintu
- Výpadok tesne pred odovzdaním, napríklad poslednú noc
 - Kontaktujú sa ostatní členovia tímu a hľadáme takého, ktorý má už svoju prácu v šprinte hotovú a má čas dokončiť rozpracované úlohy poškodeného
 - Dostupný voľný člen dokončí rozpracované úlohy
 - V prípade, že úlohy ešte nie sú dostatočne rozpracované na to, aby ich niekto dokončil alebo nikto nie je dostupný, úlohy budú musieť byť prenesené do ďalšieho šprintu

Problém so serverom

Spúšťač: prebiehajúce aktualizácie na serveri alebo jeho výpadok v dôsledku neznámej udalosti

Dôsledok: nemožnosť žiadnej práce týkajúcej sa projektu

Pravdepodobnosť nastania: stredná

Vyhnutie: predbežné ohlásenie blížiacej sa aktualizácie a odstávky servera

Riešenie:

1. Spozorujeme nefunkčnosť servera, program neodpovedá
2. Skontrolujeme pripojenie k internetu (problémy so školskou sieťou *Wi-Fi*, domáce výpadky)
3. Počkáme určitý čas od spozorovaného výpadku - maximálne 1 hodinu
4. Kontaktujeme iného člena tímu s otázkou ohľadom funkčnosti servera
 - Ak inému členovi pripojenie funguje, riešime individuálny problém
5. Ak nefunguje pripojenie ani inému členovi alebo nevieme samostatne nájsť riešenie, kontaktujeme tímového správcu servera
6. Ak tímový správca servera neposkytne riešenie, kontaktujeme vedúceho tímu (*supervisora*)

Zlý odhad času práce – počas a pred odovzdaním šprintu

Spúšťač: člen tímu nemá dostatok času na splnenie požadovaných úloh

Dôsledok: nedokončené úlohy v šprinte

Pravdepodobnosť nastania: stredná

Vyhnutie: každý člen musí zodpovedne vykonávať svoje úlohy a s tým súvisí aj jeho časový manažment

Riešenie:

1. Zameranie všetkých síl na vyriešenie úloh v čo najvyššej možnej miere a primeranej kvalite
2. Kontaktovanie tímu s popisom problému a jeho riešenie
 - Pomocné vyriešenie tímovými kolegami

- Rozdelenie časti práce na úlohách medzi iných členov
- V ďalšom šprinte zoberieme ohľad na poskytnutú pomoc a podľa toho rozdelíme prácu
- Úloha sa nestihne vyriešiť ani pomocou kolegov počas daného šprintu
 - Presun úlohy do ďalšieho šprintu

3. Diskusia na stretnutí ohľadom časovému manažmentu

- Dôkladnejšie prekonzultovanie časovej náročnosti úloh a ich odhad času v budúcnosti
- Vždy brať na vedomie rozdielne skúsenosti členov, podľa toho si zodpovedne vyberať a pridelať úlohy a k nim aj príslušné časové odhady

Nevyriešená úloha (*task*) v šprinte – prípad pri alebo po odovzdaní šprintu

Spúšťač: v čase odovzdania šprintu sa vyskytujú v *Productbacklogu* nevyriešené úlohy

Dôsledok: brzdenie práce na projekte nutnosťou riešenia starých úloh a nemožnosťou prideliť nové v súlade s rozumným časovým manažmentom

Pravdepodobnosť nastania: nízka

Vyhnutie: rozumné rozdelenie práce na kritických úlohách v priebehu predchádzajúceho šprintu

Riešenie:

1. Počas stretnutia sa vedie diskusia k nevyriešeným úlohám s ich riešiteľom
2. Určenie dôvodu nesplnenia úlohy
 - Nefunkčný pracovný počítač, veľa zápočtových písomiek alebo školských zadaní
 - úloha ostáva rovnakému riešiteľovi na splnenie v novom šprinte
 - riešiteľovi budú pridelené nové úlohy s ohľadom na to, aby splnil predchádzajúce
 - Nedostatok skúsenosti riešiteľa
 - Úloha bude pridelená inému, skúsenejšiemu členovi v danej oblasti
 - Ak sa v danej oblasti nikto nevyzná, pridelíme úlohu inému členovi podľa záujmu
 - Riešiteľ nepristupoval k riešeniu zodpovedne
 - Riadime sa pokynmi v popise rizika: Neplnenie si povinnosti člena tímu
3. Úloha bude vyriešená v nasledujúcom šprinte

Neplnenie si povinností člena tímu

Spúšťač: členovi ostali nesplnené úlohy s nepodloženými dôvodmi, neriadenie sa metodikami práce v tíme, neospravedlnená neúčasť na povinnom tímovom stretnutí

Dôsledok: nesplnené úlohy, strata potrebných človekohodín práce, nedostatočná kvalita produktu

Pravdepodobnosť nastania: nízka

Vyhnutie: zodpovedný prístup k práci

Riešenie:

1. Identifikujeme viacero dôvodov neplnenia povinností:
 - Člen nestíha
 - Diskusia na stretnutí k časovému harmonogramu a manažmentu člena tímu
 - Vyvodenie riešenia danej situácie
 - Ozrejmienie, že tímový projekt je jednou z prioritných školských úloh vzhľadom na prácu a nadväznosti v tíme
 - Nezájem o prácu - členovi chýba motivácia
 - Tímová diskusia k zaujímavosti riešenia a práce na tímovom projekte v školskom prostredí
 - Dohoda a naplánovanie neformálneho stretnutia (*teambuildingu*) – lepšie začlenenie problematického člena a zosilnenie kolektívu aj pomocou mimoprojektových aktivít
 - Vážne zdravotné dôvody (krátkodobé)
 - Tím musí počas nechcenej nečinnosti člena pracovať v núdzovom režime
 - Rieši sa menej úloh
 - Rolu člena preberie dočasne iný člen, v prípade časovo náročnej roly ju preberie viacero členov
2. Ak je člen tímu naďalej problematickým a situácia sa opakuje, bude upozornený podmieneným vylúčením z tímu
3. Ak sa situáciu bude opakovať viac ako dva krát, vedúci tímu po diskusií a presvedčení, že daný člen je nevyhovujúci pre prácu a prejavuje neustály nezájem, vylúči daného člena z tímu

Predčasný odchod člena tímu

Spúšťač: člen tímu opustí tím

Dôsledok: strata roly v tíme, zníženie počtu dostupných človekohodín

Pravdepodobnosť nastania: nízka

Vyhnutie: dôkladná príprava do školy na iné predmety, záujem o tímový projekt

Riešenie:

1. Identifikácia strát v podobe roly a počtu možných odpracovaných hodín
2. Pridelenie uvoľnenej roly v tíme, prípadne prerozdelenie všetkých rolí
3. Zníženie počtu úloh (požiadaviek) v šprinte
4. Pouvažovať nad možnosťami produktu, ktoré by sa dali v prípade nedostatku času vynechať a nebola by pri tom ohrozená hlavná funkcionality

Nastane udalosť, na ktorú neexistuje metodika

Spúšťač: chceme vyriešiť určitý problém, avšak nemáme presne špecifikovaný postup

Dôsledok: pri nesprávnom postupe vzniká v projekte chaos

Pravdepodobnosť nastania: stredná

Vyhnutie: popisanie čo najväčšieho množstva možných otázok a problémov ohľadom riešenia projektu

Riešenie:

1. Preštudovanie webovej stránky tímového projektu² a príslušných dokumentov, ktoré sú na nej umiestnené
2. Ak neexistuje dokument pokrývajúci problém, kontaktujeme člena zaoberajúceho sa danou oblasťou. Ako prehľad zodpovedností v tíme poslúži okrem webovej stránky aj stručný prehľad v tabuľke č. 3

Meno	Zodpovednosti/ Metodika
Ing. Karol Rástočný	supervisor, ProductOwner, stará sa o AST-RCS - kontakt s jeho vývojármi
Bc. Zuzana Grešlíková	vedúca tímu, zodpovedná za komunikáciu Metodika zberu a spracovania požiadaviek zákazníka
Bc. Matej Chlebana	správa servera, podpora vývoja Metodika pre verziovanie a commitovanie
Bc. Tomáš Kepič	riešenie kritických situácií Metodika identifikácie a riešenia rizík v tíme
Bc. Patrik Oriskó	dokumentarista Metodika tvorby dokumentácie
Bc. Patrik Samuhel	plánovač Metodika iterácií projektu (priebeh šprintu a stretnutí)
Bc. Michael Scholtz	webmaster stránky tímu, zodpovedný za kvalitu produktu Metodika o konvenciách písania zdrojového kódu
Bc. Július Skrisa	monitorovanie projektu Metodika testovania

Tab. č. 3: Prehľad zodpovedností a metodík v tíme

3. Ak nevyhovuje žiadna konkrétna metodika a ani žiadna zodpovedná osoba nevie poskytnúť riešenie problému, kontaktujeme člena zaoberajúceho sa riešením kritických udalostí, ktorý nám pomôže vyriešiť danú situáciu

²<http://labss2.fiit.stuba.sk/TeamProject/2013/team07is-si/>

12.2 METODIKA TVORBY DOKUMENTÁCIE

Cieľom metodiky je definovať postup pre vytvorenie projektovej dokumentácie a dokumentácie riadenia projektu. Definuje štýl dokumentov, ako aj postup pre písanie jednotlivých kapitol dokumentácie. Metodika definuje i základnú štruktúru projektovej dokumentácie a dokumentácie riadenia projektu.

12.2.1 Dedikácia

Táto metodika je určená pre každého člena tímu č.7 *Lucky Seven* na predmete Tímový projekt I, II na FIIT STU v akademickom roku 2013/2014.

12.2.2 Súvisiace metodiky

- Metodika iterácii projektu (tím č.7 FIIT STU 2013/2014)
- Metodika testovania (tím č.7 FIIT STU 2013/2014)
- Konvencie písania kódu v projekte (tím č.7 FIIT STU 2013/2014)

12.2.3 Slovník pojmov

- Šprint – jedna iterácia vývoja softvérového produktu
- Príbeh šprintu – identifikovaná požiadavka na softvérový produkt
- Prototyp – funkčná verzia softvérového produktu

12.2.4 Roly a zodpovednosti

Rola	Zodpovednosti
Manažér dokumentácie	<ul style="list-style-type: none"> • Vytvorenie základnej štruktúry dokumentov • Opis šprintu • Zjednotenie štýlu dokumentov
Manažér plánovania	<ul style="list-style-type: none"> • Vytvorenie plánu
Člen tímu	<ul style="list-style-type: none"> • Opis príbehu na ktorom člen tímu pracoval
Jazykový korektor	<ul style="list-style-type: none"> • Odstránenie gramatických chýb v dokumente

12.2.5 Tvorba dokumentácie

Táto kapitola je zameraná na definovanie štýlu písania dokumentov.

12.2.5.1 Štýl písania textu dokumentu

Zásady pre písanie textu:

- pre normálny text sa používa písmo Calibri 11pt
- pre text sa používa riadkovanie 1,15
- pri písaní vlastných mien napríklad meno funkcie alebo meno nástroja sa používa *kurzíva*
- súvislý text je vždy zarovnaný do bloku

12.2.5.2 Štýl písania nadpisov

V názvoch kapitol sa používa číslovanie. Všetky kapitoly sa identifikujú číslom a následným názvom kapitoly, kde prvé písmeno je veľké. Číslované nadpisy tvoria základ pre obsah dokumentu.

V podkapitolách sa môžu používať aj ďalšie nadpisy písané tučným písmom, ktoré nebudú v obsahu dokumentu viditeľné. Všetky nadpisy sú zarovnané doľava.

Príklad nadpisov:

<p>1 KAPITOLA</p> <hr/> <p>1.1 PODKAPITOLA</p> <p>1.1.1 Podpodkapitola</p> <p>Nečíslovaný nadpis</p>
--

12.2.5.3 Štýl písania číselných a nečíselných zoznamov

Dokument môže obsahovať číselné aj nečíselné zoznamy viacerých úrovní. Nečíselné zoznamy sa používajú pri zozname, kde nezávisí na poradí prvkov v zozname. Číselní zoznam sa používa v prípade, že prvky majú určité poradie. Je možné aj prípadná kombinácia číselných a nečíselných zoznamov.

Príklad zoznamov:

<ul style="list-style-type: none">• Prvok 1<ul style="list-style-type: none">○ Prvok 1-1○ Prvok 1-2• Prvok 3 <ol style="list-style-type: none">1. Prvok 1<ol style="list-style-type: none">1.1. Prvok 1-11.2. Prvok 1-22. Prvok 2
--

<ol style="list-style-type: none">1. Prvok 1<ul style="list-style-type: none">• Prvok 1-1• Prvok 1-22. Prvok 2
--

12.2.5.4 Pridávanie obrázkov

Obrázky v dokumente musia byť jednoznačne identifikované. Na takéto identifikovanie sa používa číslovanie obrázkov a má formu Obr. č. X. Ku každému obrázku patrí aj opis obrázku. Identifikátor a opis obrázku je písané tučným písmom. Pri odkazovaní na obrázky v texte sa používa identifikátor obrázku.

Príklad obrázku:



Obr. č.1 Vzorový obrázok

Príklad odkazovania na obrázok:

Na obrázku č. 1 je znázornený vzorový obrázok.

12.2.5.5 Pridávanie tabuliek

Tabuľky sa identifikujú značkou Tab. č. X. Ku každej tabuľke patrí stručný opis tabuľky. Tieto informácie sú napísané pod každú tabuľku tučným písmom a sú centované spolu s tabuľkou. V príklade nižšie je uvedený štýl tabuľky.

Príklad tabuľky:

Názov stĺpca 1	Názov stĺpca 2	Názov stĺpca 3	Názov stĺpca 4
Názov riadku 1	Prvok 1	Prvok 3	Prvok 5
Názov riadku 2	Prvok 2	Prvok 4	Prvok 6

Tab. č. 1 Vzorová tabuľka

Príklad odkazovania na tabuľku:

V tabuľke č.1 sú uvedené vzorové prvky tabuľky.

12.2.6 Štruktúra projektovej dokumentácie

Technická dokumentácia obsahuje nasledujúce časti:

1. Titulná strana
2. Úvod – účel dokumentu
3. Stanovenie cieľov projektu – plán tímu
4. Dokumentované šprinty – opisy šprintov
5. Celkový pohľad na projekt – opis vytvoreného prototypu

12.2.6.1 Opis šprintu

X. NÁZOV ŠPRINTU
Číslo šprintu: (číslo)
Začiatok šprintu: (dátum)
Koniec šprintu: (dátum)
Príbemy:
<ul style="list-style-type: none">• Zoznam príbehov

Príklad pre opis šprintu:

12.2.6.2 Opísanie príbehov šprintu v technickej dokumentácii

Celkový opis príbehov šprintu sa delí do viacerých častí. Tieto časti sú nasledujúce:

1. Stručný opis príbehu
2. Analýza
3. Návrh
4. Riešenie
5. Testovanie

Poradie týchto častí v opise príbehu musí dodržať poradie, ktoré je definované v zozname vyššie. V každom príbehu je nutné vyjadriť sa k bodom „Stručný opis“ a „Riešenie“, pričom ostatné body nie je nutné uviesť.

Príklad pre opis príbehu šprintu:

X.X NÁZOV PRÍBEHU

Stručný opis príbehu.

Analýza

Analýza problémovej oblasti, nástrojov, externých knižníc.

Návrh

Opis navrhnutého riešenia.

Riešenie

Opis riešenia, implementačné detaily.

Testovanie

Opis testovacieho procesu, výsledkov.

12.2.7 Štruktúra dokumentácie riadenia projektu

Technická dokumentácia obsahuje nasledujúce časti:

1. Titulná strana
2. Úvod – účel dokumentu
3. Zoznam kompetencií tímu
4. Záznamy zo stretnutí
5. Úlohy členov tímu
6. Manažment monitorovania a testovania projektu
7. Manažment kvality projektu
8. Manažment rizík a rozsahu
9. Manažment rozvrhu (plánovanie)
10. Manažment podpory vývoja a integrácie
11. Manažment komunikácie a ľudských zdrojov
12. Manažment tvorby dokumentácie
13. Preberacie protokoly

12.2.7.1 Šablóna zápisnice zo stretnutia



Zápis zo stretnutia tímového projektu

Dátum: (dd.mm.rrrr)

Miesto: Jobs. štúdio

Číslo tímu: 7

Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

1. **Názov 1. bodu stretnutia**
 - Prvok 1
 - Prvok 2
2. **Názov 2. bodu stretnutia**
3. ..
- X. **Názov X. bodu stretnutia**

Zapísal: (priezvisko)

Overil: (priezvisko)

Trvanie stretnutia: (trvanie v hodinách)

12.3 METODIKA ZBERU A SPRACOVANIA POŽIADAVIEK

Cieľom tejto metodiky je definovať postup pri vytváraní nových požiadaviek a ich spracovaní pri vývoji softvéru metódou Scrum. V metodike sú popísané postupy definovania nových požiadaviek, schválenia požiadaviek, ohodnotenia požiadaviek, priradenia požiadaviek, dekompozície požiadaviek, riešenia podúloh, splnenia podúloh a splnenia požiadaviek.

