

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Tím č. 7

System pre podporu tvorby rozvrhov

Dokumentácia k riadeniu tímového projektu

Vedúci projektu: Ing. Miroslav Galbavý

Členovia tímu: Bc. Marcel Baláž, Bc. Miroslav Beno, Bc. Alojz Gomola, Bc. Peter Korenek, Bc. Ján Kvak, Bc. Ján Kováč, Bc. Roman Meszároš

Študijný program: Softvérové inžinierstvo / Informačné systémy

Predmet: Tímový projekt I

Ak. rok: 2010/11

História vývoja dokumentu

Dátum zmeny	Verzia dokumentu	Opis	Autor
31.10.2010	0.5	Vytvorenie dokumentu, úvod, vloženie zápisníc zo stretnutí a úloh v tíme	Bc. Miroslav Beno
1.11.2010	0.7	Plán projektu	Bc. Miroslav Beno
2.11.2010	0.9	Podporné prostriedky projektu a priloženie ponúk	Bc. Miroslav Beno
3.11.2010	1.0	Verifikácia a opravenie chýb dokumentu	Bc. Alojz Gomola
1.12.2010	1.2	Aktualizácia zápisníc zo stretnutí	Bc. Miroslav Beno
2.12.2010	1.4	Prepracovanie plánu projektu	Bc. Alojz Gomola
4.12.2010	1.5	Vypracovanie štábnej kultúry – zdrojové kódy	Bc. Miroslav Beno
4.12.2010	1.6	Vypracovanie štábnej kultúry – dokumentácia implementácie, dokumentácia integrácie	Bc. Alojz Gomola
6.12.2010	1.9	Vloženie metodík	Bc. Miroslav Beno Bc. Ján Kvak Bc. Peter Korenek
11.12.2010	2.0	Verifikácia a opravenie chýb dokumentu	Bc. Alojz Gomola

Obsah

História vývoja dokumentu	2
Úvod.....	1
1 Úlohy členov tímu	2
2 Plán projektu.....	3
2.1 Rámcový plán projektu.....	3
2.2 Podrobný plán projektu.....	4
2.2.1 Získanie projektu a prostriedkov	4
2.2.2 Analýza existujúceho riešenia	6
2.2.3 Návrh riešenia.....	7
2.2.4 Výber častí do prototypu a ich návrh	8
2.2.5 Implementácia vybraných častí prototypu.....	8
3 Podporné prostriedky riadenia.....	10
3.1 Manažment úloh	10
3.2 Komunikácia v tíme	10
4 Zápisnice zo stretnutí.....	11
4.1 Zápisnica z tímového stretnutia č. 1.....	11
4.2 Zápisnica z tímového stretnutia č. 2.....	13
4.3 Zápisnica z tímového stretnutia č. 3.....	15
4.4 Zápisnica z tímového stretnutia č. 4.....	17
4.5 Zápisnica z tímového stretnutia č. 5.....	19
4.6 Zápisnica z tímového stretnutia č. 6.....	20
4.7 Zápisnica z tímového stretnutia č. 7.....	22
4.8 Zápisnica z tímového stretnutia č. 8.....	24
4.9 Zápisnica z tímového stretnutia č. 9.....	26
5 Štábná kultúra.....	29
5.1 Písanie zdrojových kódov	29
5.1.1 JavaScript.....	29
5.1.2 PHP	32
5.2 Písanie dokumentácie implementácie	34
5.2.1 Obsah a forma dokumentácie k implementácii	34
5.2.2 Citovanie implementačných prvkov v dokumentácií	37
5.3 Integrácia projektu	39

5.3.1 Načítanie tried.....	39
5.3.2 Spracovanie URL.....	40
5.3.3 Pridanie novej funkcionality	41
Príloha A – Ponuky	A-1
Profil tímu.....	A-1
Ponuka na tému 19 - Model používateľa pre jeho identifikáciu	A-3
Ponuka na tému 3 - Evolučný simulátor umelého života založený na heuristických pravidlách	A-5
Ponuka na tému 7 - Tréner mentálnych schopností	A-6
Príloha ponuky A - Zoradenie všetkých tém podľa priority.....	A-9
Príloha ponuky B - Aktuálny rozvrh všetkých členov tímu	A-10
Príloha B – Použité metodiky	B-1
Manažment úloh prostredníctvom nástroja dotProject	B-2
Metodika manažmentu verzií zdrojového kódu v programe GIT.....	B-13
Evidencia chýb pomocou nástroja Bugzilla	B-25
Príloha C – Preberacie protokoly	C-1
Preberací protokol.....	C-2
Preberací protokol.....	C-3

Úvod

Tento dokument obsahuje informácie súvisiace s riadením a manažmentom tímového projektu s názvom Systém na podporu tvorby rozvrhov. Tento projekt realizuje tím č.7 „Schedule of Pain“ v rámci predmetu Tímový projekt v akademickom roku 2010/2011. Dokument zároveň zahŕňa softvérové artefakty, ktoré vznikli počas riešenia projektu a boli priebežne dopĺňané.

V kapitole číslo 1 sú identifikované role a úlohy jednotlivých členov tímu. V kapitole číslo 2 je uvedený plán projektu, jeho postupné plnenie a aktuálny stav. V kapitole číslo 3 sa nachádza opis použitých podporných prostriedkov. Kapitola číslo 4 zhromažďuje podrobné zápisnice z oficiálnych stretnutí tímu, počnúc prvou zápisnicou. Príloha A obsahuje vypracované ponuky na tri zadania tým tímových projektov. V prílohe B sú šablóny preberacích protokolov.

1 Úlohy členov tímu

V tíme boli rozdelené medzi členov jednotlivé zodpovednosti a pozície podľa špecifických rolí (Tab. 1). Vykonávané činnosti na projekte však nemusia spadať len do týchto kategórií, jedná sa o oblasti z hľadiska manažmentu, ktorým sa členovia tímu venujú primárne.

Tab. 1 Role v tíme

Člen tímu	Pozícia
Bc. Alojz Gomola	Vedúci manažér tímu
Bc. Ján Kvak	Zástupca vedúceho manažéra
Bc. Marcel Baláž	Manažér vývoja
Bc. Roman Meszároš	Manažér plánovania
Bc. Miroslav Beno	Manažér kvality
Bc. Peter Korenek	Manažér podporných metód
Bc. Ján Kováč	Manažér dokumentácie

Nasledujúca tabuľka uvádza autorstvo konkrétnych častí projektovej dokumentácie, tak ako boli v rámci projektu rozdelené a vypracované (Tab. 2).

Tab. 2 Autorstvo častí dokumentácie

Člen tímu	Časť dokumentácie
Bc. Alojz Gomola	Špecifikácia a návrh učiteľskej časti
Bc. Ján Kvak	Návrh študentskej časti
Bc. Marcel Baláž	Analýza študentskej časti
Bc. Roman Meszároš	Špecifikácia požiadaviek študentskej časti
<i>pozri História vývoja dokumentu</i>	Dokumentácia riadenia
Bc. Peter Korenek	Analýza súčasného stavu učiteľskej časti
Bc. Ján Kováč	Analýza minuloročného riešenia a problémovej oblasti

2 Plán projektu

2.1 Rámcový plán projektu

Navrhnutý všeobecný plán tímového projektu (Tab. 3) na zimný semester, vypracovaný pri inicializácii projektu, vychádza zo stanoveného rámcového harmonogramu predmetu Tímový projekt.

Tab. 3 Rámcový plán projektu

Týždeň	Dátum	Popis činnosti
1.	20.9. – 26.9.	Zostavenie tímu Výber tém Vypracovanie ponúk
2.	27.9. – 3.10.	Oboznámenie sa zo získanou témou Stretnutie s pedagogickým vedúcim Rozdelenie rolí a úloh v tíme
3.	4.10. – 10.10.	Prvé oficiálne stretnutie tímu Podrobné oboznámenie sa s problematikou projektu Konzultácie s členmi minuloročného tímu Vytvorenie všeobecného plánu Vytvorenie webovej prezentácie tímu
4.	11.10. – 17.10.	Analýza problému Štúdium problematiky Konzultácie s členmi minuloročného tímu Nasadenie minuloročného výstupu Testovanie minuloročného výstupu
5.	18.10. – 24.10.	Analýza problému Špecifikácia požiadaviek na systém Konkretizácia zvažovaných technológií Nasadenie podporných prostriedkov
6.	25.10. – 31.10.	Analýza problému Špecifikácia požiadaviek na systém Návrh riešenia Kompletizovanie dokumentácie k 1. kontrolnému bodu
7.	1.11. – 7.11.	Odovzdanie systémovej dokumentácie a dokumentácie riadenia k 1. kontrolnému bodu Návrh riešenia
8.	8.11. – 14.11.	Návrh riešenia Návrh prototypu častí systému
9.-11.	15.11. – 5.12.	Prototypovanie vybraných častí Implementácia Testovanie Dokumentovanie
12.	6.12. – 12.12.	Odovzdanie prototypu vybraných častí systému Odovzdanie dokumentácie Používateľská prezentácia

2.2 Podrobný plán projektu

Nasledujúca podkapitola obsahuje podrobný plán projektu rozdelený do šiestich etáp, ktorý vyplynul zo spresňovania rámcového plánu minimálne raz za 4 týždne. Zároveň je uvedené plnenie a zhodnotenie plánov po uplynutí ich stanovenej doby.

Základné aktivity realizujúce projekt sú vykonávané v rámci oficiálnych spoločných stretnutí z rozsahom 3 hodiny týždenne, na ktorých sú prítomní optimálne všetci členovia tímu spolu s pedagogickým vedúcim. Konkrétne úlohy sú pridelované na tomto stretnutí a uskutočňované zodpovednými členmi tímu zvyčajne s termínom do ďalšieho stretnutia. Vedúci tímu dohliadal na plnenie podrobného plánu a spolupodielal sa na tvorbe podrobného plánu projektu.

Termíny podrobného plánovania boli postupne zhusťované, čo bolo spôsobené zväčšujúcou sa zložitou projektom. Prvé tri týždne neexistovalo reálne plánovanie projektu, preto plán na prvé štyri týždne bol doplnený spätne. Podrobný plán projektu je delený na etapy podrobného plánovania podľa (Tab. 4). Pri plánovaní jednotlivých etáp sme využívali systém rezerv, plánovali sme viac ako sme potrebovali zvládnuť.

Tab. 4 Podrobné plánovanie rámcového plánu.

Týždeň	Názov etapy	Vypracoval
1. 20.9. – 26.9. 2. 27.9. – 3.10 3. 4.10. – 10.10. 4. 11.10. – 17.10.	Získanie projektu a prostriedkov	Alojz Gomola
5. 18.10. – 24.10.	Analýza existujúceho riešenia	Alojz Gomola
6. 25.10. – 31.10. 7. 1.11. – 7.11.	Návrh riešenia	Miroslav Beno Alojz Gomola
8. 8.11. – 14.11.	Výber častí do prototypu a ich návrh	Miroslav Beno
9. 15.11. – 21.11. 10. 22.11– 28.11 11. 29.11.–5.12	Implementácia vybraných častí prototypu	Miroslav Beno Ján Kvak Marcel Baláž
12. 6.12. – 12.12.	Testovanie , dokončovanie dokumentácie	Alojz Gomola

2.2.1 Získanie projektu a prostriedkov

Plán bol navrhnutý z ohľadom na rámcový plán, plán sme preplánovali druhý týždeň po stretnutí s našim vedúcim.

ID	Názov úlohy	Vypracuje	19 Sep 2010					26 Sep 2010									
			20	21	22	23	24	25	26	27	28	29	30	1	2	3	
1	Ponuka	Všetci	[Blue bar]														
2	Ponuka HERBAL	Alojz Gomola	[Blue bar]														
3	Ponuka Mentálny Tréner	Ján Kvak Miro Beno	[Blue bar]														
4	Ponuka Sledovanie používateľa	Peter Korenek Roman Meszároš	[Blue bar]														
5	Kompletizácia ponuky	Alojz Gomola					[Blue bar]										
6	Stretnutie s pedagogickým vedúcim	Všetci											[Blue bar]				
7	Samoštúdium problematiky	Všetci												[Blue bar]			
8	Rozdelenie úloh v tíme	Alojz Gomola Roman Meszároš												[Blue bar]			

Obr. 1 Plán na prvé dva týždne.

Plán na Obr. 1 bol prvoplánový a držali sme sa toho, že dostaneme nový projekt, získali sme však už hotový projekt, tak sme boli nútení preplánovať ako vidíme na Obr. 2, ktorý zobrazuje plán na ďalšie dva týždne.