12.3.1 Dedikácia

Táto metodika je primárne určená pre členov tímu č.7 *Lucky Seven* na predmete Tímový projekt I, II na FIIT STU v akademickom roku 2013/2014.. Môže byť však zdrojom informácií aj pre iný menší tím ľudí, vyvíjajúci softvér metódou Scrum.

12.3.2 Súvisiace metodiky

- Metodika plánovania (tím č.7 FIIT STU 2013/2014)

12.3.3 Slovník pojmov

- Scrum – Metóda agilného procesu vývoja softvéru
- ScrumMaster – Kľúčová rola v agilnom tíme
- Šprint – Jedna iterácia procesu tvorby softvéru Scrum metódou, počas ktorej prebehne celý zmenšený vodopádový vývoj, pričom jeho výsledkom je produkt v určitom štádiu funkčnosti.
- Backlog projektu – Zoznam požiadaviek pre celý projekt, prioritizovaných produktovým vlastníkom
- Backlog šprintu – Zoznam požiadaviek, ktoré bude tím riešiť v nadchádzajúcom šprinte, vyberá sa z Backlogu projektu
- Požiadavka - definuje prácu, ktorá musí byť vykonaná

12.3.4 Roly a zodpovednosti

Rola	Zodpovednosti
Produktový vlastník	<ul style="list-style-type: none"> • Zadanie novej požiadavky • Určenie priority požiadavky • Schválenie požiadavky
Členovia tímu	<ul style="list-style-type: none"> • Zadanie novej požiadavky • Ohodnotenie náročnosti požiadavky • Určenie zodpovednej osoby za riešenie danej požiadavky z Backlogu šprintu • Riešenie pridelených úloh • Dekompozícia požiadavky
Scrummaster	<ul style="list-style-type: none"> • Vedenie ohodnotenia položky • Vedenie stretnutí tímu
Vedúci tímu	<ul style="list-style-type: none"> • Určenie zodpovednej osoby za danú požiadavku, pokiaľ o riešenie nemá záujem žiaden člen tímu, alebo má o požiadavku záujem viac než jeden člen tímu

12 Metodiky

12.3.5 Zber a spracovanie požiadaviek v TFS

12.3.5.1 Vytvorenie novej požiadavky v TFS

Požiadavka je do TFS zadávaná produktovým vlastníkom, alebo členom tímu. Požiadavky môžu byť vytvárané počas celého behu projektu. Názov požiadavky je zadávaný v slovenskom jazyku, nesmie obsahovať viac ako 10 slov. Názov obsahuje jedno slovesné podstatné meno a má jasne vyjadrovať danú požiadavku. Stav požiadavky „State“ je nastavený na „New“. V časti „Description“ je zadaný popis danej požiadavky. Popis je zadávaný v slovenskom jazyku a bližšie popisuje danú požiadavku. Po uložení je požiadavka pridaná do Backlogu projektu. Náhľad sa nachádza na obrázku č. 1.

Product Backlog Item 317*: Oddelenie RCS projektov od logických projektov

Tags Add...

Oddelenie RCS projektov od logických projektov

Iteration WhereIsMyCode\Zima

STATUS

Assigned To

State New

Reason New backlog item

DETAILS

Effort

Business Value

Area WhereIsMyCode

Backlog Priority

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

Zadávateľ vyžaduje oddelenie RCS projektov od logických projektov, aby systém reálne kopiroval štruktúru projektov a používateľ mal možnosť spravovať prístup iných používateľov k projektu.

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

Save Save and Close Cancel

Obr. č.1 Vytvorenie novej požiadavky v TFS

12.3.5.2 Schválenie novej požiadavky v TFS

Požiadavky sú schvaľované produktovým vlastníkom po ich pridaní do Backlogu projektu. Pri schválení požiadavky je zadaná priorita požiadavky „Backlog Priority“. Priorita požiadavky je zvolená v rozhraní od 1 do 1000000, pričom 1 predstavuje najvyššiu prioritu. Toto číslo je určené produktovým vlastníkom. Stav požiadavky „State“ je nastavený na „Approved“. Náhľad sa nachádza na obrázku č. 2.

Product Backlog Item 317*: Oddelenie RCS projektov od logických projektov

Tags Add...

Oddelenie RCS projektov od logických projektov

Iteration WhereIsMyCode\Zima

STATUS

Assigned To

State Approved

Reason Approved by the Product Owner

DETAILS

Effort

Business Value

Area WhereIsMyCode

Backlog Priority 205506

DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

Zadávateľ vyžaduje oddelenie RCS projektov od logických projektov, aby systém reálne kopiroval štruktúru projektov a používateľ mal možnosť spravovať prístup iných používateľov k projektu.

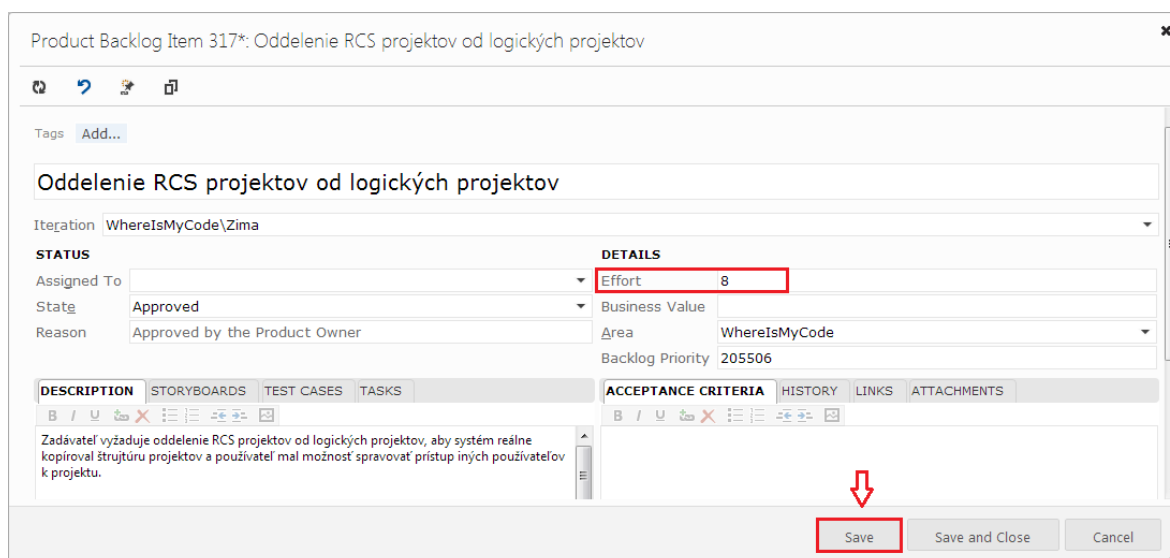
ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

Save Save and Close Cancel

Obr. č.2 Schválenie novej požiadavky v TFS

12.3.5.3 Ohodnotenie novej požiadavky v TFS

Po schválení novej požiadavky je náročnosť danej požiadavky ohodnotená členmi tímu na najbližšom stretnutí tímu. Pokiaľ sa daná požiadavka tímom na nasledujúcom stretnutí neohodnotí, ohodnotenie sa presunie na nasledujúce stretnutie. Hodnotenie prebieha na základe hodnôt fibonacciho postupnosti. Členovia tímu majú možnosť vybrať z hodnôt 0,5, 1, 2, 3, 5, 8, 13, 20, 40, 100, pričom hodnota 100 predstavuje najvyššiu zložitosť. Všetci členovia tímu, po vyzvaní Scrummastra, vyberú jednu z kariet, ktoré sú označené hodnotami fibonacciho postupnosti. Pokiaľ sa členovia nezhodnú na hodnote náročnosti, prebieha diskusia medzi členmi tímu, v ktorej sa vyjadria k náročnosti požiadavky tí členovia tímu, ktorí zvolili minimálne a maximálne hodnoty náročnosti. Po ukončení diskusie prebieha opätovné hlasovanie. Ak sa členovia tímu nezhodnú, opätovne sa začína diskusia a hlasovanie. Ak medzi členmi tímu dôjde k zhode, v TFS, v časti „Effort“ hodnotenej požiadavky, je Scrummastrom nastavená hodnota, na ktorej sa členovia tímu zhodli. Náhľad sa nachádza na obrázku č. 3.



Obr. č.3 Ohodnotenie novej požiadavky v TFS

12.3.5.4 Priradenie požiadavky členovi tímu

Na stretnutí tímu, na začiatku každého šprintu je vytváraný Backlog šprintu. Do Backlogu šprintu sú požiadavky vkladané na základe priority z Backlogu projektu Scrummastrom. Požiadavky vkladané do Backlogu šprintu musia byť ohodnotené (časť 6.3). Člen tímu má pravo vybrať si, ktorú požiadavku chce riešiť. Pokiaľ nechce danú požiadavku riešiť žiaden z členov tímu o jej priradení rozhodne vedúci tímu. Vedúci tímu tiež rozhodne o priradení požiadavky pokiaľ má o riešenie záujem viac než jeden člen tímu.

Po priradení požiadavky členovi tímu je stav požiadavky „State“ nastavený na „Committed“ a je jej priradené meno jej riešiteľa v časti „Assigned to“. Tento úkon vykoná člen tímu, ktorému je požiadavka priradená. Náhľad sa nachádza na obrázku č. 4.

12 Metodiky

The screenshot shows the 'Product Backlog Item 317*: Oddelenie RCS projektov od logických projektov' in TFS. The 'Assigned To' field is set to 'Bc. Zuzana Greslikova' and the 'State' is 'Committed'. A red box highlights these two fields. A red arrow points to the 'Save' button at the bottom right of the form.

Obr. č.4 Priradenie požiadavky členovi tímu v TFS

12.3.5.5 Dekompozícia požiadaviek

Člen tímu, ktorému bola priradená požiadavka, spraví k danej požiadavke dekompozíciu a rozdelí ju na podúlohy do 24 hodín po jej priradení. Názov podúlohy je zadávaný v slovenskom jazyku, nesmie obsahovať viac ako 10 slov, pričom musí obsahovať slovesné podstatné meno. Podúlohe je riešiteľom stanovená dĺžka práce, potrebná na jej vyriešenie. Pri stanovení dĺžky práce sa berie do úvahy celková náročnosť celej požiadavky, stanovená tímom pri ohodnotení požiadavky. Náhľad sa nachádza na obrázku č. 5.

The screenshot shows the 'New Task 2*: Vytvorenie rozhrania' form in TFS. The title is 'Vytvorenie rozhrania'. The 'Assigned To' is 'Bc. Zuzana Greslikova' and the 'Remaining Work' is '4'. Red boxes highlight the title, 'Assigned To', and 'Remaining Work' fields. A red arrow points to the 'Save' button at the bottom right.

Obr. č.5 Vytvorenie podúlohy v TFS

12.3.5.6 Riešenie podúloh požiadavky

Po začatí riešenia danej podúlohy riešiteľ zmení jej stav „State“ na „In progress“. Počas riešenia riešiteľ znižuje zostávajúci čas na riešenie podúlohy v časti „Remainingwork“. Náhľad sa nachádza na obrázku č. 6.

12 Metodiky

The screenshot shows a TFS task titled "Task 250*: Vytvorenie rozhrania". The task is assigned to "Bc. Zuzana Greslikova" and is currently in the "In Progress" state. The "Remaining Work" is set to 2. The "Reason" is "Work started" and the "Area" is "WhereIsMyCode". The "Save" button at the bottom is highlighted with a red box and a red arrow pointing to it.

Obr. č.6 Riešenie podúlohy v TFS

12.3.5.7 Splnenie podúloh

Pokiaľ člen tímu splní danú podúlohu. Zmení jej stav „State“ na „Done“. Náhľad sa nachádza na obrázku č. 7.

The screenshot shows the same TFS task "Task 250*: Vytvorenie rozhrania". The status has changed to "Done" and the "Reason" is now "Work finished". The "Remaining Work" field is empty. The "Save" button at the bottom is highlighted with a red box and a red arrow pointing to it.

Obr. č.7 Splnenie podúlohy v TFS

12.3.5.8 Splnenie požiadavky

Pokiaľ člen tímu splní všetky podúlohy, priradené k požiadavke, podľa zadania v celom rozsahu. Zmení stav danej požiadavky „State“ na „Done“. Náhľad sa nachádza na obrázku č. 8.

12 Metodiky

Product Backlog Item 317*: Oddelenie RCS projektov od logických projektov

Tags Add...

Oddelenie RCS projektov od logických projektov

Iteration WhereIsMyCode\Zima

STATUS	DETAILS
Assigned To Bc. Zuzana Greslikova	Effort 8
State Done	Area WhereIsMyCode
Reason Work finished	Backlog Priority 205506

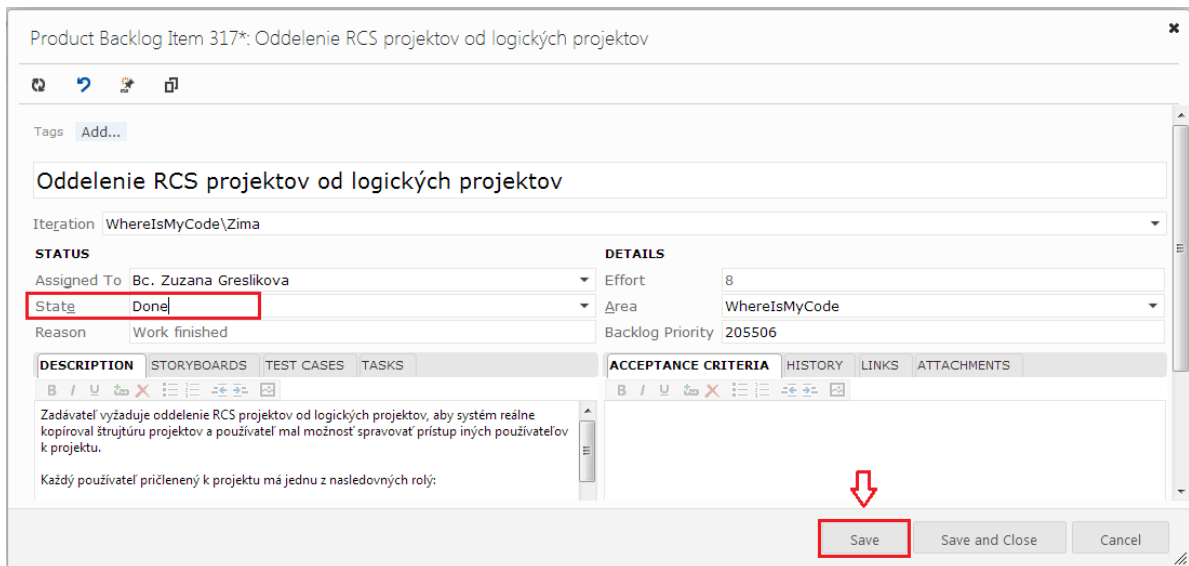
DESCRIPTION | STORYBOARDS | TEST CASES | TASKS

Zadávateľ vyžaduje oddelenie RCS projektov od logických projektov, aby systém reálne kopiroval štruktúru projektov a používateľ mal možnosť spravovať prístup iných používateľov k projektu.

Každý používateľ priradený k projektu má jednu z nasledovných rolí:

ACCEPTANCE CRITERIA | HISTORY | LINKS | ATTACHMENTS

Save Save and Close Cancel



Obr. č.8 Splnenie požiadavky v TFS

12.4 METODIKA VERZIOVANIA A PÍSANIA KOMENTÁROV K ODOVZDANIAM

Účelom tejto metodiky je definícia postupu pri vytváraní verzií kódu. Popisuje vytváranie verzií pomocou nástroja na správu verzií zdrojových kódov - Team Foundation Server (TFS). Táto metodika popisuje kedy a ako správne vytvárať nové verzie, ako písať komentáre k novým verziám a ktoré súbory commitovať.

12.4.1 DEDIKÁCIA

Táto metodika je určená pre všetkých vývojárov v tíme, ktorí sa podieľajú na vývoji produktu, vytvárajú nové verzie alebo vykonávajú revíziu zdrojových kódov.

Touto metodikou sa je nutné riadiť v prípade, ak vývojár vytvoril novú verziu zdrojového kódu a chce ju odoslať do systému TFS.

12.4.2 ZOZNAM SÚVISIACICH DOKUMENTOV

Dokumentácia TFS 2012 - <http://msdn.microsoft.com/sk-sk/vstudio/ff637362.aspx>

12.4.3 SLOVNÍK POJMOV

- VisualStudio 2012 – použité vývojové prostredie;
- Team Foundation Server – systém na správu verzií zdrojových kódov;
- Commit / Check In – odoslanie novej verzie zdrojového kódu do TFS;
- Get LatestVersion – získanie najnovšej verzie zdrojového kódu z TFS;
- Solution (*.sln) – riešenie, ktoré obsahuje všetky súbory projektu;
- „csproj“ – súbor, ktorý obsahuje informácie o súboroch v projekte;
- Team Explorer – okno vo VS, ktoré zobrazuje informácie o tímovom projekte;
- Solution Explorer – okno vo VS, v ktorom sú zobrazené zdrojové kódy a všetky ostatné súbory, ktoré patria do Solution;
- Entita – súbor, trieda, metóda, funkcionálna;
- Build – vytvorenie spustiteľnej verzie produktu;
- Konflikt – rozdiel medzi lokálnou a serverovou verziou zdrojového kódu
- Revízia zdrojových kódov – prezeranie vytvorených zdrojových kódov za účelom odhalenia chýb a zachovania udržiavateľnosti kódu;

12.4.4 Vymedzenie skratiek

Celý názov pojmu	Použitá skratka
VisualStudio	VS
Team Foundation Server	TFS
Identifikačné číslo	ID

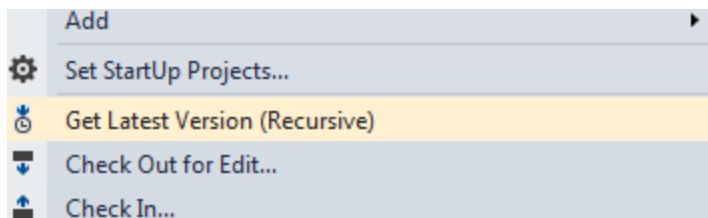
12.4.5 Opisy postupov

Táto kapitola opisuje postupy jednotlivých činností, potrebných pre správne vytvorenie novej verzie produktu a je určená pre všetkých programátorov. Nová verzia zdrojového kódu vzniká vo vývojovom prostredí VS 2012. Na prácu s tímovým projektom musí byť VS pripojené na TFS server.

12.4.5.1 Získanie poslednej verzie zdrojového kódu

Po spustení projektu musí programátor pracovať nad poslednou verziou projektu, aby nevznikali problémy súvisiace s prácou nad neaktuálnou verziou. Je nutné synchronizovať celú „Soloution“, pretože obsahuje súbor „csproj“.

Pre stiahnutie poslednej verzie zdrojového kódu sa na celú „Soloution“ vyberie možnosť „Get LatestVersion (Recursive)“ – prebehne synchronizácia lokálnej verzie so serverovou verziou.

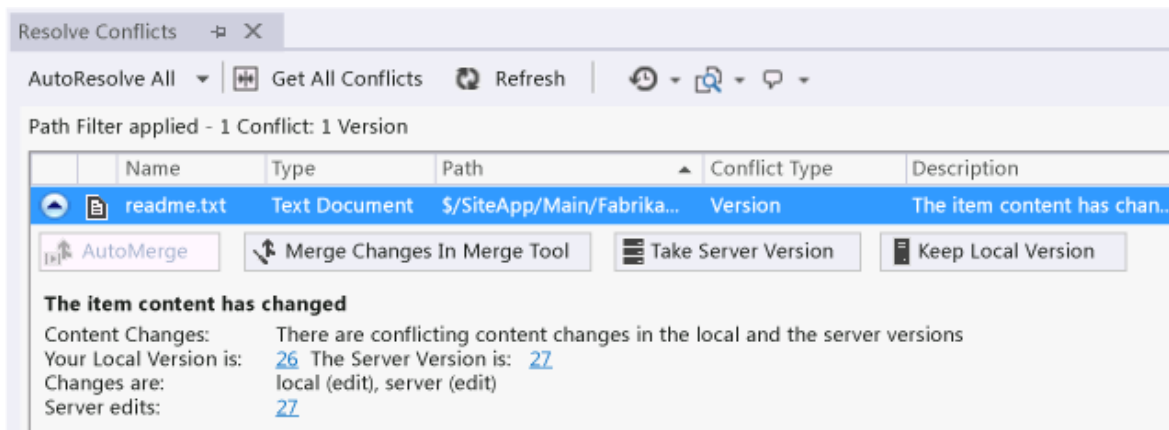


Obr. č.9. Get LatestVersion

Ak sa pri synchronizácii vyskytnú konflikty, TFS sa pokúsi o korektné spojenie („merge“) lokálnej verzie so serverovou. Ak spojenie súbor prebehne bez problémov, vývojár úspešne získal najnovšiu verziu zdrojového kódu. Pokiaľ spojenie súborov neprebehne úspešne, pokračuje sa časťou „2.2“

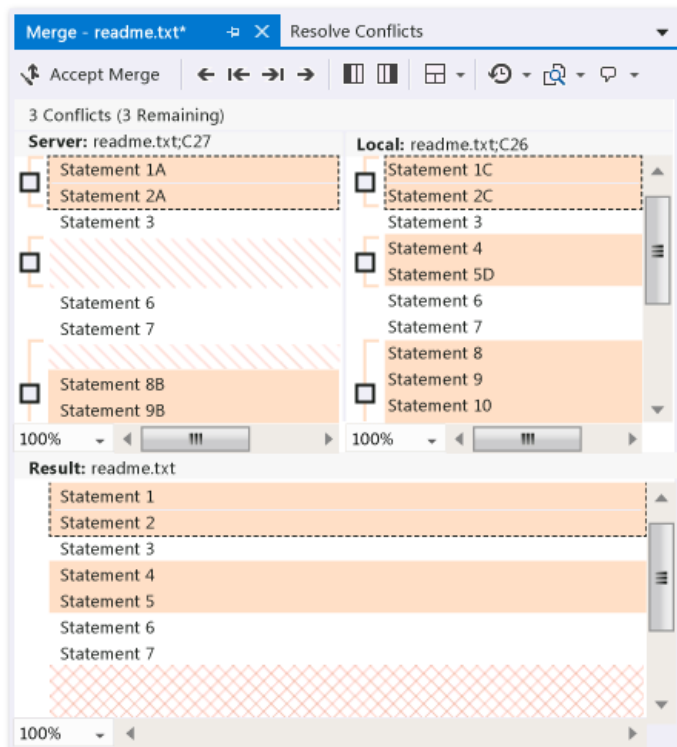
12.4.5.2 Riešenie konfliktov pri spájaní lokálnej verzie so serverovou

Pokiaľ pri získavaní novej verzie zo servera vznikne konflikt, ktorý sa nedá automaticky vyriešiť, zobrazí sa okno pre riešenie konfliktov („ResolveConflicts“). V okne sú zobrazené všetky automaticky nevyriešiteľné konflikty.



Obr. č.10. Zobrazenie konfliktných súborov

Každý konfliktný kód je potrebné prezrieť a ručne vyriešiť. Programátor musí sám rozhodnúť, či sa použije serverová verzia alebo lokálna. Programátor vyberie z oboch častí zdrojového kódu tie časti, ktoré zostanú vo výslednej verzii a potvrdí akceptovanie „merge“.



Obr. č.11. Riešenie konfliktov

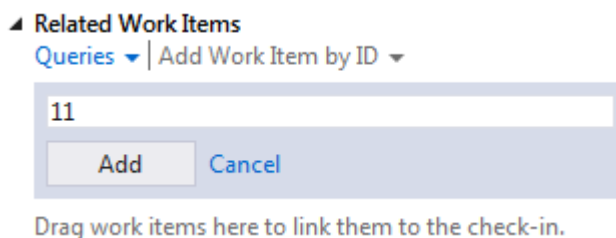
12.4.5.3 Commit / Check In

Programátor spraví „Check In“ nad celou „Soloution“, aby sa commitli aj súbory „csproj“. Programátor commituje súbory vždy, keď pridá nový súbor alebo dokončí úlohu na ktorej pracuje.

Pred commitom musí programátor spustiť všetky testy, ktoré musia korektne zbehnúť. Pokiaľ testy nezbehnú korektne, je nutné opraviť tieto chyby a až potom spraviť „Check In“. Commitovaný kód musí byť skompilovateľný a spustiteľný.

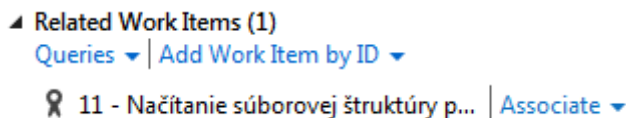
12.4.5.4 Priradenie riešenej úlohy k odovzdaniu

Pokiaľ programátor rieši konkrétnu úlohu („workitem“) v poli pre „AddWorkItem by ID“ vloží ID riešenej úlohy.



Obr. č.12. Vyhľadanie úlohy

Server TFS vyhľadá úlohu v zozname úloh a pokiaľ úloha existuje, pridá úlohu medzi súvisiace úlohy k novej verzii. Týmto spôsobom je možné priradiť viacero súvisiacich úloh k jednému commitu.



Obr. č.13. Asociácia odovzdania s úlohou**12.4.5.5 Štruktúra popisu zmien – komentár ku commitu**

Pri každom commitu je potrebné zadať komentár. Pre čo najlepšiu prehľadnosť sa treba riadiť nasledujúcou štruktúrou (vec – funkcionálna, súbor...). Pre komentár je zásadné kľúčové anglické slovo vo formáte „(Add / Change / Delete / Fix / Test) :“. Text komentárov sa píše po slovensky bez diakritiky.

Komentár	Popis ku komentáru
Add: - pridaná entita 1 - pridaná entita 2 ...	Stručný popis každej pridanej entity (názov + čo robí), oddelených odrážkami, každá na novom riadku.
Change: - upravená entita 1 - upravená entita 2..	Stručný popis každej zmenenej entity (názov + čo robí), oddelených odrážkami, každá na novom riadku.
Delete: - vymazaná entita 1 - vymazaná entita 2..	Názov vymazanej entity + dôvod prečo bola vymazaná, oddelených odrážkami, každá na novom riadku.
Fix: - opravená entita 1 - opravená entita 2..	Názov opravenej entity + popis chyby, ktorá bola opravená, oddelená odrážkami, každá na novom riadku.
Test: - pridaný / zmenený / opravený - test	Typ: „pridaný/zmenený/opravený“ + stručný opis čo testuje. Každý test oddelený odrážkou, každý na novom riadku.
Codereview - otagovaný kód	Popis codereview – pridané tagy / opravené tagy + popis

Príklad komentára k odovzdaniu:

<p>Add: - rememberme - zapamataniepouzivatela aj po vypnutiprehliadaca (volitelne) - settings - pridane AstEndpointClient - migracie – pridane exceptions</p> <p>Change: - migracie - vykonanie codereviewtagov od karola - migracie - zmenenyinitscript - doplneny o versiontabulku - serviceclient - komentare</p> <p>Delete: - createversion table migrationscript – nepotrebne</p> <p>Fix: - prihlasovanie – opravena chyba, ktoraumoznovaloprihlasenie s nespravnym heslom</p> <p>Test: - pridany test pre login</p> <p>Codereview – pridane tagy - bugs</p>

12.5 METODIKA TESTOVANIA

Účelom a cieľom metodiky je vytvoriť jednotný postup testovania v priebehu vývoja softvérového produktu. Tento postup zahŕňa definíciu procesov, z ktorých sa skladá. Definovaný proces sa špecializuje na tvorbu jednotkových testov v prostredí frameworku MVC 4.

12.5.1 Dedikácia

Touto metodikou sa riadia členovia tímu po tom ako dokončia časť kódu, ktorá zodpovedá konkrétnemu „user-story“. Člen tímu (autor kódu) po dokončení daného kódu je zodpovedný za tvorbu automatického testu, ktorý bude testovať funkčnosť práve dokončeného „controllera“.

Pre každú metódu vytvoreného „controllera“ je nutné vytvoriť jednotkový test (metódu) v testovacej triede určenej pre daný „controller“.

12.5.2 Súvisiace metodiky

- Konvencie písania kódu v projekte
- Metodika používania verziovacieho systému – písanie komentárov

12.5.3 Pojmy

- MVC : model viewcontroller je návrhový vzor softvéru, ktorý je často vytvorený ako balíček funkcií pre zjednodušenú implementáciu.
- jednotkový test: metóda nízkoúrovňového testovania „jednotiek“ softvéru. V tomto prípade jednotky zväčša predstavujú „Controllers“ teda triedy s naprogramovanou logikou.
- „framework“: balíček funkcií a nástrojov.
- „user story“: je prípad použitia alebo funkcia softvéru.
- MVC 4: „framework“ pre tvorbu web stránok v prostredí MS VisualStudio v programovacom jazyku C#
- test projekt: projekt v rámci „solution“ daného MVC 4 produktu, tento projekt obsahuje triedy s jednotlivými jednotkovými testami k daným funkciám hlavného projektu.
- „controller“: jednotka kódu (trieda), ktorá ovláda istú funkciu (user-story) na web-stránke.
- „assert“: funkcie pre overovanie vlastností dát.
- „breakpoint“: bod v kóde, kde sa zastaví vykonávanie programu pri „debugu“

12.5.4 Vytvorenie jednotkového testu v MVC 4

Táto metodika sa zameriava na tvorbu jednotkových automatických testov v prostredí MVC4 framework-u. Proces sa skladá z krokov, ktoré sú opísané v nasledujúcich kapitolách.

12.5.4.1 Určenie zamerania testu

V tomto kroku je nutné aby sa tvorca testu zamyslel nad cieľom a funkciou testovaného „controllera“ a jeho metód, tak aby bol schopný vytvoriť efektívny automatický test. Najprv je teda nutné si určiť čo sa ide testovať.

Zásady použitia funkcií „assert“ (ďalej len funkcia):

- Nepoužívať viac ako 3 funkcie v jednom teste.
- Nepoužívať viac funkcií na testovanie toho istého kódu.
- Používať funkcie v správnom poradí a brať ohľad na to, že už po prvom neúspešnom výsledku funkcie test končí so statusom „fail“.

Príklad:

Ak použijeme AreEqual() a zároveň IsNotNull(). Nemá zmysel použiť IsNotNull ako druhé v poradí na ten istý objekt, ktorý bol použitý v prvej funkcií AreEqual().

- V každej funkcií vložiť komentár (posledný atribút funkcie), ktorý opíše situáciu po neúspešnom výsledku funkcie v slovenčine.

Príklad:

```
Assert.AreEqual(data.Count, expectedCount, "Očakávaný počet dát sa nerovnal reálnemu");
```

Voľby vhodnej funkcie „assert“:

- Pri očakávanom veľkom množstve dát použiť:

```
Assert.AreEqual(x, y, "Komentár");
```

Kde „x“ je reálny počet vrátených entít v dátach a „y“ je očakávaný počet.

- V prípade, že v model ma obsahovať špecifický typ premennej použiť:

```
var model = result.ViewData.Model.x as y
Assert.IsNotNull(model, "Komentár");
```

Kde „x“ je atribút modelu a „y“ je jej špecifický typ.

- Ak testovaná trieda nemá väčšiu logiku a neprenáša špecifické dáta ale iba zobrazuje statické údaje použiť:

```
Assert.IsNotNull(x, "Komentár");
```

Kde „x“ je výsledok testovanej metódy „controllera“.

V prípade nešpecifikovaného prípadu je na programátorovi aby vhodne zvolil funkciu na overenie výstupu testovanej metódy podľa uvedených zásad.

12.5.4.2 Tvorba testu**1. Vytvorenie testovacej triedy pre testovaný „controller“**

- Programátor – tvorca testu vytvorí triedu v testovacom projekte skopírovaním základnej testovacej triedy „HomeControllerTest“ a jej premenovaním.
- Meno novej testovacej triedy: <menoTestovanehoControllera>Test.cs
- Metódy v rámci skopírovanej triedy budú premenované podľa metód testovaného „controllera“ a budú obsahovať kód, ktorý bude testovať danú metódu. Pred každou metódou, ktorá je testovacia bude atribút v tvare: [TestMethod].
- Programátor môže v prípade rozsiahlejšieho testu vytvoriť pomocné metódy v testovacej triede. V tomto prípade sa pred metódami, ktoré nie sú testovacie nebude nachádzať atribút

[TestMethod]. Pred takýmito metódami sa bude nachádzať komentár v tvare: „// <testMetoda1>,<testMetoda2>,...,<testMetodaN>“. V tomto komentári budú uvedené všetky testovacie metódy, ktoré volajú danú pomocnú metódu .

- Pri písaní kódu, ktorý bude obsahovať testovacia metóda, sa bude programátor okrem tejto metodiky riadiť aj metodikou: Konvencie písania kódu v projekte.

2. Štruktúra testovacej metódy

- Prvá časť kódu testovacej metódy má názov: „Arrange“. V časti sa inicializujú a vytvoria časti potrebné na testovanie, minimálne tu bude vytvorená inštancia testovanej triedy.
- Druhá časť metódy je „Act“. V časti sa vykonajú potrebné úkony aby boli známe reálne vrátená hodnota a očakávané vrátená hodnoty aby mohli byť porovnané v ďalšej časti.

Reálna hodnota:

```
ViewResult result = controller.Index() as ViewResult
```

Očakávaná hodnota:

Ak je potrebná očakávaná hodnota: Programátor **využije iba elementárne** metódy na získanie tejto hodnoty.

- Posledná časť testovacej metódy je „Assert“. V tejto časti programátor overí správnosť údajov získaných v predchádzajúcej časti „Act“.