ID	Názov úlohy	Vypracuje	9 Oct 2010							10 Oct 2010							17 Oct 2010			
			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Ponuka	Všetci																		
2	Ponuka HERBAL	Alojz Gomola																		
3	Ponuka Mentálny Tréner	Ján Kvak Miro Beno																		
4	Ponuka Sledovanie používateľa	Peter Korenek Roman Meszároš																		
5	Kompletizácia ponuky	Alojz Gomola																		
6	Stretnutie s pedagogickým vedúcim	Všetci																		
7	Samoštúdium problematiky	Všetci	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
8	Rozdelenie úloh v tíme	Alojz Gomola Roman Meszároš																		
9	Webová prezentácia projektu	Alojz Gomola Ján Kvak				[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
10	Vytvorenie kópie na LABBS2	Alojz Gomola				[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
11	Metodika inštalácie prostredia WAMP	Peter Korenek Roman Meszároš				[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
12	Výber vhodných nástrojov na manažment	Miro Beno, Ján Kvak Ján Kováč				[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
13	Získať informácie od minuloročného tímu	Marcel baláž				[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
14	Konfigurácia LDAP/HTTPS	Alojz Gomola											[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
15	Abstrakt alternatívneho riešenia	Peter Korenek											[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
16	Analýza prokročilých Web technológií	Roman Meszároš											[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
17	Abstrakt používateľského rozhrania	Ján Kvak											[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]
18	Testovanie minuloročnej verzie	Ján Kováč, Marcel Baláž, Miro Beno											[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]	[Blue bar]

Obr. 2 Plán na tretí a štvrtý týždeň.

Zhodnotenie plnenia plánu

Prvý týždeň prebiehalo obstarávacie konanie na získanie projektu, tu sme bez podporného nástroja rozdelili jednotlivé úlohy.

Druhý týždeň sme plánovali stretnutie s vedúcim, na základe výberového konania nám bol pridelený projekt na systém na podporu tvorby rozvrhov. Dňa 27.10. došlo k prvému preplánovaniu úloh, na základe bližšej znalosti témy projektu. Rozhodli sme sa získať, spojzdať a študovať minuloročné riešenie. Následne sme zjemnili rámcový plán, tak aby zodpovedal nášmu zadaniu. Stanovili sme jednotlivé roly v tíme. Plán z prvého týždňa sme splnili. Podarilo sa nám získať projekt.

Tretí týždeň sme pokračovali v plnení plánu z druhého týždňa, nebolo potrebné žiadne preplánovanie. Zistili sme, že niektorí členovia tímu majú tendenciu pracovať štýlom „as late as possible“. Preto sme zaviedli nedeľné podávanie výkazu činnosti.

Štvrtý týždeň sme pokračovali v plnení plánu takmer bez porušenia, Roman Meszároš a Peter Korenek splnili zadané úlohy s mimoriadnym predstihom. Došlo k preplánovaniu, lebo vznikli drobné technické problémy z serverom LABSS2, ktoré sa nám podarilo vyriešiť.

2.2.2 Analýza existujúceho riešenia

Piaty týždeň sme naplánovali analýzu existujúceho riešenia a jeho možných východísk (Obr. 3). Vzhľadom na hlbšiu znalosť schopností členov tímu, vedúci tímu Alojz Gomola prideliť úlohy tak aby zodpovedali špecializácii a záujmom jednotlivých členov tímu. Plán bol dimenzovaný na jeden týždeň s ohľadom na možné riziká spojené s dlhou analýzou. Ján Kováč nepracoval na žiadnej špecifickej úlohe, ale bol priradený na analýzu minuloročného zadania a jeho sumarizáciu.

ID	Názov úlohy	Vypracuje	17 Oct 2010					24 Oct 2010				
			18	19	20	21	22	23	24	25	26	27
1	Webová prezentácia projektu	Alojz Gomola Ján Kvak	■	■	■	■	■	■	■	■	■	
2	Metodika inštalácie prostredia WAMP	Peter Korenek Roman Meszároš	■	■	■	■	■	■	■	■	■	
3	Abstrakt alternatívneho riešenia	Peter Korenek	■	■	■	■	■	■	■	■	■	
4	Abstrakt používateľského rozhrania	Ján Kvak	■	■	■	■	■	■	■	■	■	
5	Testovanie minuloročnej verzie	Ján Kováč, Marcel Baláž, Miro Beno	■	■	■	■	■	■	■	■	■	
6	Matematický model, predbežný opis procesu	Alojz Gomola										
7	Analýza študentskej časti	Miroslav Beno Marcel Balaz										
8	Analýza učiteľskej časti	Peter Korenek Roman Meszároš										
9	Funkcionálny model študentskej časti	Ján Kvak										

Obr. 3 Plán na piaty týždeň.

Zhodnotenie plnenia plánu

Analýza častí bola hotová v utorok, čiže tesne pred termínom. Funkcionálny a matematický model boli dokončené v pondelok. Všetky požadované výstupy boli dodané do termínu ďalšieho cvičenia. Úroveň výstupov bola pomerne dobrá, aj keď bolo pomerne veľa gramatických chýb. Kvalitár musel

nad rámec plánu zrevidovať niektoré dokumenty. Pre niektoré postupy bola potrebná konzultácia s prodekanou pre vzdelávanie, tieto boli vykonané nad rámec plánu. Postupom času sme sa dopracovali k ucelenej analýze problémovej oblasti. Túto sme prezentovali nášmu vedúcemu na tímovom stretnutí. Daný postup bol odobrený. Celkové plnenie plánu bolo dobré, aj keď bolo nutné vykonať niektoré úkony navyše.

2.2.3 Návrh riešenia

Návrh riešenia bol dimenzovaný na dva týždne, do dvoch fáz, návrh teoretický, do 3.11. a návrh praktický od 4.11. V siedmom týždni bol termín odovzdávania dokumentácie, preto sme plánovali vytvorenie návrhov jednotlivých častí systému do pondelka 1.11. najneskôr utorka 2.11. Keďže sme už dlhšie mali identifikované niektoré záležitosti z technickej stránky, tak sme ich plánovali na druhý týždeň. Medzi tieto záležitosti patrili nástroje na tvorbu nového systému, opravy v starom systéme, nevyhovujúca databáza a jej refaktoring. Plán návrhu je možno vidieť na obrázku Obr. 4.

ID	Názov úlohy	Vypracuje	24 Oct 2010				31 Oct 2010						7 Nov 2010					
			27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	
1	Matematický model, predbežný opis procesu	Alojz Gomola	■															
2	Analýza študentskej časti	Alojz Gomola Peter Korenek	■	■	■	■												
3	Analýza učiteľskej časti	Ján Kvak Roman Meszároš	■	■	■	■												
4	Funkcionálny model študentskej časti	Ján Kvak	■	■	■	■												
5	Vytvorenie Plánov	Miro Beno	■	■	■	■												
6	Integrácia dokumentácie	Ján Kováč																
7	Analýza existujúcich formulárov	Beno Miroslav	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
8	Metodika písania zdrojových kódov	Beno Miroslav																
9	Refaktoring Datab=azy	Gomola,Korenek, Meszároš																
10	Nástroje JSCRIPT	Ján Kováč																
11	Export Dokumentov	Ján Kováč																
12	Záťažové testy	Marcel Baláž																
13	PHP Framework + LDAP	Miro Beno																

Obr. 4 Plán na šiesty a siedmy týždeň.

Zhodnotenie plnenia plánu

Šiesty týždeň bolo všetko odovzdané s jednoduchým predstihom, zjavne boli členovia tímu motivovanejší ako inokedy. To mohlo byť spôsobené zvýšeným pracovným nasadením. Dokumentácia bola odovzdaná do rúk pedagogického vedúceho na čas a v poriadku, jednotlivé časti prešli kontrolu kvalitéra a boli schválené vedúcim tímu. Pedagogický vedúci podpísal preberací protokol.

Siedmy týždeň sme začali s podpornými úlohami a naplánovaným refaktoringom databázy. Daný refaktoring bol potrebný od začiatku projektu. Refaktoring a jeho návrh boli hotové s malým sklzom. Bol pripravený plán na preklopenie databázy. Ostatní členovia plnili úlohy načas. Bola prijatá metodika na písanie zdrojového kódu, vo veľmi vysokej kvalite.

2.2.4 Výber častí do prototypu a ich návrh

Táto pomerne krátka etapa plánu trvala iba týždeň zato bola dosť dôležitá nato, aby si zaslúžila samostatný plán. V niektorých úlohách sa pokračovalo, navyše absencia spoločného stretnutia bola citeľná. Toto sme riešili intenzívnejšou komunikáciou cez ostatné komunikačné kanály. Nakoniec vznikol nasledovný plán (Obr. 5).

ID	Názov úlohy	Vypracuje	7 Nov 2010					14 Nov 2010				
			8	9	10	11	12	13	14	15	16	17
1	Integrácia dokumentácie	Ján Kováč	[Blue bar]					[Yellow bar]				
2	Metodika písania zdrojových kódov	Beno Miroslav	[Blue bar]					[Yellow bar]				
3	Refaktoring Datab=azy	Peter Korenek Roman Meszároš	[Blue bar]									
4	Nástroje JSCRIPT	Ján Kováč	[Blue bar]					[Yellow bar]				
5	Záťažové testy	Marcel Baláž	[Blue bar]					[Yellow bar]				
6	PhP Framework + LDAP	Miro Beno	[Blue bar]					[Yellow bar]				
7	Úprava úložiska	Ján Kvak	[Blue bar]					[Yellow bar]				
8	Intergrácia JQUERY	Ján Kvak	[Blue bar]					[Yellow bar]				
9	Príprava integrácie MVC	Alojz Gomola	[Blue bar]					[Yellow bar]				
10	Prieskum Existujúcich technológií kolaborácie	Marcel Baláž	[Blue bar]					[Yellow bar]				
11	Návrh formulárov pre ucitelskú cast	Kováč, Kvak , Beno, Baláž	[Blue bar]					[Yellow bar]				
12	Inštalácia Tortoise GIT	Všetci	[Blue bar]					[Yellow bar]				

Obr. 5 Plán na ôsmy týždeň.

Zhodnotenie plnenia plánu

Plán sme plnili dobre niektoré úlohy ako refaktoring databázy sme museli kvôli komplexnosti natiahnuť. Zaviedli sme viacúrovňovú hierarchiu do vývoja, rozdelili sme sa do funkčných blokov. V týchto blokoch sme viac menej pracovali samostatne.

2.2.5 Implementácia vybraných častí prototypu

Plán na implementáciu prototypu (Obr. 6) bol veľmi hrubo načrtnutý, takisto sa v implementácii prototypu vyskytlo najviac zmien.

ID	Názov úlohy	Vypracuje	14 Nov 2010				21 Nov 2010							28 Nov 2010							5 Dec 2010					
			17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8		
1	Refaktoring Datab=azy	Peter Korenek RomanMeszároš	[Blue bar from 17 Nov to 4 Dec]																							
2	Intergrácia JQUERY	Ján Kvak	[Blue bar on 17 Nov]																							
3	Príprava integrácie MVC	Alojz Gomola	[Blue bar on 17 Nov]																							
4	Prieskum Existujúcich technológi kolaborácie	Marcel Baláž	[Blue bar on 17 Nov]																							
5	Návrh formulárov pre ucitelskú cast	Kováč, Kvak , Beno, Baláž	[Blue bar from 17 Nov to 21 Nov]																							
6	Metodika integrácie do MVC	Alojz Gomola	[Blue bar from 17 Nov to 20 Nov]																							
7	Metodika dokumentácie implementácie	Alojz Gomola	[Blue bar from 17 Nov to 20 Nov]																							
8	Implementácia Kolaborácie	Marcel Baláž	[Blue bar from 21 Nov to 30 Nov]																							
9	Dokumentácia implementácie Kolaborácie	Alojz Gomola	[Blue bar from 21 Nov to 4 Dec]																							
10	Dokumentacia implementacie ucitelskeho prostredia	Jan Kvak	[Blue bar from 21 Nov to 4 Dec]																							
11	Implementacia ucitelskeho prostredia	Kvak, Kováč, Korenek, Beno	[Blue bar from 21 Nov to 4 Dec]																							
12	Dokumentácia riadenia	Alojz Gomola Miro Beno	[Blue bar on 17 Nov]																							

Obr. 6 Plán na deviaty, desiaty, jedenásty týždeň.

Zhodnotenie plnenia plánu

Deviaty týždeň sa vyskytli problémy pri preklápaní databázy, niektoré veci, ktoré boli implementované v databáze sme odstránili z kódu a nahradili ich alternatívnou funkčnosťou. Veľa nadčasov odpracovali najmä Peter Korenek, Roman Meszároš. Všeobecne sa plán podarilo splniť.

Desiaty týždeň sme plnili úlohy na čas, Marcel Baláž implementoval Kolaboratívne prostredie. Ostatné úlohy sa plnili, bolo spresnené rozdelenie formulárov.