Pred každou časťou metódy bude komentár v tvare „// <názovČasti>“ príklad štruktúrovanej testovacej metódy:

```
[TestMethod]
public void Index()
{
    // Arrange
    HomeController controller = new HomeController();

    // Act
    ViewResult result = controller.Index() as ViewResult;

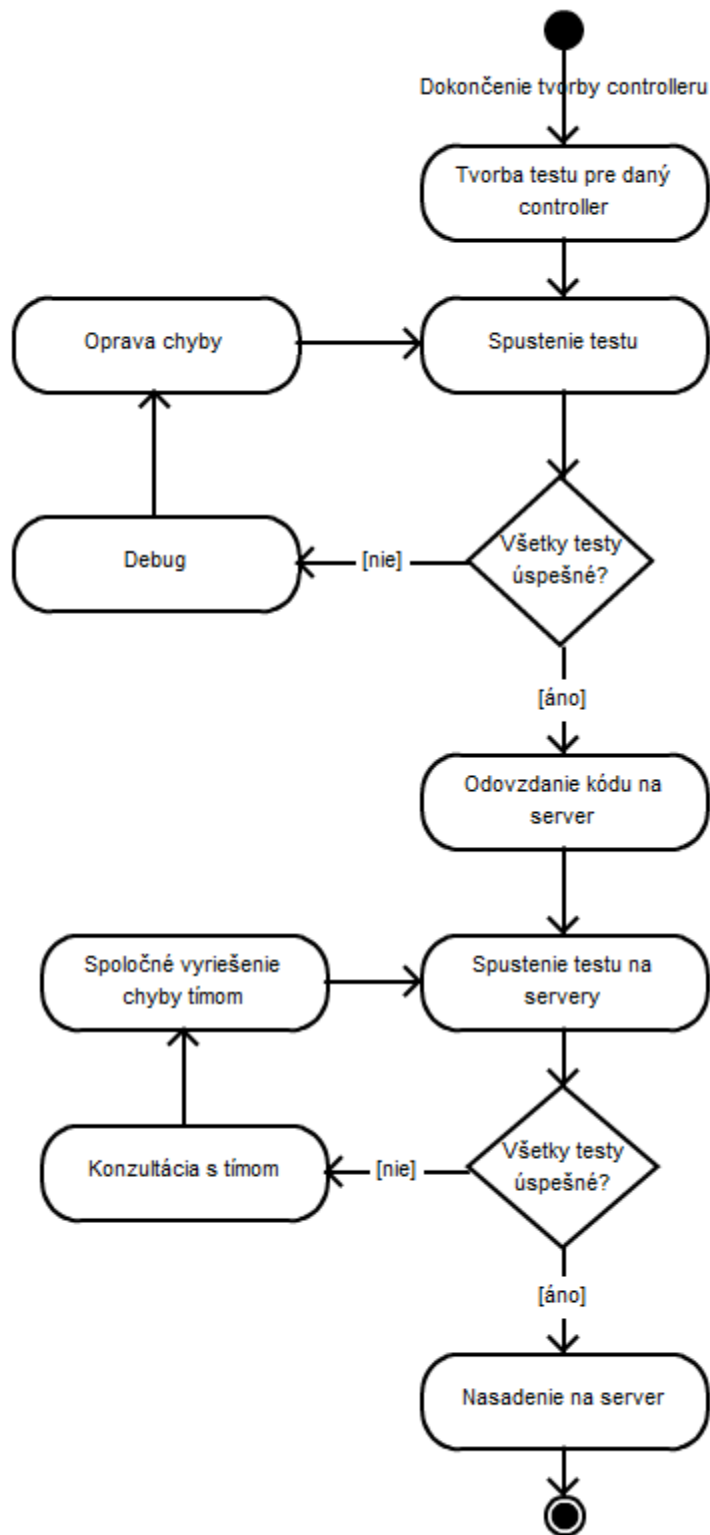
    // Assert
    Assert.IsNotNull(result, "Metóda vrátila null");
}
```

12.5.4.3 Spúšťanie a „debug“ testu

1. Po naprogramovaní testovacej triedy pre daný „controller“ programátor spustí test a overí jeho funkčnosť.
2. V prípade že test prebehol podľa očakávania programátor pristúpi ku kroku č. 4. Ak test neuspel programátor vykoná „debug“ (krok č. 3).

12 Metodiky

3. Programátor spustí daný neúspešný test v „debug“ móde a nastaví si „breakpoint“ na funkcií „assert“ aby zistil príčinu neúspechu. Ďalej programátor investiguje dôvod zlyhania či už v testovacej metóde alebo v „controllery“. Po nájdení chyby ju ihneď odstráni a znova spustí testovanie.
4. Po úspešnom zbehnutí testov programátor podľa metodiky: „Metodika používania verziovacieho systému – písanie komentárov“ odovzdá kód. Po odovzdaní kódu sa spustí nasadzovanie najnovšej verzie na server a zároveň sa automaticky spustia všetky testy, preto programátor skontroluje výsledky testov taktiež na tfs servery, kde sa tieto výsledky zobrazia. V prípade neúspešného testu iba na servery investiguje príčinu a konzultuje s tímom.



Obr. č. 1 Proces testovania.

12.6 KONVENCIE PÍSANIA ZDROJOVÉHO KÓDU V PROJEKTE

Účelom tejto metodiky je popis postupov písania zdrojového kódu v jazyku C# s cieľom zjednotenia zdrojového kódu písaného viacerými členmi jedného tímu. Taktiež definuje spôsob písania kódu v súboroch CSHTML a CSS.

12.6.1 Slovník pojmov

C#	programovací jazyk zameraný na objektovo orientovanú paradigmu
MVC	architektonický vzor určený na tvorbu komplexného softvéru
XML	značkovací jazyk zameraný na uchovanie dát
HTML	značkovací jazyk určený na tvorbu webových stránok
CSHTML	HTML skombinovaný s jazykom C# určený na tvorbu dynamických stránok
CSS	jazyk určený na tvorbu štýlov webových stránok

12.6.2 Súvisiace metodiky

- Metodika používania verziovacieho systému
- Metodika testovania
- Metodika tvorby dokumentácie

12.6.3 Dedikácia metodiky

Táto metodika je určená pre všetkých členov tímu, ktorých úlohou je tvorba zdrojového kódu a k nemu prislúchajúcich komentárov. Riadia sa ňou členovia tímu, programátori, pri písaní zdrojového kódu.

12.6.4 Písanie zdrojového kódu v jazyku C#

Táto kapitola sa venuje opisu konvencií používaných pri písaní zdrojového kódu v jazyku C#, vďaka čomu je zabezpečená konzistencia zdrojového kódu, jeho lepšia prehľadnosť a čitateľnosť.

12.6.4.1 Editor

Pre maximálnu konzistenciu v tíme používajú všetci členovia ten istý editor zdrojových kódov, Microsoft VisualStudio 2012. Tento editor je voľne dostupný zadarmo pre všetkých členov tímu. Obsahuje taktiež prednastavené automatické formátovanie.

12.6.4.2 Formátovanie kódu

Formátovanie kódu má presne stanovené pravidlá a dodržiavajú sa konvencie odporúčané pre písanie zdrojového kódu v jazyku C#.

Odsadzovanie textu

Na odsadzovanie sa striktnie používajú 4 medzery, nie znak tabulátora (obr. č. 1).

Deklarácia/priradenie

Na jednom riadku sa nachádza maximálne jedna deklarácia/priradenie.

Podmienka

Na jednom riadku sa nachádza maximálne jedna podmienka. Pokiaľ je podmienka komplexná, je vhodné zapísať každú časť podmienky na samostatný riadok.

Rozdelenie zdrojového textu

Medzi jednotlivými metódami sa nachádza aspoň jeden voľný riadok (obr. č. 3).

Zložené zátvorky

Každá zložená zátvorka (znaky '{' a '}') sa nachádza na samostatnom riadku (obr. č. 2).

Kľúčové slová

Kľúčové slová sa vkladajú vždy na nový riadok.

```
if (condition)
    Action();
```

Obr. č. 1 Jednoduchá podmienka a odsadenie textu.

```
if (condition)
    Action();
}
else
{
    SomethingElse();
}
```

Obr. č. 2 Vetvenie kódu.

```
private string Hello()
{
    return "Hello ";
}

public string SayHelloTo(string someone)
{
    return Hello() + someone;
}
```

Obr. č. 3 Zápis metód.

12.6.4.3 Názvy

Pri tvorbe názvov tried, metód a premenných sa dodržiavajú striktné pravidlá. Všeobecne platí, že sa používa iba základných 26 znakov abecedy. Názvy sú zvolené tak, aby čo najlepšie zachytili myšlienku, aby boli dobre čitateľné a aj na prvý pohľad ľahko pochopiteľné. Nepoužívajú sa žiadne oddeľovače slov, ako napr. znaky '-' (pomlčka) a '_' (podčiarkovník).

Názov triedy

Názov triedy začína vždy veľkým písmenom. V prípade ak je názov viacslovný, každé začiatkové písmeno nového slova je napísané veľkým písmenom (obr. č. 4). Niektoré triedy v názve nesú ich špeciálny význam. V tom prípade sa názov skladá z dvoch častí - zvolený názov + špeciálny význam. Na obr. 4 môžeme vidieť príklad takejto triedy - kontroléra.

Názov metódy

Tu platia rovnaké pravidlá ako pri názvoch tried. Každé prvé písmeno nového slova je veľké. Ostatné písmená sú malé (obr. č. 4).

Názov konštanty

Prvé písmeno konštanty je veľké, ostatné písmená sú malé. Ak má konštanta viacslovný názov, každé prvé písmeno slova je veľké (obr. č. 4).

Názov premennej

Prvé písmeno v názve premennej je malé. Ak má premenná viacslovný názov, ostatné slová začínajú vždy veľkým písmenom (obr. č. 4). Jednopísmenné premenné sú rezervované pre indexovanie v cykloch a nesmú byť inak používané.

```
// Nazov specialnej triedy.
UserPanelController {};

// Nazov triedy.
ClassName {};

// Nazov metody.
MethodName();

// Nazov konstanty.
ThisIsPi;

// Nazov premennej.
someVariable;
```

Obr. č. 4 Možné zápisy názvov.

12.6.4.4 Štruktúra tried

Každá trieda by mala byť štruktúrovaná uniformne. Na začiatku sú definované vlastnosti jej triedy, za nimi nasledujú konštruktory, po nich nasledujú definície verejných metód a nakoniec sú uvedené prívátne metódy. Vlastnosti definujeme jednotným štýlom (obr. č. 5).

```
public int number { get; set; }
public string descriptionTag { get; private set; }
```

Obr. č. 5 Príklady definovania vlastnosti triedy.

12.6.4.5 Jazyk

Kód aj všetky jeho časti sa píše v anglickom jazyku. Všetky premenné, názvy metód aj tried sú písane po anglicky pričom sa dbá, aby bolo anglické slovo korektne napísané. Výnimkou sú komentáre, ktoré sú popísané v kapitole 5.6. Na pomenovanie sa používajú slová, ktoré majú jednoduchý preklad a zachytávajú podstatu veci, ktorú pomenávajú (obr. č. 6). Skratky a akronymy sa v názvoch nepoužívajú.

```
// Nekorektny príklad.
PouzivatelskyPanel

// Nekorektny príklad.
UserCtrlPanel

// Korektny príklad.
UserControllerPanel
```

Obr. č. 6 Príklady pomenovávaní.

12.6.4.6 Komentáre v zdrojovom kóde

Veľmi dôležitou súčasťou zdrojového kódu sú komentáre, ktoré objasňujú jeho fungovanie. Komentáre sú písane v slovenskom jazyku bez diakritiky. Telo komentára začína veľkým písmenom a končí bodkou.

12.6.4.6.1 Jednoduchý komentár

Jednoduchý komentár je určený na krátke objasnenie nasledujúceho kódu, prípadne na poznačenie určitej informácie. Krátky komentár začína dvomi lomítkami ("//"), za nimi nasleduje medzera a po nej nasleduje text komentára. Komentár je vždy umiestnený na samostatnom riadku. Ak je komentár viacriadkový, každý nový riadok začína dvomi lomítkami (obr. č. 7).

```
// Toto je kratky komentar.  
// Druhy riadok.
```

Obr. č. 7 Jednoduchý komentár.

12.6.4.6.2 Komentár k metóde

Ku každej verejnej (public) metóde je povinný jej autor napísať komentár, ktorý ju stručne a jasne popisuje. Komentáre sú vo formáte XML a majú presne stanovenú štruktúru. Pre vygenerovanie šablóny stačí vložiť kurzor na prázdny riadok pred metódu, ktorú chcem popísať, a napísať tri lomítka ("///"). Editor následne vygeneruje korektnú XML šablónu. XML šablóna obsahuje polia, do ktorých sa vloží stručný popis metódy, popis jednotlivých parametrov, a popis objektu, ktorý metóda vracia (obr. č. 8).

```
/// <summary>  
/// Zobrazí suborovu strukturu projektu na zaklade jeho ID.  
/// </summary>  
/// <param name="projectId">ID projektu.</param>  
/// <param name="projectName">Nazov projektu.</param>  
/// <returns>View, ktore zobrazí suborovu strukturu a obsah suboru.</returns>
```

Obr. č. 8 Komentár k metóde vo formáte XML.

12.6.4.6.3 Komentár k triede

Pri vytváraní komentára k triede sa postupuje identicky ako pri vytváraní komentára k metóde. Opäť si vygenerujeme šablónu vložením troch lomítkov na prázdny riadok pred názov triedy. Následne vyplníme vygenerovanú XML šablónu (obr. č. 9).

```
/// <summary>  
/// Trieda predstavuje jeden uzol v strome.  
/// </summary>
```

Obr. č. 9 Komentár k triede.

12.6.5 Písanie zdrojového kódu v jazyku CSS

V rámci nášho projektu sa upravuje iba jediný súbor CSS, ktorý je globálny pre celý projekt a obsahuje všetky definície vizuálneho štýlu nášho projektu. Ako jazyk je opäť použitá angličtina.

12.6.5.1 Názvy

Názvy sa píšú iba malými písmenami. Pokiaľ je názov viacslovný, slová sa oddeľujú jediným povoleným oddeľovačom, pomlčkou (znak '-').

Názvy identifikátorov

Názov identifikátora vždy začína znakom mriežky ('#') a hneď za ním nasleduje samotný názov identifikátora (obr. č. 10). Tento identifikátor je jedinečný pre daný prvok v HTML súbore.

Názvy tried

Názov triedy začína znakom bodky ('.') a hneď za ním nasleduje názov triedy (obr. č. 10). Názov triedy sa môže vzťahovať ku viacerým prvkom v rámci viacerých HTML súborov.

```
/* Príklady identifikatorov. */
#tree-node
#id-name

/* Príklady tried. */
.table-class
.clickable
```

Obr. č. 10 Príklady názvov.

12.6.5.2 Komentár

Komentár začína znakmi lomítka a hviezdičky ("/*"), za nimi nasleduje jedna medzera a po nej nasleduje samotný text komentára. Po texte nasleduje opäť jedna medzera a celý komentár uzatvárajú opäť znaky hviezdičky a lomítka ("*/") (obr. č. 11). Komentár môže byť viacriadkový.

```
/* Kratky komentar. */

/* Viacriadkovy
komentar. */
```

Obr. č. 11 CSS komentár.

Komentár je písaný v slovenskom jazyku bez diakritiky.

12.6.6 Písanie zdrojového kódu v súboroch CSHTML

CSHTML súbory sú špeciálne súbory, ktoré kombinujú C# kód s HTML značkovacím jazykom. V týchto súboroch sa aplikujú pravidlá identicky ako v prípade súborov, ktoré obsahujú iba C# kód. Čiastkový HTML kód je formátovaný prehľadne a odsadzovaný štyrmi medzerami ako C# kód, aby sa zabezpečila lepšia prehľadnosť (obr. č. 12). HTML kód sa dá čiastočne predgenerovať priamo v textovom editore a to pravým kliknutím na akciu v kontroléry, zvolením položky "AddView".

```
<div class="title">
  <p>@Model.projectName</p>
</div>
```

Obr. č. 12 Ukážka kódu v CSHTML dokumente.

Najhlavnejšou zásadou je nevykonávanie žiadnej logickej a výpočtovej úlohy v CSHTML súboroch. Tieto súbory slúžia iba na zobrazenie stavu systému koncovému používateľovi.

12.6.6.1 Komentár

Komentár v CSHTML súboroch sa dá vytvoriť dvomi spôsobmi. Ako súčasť HTML kódu. začína sekvenciou znakov "<!--" a končí znakmi "-->". Telo komentára tvorí krátky text, ktorý je na začiatku aj na konci sprevádzaný jednou medzerou (obr. č. 13).

Ako súčasť C# kódu začína komentár znakmi "@*" a končí znakmi "*@". Telo komentára tvorí krátky text, ktorý je na začiatku aj na konci sprevádzaný jednou medzerou (obr. č. 13).

Obidva typy komentárov môžu obsahovať v rámci tela komentára znak nového riadku.

```
<!-- Kratky komentar. -->
@* Kratky komentar 2. *@
```

Obr. č. 13 Komentár v CSHTML súbore.

Je potrebné brať na vedomie, že vzhľadom na to, že tieto súbory nevykonávajú žiadnu logiku, komentáre by nemali slúžiť na objasnenie akcie, ktorá je v danom CSHTML súbore vykonávaná.

12.6.7 Fyzické rozdelenie zdrojového kódu v projekte

Hlavný projekt sa skladá z dvoch podprojektov a to "Web" a "Tests". Tieto podprojekty majú pevne zadefinovanú štruktúru a pri vytvorení nového zdrojového súboru je potrebné takýto súbor zaradiť do správneho adresára.

12.6.7.1 Web

Zdrojové kódy v tomto projekte sú vďaka vzoru MVC rozdelené do troch základných adresárov:

1. Models - práva s databázou, entitno-relačné triedy
2. Views - CSHTML súbory, ktoré zobrazujú údaje v prehliadači
3. Controllers - prepojenie medzi "Model" a "View". Vykonáva komplexnú logiku.

Hlavný CSS súbor sa nachádza v adresári "Content". Všetky JavaScriptové súbory sa nachádzajú v adresári "Scripts".

Dodatočné zdrojové súbory sú zaradované do logicky prislúchajúcich adresárov:

- AST - prístup a práca so službou AST-RCS
- Database - špeciálne úkony nad databázou

V prípade, ak súbor neprislúcha ani do jedného z týchto adresárov, člen tímu vytvorí nový adresár do ktorého zaradi jeho zdrojový kód.

12.6.7.2 Tests

Zdrojové kódy v tomto projekte obsahujú prislúchajúce testy k jednotlivým kontrolérom v projekte Web. Tieto testy sa vkladajú do adresára "Controllers".

12.7 METODIKA ITERACÍÍ PROJEKTU

Metodika sa zaoberá plánom jednotlivých iterácií, ktorým sa bude riadiť celý priebeh tvorby softvérového produktu. Obsahom metodiky sú jednotlivé kroky jednej iterácie (šprintu) vývoja. Tieto postupy slúžia na zosúladenie práce členov tímu. Pri dodržiavaní metodiky by nemali vznikajú problémy s nedodržaním stanovených cieľov v každej iterácii. Jednotlivé iterácie sú manažované v systéme TFS, ktorý slúži na správu zdrojového kódu.

12.7.1 Dedikácia metodiky

Postupy uvedené v metodike sú určené pre všetkých členov tímu pracujúcich metódou vývoja SCRUM. Do rozhodnutí v priebehu plánu zasahuje productowner.

12.7.2 Zoznam nadväzujúcich metodík

- Metodika zberu a spracovania požiadaviek
- Metodika komunikácie
- Metodika tvorby dokumentácie
- Metodika identifikácie a riešenia rizík v tíme

12.7.3 Vymedzenie pojmov

Šprint – časový úsek vývoja; jedna iterácia

User story – požiadavka na systém; komponent alebo funkcionality systému

Backlogitem – požiadavka evidovaná v systéme TFS

Productowner – vlastník produktu; určuje požiadavky a zmeny

TFS - Team Foundation Server – systém na správu projektov a manažment zdrojových kódov

Productbacklog – zoznam user stories v systéme TFS. Obsahuje nové, schválené, aktuálne riešené a vykonané požiadavky

Burndownchart – graf práce na projekte. Zobrazuje množstvo priebeh práce na úlohách

12.7.4 Plánovanie šprintu (SCRUM)

Etapy šprintu

1.	Začiatok šprintu	5.1.1
2.	Priebeh prvého týždňa	5.1.2
3.	Stretnutie v priebehu šprintu	5.1.3
4.	Priebeh druhého týždňa	5.1.4
5.	Ukončenie šprintu	5.1.5

12.7.4.1 Začiatok šprintu

Šprint začína v priebehu spoločného tímového stretnutia. Na tomto stretnutí rovnako prebieha ukončovanie predchádzajúceho šprintu. Po uzatvorení predchádzajúceho šprintu (8.5.4.5 Ukončenie šprintu) sa definuje nový šprint v systéme TFS. Productowner určí user stories, ktoré sa budú v aktuálnom šprinte riešiť. V následnej diskusii sa určia členovia, ktorý budú jednotlivé user stories riešiť. Postupy určenia, ohodnotenia a pridelenia user storie sú popísané v **Metodike zberu a spracovania požiadaviek**. Podľa odhadu na základe predchádzajúcej práce je pridelené určité množstvo user stories každému z členov tímu. Odhad množstva úloh sa prispôsobí už známym

povinnostiam v nasledujúcich dvoch týždňoch. V nasledujúcom priebehu stretnutia sú definované, ohodnotené a pridelené jednotlivé user stories, ktoré productowner určil na spracovanie.

Na začiatku stretnutia je členmi tímu určený jeden, ktorý spisuje zápisnicu. Pridelené user stories a problematika riešená na stretnutí sa zapisuje do tejto zápisnice. Zápisnica sa spisuje na každom tímovom stretnutí. Šablóna zápisnice a jej forma sú uvedené v **Metodike tvorby dokumentácie**.

Definovanie novej iterácie

Manažér plánovania v systéme TFS zaeviduje začiatok novej iterácie. Po prihlásení do systému TFS zvolí *administratívne nastavenia*. Následne v menu *iterations* vykoná dvojklik na aktuálne začínajúcu iteráciu.

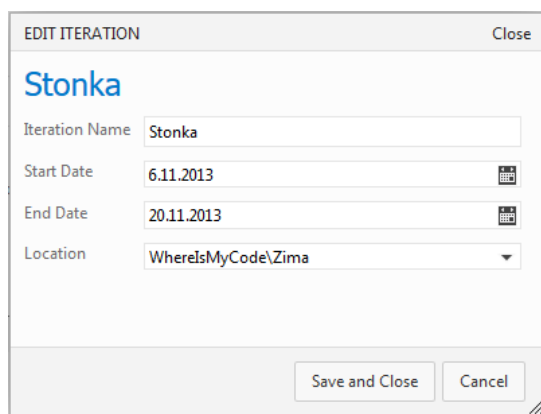
The screenshot shows the 'Iterations' page in TFS. At the top, there is a navigation bar with tabs: 'overview', 'iterations' (selected), 'areas', 'security', 'alerts', and 'services'. Below the navigation bar, the title 'Iterations' is displayed. A sub-tab 'iterations' is also visible. A message states: 'Select the iterations you want to use for iteration planning (sprint planning). Selected iterations will appear in your backlog view as iterations available for planning.' Below this, there are buttons for 'New' and 'New child'. The main content is a table of iterations:

Iterations	Start Date	End Date	
WhereIsMyCode			Backlog iteration for this team
<input type="checkbox"/> Zima	9.10.2013	12.12.2013	
<input checked="" type="checkbox"/> Semienko	9.10.2013	23.10.2013	
<input checked="" type="checkbox"/> Korienok	23.10.2013	6.11.2013	
<input checked="" type="checkbox"/> Stonka	6.11.2013	20.11.2013	
<input checked="" type="checkbox"/> Sprint 4	20.11.2013	4.12.2013	
<input checked="" type="checkbox"/> Sprint 5	4.12.2013	11.12.2013	
<input type="checkbox"/> Leto	17.2.2014	26.5.2014	

Obr. č. 8 Plán iterácií v systéme TFS

12 Metodiky

Po zobrazení okna *Edititeration* určí názov iterácie a dátumy odkedy a dokedy bude aktuálne začínajúci šprint prebiehať.



Obr. č. 9 Okno pre definovanie šprintu

12.7.4.2 Priebeh prvého týždňa

Do 24 hodín od stretnutia na začiatku šprintu je potrebné určiť úlohy, na ktoré sa rozdelia jednotlivé user stories. Člen tímu, ktorému bol user story pridelený určí v systéme TFS, ktoré úlohy je v rámci user story potrebné splniť. Prioritne začnú členovia tímu riešiť úlohy, ktoré súvisia alebo na ne nadväzujú iné úlohy.

Každý člen tímu, ktorí ide riešiť úlohu, ju v systéme TFS určí ako „In progress“.

	TO DO 33 h	IN PROGRESS 10 h	DONE
✓ Vytvorenie vizuálu stránky 2 h		Dizajn home stránky projektu 2 Bc. Patrik Sam...	Vytvorenie menu Bc. Patrik Sam... Úprava dizajnu Bc. Patrik Sam... Prepojenie položiek menu a viewov Bc. Patrik Sam...
✓ Vytvorenie kontextového menu pre položky 6 h	Vytvorenie HTML kódu 2 Bc. Michael Sc... Vytvorenie JS kódu 2 Bc. Michael Sc... Úprava CSS + štýlovanie 2 Bc. Michael Sc...		
✓ Kontrola prístupu k dátam 2 h		Testovanie prav 2 Bc. Matej Chle...	Vytvorenie metódy na kontrolu prav k prístupu k projektu Bc. Matej Chle... Vytvorenie metódy na kontrolu prav na prístup ku channacatu Bc. Matej Chle...
✓ Filtrácia histórie odovzdaní 9 h	Preskúmanie aktuálneho riešenia a kódu 2 Bc. Tomas Kopic Tvorb a nasadenie filtra 4 Bc. Tomas Kopic Testovanie správnosti filtrácie 3 Bc. Tomas Kopic		

Obr. č. 10 Nástenka so zoznamom usertstories, úlohami a ich stavom v systéme TFS

Priebeh riešenia úlohy začína analýzou prístupov ku konkrétnej problematike. Po analýze navrhne riešiteľ spôsob, akým chce úlohu riešiť. V prípade, že sa téma úlohy prekrýva s témou iného člena tímu, musí riešiteľ spôsob riešenia prekonzultovať aj s ostatnými a dohodnúť ako budú postupovať. Všetci členovia analyzujú problematiku ich riešenia do maximálne 3 dní. Do najbližšieho stretnutia jednotliví členovia riešia pridelené úlohy.

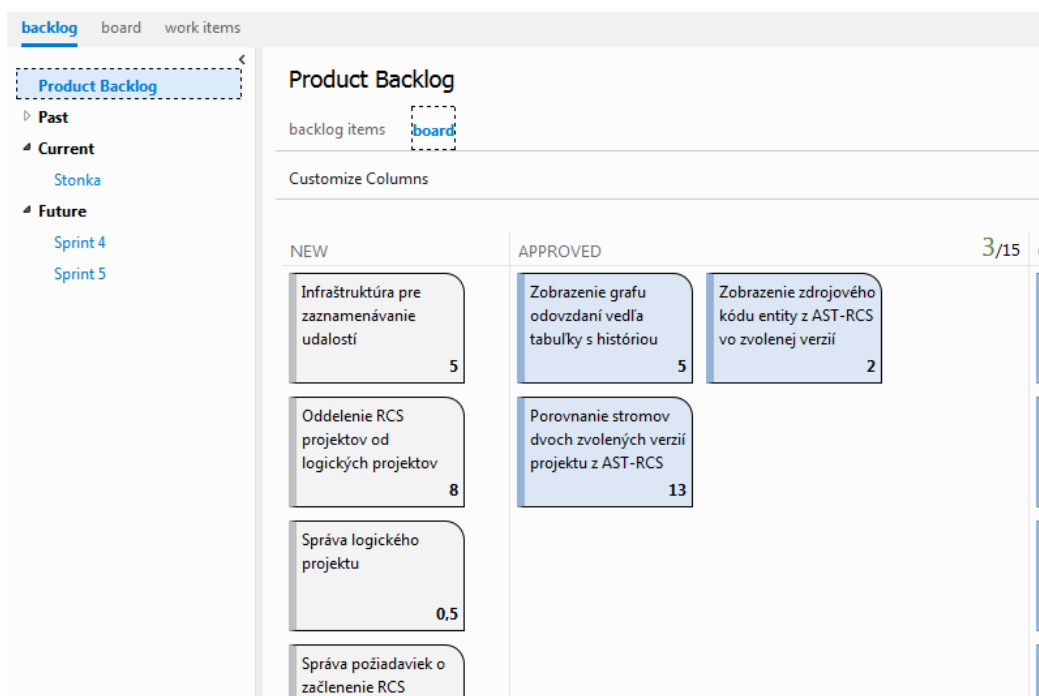
Do konca prvého týždňa musia byť analyzované problematiky, ktoré budú jednotliví členovia tímu riešiť a definované problémy, ktoré by mohli v priebehu šprintu vzniknúť. Ak niektorý člen tímu odhalí problém, snaží sa ho riešiť elektronickou komunikáciou (spôsoby komunikácie sú opísané v **Metodike komunikácie**) s ostatnými členmi tímu, prípadne s productownerom.

12.7.4.3 Stretnutie v priebehu šprintu

Po prvom týždni trvania šprintu prebieha stretnutie tímu. Na začiatku stretnutia sa riešia problémy, ktoré vznikli v prvom týždni šprintu a nepodarilo sa ich vyriešiť v priebehu prvého týždňa. Problémy sa konzultujú medzi členmi tímu a s productownerom. Ak je problémov viac, poradie riešenia určí productowner. Po vyriešení problémov jednotliví členovia konzultujú ich nápady a návrhy na softvér s productownerom. Productowner následne určí, ktoré zmeny a návrhy sa budú v ďalšom priebehu vykonávať.

Ďalším bodom stretnutia je zhodnotenie práce v prvom týždni. Na základe diskusie všetkých zúčastnených a nadväznosti úloh sa určí, ktoré úlohy je potrebné riešiť prioritne.

V prípade, že boli na stretnutí navrhnuté a schválené nové user stories, navrhovateľ ich zaeviduje do systému TFS ako nový backlogitem. Tieto nové user stories sú evidované v systéme TFS ako *new* v produktovom backlogu a odložené na riešenia v ďalších šprintoch.



Obr. č. 11 Nástenka s novými zaevidovanými požiadavkami

Postup pridania nového backlogitemu je popísaný v **Metodike zberu a spracovania požiadaviek**

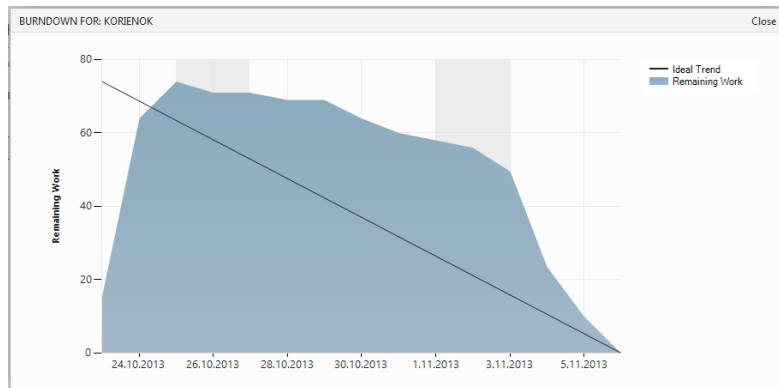
12.7.4.4 Priebeh druhého týždňa

Druhý týždeň šprintu každý člen tímu rieši pridelené úlohy. V prípade vzniku problému sa problém prediskutuje s členmi tímu, prípadne s productownerom, prostredníctvom elektronickej komunikácie. Členovia tímu by si rozvrhnu čas vykonávania jednotlivých úloh tak aby ich finálne dokončenie bolo najneskôr 24 hodín pred stretnutím na konci šprintu. V priebehu posledných 24 hodín sa hľadajú chyby a nedostatky, ktoré boli prehliadnuté. Túto činnosť spravidla vykonáva Manažér kvality a Manažér podpory vývoja.

Každý člen tímu, ktorý dokončí všetky úlohy v jednotlivých user stories, zdokumentuje činnosť v priebehu šprintu. Jednotliví členovia zdokumentujú vykonané požiadavky (user story) podľa postupov v **Metodike tvorby dokumentácie**.

12.7.4.5 Ukončenie šprintu

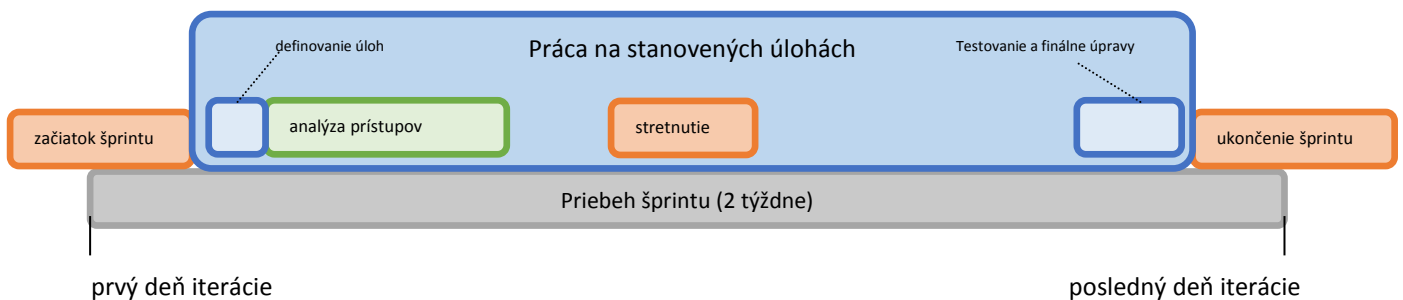
Ukončenie šprintu prebieha opäť na spoločnom stretnutí tímu. Na začiatku stretnutia členovia tímu zhodnotia priebeh šprintu. Prediskutuje sa burndownchart so systému TFS. V prípade, že sa nestihli vykonať všetky úlohy zadané na začiatku šprintu, spoločne sa prediskutuje a upraví množstvo úloh tak aby sa v nasledujúcom šprinte stihli vykonať.



Obr. č. 12 Burndownchart úspešne ukončeného šprintu

Nesplnenie stanoveného množstva úloh je riziko, ktoré môže mať za následok nedodržanie termínov pre včasné dokončenie projektu. Spôsoby riešenia problému sú popísané v **Metodike identifikácie a riešenia rizík v tíme**

12.7.5 Diagram znázorňujúci priebeh iterácie



- Šprint trvá 2 týždne
- Začiatok jedného šprintu a ukončenie predchádzajúceho prebieha na jednom stretnutí

12.8 METODIKA LOGOVANIA

Účelom a cieľom metodiky je vytvoriť jednotný postup logovania informácií, chýb a debugových výpisov v priebehu vývoja softvérového produktu. Tento postup zahŕňa definíciu procesov, z ktorých sa skladá.