3 Podporné prostriedky riadenia

3.1 Manažment úloh

Na manažment úloh tím využíva voľne dostupný nástroj *dotProject*. Jedná sa o webovú aplikáciu, ktorá umožňuje spravovanie viacerých užívateľov, a rôznych lokalizačných prostredí. Ponúka pritom rozšírené možnosti:

- manažovania projektov a úloh
- vedenia spoločností, oddelení, kontaktov
- zaznamenávania „todo“ zoznamov
- evidencie zdrojov

Dá sa sledovať vykonávanie úloh, plnenie plánu, vyťaženosť jednotlivých členov tímu a podobne. Nástroj umožňuje generovať okrem iných užitočných štatistík projektu napríklad aj Ganttové diagramy, ktoré boli použité aj na vizualizovanie postupu projektu v tejto dokumentácii (Obr. 1). Prístupný je kalendár, história a zálohovanie informácií o stave projektu v určitých časových bodoch pre spätné vyhodnotenie.

Z nástroja využívame funkcionality potrebnú pre tímy menšieho rozsahu, čiže evidujeme iba jednu spoločnosť, oddelenie a projekt. V rámci neho boli vytvorené záznamy o jednotlivých členoch, spolu s príslušnými kontaktmi a rolami. Intenzívne je využívaná časť pridelovania úloh, kedy špecifikujeme zodpovednosti za úlohy a časový rozsah pridelený na splnenie týchto úloh. Následne sledujeme vykonávanie úloh a reálnu spotrebu času spojenú s týmto výkonom. Vynechávame niektoré podrobnosti špecifické pre biznis aplikácie, ako napríklad prácu s rozpočtom.

3.2 Komunikácia v tíme

Okrem oficiálnych či neoficiálnych osobných stretnutí prebieha komunikácia medzi členmi tímu primárne pomocou distribuovanej mailovej prispievacej skupiny v *Google Groups*. Vedené sú tematické bloky, do ktorých prispievajú s témou zainteresovaní členovia tímu. Uložené sú rôzne súbory súvisiace s projektom, prístupné na stiahnutie. Ak je potrebná intenzívnejšia komunikácia v reálnom čase pri riešení konkrétneho problému, využívajú sa nástroje podporujúce „instant messaging“ ako *ICQ* a *Skype*, predpokladom čoho bola vzájomná výmena kontaktov na tieto siete. Komunikácia medzi tímom a pedagogickým vedúcim prebieha na oficiálnych stretnutiach osobne, kedy odovzdávame na posúdenie vypracované dokumenty, a diskutujeme ďalší postup a riešenia, poprípade dodatočné otázky konzultujeme mailovou formou.

4 Zápisnice zo stretnutí

4.1 Zápisnica z tímového stretnutia č. 1

Dátum: 6. 10. 2010
Čas: 18:00 – 19:15
Miesto: Softvérové štúdio FIIT ST² BA
Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý
Členovia tímu: Bc. Marcel Baláž Bc. Miroslav Beno
 Bc. Alojz Gomola Bc. Peter Korenek
 Bc. Ján Kvak Bc. Roman Meszároš
 Bc. Ján Kováč
Ďalší: Predchádzajúci tím: Bc. Peter Mikuška, Bc. Dušan Palo, Bc. Matej Krchniak, Bc. Samuel Števaňák
Moderátor: Ing. Miroslav Galbavý
Zápis: Vyhotovil: Bc. Miroslav Beno
Overil: Bc. Alojz Gomola

Téma stretnutia:

Stretnutie s minuloročným tímom a základné oboznámenie sa s požiadavkami na systém.

Priebeh stretnutia:

- Stretnutie prebiehalo s vopred neurčeným programom prostredníctvom voľnej diskusie medzi účastníkmi. Zo stretnutia vyplynuli nasledovné poznatky:
- zoznámenie sa s minuloročnými členmi tímu, ktorý vyvíjal Systém na podporu tvorby rozvrhov + výmena kontaktov
- odporúčané nástroje na vývoj: NetBeans, Notepad++, PSPad, EclipsePHP, XAMPP, WAMP
- administráciu serverov v softvérovom štúdiu (SŠ) má na starosti Ing. Peter Lacko, PhD., snapshot systému je už na serveri uložený, ale treba vybaviť prístupové práva
- je dobré mať aj záložný server v prípade nedostupnosti SŠ, možnosťou je Ynet, systém nemá byť prístupný verejne
- repozitáre SVN je možné realizovať tiež na serveroch SŠ
- v existujúcom systéme by mali byť formuláre (rozhranie) hotové, primárne sa už budeme venovať úprave kódu a databázy

- systém už bol podrobený v decembri 2009 testovacej prevádzke, a pripomienky boli zapracované
- existujúce kódy systému by mali byť do istej miery refaktorované a komentované
- autentifikácia v systéme prebieha cez systém LDAP - prístup naň je len z IP fakulty, autentifikácia sa dá vypnúť
- v databáze by sa malo počítať s archiváciou dát a následnými iteráciami v jednotlivých akademických obdobiach
- prepojenie so systémom AIS mali na starosti minulý rok Dušan a Samo, toto prepojenie je komplikované kvôli neflexibilitnosti AIS
- primárnou úlohou systému je zber požiadaviek na rozvrh a následne podpora pri jeho vytváraní (nie automatizovane)
- našimi prvými úlohami bude rozchodiť systém, oboznámiť sa s jeho funkciami, a prejsť už vloženými pripomienkami
- Ing. Galbavý nám poskytne neskôr aj reálne požiadavky, s ktorými sa pracovalo pri vytváraní rozvrh na aktuálne akademické obdobie
- pracovať budeme s dátami a charakteristikami novej budovy FIIT, keďže sa plánuje jej používanie už budúci rok

Zhodnotenie úloh z predchádzajúceho stretnutia:

-

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
1.1	Naštudovať minuloročnú systémovú dokumentáciu	Všetci	6. 10. 2010	13. 10. 2010
1.2	Pripraviť webovú prezentáciu tímu	Bc. Alojz Gomola, Bc. Ján Kvak	6. 10. 2010	13. 10. 2010
1.3	Kontaktovať p. Lacka ohľadom servera v SŠ	Bc. Alojz Gomola	6. 10. 2010	13. 10. 2010
1.4	Nainštalovať a rozchodiť lokálny snapshot systému na prostredí X/WAMP	Bc. Peter Korenek, Bc. Roman Meszároš	6. 10. 2010	13. 10. 2010
1.5	Vytvoriť a naplniť šablónu na zápisnice stretnutí tímu	Bc. Miroslav Beno	6. 10. 2010	13. 10. 2010
1.6	Nakontaktovať minuloročný tím	Bc. Marcel Baláž	6. 10. 2010	13. 10. 2010
1.7	Podrobne preštudovať analýzu minuloročného výstupu	Bc. Ján Kováč	6. 10. 2010	13. 10. 2010
1.8	Nájsť vhodné nástroje na manažment úloh, manažment verzíí a zdieľanie zdrojových kódov a dokumentácie	Bc. Marcel Baláž, Bc. Miroslav Beno, Bc. Ján Kvak, Bc. Ján Kováč	6. 10. 2010	13. 10. 2010

Poznámky k stretnutiu:

Dohodli sme sa na budúcom abecednom poradí zapisovateľov – najbližšie zapisuje Marcel.

Prílohy:

4.2 Zápisnica z tímového stretnutia č. 2

Dátum: 13. 10. 2010
Čas: 18:00 – 20:40
Miesto: Softvérové štúdio
Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý
Členovia tímu: Bc. Marcel Baláž Bc. Miroslav Beno
 Bc. Alojz Gomola Bc. Peter Korenek
 Bc. Ján Kvak Bc. Roman Meszároš
 Bc. Ján Kováč
Ďalší: Bc. Matej Krchniak, Bc. Peter Mikuška, Bc. Samuel Števaňák,
Moderátor: Ing. Miroslav Galbavý
Zápis: Vyhotovil: Bc. Marcel Baláž
Overil: Vyberte položku.

Téma stretnutia:

Konzultácia technologických problémov s minuloročným tímom a návrh alternatívneho riešenia projektu.

Priebeh stretnutia:

Ku stretnutiu boli pripravené otázky na minuloročný tím, ktoré boli v krátkom časovom intervale zodpovedané. Zvyšok stretnutia prebiehala diskusia o inovatívnom alternatívnom riešení projektu. Tému alternatívneho riešenia predložil Ing. Galbavý.

Poznámky zo stretnutia:

- Technologické problémy s LDAP a HTTPS na strane školského testovacieho servera je potrebné vyriešiť s Ing. Lackom

Návrh alternatívneho riešenia:

- Rozvrh ako hra
- Hrajú pedagógovia aj žiaci
- Počet hráčov až niekoľko tisíc
- V hre sa „bojuje“ proti časovým a priestorovým obmedzeniam, ktoré ponúka kapacita školy, s cieľom vytvoriť rozvrh, s ktorým „každý vyhráva“
- Hráči s najvyššou prioritou na začiatku určia základné pravidlá hry (predmety, počty ľudí na predmet, počet hodín prednášok a cvičení, vyhradené priestory a časy)
- Následne sa hlasuje o jednotlivé termíny, najskôr asi cvičiaci, potom žiaci (?)

- Hlasovanie na spôsob hry s bodmi, alebo žetónmi.
- Počet bodov napr. podľa VŠP
- Ďrčovanie priorit a alternatív na jednotlivé termíny
- Riadené riešenie konfliktov
- Sociálna a kolaboratívna tvorba rozvrhu
- Vysoký dôraz na Ďl a použiteľnosť
- potrebná dôkladná analýza

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
1.1	Naštudovať minuloročnú systémovú dokumentáciu	Všetci	6. 10. 2010	13. 10. 2010	Rozpracovaná
1.2	Pripraviť webovú prezentáciu tímu	Bc. Alojz Gomola, Bc. Ján Kvak	6. 10. 2010	13. 10. 2010	Splnená
1.3	Kontaktovať p. Lacka ohľadom servera v SŠ	Bc. Alojz Gomola	6. 10. 2010	13. 10. 2010	Splnená
1.4	Nainštalovať a rozchodiť lokálny snapshot systému na prostredí X/WAMP	Bc. Peter Korenek, Bc. Roman Meszároš	6. 10. 2010	13. 10. 2010	Splnená
1.5	Vytvoriť a naplniť šablónu na zápisnice stretnutí tímu	Bc. Miroslav Beno	6. 10. 2010	13. 10. 2010	Splnená
1.6	Nakontaktovať minuloročný tím	Bc. Marcel Baláž	6. 10. 2010	13. 10. 2010	Splnená
1.7	Podrobne preštudovať analýzu minuloročného výstupu	Bc. Ján Kováč	6. 10. 2010	13. 10. 2010	Rozpracovaná
1.8	Nájsť vhodné nástroje na manažment úloh, manažment verzií a zdieľanie zdrojových kódov a dokumentácie	Bc. Marcel Baláž, Bc. Miroslav Beno, Bc. Ján Kvak, Bc. Ján Kováč	6. 10. 2010	13. 10. 2010	Rozpracovaná

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
2.1	Kontaktovať Ing. Lacka ohľadom konfigurácie LDAP a HTTPS	Bc. Alojz Gomola	13. 10. 2010	20. 10. 2010
2.2	Opísanie alternatívneho riešenia	Bc. Peter Korenek	13. 10. 2010	20. 10. 2010
2.3	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	13. 10. 2010	20. 10. 2010
2.4	Vypracovanie zápisnice zo stretnutia a rozdelenie úloh	Bc. Marcel Baláž	13. 10. 2010	20. 10. 2010
2.5	Návrh Ďl elementov pre alt. riešenie	Bc. Ján Kvak	13. 10. 2010	20. 10. 2010
2.6	Nápady na funkcionality alt. riešenia	Všetci	13. 10. 2010	20. 10. 2010
2.7	Testovanie verzie z minulého roka	Všetci	13. 10. 2010	20. 10. 2010

Poznámky k stretnutiu:

Stretnutie sa nieslo v duchu brainstormingu.

Prílohy:

-

4.3 Zápisnica z tímového stretnutia č. 3

Dátum: 20. 10. 2010
Čas: 18:00 – 20:40
Miesto: Softvérové štúdio
Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý
Členovia tímu: Bc. Marcel Baláž Bc. Miroslav Beno
 Bc. Alojz Gomola Bc. Peter Korenek
 Bc. Ján Kvak Bc. Roman Meszároš
 Bc. Ján Kováč
Ďalší: Bc. Matej Krchniak
Moderátor: Ing. Miroslav Galbavý
Zápis: Vyhotovil: Bc. Alojz Gomola
Overil: Bc. Ján Kvak

Téma stretnutia:

Konzultácia alternatívneho riešenia, prezentovanie prototypu nového rozhrania, prezentovanie koncepcie celkového riešenia.

Priebeh stretnutia:

Začali sme objasnením koncepcie riešenia vedúcemu projektu, viedli sme konštruktívnu diskusiu a doladzovali sme detaily stretnutia. Vzhľadom na komplexnosť analýzy sme začali s analýzou predchádzajúceho riešenia, prebrali sme znalostnú časť analýzy doladili sme niektoré drobnosti.