12.8.1 Dedikácia

Touto metodikou sa riadia členovia tímu pri tvorbe programu, pri určených udalostiach, ktoré je nevyhnutné zalogovať. Logovanie slúži na investigovanie chýb programu, informácie o behu programu a debug výpisy slúžiace samotným programátorom.

12.8.2 Súvisiace metodiky

Konvencie písania kódu v projekte

12.8.3 Pojmy

„logovanie“: zaznamenávanie potrebných informácií o programe.

12.8.4 Používanie triedy „Logger“

Všeobecné použitie

Trieda „Logger“ ma definované rôzne prekonania svojich metód. Každá metóda ma parameter „msg“, kde programátor pri logovaní zadá správu v slovenskom jazyku ako sa program dostal k aktuálnemu kódu.

Príklad: „Nastala chyba pri pripojení do databázy.“

Použiť statickú triedu „Logger“ je možné pri logovaní troch druhov informácií:

Error - chyba:

- Zachytenie výnimky („Exception“)
 - Ak je zachytená výnimka použije sa metóda `Logger.Error()` s nasledujúcimi parametrami:

```
publicstaticvoidError(objectmsg, Exception ex)
```

- Prípád keď sa program dostane do neočakávaného stavu.
 - Ak nie je zachytená výnimka a program by sa mohol dostať do chyby, je nutné čo najskôr doplniť výnimku, zatiaľ použiť metódu `Logger.Error()` s parametrami:

```
publicstaticvoidError(objectmsg)
```

Info – informácia:

- Zmena stavu programu
 - V správe je nutné opísať stav programu v akom sa od volania loggeru nachádza.
- Začiatok/Koniec istého procesu
 - V správe je nutné opísať aký proces práve v programe začak.

```
publicstaticvoidInfo(objectmsg)
```

Debug – pomocné výpisy :

- Proces, ktorý je nutné debugovať – závisí od programátora.
 - Obsah správy je na programátorovi, avšak je nutné udržiavať poriadok v debug výpisoch.
 - Správy, ktoré slúžia výhradne na krátkodobé účely je nutné po ukončení debugu zmazať.

```
publicstaticvoidDebug(objectmsg)
```

13 ROZVRH LETNÉHO SEMESTRA

13.1 HRUBÝ PLÁN SEMESTRA

13.1.1 Prvý šprint

Oprava objavených chýb z predchádzajúceho semestra. Príprava na aktívne dopĺňanie značiek do kódu. Doplnenie kompatibility s najpoužívanejšími prehliadačmi.

13.1.2 Druhý šprint

Aktívne pridávanie značiek do kódu. Evidencia pridávaných značiek. Možnosť odkazovania sa na značky (link)

13.1.3 Tretí šprint

Notifikácie o novo pridaných značkách. Doplnenie špeciálnych vizuálnych prvkov značiek.

13.1.4 Štvrtý šprint

Úprava pluginu pre VisualStudio. Doplnenie ďalších štatistík ku projektom.

13.1.5 Piaty šprint

Zobrazenie používateľskej aktivity. Testovanie použiteľnosti systému.

13.1.6 Šiesty šprint

Dokončenie finálnej verzie systému. Finalizácia dokumentácie. Doladenie chýb. Predstavenie produktu.

13.2 DETAILNÝ OPIS PLÁNU NA PRVÉ TÝŽDNE LETNÉHO SEMESTRA

Nasledujúce komponenty a vlastnosti zakomponujeme do systému v prvých týždňoch letného semestra:

- Aktívne pridávanie značiek
 - evidencia značiek
 - vizuálna úprava značiek podľa typu
- Pridávanie komentárov ku kódu
- Notifikácie o pridávaných značkách
 - emailová notifikácia
 - zobrazenie zoznamu naposledy pridaných značiek
- Vytvorenie odkazov pre navigáciu ku značkám
- Vyhľadávanie na stránke

Budeme sa snažiť o čo najväčšie pokrytie požiadaviek na systém. Chceme čo najskôr dosiahnuť funkčnosť systému pre použitie ostatnými tímami. Postup práce ale závisí od množstva požiadaviek, ktoré budeme schopný v danom čase splniť.

14 PRÍLOHY

V tejto kapitole sa nachádzajú zápisy zo stretnutiach, ako aj preberací protokol. Zoznam príloh:

- **Príloha A** – Zápisy zo stretnutiach
- **Príloha B** – Preberací protokol

PRÍLOHA A



Zápis zo stretnutia tímového projektu

Dátum: 2.10.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

1. Obsah stretnutia

- Výber rolí v tíme: Grešílková – Vedúci tímu
Chlebana – Podpora vývoja
Kepič – Manažment rizík
Samuhel – Manažment plánovania
Scholtz – Manažment kvality
Skrisa – Manažment monitorovania projektu
Oriskó – Manažment dokumentácie
- TFS technológia
- PerConIK – monitorovanie používateľa
- Zdrojáky od Gratexu (len pre inšpiráciu)
- Našou úlohou je vytvoriť webový portál – prihlásenie s autentifikáciou AIS a štandardnou autentifikáciou
- Dostaneme server:
 - IIS8
 - Remote desktop - VNC
 - Konfigurácia používateľov 7+1
 - Databázový systém
- Obsah dokumentácie:
 - jednotlivé metodiky
 - grafy postupu práce
 - technická dokumentácia
 - Model
- Itemtracking systém (Gira, Redmain, TFS):
 - definovanie taskov
 - čas strávený na tasku a stav tasku
 - označenie chýb

Príloha A

- Web tímu:
 - len HTML, CSS a obrázky
 - členovia tímu
 - Žiadny JS, PHP
- Teambuilding

2. Úlohy

- Zvoliť metodiku dokumentácie
- Dohodnutie architektúry
- Do prednášky TP urobiť plagát
- Rozbehať TFS
- Zistiť koľko pamäte je treba na servery
- Spoločný komunikačný kanál – FB, DB, Google Drive Dokumenty
- Prejsť dokumentácie:
 - AST-RCS
 - Inštalačný manuál k VS
- Rozhodnúť či ideme do TP Cup-u

3. Témy ďalšieho stretnutia

- Prvý backlog
- Brainstorming – čo vlastne ideme robiť?
- Každý oboznámi s tým čo analyzoval

Zapísal: Samuhel

Overil: Chlebana

Trvanie stretnutia: 2:00



Zápis zo stretnutia tímového projektu

Dátum: 9.10.2013
Miesto: Jobsovo softvérové štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

Obsah stretnutia:

4. Zhrnutie od predchádzajúceho stretnutia

- prešli sme dokumentácie

5. Definovanie user stories

- autorizácia z AIS – openeldap – heslo môžeme brať aj z AIS aj uložiť u nás a pracovať u nás
- priradenie ku projektu
 - tabuľka používateľ projekt v DB -> môže mať podstories v iteráciach
 - tabuľka projektov, AST RCS. Vyťahovanie projektov, formulár pre administrátora priradenie ľudí ku projektu, kto kde patrí. Okrem programátorov môže byť priradený aj manažér
- zobrazenie projektov
 - prihlásim sa a zobrazí sa mi zoznam projektov
- prehliadanie súborovej štruktúry projektu
 - používateľ sa môže pozrieť na súbory organizované v adresároch
 - zobrazí strom kde bude vidno súbory
 - algoritmus na spracovanie stromu, treba ho spracovať
- prehliadanie AST
 - aby vedel user nájsť triedu a podobne, na logickej vrstve nie fyzickej ako zobrazenie súborov
 - uzol projektu, namespaces, triedy, netreba nič špeciálne parsovať
- určenie roly a administrátora
 - bežný používateľ a administrátor, v DB tabuľka s rolami a priradenie používateľa k roly
- zobrazenie kódu
 - vidieť čisto kód, nemusí byť ešte highlightované, len text
- zobrazenie histórie entity
 - súbor, trieda, metóda
 - zoznam commitov kedy bolo čo zmenené
 - dátum, meno - kto ho urobil, commit
 - informácie o entite, kedy ako čo, jednoducho všetko, kde je a tak
- zobrazenie kódu pre entitu v danej verzii
- zobrazenie syntaxe kódu C#, JAVA
- porovnanie verzií kódu
- zobrazenie changesetov projektu

- zoznam všetkých commitov, kedy zmena kto ju urobil a aký dal k tomu komentár
- jednoduchšie ako zobrazenie histórie
- porovnanie AST stromov medzi verziami projektu (changesetmi)
 - vypísať všetky rozdiely medzi dvoma verziami projektu, dvoma stromami, rozdiely v triedach, metódach a podobne. Napríklad zmenami farby v strome štruktúry
 - výsledok je ofarbený strom
- zmena hesla používateľa
- registrácia emailu

6. SCRUM

- prehľad kartičiek s číslami
- diskusia o scrume, scrummaster
- čísla sa dávajú len v rámci stretnutia, na inom stretnutí sa môžu meniť
- rozdelenie úloh v rámci šprintu – trochu ohnuté na potreby tímového projektu, vstupujú ďalšie roly – ako plánovač a vedúci čo nie je scrum
- určenie prvého šprintu – Semienko
- vytvorenie taskov
- odhad okolo 6-8 na osobu
- odhad náročnosti úloh pomocou kartičiek

7. Diskusia o user stories , časť z Backlog

- AIS login – 3
- Priradenie ku projektu – 5
- Zobrazenie projektov – 2
- Prehliadanie súborovej štruktúry projektu – 8
- Prehliadka AST – 8
- Určenie roly a administrátora – 1
- Zobrazenie kódu -2
- Zobrazenie histórie entity - 13
- Zobrazenie kódu pre entitu v danej verzii - 2
- Zobrazenie syntaxe kódu C#, JAVA – 5
- Porovnanie verzií kódu - 8
- Zobrazenie changesetov projektu – 8
- Porovnanie AST stromov medzi verziami projektu (changesetmi) – 13
- Zmena hesla používateľa – 1
- Registrácia emailu – 1

8. Iné

- komunikačný kanál – na dohadovanie taskov použiť issuetracker
- veci ťaháme z AST RCS a ukladáme do našej databázy, projekty nevkladáme, tie tam už budú

Úlohy:

- rozdelenie jednotlivých úloh:
 - AIS login – 3 - Matej
 - Zmena hesla - 1 - Matej
 - Email – 1 - Matej
 - Admin – 1 –Matej
 - Priradenie – 5 - Zuzka
 - Zobrazenie projektov – 2 - Zuzka
 - Načítanie súborov – 5 - Norti
 - Načítanie AST – 5 - Julio
 - Zobrazenie kódu – 2 - Paťo
 - Syntax – 5- Paťo
 - Zobrazenie zmien v projekte (changeset) – 8 – Kepo – urobiť iba tabuľku čo kto
 - Prehliadanie stromu – 5 – Mišo

Témy ďalšieho stretnutia:

- Diskusia k vykonanej práci na zadaných úlohách
- Riešenie problémov pri práci na taskoch

Zapísal: Kepič

Overil: Samuhel

Trvanie stretnutia: 3 hodiny



Zápis zo stretnutia tímového projektu

Dátum: 16.10.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

Obsah stretnutia

1. Zhrnutie od predchádzajúceho stretnutia

- bol spustený šprint č.1 - Semienko
- začali sme pracovať na definovaných user stories
- zvolené user stories boli rozdelené na task-y
- nasadili sme nástroj PerConIK

2. Diskusia k vykonanej práci na úlohách

- diskusia o problémoch ktoré boli počas práce na úlohách identifikované, každý individuálne

3. Definovanie a ohodnotenie user stories pre ďalšie šprinty

- Zobrazenie zoznamu zmien vykonaných vo zvolenom odovzdaní – 13
- Zobrazenie zdrojového kódu zmenenej entity v odovzdaní so zvýraznenými zmenami – 8
- Načítanie zoznamu značiek priradených prehliadanej entite – 3
- Zvýraznenie riadkov zdrojového kódu, ktoré sú označované značkami v zozname značiek – 5
- Filtrovanie zoznamu značiek podľa ich typu – 5
- Presunutie na miesto v kóde, ktorému je priradená vybraná značka - 5
- Fulltextové vyhľadávanie nad entitami a ich zdrojovými kódmi – 100

Úlohy

- ukončiť šprint č.1 - Semienko
- vybaviť povolenie portu pre remote desktop

Témy ďalšieho stretnutia

- vyhodnotenie ukončeného šprintu č. 1 - Semienko
- rozdelenie úloh pre šprint č. 2-2Z

Zapísal: Oriskó

Overil: Kepič

Trvanie stretnutia: 2 hodiny



Zápis zo stretnutia tímového projektu

Dátum: 23.10.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení: -

1. Zhrnutie a overenie splnenia úloh v sprinte

- Prihlásenie používateľa AIS loginom – Done
- Pridanie informácií o tíme na stránku projektu – Done
- Určenie administrátora systému – Done
- Priradenie používateľa k projektu – In progress
- Zobrazenie zoznamu projektov používateľa – Done
- Načítanie súborovej štruktúry projektu do stromovej reprezentácie – Done
- Načítanie AST štruktúry projektu do stromovej reprezentácie – Done
- Prehliadanie stromovej reprezentácie – Done
- Zobrazenie zdrojového kódu zvolenej entity z AST-RCS – Done
- Zmena hesla používateľa - Done
- Registrácia emailu používateľa – Done
- Zvýraznenie syntaxe zobrazeného kódu – Done
- Zobrazenie zoznamu odovzdaní (changesetov) projektu – Done

2. Diskusia o novej verzii AST-RCS

- nová verzia a dokument k nej
- nové položky v „backlog“ na update s novou verziou AST-RCS.

3. Prehliadka dokončených „user stories“

- diskusia k jednotlivým častiam
- odporúčenie efektívnejšieho prístupu vedúcim projektu k istým častiam

4. Pridávanie nových „user stories“

- Dynamické expandovanie stromovej štruktúry – 3
- Zobrazenie stromovej štruktúry vedľa zdrojového kódu – 8
- Zobrazenie grafu odovzdaní projektu – 13
- Zobrazenie grafu entít projektu - 5

5. Vytvorenie nového sprintu a pridelenie úloh

- Dynamické expandovanie stromovej štruktúry – Michael Scholtz
- Zobrazenie stromovej štruktúry vedľa zdrojového kódu – Michael Scholtz
- Zobrazenie grafu odovzdaní projektu – Július Skrisa
- Zobrazenie zoznamu zmien vykonaných vo zvolenom odovzdaní – Patrik Samuhel
- Zobrazenie zdrojového kódu zmenenej entity v odovzdaní so zvýraznenými zmenami – Matej Chlebana
- Načítanie histórie súborovej entity z AST-RCS – Zuzana Grešlíková

Príloha A

- Načítanie histórie entity zdrojového kódu z AST-RCS – Tomáš Kepič
- Zobrazenie histórie entity vo forme tabuľky – Tomáš Kepič
- Porovnanie zdrojového kódu dvoch zvolených verzií entity z AST-RCS – Patrik Oriskó

Úlohy:

- prispôbiť projekt s novou verziou AST-RCS
- riešenie nového šprintu „Korienok“

Témy ďalšieho stretnutia:

- Zhodnotiť priebeh a progres v šprinte
- Riešiť vyskytnuté problémy so zadanými úlohami

Zapísal: Skrisa

Overil: Oriskó

Trvanie stretnutia: 2:00



Zápis zo stretnutia tímového projektu

Dátum: 30.10.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešliková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení: -

Preberané témy:

1. Riešenie problémov v 2. šprinte

- Ujasnenie typov entít
- Zodpovedanie otázok ohľadom upraveného AST-RCS
- Prejdenie položiek 2. šprintu a ich objasnenie
- Vyriešenie problémov s check-inom
- Fixnutý problém s web.config („receivedmessagequotaexceeded“)
- Riešenie nasadzovania webu; testovanie loginu

2. Diskusia ku testom

- Otestovať, či controller správne plní model
- Pozrieť si dostupné materiály k testovaniu
- Určiť si metodiku testovania

3. Rozdelenie metodík v tíme

- Grešliková – Metodika komunikácie v tíme
- Skriňa – Metodika testovania
- Scholtz – Metodika kvality kódu, písania kódu a jeho konvencii
- Kepič – Metodika identifikácie rizík
- Oriskó – Metodika písania dokumentácie
- Samuhel – Metodika plánovania
- Chlebana – Metodika commitovania

Úlohy:

- Ukončenie druhého šprintu „Korienok“

Témy ďalšieho stretnutia:

- Zhodnotiť priebeh a progres v 2. Šprinte
- Zčať tretí šprint; určiť si backlogitemy