Poznámky zo stretnutia:

- Technologické problémy s LDAP a HTTPS na strane školského testovacieho servera je potrebné vyriešiť s Ing. Lackom
- Výber vhodnej technológie AJAX alebo CakePHP, vzhľadom na náročnosť grafického prostredia sme uvažovali aj Flash
- Identifikovali sme CakePHP ako framework na ktorom bol vyvíjaný systém, tento framework je dobrý a zvažujeme jeho zachovanie v učiteľskej časti

Poznámky ku riešeniu:

- Systém sme rozdelili na učiteľskú a študentskú časť, učiteľskú časť sme sa rozhodli zachovať v konzervatívnom duchu

- ě čiteľskú časť sme pojali formou kartovej hry, učiteľ dostane určitý obnos kreditov, ktorý môže investovať do rozvrhu alebo rôznych “vymožeností”
- Študentskú časť sme poňali formou stávkovej hry
- Rozhodli sme sa prirobiť aj skupinové rezervácie, každý študent si môže založiť skupinu kde si demokratickým systémom zvolia spoločné cvičenie, každý prihodí do banku určité množstvo kreditu

Dohodli sme sa na:

- Zvažovali sme nadstavbu rozvrhových variant
- Zvažovali sme nadstavbu burzy cvičení
- Vypracovaní analýzy
- Výbere vývojového prostredia a technológie
- Prestavbe dátového modelu

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
2.1	Kontaktovať Ing. Lacka ohľadom konfigurácie LDAP a HTTPS	Bc. Alojz Gomola	13. 10. 2010	20. 10. 2010	Splnená
2.2	Opísanie alternatívneho riešenia	Bc. Peter Korenek	13. 10. 2010	20. 10. 2010	Splnená
2.3	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	13. 10. 2010	20. 10. 2010	Rozpracovaná
2.4	Vypracovanie zápisnice zo stretnutia a rozdelenie úloh	Bc. Marcel Baláž	13. 10. 2010	20. 10. 2010	Splnená
2.5	Návrh ě elementov pre alt. riešenie	Bc. Ján Kvak	13. 10. 2010	20. 10. 2010	Splnená
2.6	Nápady na funkcionálnosť alt. riešenia	Všetci	13. 10. 2010	20. 10. 2010	Rozpracovaná
2.7	Testovanie verzie z minulého roka	Všetci	13. 10. 2010	20. 10. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
3.1	Vypracovanie matematického modelu, príprava opisu procesu, vypracovanie zápisnice	Bc. Alojz Gomola	20. 10. 2010	27. 10. 2010
3.2	Predbežná analýza študentskej časti	Bc. Miroslav Beno	20. 10. 2010	27. 10. 2010
3.3	Predbežná analýza študentskej časti	Bc. Marcel Baláž	20. 10. 2010	27. 10. 2010
3.4	Predbežná analýza učiteľskej časti	Bc. Peter Korenek	20. 10. 2010	27. 10. 2010
3.5	Funkcionálny model, návrh vhodnej technológie	Bc. Ján Kvak	20. 10. 2010	27. 10. 2010
3.6	Predbežná analýza učiteľskej časti	Bc. Roman Meszároš	20. 10. 2010	27. 10. 2010
3.7	Vypracovanie kostry dokumentácie	Bc. Ján Kováč	20. 10. 2010	27. 10. 2010
3.8	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	13. 10. 2010	27. 10. 2010
3.9	Nápady na funkcionálnosť alt. riešenia	Všetci	13. 10. 2010	20. 10. 2010

Poznámky k stretnutiu:

Stretnutie sa nieslo v duchu jednanja zo zákazníkom, stmeľovali sme našu predstavu o systéme.

Prílohy:

–

4.4 Zápisnica z tímového stretnutia č. 4

Dátum: 27. 10. 2010
Čas: 18:00 – 20:40
Miesto: Softvérové štúdio
Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý
Členovia tímu: Bc. Marcel Baláž Bc. Miroslav Beno
 Bc. Alojz Gomola Bc. Peter Korenek
 Bc. Ján Kvak Bc. Roman Meszároš
 Bc. Ján Kováč
Ďalší: -
Moderátor: Ing. Miroslav Galbavý
Zápis: Vyhotovil: Bc. Ján Kováč
Overil: Bc. Miroslav Beno

Téma stretnutia:

Konzultácia k analýze a návrhu riešenia, príprava dokumentov na 1. kontrolný termín.

Priebeh stretnutia:

Na začiatku stretnutia priniesol vedúci tímu opoznámkovú analýzu riešenia, ktorú sme mu týždeň pred tým spísali. Vedúci tímu mal pripomienky k formulácií niektorých viet. Ďalšia vec ktorú je nutné opraviť je to, že vyučujúci aj študenti by nemali vidieť svoj počet herných kreditov (rank-hodnosť). Jedna z ďalších pripomienok sa týkala hierarchie všetkých zainteresovaných osôb používajúcich danú aplikáciu. Zrejmi sme si všetky tieto typy užívateľov – k dostupným pridáme rolu riaditeľa ústavu.

Potom sme sa venovali 1. kontrolnému dátumu – nasledujúci týždeň. Do tejto doby je nutné všetky dokumenty dať do prezentovateľnej podoby. Jedna z dôležitých častí ktorú musíme upraviť je webová prezentácia nášho tímu. Dohodli sme sa na nasledujúcej aktualizácii resp. úprave webovej stránky:

- Sprehľadnenie celej webovej prezentácie
- Pridanie plánov
- Pridanie názvu pre náš projekt
- Pridanie 4. zápisnice

Ďalšou preberanou témou bolo ujasnenie si niektorých zmien v návrhu riešenia projektu:

- Pre vyučujúcich bude dostupný žolík na riešenie kolízií v rozvrhoch.
- Vyučujúci a študenti nebudú môcť vidieť svoj ranking (počet herných kreditov).
- ☒ študentov bude čisto vážený študijný priemer (VŠP) určovať istý druh výhody oproti ostatným s horším VŠP.
- Pri kolaborácii študentov pri tvorbe rozvhov budú mať k dispozícii chat – dohodli sme sa na jednoduchom chate, kde sa budú najnovšie, aktuálne príspevky zobrazovať hore a staršie nebudú ďalej zobrazené a tým pádom ani potrebné.
- Pre vyučujúcich bude dostupné tlačítko pre jednoduché načítanie minuloročných požiadaviek.

Nakoniec sme sa rozhodli, že v ZS budeme implementovať učiteľskú časť aplikácie a v LS časť pre študentov. Následne sme si už iba pridelili úlohy na nasledujúci týždeň.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
3.1	Vypracovanie matematického modelu, prípravaopisu procesu, vypracovanie zápisnice	Bc. Alojz Gomola	20. 10. 2010	27. 10. 2010	Splnená
3.2	Predbežná analýza študentskej časti	Bc. Miroslav Beno	20. 10. 2010	27. 10. 2010	Splnená
3.3	Predbežná analýza študentskej časti	Bc. Marcel Baláž	20. 10. 2010	27. 10. 2010	Splnená
3.4	Predbežná analýza učiteľskej časti	Bc. Peter Korenek	20. 10. 2010	27. 10. 2010	Splnená
3.5	Funkcionálny model, návrh vhodnej technológie	Bc. Ján Kvak	20. 10. 2010	27. 10. 2010	Rozpracovaná
3.6	Predbežná analýza učiteľskej časti	Bc. Roman Meszároš	20. 10. 2010	27. 10. 2010	Splnená
3.7	Vypracovanie kostry dokumentácie	Bc. Ján Kováč	20. 10. 2010	27. 10. 2010	Splnená
3.8	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	20. 10. 2010	27. 10. 2010	Rozpracovaná
3.9	Nápady na funkcionálnosť alt. riešenia	Všetci	20. 10. 2010	27. 10. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
4.1	Aktualizovanie webovej stránky	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010
4.2	Vytvorenie plánov	Bc. Miroslav Beno	27. 10. 2010	3. 11. 2010
4.3	Integrovanie dokumentácie	Bc. Ján Kováč	27. 10. 2010	3. 11. 2010
4.4	Analýza študentskej časti	Bc. Marcel Baláž	27. 10. 2010	3. 11. 2010
4.5	Analýza učiteľskej časti	Bc. Peter Korenek	27. 10. 2010	3. 11. 2010
4.5	Vytvorenie hrubého návrhu herného prostredia	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010
4.6	Špecifikácia riešenia	Bc. Roman Meszároš	27. 10. 2010	3. 11. 2010
4.7	Funkcionálny model, návrh vhodnej technológie	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010
4.8	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	27. 10. 2010	3. 11. 2010
4.9	Analýza študentskej časti	Bc. Miroslav Beno	27. 10. 2010	3. 11. 2010

Poznámky k stretnutiu:

-
Prílohy:

4.5 Zápisnica z tímového stretnutia č. 5

Dátum: 3. 11. 2010

Čas: 18:00 – 20:40

Miesto: Softvérové štúdio

Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý

Členovia tímu: Bc. Marcel Baláž

Bc. Alojz Gomola

Bc. Ján Kvak

Bc. Ján Kováč

Bc. Miroslav Beno

Bc. Peter Korenek

Bc. Roman Meszároš

Ďalší: -

Moderátor: Ing. Miroslav Galbavý

Zápis: Vyhotovil: Bc. Peter Korenek

Overil: Bc. Roman Meszároš

Téma stretnutia:

Odovzdanie dokumentácie, refaktoring dátového modelu

Priebeh stretnutia:

Na začiatku stretnutia sme odovzdali podľa pokynov vytlačenú analýzu, návrh a špecifikáciu projektu. Vedúci tímu nám písomne potvrdil odovzdanie. Dohodli sme sa, že do najbližšieho stretnutia už začneme s implementáciou učiteľskej časti.

Vedúci nás upozornil na blížiaci sa termín podania prihlášky do súťaže TP Cup. Zopakoval nám, aké podmienky by mala spĺňať prihláška do tejto súťaže.

V rámci implementácie sme sa začali zaoberať postupmi pri tvorbe používateľského prostredia pomocou technológií flash a javascript.

Ďalej sme sa zaoberali konvenciami, ktoré by mali byť dodržiavané pripísaní zdrojových kódov v jazykoch PHP, flash a javascript. Zhodli sme sa, že bude vhodné, ak sa z týchto postupov vytvorí metodiky, ktoré sa budú dodržiavať pri programovaní. Ďalej sme sa dohodli, že pri evidencii úloh v nástroji .Project nebudeme používať diakritiku.

Nakoniec sme sa zaoberali nutnosťou refactoringu dátového modelu. Dôvodom sú nové funkcionálne požiadavky na aplikáciu. Ide hlavne o rozšírenie dátového modelu o študentskú časť aplikácie. Z týchto dôvodov je nutné sa zamerať na efektívnejšie a prehľadnejšie rozdelenie dát do tabuliek. Refaktoring považujeme v tejto chvíli za urgentný, pretože od dátového modelu sa bude ďalej odvíjať celá aplikácia

Na záver sme si rozdelili úlohy a dohodli termíny, do kedy treba úlohy vypracovať.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
4.1	Aktualizovanie webovej stránky	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010	Splnená
4.2	Vytvorenie plánov	Bc. Miroslav Beno	27. 10. 2010	3. 11. 2010	Splnená
4.3	Integrovanie dokumentácie	Bc. Ján Kováč	27. 10. 2010	3. 11. 2010	Splnená
4.4	Analyzovanie študentskej časti	Bc. Marcel Baláž	27. 10. 2010	3. 11. 2010	Splnená
4.5	Analyzovanie učiteľskej časti	Bc. Peter Korenek	27. 10. 2010	3. 11. 2010	Splnená
4.5	Vytvorenie hrubého návrhu herného prostredia	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010	Splnená
4.6	Špecifikácia riešenia	Bc. Roman Meszároš	27. 10. 2010	3. 11. 2010	Splnená
4.7	Funkcionálny model, návrh vhodnej technológie	Bc. Ján Kvak	27. 10. 2010	3. 11. 2010	Splnená
4.8	Analýza technológií na tvorbu pokročilých interaktívnych prvkov na webových stránkach	Bc. Roman Meszároš	27. 10. 2010	3. 11. 2010	Splnená
4.9	Analyzovanie študentskej časti	Bc. Miroslav Beno	27. 10. 2010	3. 11. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
5.1	Vytvorenie metodík na písanie zdrojových kódov	Bc. Miroslav Beno	3. 11. 2010	10. 11. 2010
5.2	Analýza možností nástrojov pre prácu s jazykom flash	Bc. Ján Kvak	3. 11. 2010	5. 11. 2010
5.3	Analýza možností nástrojov pre prácu s jazykom javascript	Bc. Ján Kováč	3. 11. 2010	5. 11. 2010
5.4	Analýza vhodnosti použitia openSource prostredí a odhad cenovej hladiny produktov	Bc. Marcel Baláž	3. 11. 2010	5. 11. 2010
5.5	Refaktoring dátového modelu	Bc. Peter Korenek	3. 11. 2010	10. 11. 2010
5.6	Refaktoring dátového modelu	Bc. Roman Meszároš	3. 11. 2010	10. 11. 2010
5.7	Refaktoring dátového modelu	Bc. Alojz Gomola	3. 11. 2010	10. 11. 2010
5.8	Analýza možnosti využitia záťažových testov pri testovaní produktu	Bc. Marcel Baláž	5. 11. 2010	10. 11. 2010
5.9	Analýza protokolu LDAP	Bc. Miroslav Beno	5. 11. 2010	10. 11. 2010
5.10	Analýza možnosti exportu dokumentov zo systému	Bc. Ján Kováč	5. 11. 2010	10. 11. 2010
5.11	Návrh používateľského prostredia	Bc. Ján Kvak	5. 11. 2010	10. 11. 2010