Zapísal: Scholtz

Overil: Skrisa

Trvanie stretnutia: 2:00



Zápis zo strenutia tímového projektu

Dátum: 6.11.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení: -

1. Zhrnutie a overenie splnenia úloh v druhom šprinte

- Zobrazenie zoznamu projektov používateľa z novej verzie AST-RCS – Done
- Rozbaľovaná stromová štruktúra – Done
- Priradenie používateľa k projektu v novej verzii AST-RCS– Done
- Zobrazenie stromovej štruktúry vedľa zobrazenia zdrojového kódu– Done
- Zobrazenie grafu odovzdaní projektu – Done
- Zobrazenie zoznamu zmien vykonaných vo zvolenom odovzdaní – Done
- Zobrazenie zdrojového kódu zmenenej entity v odovzdaní so zvýraznenými zmenami – Done
- Načítanie histórie súborovej entity z AST-RCS – Done
- Načítanie histórie entity zdrojového kódu z AST-RCS – Done
- Zobrazenie histórie entity vo forme tabuľky - Done
- Porovnanie zdrojového kódu dvoch zvolených verzií entity z AST-RSC – Done
- Načítanie súborovej štruktúry projektu do stromovej reprezentácie pre novú verziu AST-RCS – Done
- Načítanie AST štruktúry projektu do stromovej reprezentácie pre novú verziu AST-RCS – Done
- Zobrazenie zdrojového kódu zvolenej entity z novej verzie AST-RCS – Done
- Zobrazenie zoznamu odovzdaní projektu v novej verzii AST-RCS – Done
- Úspešné prihlásenie s neplatným heslom – Done

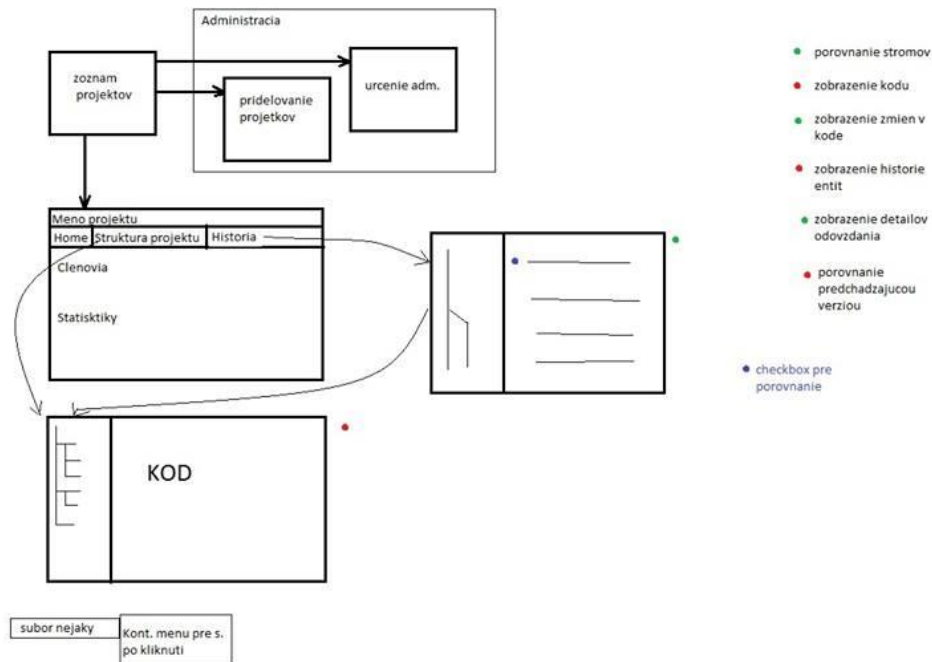
2. Prehliadka dokončených „user stories“

- diskusia k jednotlivým častiam druhého šprintu. Diskusia o testovaní. Porada s vedúcim projektu
- odporúčenie efektívnejšieho prístupu vedúcim projektu k istým častiam
- Upozornenie na nedostatky vedúcim projektu:
 - Bude potrebné upraviť program pre rolu, keď je používateľ zároveň aj admin. Aby sa vedel dostať aj do administrátorovho menu.
 - Pridanie multiselect list pre viac projektov a viac používateľov.
 - Security - Treba vyriešiť bezpečnosť pri logovaní a tiež keď si pozriem cudzí projekt nemalo by to ísť ani pre entity ani pre changesety.
 - Stránkovanie - Doplniť tam, kde ešte nie je. Napríklad v rámci Changesetov
 - Rozbaľovací strom by mal byť na začiatku zbalený. Pridanie + - ikon
 - V časti kde sú zobrazené rozdiely v dvoch stromoch bude potrebné iné rozlíšenie editovaných súborov.
 - Číslo riadkov - Aby boli zobrazené všetky čísla v jednom stĺpci vo FileViewer a pri zobrazení kódu

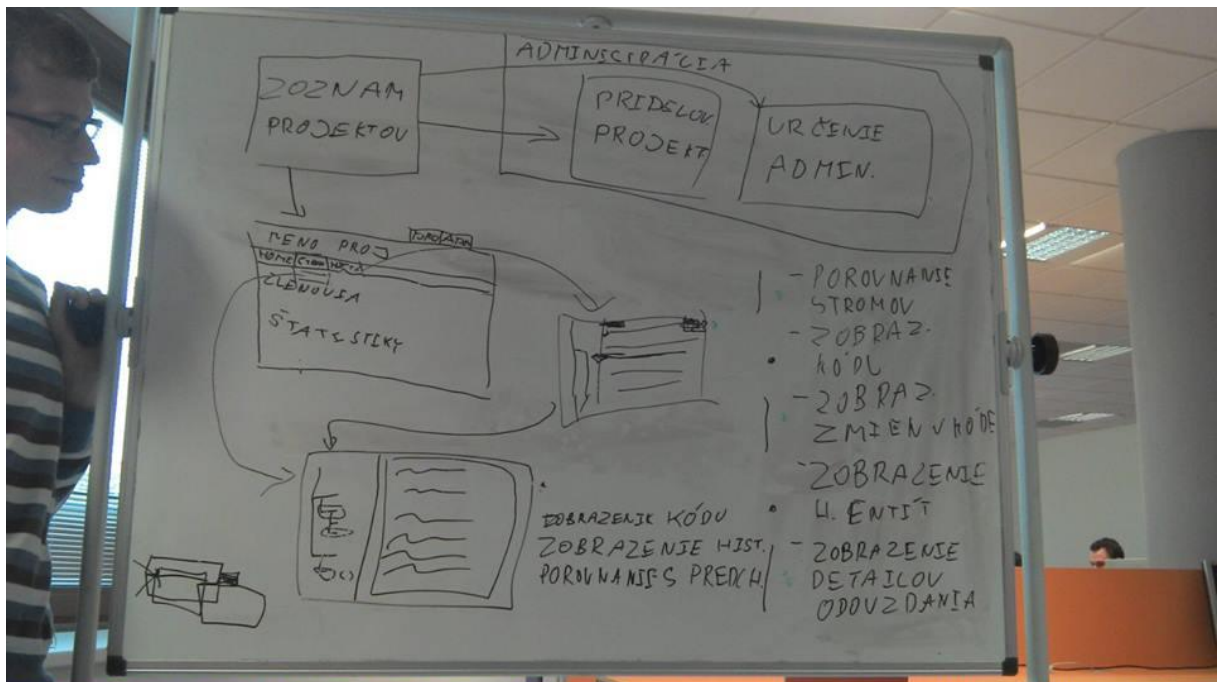
- Vizual stránky – bude potrebné dohodnúť sa na štruktúre stránky
- Postupne je potrebné vytvárať technickú dokumentáciu k jednotlivým „user stories“. Robiť komentáre v kóde. Podpisovať sa pod kód a testovať

3. Vizual stránky

- Bol vytvorený návrh vizualu pre stránku. Ktorý ja načrtnutý na obrázku 1 a na obrázku 2.



Obrázok 2



Obrázok 3

4. Pridávanie nových „user stories“

- Security 5
- Stránkovanie changesetov 0,5
- Strom +- 1
- Strom [] 1
- Vizuál stránky 5
- Menu 5
- ActionLinks 8
- Multiselect projektov a používateľov 5
- Určenie aministrátorov 5

5. Vytvorenie nového sprintu a pridelenie úloh

- Vytvorenie vizuálu stránky – Patrik Samuhel
- Vytvorenie kontextového menu pre položky – Michael Scholtz
- Kontrola prístupu k dátam– Matej Chlebana
- Filtrácia histórie odovzdaní – Tomáš Kepič
- Stránkovanie histórie odovzdaní – Július Skrisa
- Priradenie projektov k používateľovi– Zuzana Grešlíková
- Priradenie používateľov k projektu – Zuzana Grešlíková
- Určenie administrátorov systému - Zuzana Grešlíková
- Definovanie aktívnych odkazov v kontextovom menu - Patrik Oriskó
- Úplné zobrazenie číslovania riadkov – Michael Scholtz
- Vizualizácia zmeny v strome – Matej Chlebana
- Vizualizácia rozbalenia uzla stromu – Michael Scholtz

Úlohy:

- riešenie nového šprintu „Stonka“

Témy ďalšieho stretnutia:

- Zhodnotiť priebeh a progres v treťom šprinte
- Riešiť problémy, ktoré sa vyskytli v treťom šprinte

Zapísal: Grešlíková

Overil: Scholtz

Trvanie stretnutia: 3:00



Zápis zo stretnutia tímového projektu

Dátum: 13.11.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

1. Zhrnutie od predchádzajúceho stretnutia

- Diskusia o tom, komu sa čo podarilo stihnúť a aké problémy boli
- Riešili sme testovanie – problém s vytváraním http session

2. Metodiky

- Tagovanie kódu
- Diskusia o metodikách - kto akú metodiku robí
- Rozoberanie jednotlivých metodik
 - písanie kódu
 - zjednotenie css a cshtml
 - zjednotenie písania komentárov
 - codereview
 - rizika

3. Kto bude komu robiťcodereview

Scholtz -> Chlebana -> Samuhel -> Grešílková -> Kepič -> Oriskó -> Skrisa -> Scholtz

4. Čo je potrebné robiť / urobiť

- Kontrolovanie parametrov controllera
- Zjednotiť sa pri písaní komentárov
- Naučiť sa používať trycatch
- Vytvoriť naše exceptions
- Chýba plán na stránke – Patrik Samuhel
- Tagovanie kódu
- Rozmiestňovanie súborov – treba rozčleniť
- Zle umiestnene security
- Upratovanie v kódoch classach
- Emaily používateľov podľa AIS loginu

5. Dokumentácia

- Ako ma vyzerať dokumentácia – štruktúra
 - číslovanie strán – „číslovanie kapitoly – číslovanie strany“
 - náhľad na víziu produktu
 - plán
 - rozdelenie roli
 - každý šprint ma mať vlastnú kapitolu

- monitorovanie
 - Odôvodňovanie grafov – (podľa grafu – dôvod prečo (zápočtovky, projekty atd..))

6. Problémy

- Komunikácia – facebook nie je dobrá voľba, treba zvoliť inú
- Hľadanie iných alternatív
 - trello - nie to čo potrebujeme
 - email – googlegroup - Zuzka vytvorila
- riešenie značiek

7. Hodnotenie backlog

- Logovanie do txt (log for net) - 5
 - požiadavky od používateľov
 - úspešnosť a neúspešnosť prihlásenia
 - výnimky
- Logické projekty (solution) - 8
 - Roly pre projekt (projectAdmin, projectUser, projectEditor)
 - modifikácia databázy - migrácia
 - úprava logiky prístupu k dátam (zmeny v atribútoch)
- Sprava požiadaviek vytvorenia logických projektov - 8
 - databáza – požiadavka na pridelenie projektu
 - emailová notifikácia
 - používateľ (project Admin) môže pridávať projekty – notifikácia adminovi
 - vybavenie požiadavky
 - Zaradenie projektu do repozitára
 - notifikácia o priradení do repozitára
- Rozhranie pre správu používateľov logického projektu - 5
 - projektový admin prideluje projektové práva userom ku projektom
 - začlenenie do menu
- Pridelenie správu do menu – 0.5
- Sprava aliasov – priradiť do menu pre úpravu info u userovi - 5
 - pridanie tabuľky pre aliasy
 - Meno aliasu, serverId
 - pridanie do userproperty – alias user activity

Zapísal: Matej Chlebana

Overil: Zuzana Grešlíková

Trvanie stretnutia: 3:0



Zápis zo stretnutia tímového projektu

Dátum: 13.11.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič,

Ospravedlnení:

1. Zhrnutie od predchádzajúceho stretnutia

- Diskusia o tom, komu sa čo podarilo stihnúť a aké problémy boli
- Riešili sme testovanie – problém s vytváraním http session

2. Metodiky

- Tagovanie kódu
- Diskusia o metodikách - kto akú metodiku robí
- Rozoberanie jednotlivých metodik
 - písanie kódu
 - zjednotenie css a cshtml
 - zjednotenie písania komentárov
 - codereview
 - rizika

3. Kto bude komu robiťcodereview

Scholtz -> Chlebana -> Samuhel -> Grešílková -> Kepič -> Oriskó -> Skrisa -> Scholtz

4. Čo je potrebné robiť / urobiť

- Kontrolovanie parametrov controllera
- Zjednotiť sa pri písaní komentárov
- Naučiť sa používať trycatch
- Vytvoriť naše exceptions
- Chýba plán na stránke – Patrik Samuhel
- Tagovanie kódu
- Rozmiestňovanie súborov – treba rozčleniť
- Zle umiestnene security
- Upratovanie v kódoch classach
- Emaily používateľov podľa AIS loginu

5. Dokumentácia

- Ako ma vyzerá dokumentácia – štruktúra
 - číslovanie strán – „číslovanie kapitoly – číslovanie strany“
 - náhľad na víziu produktu
 - plán
 - rozdelenie roli
 - každý šprint ma mat vlastnú kapitolu
 - monitorovanie

- Odôvodňovanie grafov – (podľa grafu – dôvod prečo (zápočtovky, projekty atd..))

6. Problémy

- Komunikácia – facebook nie je dobrá voľba, treba zvoliť inú
- Hľadanie iných alternatív
 - trello - nie to čo potrebujeme
 - email – googlegroup - Zuzka vytvorila
- riešenie značiek

7. Hodnotenie backlog

- Logovanie do txt (log for net) - 5
 - požiadavky od používateľov
 - úspešnosť a neúspešnosť prihlásenia
 - výnimky
- Logické projekty (solution) - 8
 - Roly pre projekt (projectAdmin, projectUser, projectEditor)
 - modifikácia databázy - migrácia
 - úprava logiky prístupu k dátam (zmeny v atribútoch)
- Sprava požiadaviek vytvorenia logických projektov - 8
 - databáza – požiadavka na pridelenie projektu
 - emailová notifikácia
 - používateľ (project Admin) môže pridávať projekty – notifikácia adminovi
 - vybavenie požiadavky
 - Zaradenie projektu do repozitára
 - notifikácia o priradení do repozitára
- Rozhranie pre správu používateľov logického projektu - 5
 - projektový admin prideluje projektové práva userom ku projektom
 - začlenenie do menu
- Pridelenie správu do menu – 0.5
- Sprava aliasov – priradiť do menu pre úpravu info u userovi - 5
 - pridanie tabuľky pre aliasy
 - Meno aliasu, serverId
 - pridanie do useraproperty – alias user activity

Zapísal: Matej Chlebana

Overil: Zuzana Grešlíková

Trvanie stretnutia: 3:00



Zápis zo stretnutia tímového projektu

Dátum: 20.11.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešílková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

1. Nedokončené požiadavky na konci šprintu

- Vytvorenie kontextového menu pre položky - Perzistencia pre JS menu
 - neboli jasné požiadavky – perzistencia nie je potrebná
- Filtrácia histórie odovzdaní
 - zlý časový manažment riešiteľa - dokončenie v ďalšom šprinte
 - zlé pochopenie rozdeľovania vetiev projektu

2. Nájdene chyby

- Kontextové menu – nezobrazuje obrázok – zlá cesta k súboru
- Test – FileViewer – opraviť náhodný výber projektu
- CodeChanges – ak nemá predchodcu spadne
- Zobrazenie kódu – dokumentačné komentáre sa nezobrazujú

3. Budúca práca

- Úprava stromu changesetov – zobrazenie viac vetiev

4. Nové požiadavky

- Zobrazenie zdrojového kódu skrýva dokumentačné tagy – Orisko – 0,5
- Chýbajúci obrázok kontextového menu v strome – Scholtz – 0,1
- Pri zobrazení porovnania súboru bez histórie nastane chyba – Samuhel – 0,25
- Písmo v zobrazení changesetov sa prekrýva – Skrisa – 0,5
- Obnovenie sedenia pre údaje projektu – Samuhel – 1
- Súbežné scollovanie v porovnaní zdrojových kódov – Samuhel – 1
- Zvýraznenie entity v zdrojovom kóde – Orisko – 2
- Zbalenie a rozbalenie celého stromu – Scholtz – 0,5
- Infraštruktúra pre zaznamenávanie udalostí - Skrisa - 5
- Oddelenie RCS projektov od logických projektov - Chlebana - 8
- Správa logického projektu – Kepič - 0,5
- Správa požiadaviek o začlenenie RCS projektu k logickému projektu - Scholtz - 8
- Správa používateľov logického projektu - Grešílková - 5
- Správa aliasov používateľa - Orisko - 3
- Základné štatistiky nad projektom – Samuhel – 8
- Načítanie zoznamu značiek priradených prehliadanej súborovej entite – Kepič – 5

5. Poznámky ku ďalšiemu šprintu

ITM Server.CreateFIITCon(); - vytvorí pripojenie

Načítanie značiek – getInformationTags()

 Parametre – URI - identifikátor

 aj zmažane značky = false

 Constraints

General InformationTag – všetky značky

UserInfoInformationTag – značky, ktoré vytvárame keď robím CodeReview

Zapísal: Samuhel

Overil: Chlebana

Trvanie stretnutia: 3 hodiny



Zápis zo stretnutia tímového projektu

Dátum: 27.11.2013
Miesto: Jobs. štúdio
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