Poznámky k stretnutiu:

-

Prílohy:

-

4.6 Zápisnica z tímového stretnutia č. 6

Dátum: 10. 11. 2010

Čas: 18:00 – 19:45

Miesto: Softvérové štúdio

Účastníci: Vedúci pedagóg:

Ing. Miroslav Galbavý

Členovia tímu:

Bc. Marcel Baláž

Bc. Alojz Gomola

Bc. Miroslav Beno

Bc. Peter Korenek

☒ Bc. Ján Kvak
☒ Bc. Ján Kováč
Ing. Tomáš Kramár

☒ Bc. Roman Meszároš

Ďalší:
Moderátor: Bc. Alojz Gomola
Zápis: Vyhotovil: Bc. Ján Kvak
Overil: Bc. Miroslav Beno

Téma stretnutia:

Konzultácia postupu riešenia s externým poradcom, prehodnotenie použitia technológií

Priebeh stretnutia:

Externého konzultanta Ing. Tomáša Kramára sme oboznámili s doterajším postupom v projekte, s dôvodmi prečo sme sa rozhodli riešiť systém odlišne oproti predchádzajúcim tímom. Konzultant naše riešenia podrobil kritike a my sme obhajovali náš prístup aj v kontexte nášho zamýšľaného prihlásenia na TP – Cup.

Konzultant bol oboznámený s technológiami, s ktorými už pracujeme a ktoré zamýšľame použiť v konkrétnych prípadoch. Kritizoval náš pôvodný výber technológie Flash ako technológie na tvorbu rozhrania pre študentskú časť systému. Navrhol riešenie v dvojrozmernom prostredí namiesto trojrozsmernej grafiky a použitie technológie JQuery. Po diskusii sme sa rozhodli zvoliť technológiu JQuery.

Ďalej sme konzultanta oboznámili s postupmi v oblasti refaktoringu databázy, ktorý je dokončený a kde ďalšou úlohou bude zdokumentovanie.

Ako bývalý riešiteľ nás konzultant oboznámil s tým, že v systéme nebol použitý framework CakePHP ako sme boli pôvodne presvedčení.

Úložisko systému na kontrolu verzií nám odporučil upraviť a v ňom obsiahnuté súbory posunúť o adresár vyššie kvôli prehľadnosti.

Na záver sme si rozdelili úlohy a dohodli termíny, do kedy treba úlohy vypracovať.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
5.1	Vytvorenie metodík na písanie zdrojových kódov	Bc. Miroslav Beno	3. 11. 2010	10. 11. 2010	Splnená
5.2	Analýza možností nástrojov pre prácu s jazykom flash	Bc. Ján Kvak	3. 11. 2010	5. 11. 2010	Splnená
5.3	Analýza možností pre prácu s jazykom javascript	Bc. Ján Kováč	3. 11. 2010	5. 11. 2010	Splnená
5.4	Analýza vhodnosti použitia OpenSource prostredí a odhad cenovej hladiny produktov	Bc. Marcel Baláž	3. 11. 2010	5. 11. 2010	Splnená
5.5	Refaktoring dátového modelu	Bc. Peter Korenek	3. 11. 2010	10. 11. 2010	Splnená
5.6	Refaktoring dátového modelu	Bc. Roman Meszároš	3. 11. 2010	10. 11. 2010	Splnená

5.7	Refaktoring dátového modelu	Bc. Alojz Gomola	3. 11. 2010	10. 11. 2010	Splnená
5.8	Analýza možností využitia záťažových testov pri testovaní produktu	Bc. Marcel Baláž	5. 11. 2010	10. 11. 2010	Splnená
5.9	Analýza protokolu LDAP	Bc. Miroslav Beno	5. 11. 2010	10. 11. 2010	Splnená
5.10	Analýza možnosti exportu dokumentov zo systému	Bc. Ján Kováč	5. 11. 2010	10. 11. 2010	Splnená
5.11	Návrh používateľského prostredia	Bc. Ján Kvak	5. 11. 2010	10. 11. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
6.1	Úprava úložiska systému na kontrolu verzií	Bc. Ján Kvak	10. 11. 2010	17. 11. 2010
6.2	Migrácia databázy a refaktoring funkčnej časti kódu	Bc. Peter Korenek	10. 11. 2010	17. 11. 2010
6.3	Migrácia databázy a refaktoring funkčnej časti kódu	Bc. Roman Meszároš	10. 11. 2010	17. 11. 2010
6.4	Prieskum existujúcich riešení a príprava fóra pre systém	Bc. Alojz Gomola	10. 11. 2010	17. 11. 2010
6.5	Prieskum existujúcich riešení a príprava fóra pre systém	Bc. Marcel Baláž	10. 11. 2010	17. 11. 2010
6.6	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Ján Kvak	10. 11. 2010	17. 11. 2010
6.7	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Miroslav Beno	10. 11. 2010	17. 11. 2010
6.8	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Ján Kováč	10. 11. 2010	17. 11. 2010
6.9	Inštalácia systému na kontrolu verzií GIT na lokálnej úrovni	Všetci	10. 11. 2010	17. 11. 2010

Poznámky k stretnutiu:

-

Prílohy:

-

4.7 Zápisnica z tímového stretnutia č. 7

Dátum: 24. 11. 2010

Čas: 18:00 – 20:40

Miesto: Softvérové štúdio

Účastníci: Vedúci pedagóg:

Ing. Miroslav Galbavý

Členovia tímu:

Bc. Marcel Baláž

Bc. Alojz Gomola

Bc. Ján Kvak

Bc. Ján Kováč

Bc. Miroslav Beno

Bc. Peter Korenek

Bc. Roman Meszároš

Ďalší:

-

Moderátor: Bc. Alojz Gomola

Zápis: Vyhotovil:

Bc. Roman Meszároš

Overil:

Bc. Peter Korenek

Téma stretnutia:

Synchronizácia práce s databázou, vytvárania formulárov a dokumentácie.

Priebeh stretnutia:

Stretnutie sa od začiatku zaoberalo najmä tvorbou formulárov a použitím refaktorovaného modelu databázy. Vzhľadom na použitie časti existujúceho systému sme sa dohodli na nutnosti refaktoriácie existujúcich databázových volaní na nový model databázy. Z dôvodu súčasnej práce na formulároch je potrebné túto refaktoriáciu skompletizovať ešte pred ďalšími stretnutiami, vzhľadom na testovanie a použitie vo formulároch.

Vývoj formulárov bude v prvej fáze prebiehať nad starou databázou, čo vzhľadom na ponechanie pôvodného rozhrania ovládača databázy nerobí prekážku. Po refaktoriácii databázových volaní budú tieto použité a testované.

Na stretnutí sme ďalej vytvorili rámcový plán, a síce v zimnom semestri máme za úlohu tvorbu učiteľského rozhrania systému. Plán na letný semester pozostáva z 1 – 2 mesiacov implementácie študentskej časti, pričom zvyšný čas je určený najmä na ďalšie ladenie systému a návrh algoritmu na automatické generovanie rozvrhu.

Na generovanie rozvrhu môžu byť použité rôzne techniky paralelizácie. Na generovanie rozvrhov by sa dali následne použiť procesorové polia. Vzhľadom na zmenu modelu rozvrhového systému, vytváranie graficky príjemného používateľského prostredia, integrácie hry a automatické generovanie rozvrhu, nám bolo vedúcim projektu odporúčané zapojiť sa s projektom do študentskej vedeckej konferencie IIT SRC, súťaže TP Cup sa v roku 2011 tím rozvrhárov nezúčastňuje.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
6.1	Úprava úložiska systému na kontrolu verzií	Bc. Ján Kvak	10. 11. 2010	17. 11. 2010	Splnená
6.2	Migrácia databázy	Bc. Peter Korenek	10. 11. 2010	17. 11. 2010	Splnená
6.3	Migrácia databázy	Bc. Roman Meszároš	10. 11. 2010	17. 11. 2010	Splnená
6.4	Prieskum existujúcich riešení a príprava fóra pre systém	Bc. Alojz Gomola	10. 11. 2010	17. 11. 2010	Splnená
6.5	Prieskum existujúcich riešení a príprava fóra pre systém	Bc. Marcel Baláž	10. 11. 2010	17. 11. 2010	Splnená
6.6	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Ján Kvak	10. 11. 2010	17. 11. 2010	Splnená
6.7	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Miroslav Beno	10. 11. 2010	17. 11. 2010	Splnená
6.8	Návrh formulárov pre učiteľskú časť systému a použitie JQuery v systéme	Bc. Ján Kováč	10. 11. 2010	17. 11. 2010	Splnená
6.9	Inštalácia systému na kontrolu verzií GIT na lokálnej úrovni	Všetci	10. 11. 2010	17. 11. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
7.1	Správa o organizácii MVC a integrácii projektu	Bc. Alojz Gomola	24. 11. 2010	1. 12. 2010
7.2	Vytváranie grafických objektov pre použitie vo formulároch, JQuery	Bc. Ján Kvak	24. 11. 2010	1. 12. 2010
7.3	Vytváranie formulárov učiteľskej časti systému	Bc. Marcel Baláž	24. 11. 2010	1. 12. 2010
7.4	Vytváranie formulárov učiteľskej časti systému	Bc. Miroslav Beno	24. 11. 2010	1. 12. 2010
7.5	Vytváranie formulárov učiteľskej časti systému	Bc. Ján Kováč	24. 11. 2010	1. 12. 2010
7.6	Refaktoring prepojenia na databázu v existujúcom systéme	Bc. Peter Korenek	24. 11. 2010	27. 11. 2010
7.7	Refaktoring prepojenia na databázu v existujúcom systéme	Bc. Roman Meszároš	24. 11. 2010	27. 11. 2010
7.8	Pridelenie práv všetkým v GIT	Bc. Ján Kvak	24. 11. 2010	1. 12. 2010
7.9	Metodika dokumentácie	Bc. Alojz Gomola	24. 11. 2010	1. 12. 2010

Poznámky k stretnutiu:

-

Prílohy:

-

4.8 Zápisnica z tímového stretnutia č. 8

Dátum: 1. 12. 2010

Čas: 18:00 – 19:30

Miesto: Softvérové štúdio FIIT ST² BA

Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý

Členovia tímu: Bc. Marcel Baláž

Bc. Alojz Gomola

Bc. Ján Kvak

Bc. Ján Kováč

Bc. Miroslav Beno

Bc. Peter Korenek

Bc. Roman Meszároš

Ďalší:

-

Moderátor: Bc. Alojz Gomola

Zápis: Vyhotovil: Bc. Miroslav Beno

Overil: Bc. Alojz Gomola

Téma stretnutia:

Informácie o priebehu implementácie a organizovaní práce; dokumentácia

Priebeh stretnutia:

Na začiatku stretnutia sa zhodnotil postup prác na implementácii prostredníctvom rozpravy pod vedením moderátora Bc. Alojza Gomolu. Podtímy zaoberajúce sa kolaboračným prostredím a refaktoriáciou databázy oznámili skorú finalizáciu týchto častí implementácie, implementácia

rozhrania upraveného učiteľského prostredia je vo fáze rozpracovanosti. Počas rozpravy boli za účasti Ing. Galbavého diskutované niektoré pripomienky a skutočnosti:

- Úložisko GitHub, na ktorom sa nachádza aktuálna verzia systému spolu s databázou je verejný. Je potrebné ošetriť prístup k citlivým osobným dátam, napríklad presunutím na neverejný server.
- Pri prenesení požiadavky z minulého akademického obdobia, je potrebné vytvárať v databáze kópiu požiadavky, pre možnú úpravu údajov, nie iba odkaz cez väzobnú tabuľku.
- Kolaboratívne prostredie je už vo vývojovej vetve v repozitári Git a pripravené na širšie testovanie ostatnými členmi tímu.
- Pri eventuálnych požiadavkách týkajúcich sa databázy treba kontaktovať vedúceho tejto časti systému Bc. Romana Meszároša, ktorý disponuje prehľadom o implementačných detailoch, poprípade konzultovať vypracovaný opis databázových tabuliek.
- Momentálne používaný nástroj na manažment úloh dotProject nespĺňa požadované kritériá, hlavne jednoduché napojenie na repozitár, a prehľadné hlásenie chýb. Pre budúci semester sme sa zhodli na použití nástroja Redmine.
- V rámci systémovej dokumentácie je potrebné uviesť samostatnú kapitolu sumarizujúcu, čo bolo v implementovanom systéme pridané a pozmenené našim tímom od prebratia minuloročnej verzie.
- V rámci podtímov je vhodná na rozdelenie zodpovednosti a úloh technika tzv. Case Points, ktorú ozrejmil Bc. Alojz Gomola.
- Do dokumentácie riadenia zahrnieme metodiky, ktoré vypracovali jednotliví členovia tímu, pričom prejdú úpravou tak, aby zodpovedali procesom vykonávaným v našom tíme.