1. Odovzdaná dokumentácia - riadenia

- Problémy : číslovanie obrázkov, strana 391, strana 1 viackrát, odvolania na obrázky v texte chýbajú, odkazy na kapitoly sú nesprávne
- Použitie crossreference vo worde na číslovanie tabuliek, odkazov na kapitoly
- Metodiky – opakujúce sa informácie – slovníky pojmov, treba jeden jednoznačný
- Obrázky napr. podľa čísel kapitol, aby sa dalo dopĺňať
- Tabuľky nemajú opisy, dávajú sa väčšinou nad tabuľky
- Tabuľka zmien ak by sme robili zmeny, čo zrejme nebudeme robiť lebo len dostávame spätnú väzbu do budúcnosti
- Popis obrázku na ďalšej strane
- Poznámky k metodikám – Zuzkina – spracovanie požiadaviek menšie nezrovnalosti
- Metodiky – množstvo vymyslených metodík
- Gramatické chyby : ii, ií
- Metodika rizík: supervisor – pojem a rola, je to dvakrát, spúšťač je nejaká udalosť(napríklad pokazil sa počítač je dôsledok), máme 4 servery, každý má iné riešenie(TFS, AST-RCS, stránka, build server), tabuľka rizík (napríklad aké, pravdepodobnosť, váha), nevyriešená úloha ako dôsledok
- **Tabuľka kto čo robil !!!!!!!!!!!!!**
- Spustiť testy – odkaz na metodiku testovania, cvičiaci MSI zavrhol, preto to tam nebolo
- V testoch chýba bodkočiarka
- Zápisnice – prezývky alebo plné mená, je to pomiešané
- Metodika monitorovania – reports vo VisualStudio – burndownchart a podobne

2. Odovzdaná dokumentácia k inžinierskemu dielu

- Iné číslovanie strán ako v predchádzajúcej
- Opisy tabuľky v DB formou tabuľky, definovať typy polí (ID ? string, int)
- Popis story – analýza, návrh atď. by malo byť povinné, teraz to je voliteľné
- Spôsob použitia migrácií a podobne, odkaz na metodiku a podobne
- Dosť vatovo, obširne, technická dokumentácia by mala byť stručnejšia
- Opisy testovania – napríklad scenár – kde na stránke sme si dali čo zobrazíť a podobne, čo je automatický jednotkový test, ktorý sa spúšťa s nasadzovaním kódu – nič to nepovedalo, čo splnil, ktorý – riešenie – táto trieda kontroluje tento controller, testujú sa tieto hodnoty, očakáva sa toto a toto
- Testovanie vieme nájsť jednoducho v dokumente
- Chýba podiel prác členov – tabuľka
- Keď je oprava bugu – popísať aj v novom sprinte

3. Coderieview

- Kontrolovať aj navzájom hocikomu, nemusí byť len určenému členovi
- 4. Technická dokumentácia – ďalšie rady**
- Backlogitem – usecase, scenár, dáta, navigácia na stránke, niečo už používateľ splnil – aké má práva a podobne, informácie o entite atď. – patrí do analýzy
 - Usecase patrí do špecifikácie/analýzy, čiže u nás v analýze
 - Scenár čo sa bude robiť, ako, vstupná a výstupná podmienka - opísať hlavný scenár
 - Testovacie casy (prípady) podľa usecase
 - Výstup analýzy – prečo niečo, prečo je toto rozhodnutie, konkrétny návrh
 - MVC - Diagram tried pre model – závislosti medzi dátami, typy
 - Controller: Lepšie sekvenčný diagram, aktivít, lepšie ako nejaký dlhší popis a kód
 - Napríklad generovanie diagramov vo Visualku, ušetrí čas písania, popis: Práca (Logika spracovania dát) je zachytená v diagrame:, obrázok
 - Dohodnúť sa na jednotnom nástroji pre diagramy – napríklad VisualStudio
 - View – dávať obrázky, z textu náročné pochopiť
 - V tomto View zobrazujeme tento a tento model, vieme čo sa v ňom zobrazuje, nielen nejaké príkladné meno
 - Dokumentácia musí byť priamočiara, vizuálne čo najrýchlejšia
 - V riešení/implementácia neopakovať analýzu, kde to je v kóde, v akej triede, kde to rieši, kde je to realizované, prípadne čísla changesetov
 - Testovanie – stačí skonštatovať funkčnosť scenáru, napríklad tento projekt, tento súbor, nerobíme akceptačné testy typu okej okejokej zle.
 - Jednotkový test – táto metóda testuje tento (určitý) prípad, aká je konfigurácia, vidíme čo sme netestovali
 - Postupujeme scrumom nie test drivendevelopment
 - 15 % metód máme aktuálne testovaných
 - Prvky rozhrania – čo dostanem, aké testy, ak je to implementované skúsím testy – test drivendevelopment
 - Očakávame že testy zbehli, ak nezbehol na konci šprintu, treba napísať poznámku, že sa nepodarilo splniť
 - Dokumentácia pre nás, má sa podobať na technickú dokumentáciu
- 5. Používateľská príručka**
- Na konci nejaká ma byť
 - Pre určitý usecase, čo sa dialo, to by sa dalo nakopírovať
- 6. Cieľ projektu**
- Urobiť program na Codereview pre študentov – nevyformované tímy s nie hlbokými vedomosťami
 - Posudzovanie a vyjadrovanie sa ku kódom
 - Práca so značkami
- 7. Scrum panel**
- Názov projektu, kto čo robí – jeden, dva slajdy – Zuzka to spraví
- 8. Otázky k urobeným veciam, problémy**
- Logický model, kto bude robiť
 - Kód ktorý je navyše – upravovať posledný týždeň
 - Napríklad ofarbovanie stromu nie je nalinkované
 - O týždeň kontrola čo je urobené, kde to je a podobne, prehliadka produktu
 - Changeset ID z TFSka
 - AST je aktualizované postupne, napríklad changesety po minúte
- 9. Obrázková príloha**



Zápis zo stretnutia tímového projektu

Dátum: 04.12.2013
Miesto: miestnosť 3.21
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Ospravedlnení:

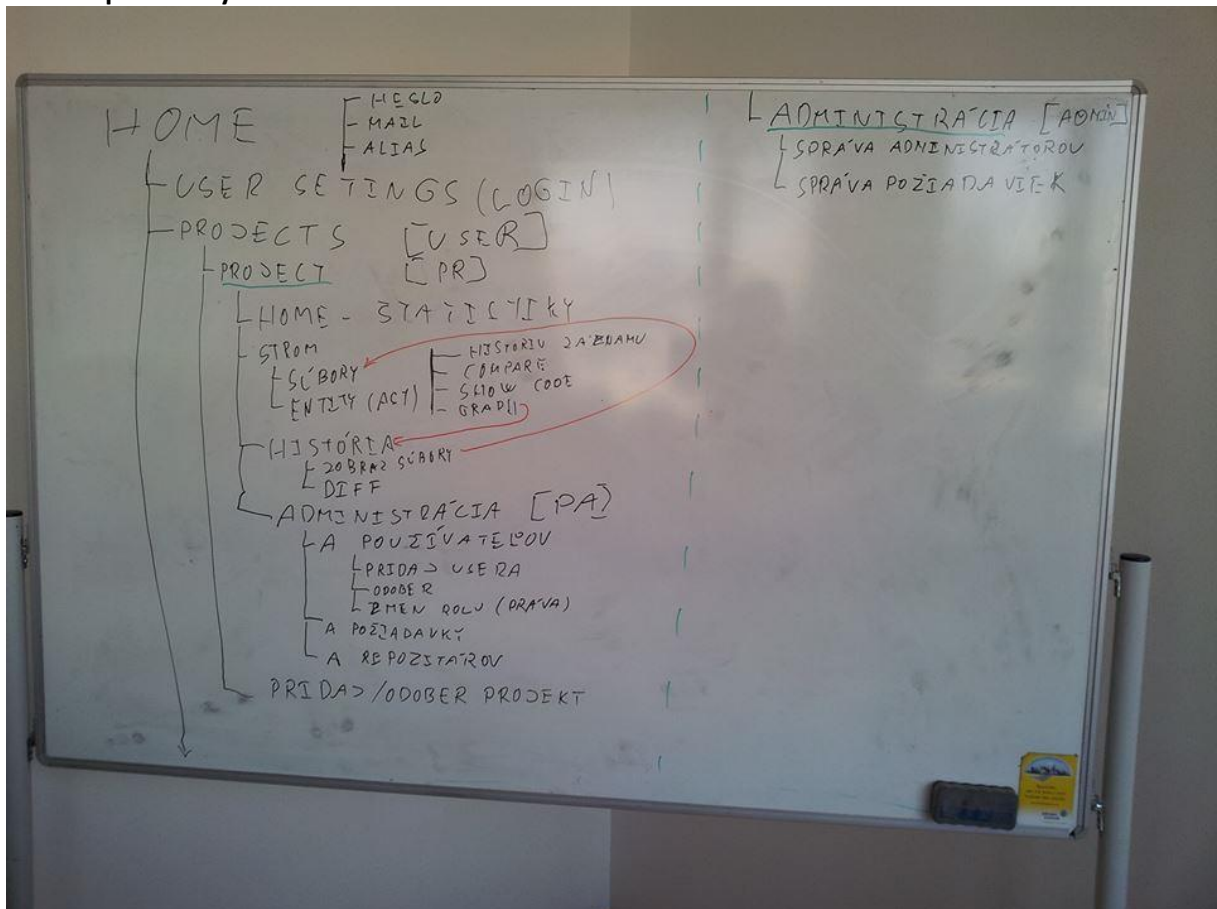
1. Priebeh stretnutia

- Stretli sme sa v zasadačke, ktorá bola vybavená projektorom
- Prešli sme všetky prvky z backlogu zo všetkých šprintov
- Pre všetky prvky sme si vyskúšali funkcionality a identifikovali sme chyby, nedostatky a navrhovali sme pre nich zmeny

2. Nájdene chyby, nedostatky

- Registrácia e-mailu – ošetriť vstup e-mailu
- Zvýraznenie kódovej entity – ofarbenie entity treba zmeniť na orámovanie
- Pri zobrazení history odstrániť čiaru z menu pri change repository
- Pri návrate späť ponechať stromovú štruktúru ako to bola otvorená
- History z fileStructure – ako partialview
- Zmeniť vizuál pri porovnávaní zdrojových kódov dvoch entít
- Pri kopírovaní odkazu sa zobrazí error
- Všetky zobrazenia kódu pridať do jedného kontrolera
- History-display files – zvýrazniť zmeny aj na najvyššej úrovni – zobrazíť zmeny v štruktúre
- Spraviť kontrolu prístupových práv aj pre nové roly
- Kontextové menu spraviť aj pre history graf okrem structure, tak isto aj pre entity structure
- Pridať menu projects aj pre administrátorov
 - admin má menu administrator
 - má menu projects
- Spraviť migráciu na vyčistenie databázy projektov
- Časy pri porovnaných súboroch sú rovnaké
- Autorizácia nefunguje
- E-mailová požiadavka na pridelenie projektov - treba viac info, minimálne Id projektu
- Upraviť vizuál pre pridelenie repositárov pre projekty
- Doimplementovať správu aliasov
- Dokončiť štatistiky projektu

3. Mapa stránky



4. Úlohy pre ďalší šprint

- Kontrola projektových práv – Chlebana
- Zmazanie starých projektov – Chlebana
- Vytvorenie a naštýľovanie systémového administračného menu – Samuhel
- Výška riadkov pri zvyrazňovaní kódu nie je určená správne – Keping
- Doplnenie informácií o požiadanie na začlenenie repozitára do projektu – Scholtz
- Zobrazenie kódu po kliknutí na meno súborovej alebo kódovej entity v strome – Scholtz
- Zvýraznenie entity v kóde žltým obdĺžnikom – Oriskó
- Zlúčenie menných priestorov v AST strome – Skrisa
- Zobrazenie tabuľky histórie entity vedľa stromu – Oriskó
- Premenovanie položiek Compare, History a Graph v kontextovom menu v strome – Oriskó
- Kontextové menu v AST strome a histórii – Scholtz
- Výber iba dvoch verzií entít na porovnanie – Oriskó
- Umožniť prechod na zadanú URL – Chlebana
- Zachovanie hlavného pohľadu po zmene repozitára – Chlebana
- Zvýraznenie entity v zdrojovom kóde – Codereview - Keping
- Infraštruktúra pre zaznamenávanie udalostí – Codereview – Oriskó
- Systémová správa projektov a repozitárov – Grešlíková
- Štatistiky značiek sú vždy počítané pre repozitár WhereIsMyCode – Samuhel
- Big picture projektu po skončení prvej etapy – Grešlíková

Príloha A

- Správa používateľov logického projektu
- Správa aliasov používateľa – Oriskó
- Základné štatistiky nad projektom – Samuhel
- User s administrátorskými právami si môže zrušiť administrátorské práva – Grešlíková
- Stránkovanie pri načítaní AST projektov do databázy – Grešlíková

Zapísal: Oriskó

Overil: Kepič

Trvanie stretnutia: 3 hodiny



Kvetináčik – retrospektíva po šprinte

Dátum: 27.02.2014 – 13.03.2014
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

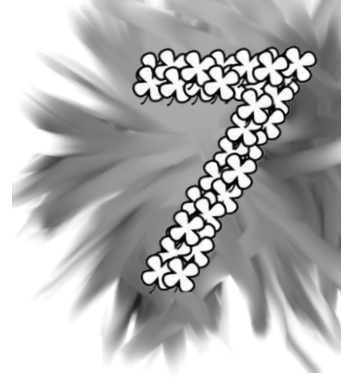
Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Vykonané v tomto šprinte:

- Zobrazenie zoznamu značiek
- Dynamické načítavanie rôznych častí HTML dokumentov
- Generovanie URL pre jednotlivé značky
- Vyhľadávanie v stromovej štruktúre
- Oprava chýb

Zhodnotenie šprintu:

V tomto šprinte sme splnili všetky úlohy tak, ako sme ich mali zadané. V ďalších šprintoch by sme mali pokračovať rovnako, a mali by sme aj priebežne písať dokumentáciu.



Prvý lístok – retrospektíva po šprinte

Dátum: 13.03.2014 – 27.03.2014
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Vykonané v tomto šprinte:

- Notifikačný systém
 - Vytváranie notifikačných filtrov a ich spravovanie
 - Registrácia notifikačného filtra na servery ITM
 - Prijímanie notifikačných správ
- Refactoring kódu
- Vytváranie značiek
 - Pridávanie priamo cez webový prehliadač
 - Komunikácia s ITM

Zhodnotenie šprintu:

V tomto šprinte sme splnili všetky úlohy tak, ako sme ich mali zadané. V ďalších šprintoch by sme mali pokračovať rovnako a vykonávať aj priebežné testovanie.



Druhý lístok – retrospektíva po šprinte

Dátum: 27.03.2014 – 10.04.2014
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Vykonané v tomto šprinte:

- Vykonalie rozsiahleho code review.
- Spracovanie notifikačných správ
 - Vytvorenie systémového endpoint-u
 - Ukladanie do databázy
- Výrazná zmena dizajnu stránky
 - Upravené vrchné menu
 - Upravená domovská stránka

Zhodnotenie šprintu:

V tomto šprinte sme splnili všetky úlohy tak, ako sme ich mali zadané. V ďalších šprintoch by sme mali pokračovať rovnako ako doteraz.



Tretí lístok – retrospektíva po šprinte

Dátum: 10.04.2014 – 28.04.2014
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Vykonané v tomto šprinte:

- Mazanie projektov
- Mazanie značiek
- Pridanie master aliasu pre používateľa
- Zlepšenie grafu vetiev v projekte
 - Farebné odlíšenie používateľov
 - Filtrácia na základe zvolených používateľov

Zhodnotenie šprintu:

V tomto šprinte sme splnili všetky úlohy tak, ako sme ich mali zadané. V ďalších šprintoch by sme mali pokračovať rovnako ako doteraz a pracovať na finalizácii projektu.



Štvrolístok – retrospektíva po šprinte

Dátum: 28.04.2014 – 09.05.2014
Číslo tímu: 7
Vedúci projektu: Ing. Karol Rástočný

Prítomní:

Z. Grešlíková, M. Chlebana, M. Scholtz, J. Skrisa, P. Oriskó, P. Samuhel, T. Kepič

Vykonané v tomto šprinte:

- Ukončenie práce na projekte
- Odstraňovanie objavených problémov
- Finalizácia projektu
- Finalizácia dokumentácie

Zhodnotenie šprintu:

V tomto šprinte sme splnili všetky úlohy tak, ako sme ich mali zadané.

PRÍLOHA B – PREBERACÍ PROTOKOL

Preberací protokol

Tímový projekt 2013/2014

Tím č.7 – Lucky Seven

Predmet odovzdávania:

- Dokumentácia riadenia - verzia na konci letného semestra
- Dokumentácia k inžinierskemu dielu - verzia na konci letného semestra

Vedúci projektu: Ing. Karol Rástočný

Podpisom potvrdzuje prevzatie vyššie uvedených častí dokumentácie

V Bratislave

.....

Dátum

.....

Podpis