Na záver stretnutia boli zrekapitulované úlohy na nasledujúci týždeň a definované ich príslušné zodpovednosti, pričom boli vykonané mierne zmeny podľa stavu oblastí a pozorovaných doterajších schopností členov tímu pre zabezpečenie maximálnej miery efektivity ich využitia. Odovzdanie prototypu spolu s dokumentáciami a používateľskou prezentáciou je naplánované o dva týždne.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
-------	-------	----------	---------------	------------------	------

7.1	Správa o organizácii MVC a integrácii projektu	Bc. Alojz Gomola	24. 11. 2010	1. 12. 2010	Splnená
7.2	Vytváranie grafických objektov pre použitie vo formulároch, JQuery	Bc. Ján Kvak	24. 11. 2010	1. 12. 2010	Splnená
7.3	Vytváranie formulárov učiteľskej časti systému	Bc. Marcel Baláž	24. 11. 2010	1. 12. 2010	Rozpracovaná
7.4	Vytváranie formulárov učiteľskej časti systému	Bc. Miroslav Beno	24. 11. 2010	1. 12. 2010	Rozpracovaná
7.5	Vytváranie formulárov učiteľskej časti systému	Bc. Ján Kováč	24. 11. 2010	1. 12. 2010	Rozpracovaná
7.6	Refaktoring prepojenia na databázu v existujúcom systéme	Bc. Peter Korenek	24. 11. 2010	27. 11. 2010	Splnená
7.7	Refaktoring prepojenia na databázu v existujúcom systéme	Bc. Roman Meszároš	24. 11. 2010	27. 11. 2010	Splnená
7.8	Pridelenie práv všetkým v GIT	Bc. Ján Kvak	24. 11. 2010	1. 12. 2010	Splnená
7.9	Metodika dokumentácie	Bc. Alojz Gomola	24. 11. 2010	1. 12. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
8.1	Implementácia formulárov učiteľskej časti	Bc. Ján Kvak, Bc. Peter Korenek, Bc. Marcel Baláž, Bc. Ján Kováč	1. 12. 2010	8. 12. 2010
8.2	Testovanie kolaboratívnej časti systému	Bc. Miroslav Beno, Bc. Ján Kováč, Bc. Alojz Gomola, Bc. Roman Meszároš	1. 12. 2010	8. 12. 2010
8.3	Spracovanie dokumentácie riadenia	Bc. Miroslav Beno, Bc. Alojz Gomola	1. 12. 2010	8. 12. 2010
8.4	Finalizácia refaktoringu databázy, servis databázy	Bc. Roman Meszároš	1. 12. 2010	8. 12. 2010
8.5	Implementácia dodatočných úprav systému mimo učiteľského prostredia	Bc. Ján Kováč	1. 12. 2010	8. 12. 2010
8.6	Odovzdanie vypracovaných metodík k procesom v tíme	Všetci	1. 12. 2010	3. 12. 2010

Poznámky k stretnutiu:

-

Prílohy:

-

4.9 Zápisnica z tímového stretnutia č. 9

Dátum: 8. 12. 2010

Čas: 18:00 – 19:05

Miesto: Softvérové štúdio FIIT ST² BA

Účastníci: Vedúci pedagóg: Ing. Miroslav Galbavý

Členovia tímu: Bc. Marcel Baláž

Bc. Alojz Gomola

Bc. Ján Kvak

Bc. Ján Kováč

Bc. Miroslav Beno

Bc. Peter Korenek

Bc. Roman Meszároš

Ďalší:

-

Moderátor: Bc. Alojz Gomola

Zápis: Vyhotovil: Bc. Marcel Baláž
 Overil: Bc. Alojz Gomola

Téma stretnutia:

Aktuálny stav implementácie a krátkodobý plán jej napredovania. Príprava na prezentáciu.

Priebeh stretnutia:

Zhodnotenie aktuálneho stavu implementácie:

- Kolaboratívne prostredie – prvá verzia dokončená a funkčná. Potrebne vykonať akceptačné testy.
- Refaktoring databázy – rozpracovaný. Úloha je časovo náročnejšia, ako sa predpokladalo.
- Refaktoring formulárov – rozpracovaný. Grafické rozhrania prakticky hotové. Funguje načítavanie dát z databázy, ale ešte je potrebné dorobiť ukladanie a vykonať testovanie.

Krátkodobý plán a príprava na prezentáciu:

- Jano Kováč: do piatku vypracovať akceptačné testy na kolaboratívne. Vytvorí testovacie dáta, ktoré sa použijú pri prezentácii. Chyby z testovania nahlásiť Marcelovi Balážovi. Od piatku pracovať na integrácii dokumentácie implementácie.
- Roman Mészáros: dopracovať refaktoring databázy, aby bolo možné prezentovať vybrané časti systému. Pripraviť notebook s verziou systému odladenou na prezentovanie. Hlavne pripraviť dáta na predvedenie funkcionality. V prípade potreby pomôžu s databázou Peter Korenek alebo Marcel Baláž.
- Ján Kvak, Peter Korenek, Marcel Baláž – dokončenie refaktoringu formulárov, hlavne prepojenie s databázou. Testovanie.
- Deadline dokončenia jednotlivých implementačných častí – nedeľa 12.12.2010 12:00.
- Alojz Gomola, Miro Beno – dopracovanie dokumentácie a príprava prezentácie. Približná osnova prezentácie:
 - 8 minút úvod a opis navrhnutého systému – Hra o rozvrh
 - 3 minúty prezentácia kolaboratívneho prostredia
 - 3 minúty prezentácia refaktoringu DB
 - 3 minúty prezentácia refaktoringu formulárov

Zhodnotenie úloh z predchádzajúceho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia	Stav
8.1	Implementácia formulárov učiteľskej časti	Bc. Ján Kvak, Bc. Peter Korenek, Bc. Marcel Baláž, Bc. Ján Kováč	1. 12. 2010	8. 12. 2010	Rozpracovaná
8.2	Testovanie kolaboratívnej časti systému	Bc. Miroslav Beno, Bc. Ján Kováč, Bc. Alojz Gomola, Bc. Roman Meszároš	1. 12. 2010	8. 12. 2010	Rozpracovaná
8.3	Spracovanie dokumentácie riadenia	Bc. Miroslav Beno, Bc. Alojz Gomola	1. 12. 2010	8. 12. 2010	Rozpracovaná
8.4	Finalizácia refaktoringu databázy, servis databázy	Bc. Roman Meszároš	1. 12. 2010	8. 12. 2010	Rozpracovaná
8.5	Implementácia dodatočných úprav systému mimo učiteľského prostredia	Bc. Ján Kováč	1. 12. 2010	8. 12. 2010	Splnená
8.6	Odovzdanie vypracovaných metodík k procesom v tíme	Všetci	1. 12. 2010	3. 12. 2010	Splnená

Úlohy do ďalšieho stretnutia:

Číslo	Popis	Vykonáva	Dátum zadania	Termín ukončenia
9.1	Vytvorenie akceptačných testov kolaboratívneho prostredia	Bc. Ján Kováč	8. 12. 2010	10. 12. 2010
9.2	Vytvorenie testovacích dát pre kolaboratívne prostredie	Bc. Ján Kováč	8. 12. 2010	10. 12. 2010
9.3	Integrácia dokumentácie implementácie	Bc. Ján Kováč	8. 12. 2010	14. 12. 2010
9.4	Príprava prezentácie	Bc. Miroslav Beno, Bc. Alojz Gomola	1. 12. 2010	12. 12. 2010
8.1	Implementácia formulárov učiteľskej časti	Bc. Ján Kvak, Bc. Peter Korenek, Bc. Marcel Baláž,	1. 12. 2010	12. 12. 2010
8.2	Testovanie kolaboratívnej časti systému	Bc. Ján Kováč	1. 12. 2010	10. 12. 2010
8.3	Spracovanie dokumentácie riadenia	Bc. Miroslav Beno, Bc. Alojz Gomola	1. 12. 2010	14. 12. 2010
8.4	Finalizácia refaktoringu databázy, servis databázy	Bc. Roman Meszároš	1. 12. 2010	12. 12. 2010

Poznámky k stretnutiu:

-

Prílohy:

-

5 Štábna kultúra

V tejto kapitole sú uvedené pokyny a odporúčania za účelom zachovania konzistentnosti v distribuovanom projekte. Konkrétne aspekty štábnej kultúry sú podrobne rozpracované v podkapitolách. Zároveň sú v prílohe B pripojené špecifické metodiky vypracované pre potreby tímu Schedule of Pain jeho jednotlivými členmi, ktoré boli následne distribuované na adaptovanie ostatným členom tímu.

5.1 Písanie zdrojových kódov

V tejto kapitole sú zhrnuté zásady písania zdrojových kódov pre ich čitateľnosť a zefektívnenie, organizáciu kódov v rámci projektu a ďalšie pravidlá. Všeobecne je stanovené používanie názvoslovia pre premenné a triedy v zdrojovom kóde, dátové objekty, a ďalšie kľúčové slová, **v angličtine**, aby nevníkali skomoleniny anglických prekladov a podobne. Vhodné je dodržiavať všeobecnú terminológiu závislú od prostredia. Výnimkou je písanie komentárov do zdrojových kódov, ktoré sú písané po slovensky, opäť ale reflektujú konkrétnu zaužívanú terminológiu prostredia, a pri referencovaní používajú presné názvy napr. tried alebo modulov. Pri týchto výnimkách platí pravidlo zachovania textu **bez diakritiky**, čím sa zjednodušuje prehľadnosť a vyhľadávanie v textoch. To platí aj pri uvádzaní mien, napríklad pri autoroch zdrojových kódov uvádzaných v komentároch sa nepoužíva diakritika.

5.1.1 JavaScript

Medzery a prázdne riadky

- Základné odsadenie je dve medzery, tabulátory sa nepoužívajú.
- Riadky by nemali byť dlhšie ako 80 znakov. Pri zalamovaní dlhších riadkov sa odsadzuje text pod súvisiacu časť kódu, na ktorú nadväzuje:

```
var result = prompt(aMessage,  
                    aInitialValue,  
                    aCaption);
```

- Pri bloku súvisiacich premenných je možné zarovnanie:

```
short           = foo(bar);  
longVariable = foo(baz);
```

- Riadky by nemali obsahovať zbytočné medzery pred ukončovacím príznakom riadku.
- Binárne operátory sú oddelené od operandov medzerou, týka sa aj priradenia:

```
string = bar() + 'foo';
```

- Medzera sa vkladá za každou čiarkou a bodkočiarkou, nikdy nie pred nimi.

- Medzera sa vkladá za kľúčovými slovami napr. `if (x > 0)`.
- Na oddelenie logicky súdržných blokov kódu sa používajú jeden alebo dva prázdne riadky.
- Súbor so zdrojovým kódom je ukončený prázdny riadkom.

Znaky

- Funkčné zátvorky pri jednoriadkových funkciách sú vymedzené medzerami, výnimkou je vymedzenie bodkočiarkou alebo čiarkou:

```
function valueObject(aValue) { return { value: aValue }; };
```

- Inak sú funkčné zátvorky vždy na samostatnom riadku:

```
function toOpenWindow(aWindow)
{
    aWindow.document.commandDispatcher.focusedWindow.focus();
}
```

- Pri normálnych zátvorkách sa medzery používajú iba na oddelenie viacerých parametrov, a nie samotných znakov zátvoriek alebo bodkočiark: `for (; ;)`

- Znak bodkočiarky sa dáva za každým príkazom.
- Na vymedzenie textových konštánt sa preferujú dvojité úvodzovky pred jednoduchými.
- Zátvorky sa odsadzujú tak aby začiatočná a ukončujúca bloková zátvorka boli na rovnakej úrovni:

```
if (condition1 || condition2)
{
    action1();
}
else if (condition3 && condition4)
{
    action2();
}
else
{
    defaultAction();
}
```

- Na znaky z diakritikou napr. pre rozhranie sa používa tvar unicode konštanty `\uXXXX`.

Komentáre

Odporúča sa komentovať každý ucelený blok kódu znakmi `/* */`. Používajú sa celé vety začínajúce veľkým písmenom a končiace bodkou. Pri komentovaní konkrétnych riadkov kódu, sa uvádza komentár v rámci príslušného riadku, predchádzajú mu znaky `//`. Pre podrobnú metodiku písania komentárov možno vychádzať z rozšírenej konvencie Doxygen.

Premenné a funkcie

- Používa sa štandardný *lower camel case* spôsob pri viacslovnom pomenovaní, premenné začínajú malým písmenom. V názvoch sa nepoužívajú podtrhovníky. Konštanty sú písané veľkými písmenami.
- Vyhýbame sa používaniu skratiek v názvosloví.
- Premenné predstavujúce uzavretú množinu hodnôt (enumeration) majú prefix *k*:
`const kDisplayModeNormal = 0;`
- Globálne premenné majú prefix *g*:
`var gFormatToolBar;`
- Funkcie spracúvajúce udalosti (event handlers) by mali mať prefix *on* nasledovaný udalosťou, na ktorú reagujú ako napríklad `onLoad`, `onDialogAccept`, `onDialogCancel`.
- Lokálne premenné deklarujeme čo najbližšie k miestu ich použitia, a inicializujeme ich. Používame kľúčové slovo *var*.

JavaScript špecifiká

- Zdrojový kód nesmie generovať JavaScript varovania, spôsobené napríklad:
 - Duplicitná deklarácia premennej.
 - Vádzenie `return hodnota;` vo funkciách bez návratovej hodnoty.
 - Deklarácia objektu ukončená čiarkou.
 - Vynechanie deklarácie premenných.
- Na vytváranie polí sa používa konštrukcia `[hodnota, ...]` namiesto `new Array(hodnota, ...)`.
- Na vytváranie objektov sa používa konštrukcia `{ premenná: hodnota, ... }` namiesto `new Object()`.
- Nepoužívajú sa obalovače `new Number`, `new String`, `new Boolean`.
- Premenné *boolean* sa neporovnávajú explicitne s hodnotami *true* a *false*, namiesto toho sa používa konštrukcia `if (ioService.offline)`. Objekty sa porovnávajú s *null*, čísla s *0*, a *stringy* s `""` pomocou operátorov `===` a `!==`.
- Pri používaní `typeof` sa nepíšu zátvorky:

```
if (typeof myVariable == 'string') {  
    // ...  
}
```

DOM elementy

- Pre efektívne vytváranie HTML elementov sa nepoužíva príkaz `document.createElement()`:

```
this.popup = document.createElement('div');  
this.popup.id = 'autocomplete';
```

ale používa sa:

```
this.popup = $('<div id="autocomplete"></div>')[0];
```

- Na zistenie existencie atribútu sa nepoužíva volanie *getAttribute*, ale *hasAttribute*.
- Preferuje sa prechádzanie cez *childNodes* namiesto používania *first/lastChild* a *next/previousSibling*.
- Preferuje sa volanie *hasChildNodes()* pred testovaním *childNodes.length > 0*.
- Preferuje sa používanie *nodeName* pred *tagName*.
- Používajú sa definované DOM konštanty ako napríklad `event.keyCode == KeyEvent.DOM_VK_RETURN` namiesto `event.keyCode == 13`.

5.1.2 PHP

Pre zachovanie konzistentnosti sa riadia pravidlá pre písanie zdrojových kódov PHP dvomi východiskami:

1. Základné pravidlá sú uvedené v časti o písaní zdrojových kódov Javascriptu. Jedná sa o odsadzovanie, používania medzier a prázdnych riadkov, názvoslovie premenných a funkcií, formátovanie funkčných blokov kódu.
2. Ďalšie pravidlá špecifické pre PHP a výnimky z predchádzajúceho bodu sú prebraté z metodiky minuloročného tímu Bug Hunters, keďže sa modifikujú a dopĺňajú práve implementované výstupy minuloročného projektu.

Názvy entít

- Ak nie je uvedené inak názvy sa riadia *camelCase* konvenciou.
- Názvy tried metód a funkcií začínajú veľkým písmenom: `Class UsersController`.
- Názvy atribútov tried začínajú malým písmenom *m*, pre zabránenie konfliktu s názvami metód:

```
class NameOneTwo  
{  
    function VarAbc() {};  
    function ErrorNumber() {};  
    var $mVarAbc;  
    var $mErrorNumber;  
    var $mrName;  
}
```

- Pri názvoch premenných sú všetky písmená malé, pri viacslovných názvoch sa oddeľuje podtrhovníkom:

```
function HandleError($errorNumber)  
{
```

```

$error = new OsError;
$time_of_error = $error->GetTimeOfError();
$error_processor = $error->GetErrorProcessor();
}

```

- Názvy elementov polí sa píše rovnako ako názvy premenných, na prístup k nim sa používajú jednoduché alebo dvojité úvodzovky:

```

$myarr['foo_bar'] = 'Hello';
$element_name = 'foo_bar';
print "$myarr[foo_bar] world";
print "$myarr[$element_name] world";
print "$myarr['$element_name'] world";
print "$myarr["$element_name"] world";

```

- Názvy premenných, ktoré sú referenciami začínajú s písmenom *r*. Názvy funkcií a metód vracajúcich referenciu tiež začínať s *r*:

```

class Test
{
    var $mrStatus;
    function DoSomething(&$rStatus) {};
    function &rStatus() {};
}

```

- Názvy globálnych premenných začínajú písmenom *g*:

```

global $gLog;
global &$grLog;

```

- Názvy globálnych konštánt sú veľkými písmenami s využitím podtrhovníka:

```

define("A_GLOBAL_CONSTANT", "Hello world!");

```

- Statické premenné sa začínajú písmenom *s*:

```

function test()
{
    static $msStatus = 0;
}

```

- Pri názvoch globálnych funkcií sa používa C GN[®] konvencia – všetky písmená malé, podtrhovník – zabezpečuje rozlíšiteľnosť funkcií od metód tried:

```

function some_bloody_function()
{
}

```

Formátovanie kódu

- Pravidlá odsadenia, zátvorkovania, a používania medzier sú rovnaké ako pri písaní JavaScriptu.
- Pri podmienkach sa konštanty uvádzajú na ľavej strane testu rovnosti:

```

if ( 6 == $errorNum ) ...

```

Komentáre

- Vždy, keď je na výber z viacerých možných riešení, treba zdokumentovať prečo je zvolená daná možnosť.
- Používať formát komentárov kompatibilný so systémom phpDocumentor.

- Treba ku každej zmene alebo pridanému kódu pridať autora, čas zmeny a ID úlohy, ku ktorej sa zmena vzťahuje
- Komentovanie súborov - treba stručne uviesť obsah súboru (aké sú tu implementované triedy – mala by byť jedna trieda na jeden súbor)
- Komentovanie tried - uviesť účel triedy (na čo je trieda určená)
- Komentovanie premenných a atribútov - uviesť význam premenných a atribútov a ich očakávané typy
- Komentovanie metód - treba uviesť význam vstupných parametrov, typ a význam všetkých možných návratových hodnôt a stručne opísať čo metóda robí, aký je jej účel
- Komentovanie kódu - treba komentovať hlavne netriviálne rozhodnutia
- Ak je známy nedostatok, prípadne plánovaná zmena časti kódu, zdokumentovať plánované/navrhované zmeny v *@todo* bloku:

```
/**
function datafunction
@todo make it do something
*/
function datafunction()
{
}
```

5.2 Písanie dokumentácie implementácie

Táto podkapitola sa zaoberá formátovaním dokumentácie k implementácií, riadia sa ňou všetci, ktorí dokumentujú nejakú implementovanú časť.

5.2.1 Obsah a forma dokumentácie k implementácii

Každý funkčný inkrement systému obsahuje nasledovné položky v dokumentácii:

1. Návrh funkcionality.
2. Návrh integrácie do existujúceho systému.
3. Návrh používateľského rozhrania (formulárov).
4. Požadované dáta.
5. Opis implementácie.
6. Akceptačné testy.
7. Chyby.

Návrh funkcionality

V tejto stati opíšete navrhovanú funkcionality, v konkrétnom implementačnom prostredí. Čiže JQuery a PHP.

Identifikujete:

1. Funkčný proces – (čo? kedy? ako?)
2. Dáta vystupujúce vo funkčnom procese
3. Formuláre vyplývajúce funkčného procesu

Funkčný proces

Funkčný proces opíšete štruktúrovane:

1. Podrobným prípadom použitia
2. Podrobným diagramom aktivít

Na **prípady použitia** používajte výlučne Cockburnovu notáciu a to nasledovne:

<Meno prípadu použitia >

Predpoklady a dôsledky

<Číslovaný zoznam predpokladov>

Účastníci

<Číslovaný zoznam účastníkov>

Krátky opis

<Krátky funkcionálny opis prípadu použitia >

Základný tok{Kroky v základnom toku sú nasledovné:}

1. Používateľ vykoná operáciu
2. Systém vykoná odozvu na základe nejakých informácií
3. Ak nastane situácia pokračujeme *krokom x základného toku*
4. Ak nastane situácia pokračujeme nasledovne
 - a. Ak splnená takáto podmienka pokračujeme *krokom x základného toku*
 - b. Ak splnená takáto podmienka pokračujeme *krokom x alternatívneho toku* *Meno alternatívneho toku*
5. Ak nastane takáto situácia pokračujeme *krokom X alternatívneho toku* *Meno alternatívneho toku*, inak pokračujeme *krokom x hlavného toku*.
6. Prípad použitia končí

Alternatívne toky

Meno vedľajšieho toku

Kroky podobné ako v hlavnom toku

Diagram aktivít vypracujte vo MS Visio 2010 alebo inom obdobnom nástroji, odporúča sa však Visio.

Diagram aktivít podrobne opíšte pre každú aktivitu nasledovne:

<nadpis n-1><Aktivita></nadpis n-1>

Popis aktivity:

<krok x.><Podrobný opis kroku x. ako pri prípade použitia>

Dáta vystupujúce vo funkčnom procese

Opíšeme jednotlivé dáta na vyššej úrovni granularity.

Formuláre vyplývajúce funkčného procesu

Formuláre opíšeme v tabuľke nasledovne:

Tab. 5 Zoznam formulárov

Číslo formuláru	Názov formuláru	Nadväzujúce prípady použitia
F-...	<Nejaký názov>	<Zoznam prípadov použitia>

Pomenovanie a číslovanie formulárov je opísané v časti *Citovanie a opis formulárov*.

Návrh integrácie do existujúceho systému

V tejto stati opíšete:

1. Ktoré súbory ovplyvní inkrement .
2. Ako bude ovplyvnená funkcionálnosť systému.
3. Ktoré časti funkcionality sa musia po inkremente upraviť/odstrániť

Návrh používateľského rozhrania (formulárov)

V tejto stati opíšete formuláre:

1. Obrázok návrhu používateľského rozhrania
2. Opis jednotlivých komponentov používateľského rozhrania
3. Funkcionálne správanie jednotlivých komponentov

Implementácia nemusí nutne byť v prostredí, stačí scratch formulára. Opis funkcionality je naviazaný na komponenty, jednotlivé komponenty nemusia byť pomenované.

Požadované dáta

Tabuľka požadovaných dát z databázy:

1. Tabuľka požadovaných údajov (Read&Write)
2. Tabuľka ovplyvňovaných údajov

Tabuľky budú vyzerať nasledovne:

Tab. 6 Zoznam požadovaných údajov

Tabuľka	Stĺpec	Typ premennej
T_...	ID_....	Char[10]...

Tabuľka je pomenovaná podľa state *citov* a *t* *tbliek SQL*, stĺpec je pomenovanie stĺpca v tabuľke typ premennej je typ v implementačnom prostredí PGSQL.

Akceptačné testy

Akceptačné testy:

1. Navrhne autor zdrojového kódu funkcionality.
2. Vykonáva používateľ, respektíve, iný človek, ktorý nie je autor .
3. Píšu po implementovaní časti funkcionality.
4. Sa vykonávajú
 - a. ručne ,
 - b. automatizovane.

Formát akceptačného testu je nasledovný:

Tab. 7 Akceptačný test.

ID	?	Názov	<Názov Testu>			
Prípado použitia	Označenie UC	Úroveň splnenia testu	<Mandatory/Optional>	Autor	<Meno autora testu>	
Rozhranie	IS/ Zamestnanec/Platby					
Účel	Overenie správnej funkčnosti žiadania platby.					
Vstupné podmienky	Zákazník sa prihlási do systému a pridá žiadosť					
Výstupné podmienky	Žiadosť prebehla úspešne a bola pridaná do systému.					
Krok	Akcia	Očakávaná reakcia			Skutočná reakcia	
<A>		<C>			<D>	

1. Formulár sa vyplní podľa metodiky *citovanie implementačných prvkov v dokumentácií*
 - a. Pole <A> je číslo kroku od 1 po N
 - b. Pole akcia je opis akcie (stlačí tlačidlo)
2. Autor vyplní formulár a očakávanú reakciu
3. Tester vyplní skutočnú reakciu

5.2.2 Citovanie implementačných prvkov v dokumentácií

Správne citovanie premenných je dôležité pre ich správne prepojenie zo zdrojovým textom.

V dokumentácií vystupujú:

1. premenné,
2. funkcie,
3. objekty,
4. tabuľky SQL,
5. kód, dopyty SQL,
6. súbory
7. formuláre,

Citovanie premenných

Premenné v texte používame v spojitosti s ich hodnotu.

Príklad: keď premenná *iCavader* nadobúda hodnotu 0 znamená, že formulár je vyplnený správne.

Pravidlá citovania a pomenovania:

1. Meno premenných je rovnaké ako v implementovanom zdrojovom texte.
2. Premennú v texte zvýrazňujeme kurzívou písma (*iCavader*).
 - a. Premenná si zachováva formát písma okolitého textu.
3. Premennú v tabuľke ponechávame v formáte tabuľky.
 - a. Premenná si zachováva formát písma okolitej tabuľky (*iCavader*).

Citovanie funkcií

Funkcie v texte používame v spojitosti z ich funkčnosťou.

Príklad: funkcia `iCvader()` spočíta všetky znaky v reťazci a následne vytvorí doplnok voči počtu znakov v druhom reťazci, ten následne zobrazí cez Kernelovu maticu do indexu podobnosti textu.

Pravidlá citovania a pomenovania:

1. Meno funkcií je rovnaké ako v implementovanom zdrojovom texte.
2. Telo funkcie sa necituje.
3. Vstupné parametre sa citujú iba keď sa na ne odvolávate neskôr.
 - a. Vstupné parametre citujeme ako zoznam premenných.
 - b. Štruktúra zoznamu premenných zachováva poradie premenných
4. Funkcie v texte zvýrazňujeme kurzívou písma (`iCvader()`).
 - a. Premenná si zachováva formát písma okolitého textu.
5. Funkcie v tabuľke ponechávame v formáte tabuľky.
 - a. Premenná si zachováva formát písma okolitej tabuľky(`iCvader()`).

Citovanie objektov

Objekty v texte používame v spojitosti s ich významom.

Príklad: objekt `Cvader()` je previazaný z reálnym objektom strúhatka na ceruzky.

Pravidlá citovania a pomenovania:

1. Meno objektu je rovnaké ako v implementovanom zdrojovom texte.
2. Telo konštruktora objektu sa necituje.
3. Objekt v texte zvýrazňujeme kurzívou písma (`Cvader()`).
 - a. Premenná si zachováva formát písma okolitého textu.
4. Objekt v tabuľke ponechávame v formáte tabuľky.
 - a. Premenná si zachováva formát písma okolitej tabuľky(`Cvader()`).

Citovanie tabuliek SQL

Na tabuľku SQL sa v texte odvolávame keď potrebujeme pracovať s nejakým stĺpcom, alebo časťou tabuľky.

Príklad: stĺpec `T_Cvader/ID_Cvader` nám jednoznačne určuje pozíciu v tabuľke `T_Cvader`

Tabuľku pomenujeme nasledovne `T_<Meno tabuľky>/<Meno stĺpca>*`:

1. `<Meno tabuľky>` odráža reálne meno tabuľky v PGSQL databáze.
2. `<Meno stĺpca>` odráža reálne meno stĺpca v PGSQL databáze.

Pravidlá citovania:

1. Na tabuľku/stĺpec sa v texte odvolávame kurzívou `T_Cvader/ID_Cvader`
2. Na tabuľku/stĺpec sa v tabuľke odvolávame vo formáte tabuľky `T_Cvader/ID_Cvader`
3. Na viacero stĺpcov sa odvolávame nasledovne `T_Cvader/ID_Cvader/UCP/Name`

Citovanie kódu SQL dopytov

Čistý zdrojový kód citujeme len keď je to nevyhnutné – čistý zdrojový kód nesmie tvoriť viac ako 10% dokumentácie implementácie.

Pravidlá citovania:

1. Kód citujeme v jednoduchom orámovaní, ktoré jasne oddeľuje, zdrojový kód od okolitého textu.
2. Kód pomenujeme pod ohraničením kódu
 - a. Kód. <poradové číslo kódu> <pomenovanie časti kódu>
3. Kód citujeme písmom `IDEA CONSOLE` veľkosť 10 pt.
4. Pokiaľ je SQL dotaz zložitejší oddeľujeme ho po funkčných častiach

Citovanie a opis formulárov (Obrazovky používateľského rozhrania)

Formulár pomenovávame nasledovne:

F_<Skratka funkčnej časti, kde sa formulár nachádza>_<číslo formulára>

1. <Skratka funkčnej časti, kde sa formulár nachádza> tvoria tri veľké písmená .
 - a. prvé tri písmená anglického pomenovania funkčnej oblasti
 - b. prvé písmená funkčnej oblasti, pri viac ako trojslovnom pomenovaní
2. <číslo formulára> je trojčísle doplnené nulami zo začiatku
 - a. číslujeme od 1.
 - b. do 999

Príklad: Formulár F-TEA-001 je zobrazením učiteľových osobných časových preferencií.

Pravidlá citovania:

1. Na tabuľku/stĺpec sa v texte odvolávame kurzívou *F_TEA_001*
2. Na tabuľku/stĺpec sa v tabuľke odvolávame vo formáte tabuľky F_TEA_001

5.3 Integrácia projektu

Pre integráciu funkčných komponentov do systému využívame modulárnu vlastnosť systému. Pridanie inkrementu spočíva v pridaní tried do modelu MVC. Rozšírenie daného modelu neovplyvňuje iné komponenty. Komponenty sú striktné separované.

5.3.1 Načítanie tried

Načítanie jednotlivých tried do systému je riešené pomocou načítania na začiatku, teda nie je potrebné v súboroch nič vyžadovať znova. Zakazuje sa používať funkciu `require()` na načítanie tried.

Triedy modelu sú načítavané pomocou `/core/init_loader.php`.

Súbory formulárov, sú volané `/core/request_handler.php` metódou `render()`.

Triedy kontrolérov sú načítavané `/core/request_handler.php`.

Triedy modelu sú pomenované `<MenoTriedy>()`.

Triedy kontrolérov sú pomenované `<MenoTriedy>Controller()`.

Súbor `/index.php`:

```
require_once("core/auto_loadable.php");
```

Súbor `/core/auto_loadable.php` vyžiada z `/core`, `/model`, `/view/helpers/` všetky triedy a súbory:

```
/*  
    CamelCase konvencia pre MVC  
*/  
function __autoload($className) {  
    $className = Inflector::underscore($className);  
    if(file_exists("core/$className.php")) {  
        require_once("core/$className.php");  
    } else if(file_exists("model/$className.php")) {  
        require_once("model/$className.php");  
    } else if(file_exists("view/helpers/$className.php")) {  
        require_once("view/helpers/$className.php");  
    } else {  
        echo "Autoload failed for class $className";  
    }  
}
```

O spracovanie požiadaviek sa stará trieda `RequestHandler()` jej metóda `handle` vyvoláva jednotlivé operácie na načítanie, zobrazenie, spracovanie formulárov.

Súbor `/core/request_handler.php`, funkcia `handle`

```
public function handle() {  
    $page = htmlspecialchars($_GET['page']);  
    $page = $this->getPageIfEmpty($page);  
  
    $view = $this->pageData($page);  
  
    $this->render($view);  
}
```

Funkcia spracuje zvyšnú časť URL, a to tak, že získa meno príslušného kontrolera a načíta jeho model a vykreslí príslušný formulár.

5.3.2 Spracovanie URL

URL môžeme chápať nasledovne:

`http://<Cestá k koreňovému adresáru>/<Meno adresára v MVC>/<Meno triedy alebo adresára>/<Meno metódy alebo súboru>`

Meno adresára v MVC

Opísané je meno adresára v MVC, kde sa nachádza požadovaný formulár, takže pre jednotlivé časti je to nasledovné:

1. Viewer `/view/<adresár>`
2. Controller `/controller/<adresár>`

Meno triedy alebo adresára

Opísané je meno triedy v kontroleri, alebo adresára vo viewer-i. Môžeme si to predstaviť nasledovne:

1. Viewer `/view/<adresár>/<meno>.php`
2. Controller `/controller/<adresár>/<meno>/`

Meno metódy alebo súboru

Posledná časť adresy určuje ktorá metóda sa zavolá v kontroleri, a ktorý formulár z viewer-a sa použije:

1. Viewer `/view/<adresár>/<meno>.php:<meno metódy>(...)`
2. Controller `/controller/<adresár>/<meno>/<meno metódy>.php`
3. Model `/model/<meno>.php`

Z tohto nám vyplýva nasledovné:

Meno metódy, ktorá je volaná na obsluhu formulára je rovnaká ako pomenovanie súboru kde sa formulár fyzicky nachádza.

Príklad

URL je nasledovné `http://<cesta k koreňovému adresáru>/suggestion/dd`

Použité sú súbory a metódy

1. Model `/model/suggestion.php`
2. Viewer `/view/suggestion/dd.php`
3. Controller `/controller/suggestion.php::dd(...)`

5.3.3 Pridanie novej funkcionality

Novú funkcionality pridávame vo viacerých krokoch

A. Pridanie záznamu do tabuľky menu

Vychádzame zo zavedených konvencií, pre každú skupinu formulárov vytvoríme záznam v tabuľke menu nasledovne:

name	<Meno ktoré sa zobrazí v menu>
href	<Odkaz na uvodny formular>
group_id	<číslo z rozsahu 0-6>
poradie	<poradové číslo v menu (celkové)>

Meno ktoré sa zobrazí v menu

Sem zadávame meno, ktoré vystihuje funkcionality skupiny formulárov

Odkaz na uvodný formulár

Sem sa vloží odkaz vo formáte:

`/<skupina pouzivatel'ov>/<meno funkcionality>/<meno uvodneho formulara>`

Pri vytváraní tohto odkazu sa riadime predchádzajúcou staťou.

Číslo z rozsahu 0-6

Reprezentuje používateľskú skupinu, ktorej je skupina formulárov určená. Skupiny sú nasledovné:

Skupina	ID Skupiny
1. Administrator	0
2. APE	1
3. Garant	2
4. Teacher	3
5. Pract	4
6. Scheduler	5
7. All	7

Poradové číslo v celkovom poradí formulárov, je úplne jedno aké tam dáte, systém si to aj tak radí abecedne po skupinách, čo môže indikovať zlú podmienku zoradenia pri výbere z tabuľky, alebo nekonzistentné poradie v databáze.

B. Pridanie triedy do /model/

Triedu do modelu pridáte ako:

1. ľubovoľne pomenovaný php súbor, nachádza sa v priečinku /model/
2. Metódy budú pracovať nad databázou, iba v tomto súbore sa môžu nachádzať SQL dopyty
3. Metódy môžu byť ľubovoľne pomenované
4. Metódy, ktoré sú interné sú označené príznakom *private*

C. Pridanie súborov formulárov do /view/

Formuláre ukladáme do adresára:

`/view/<skupina pouzivatel'ov>/<meno funkcionality>/`

Pri vytváraní tohto adresára sa riadime predchádzajúcou staťou.

Mená súborov, kde sú uložené jednotlivé formuláre pomenovávame tak aby ich mená odrážali funkcionality. Len a len vo formulároch používame html tagy, odporúča sa používať funkciu *print()* alebo *echo()*

Pri tvorbe súborov, platia nasledovné zásady:

1. Metódy môžu byť ľubovoľne pomenované
2. Metódy, ktoré sú interné sú označené príznakom *private*

D. Pridanie triedy do /controller/

Triedu do kontrolera pridávame do súboru:

`/controller/<skupina pouzivatel'ov>/`

Nutné podmienky pri tvorbe triedy:

1. Triedu pomenovávame podľa *<implementovanej funkcionality>Controller*
2. Triedu ukladáme do súboru *<implementovanej funkcionality>.php*
3. Mená funkcií, ktoré priamo pracujú nad jednotlivými formulármi sú *<meno súboru formár>(..)*
4. Metódy, ktoré sú interné sú označené príznakom *priváte*

Do tela triedy je potrebné pridať nasledovný riadok:

Deklarujeme, že dedí od triedy ApplicationController

```
class <meno funkc.>Controller extends ApplicationController
```

Do tela triedy je potrebné pridať nasledovný riadok:

```
protected $access = array('<trieda používateľa>');
```

Známe triedy používateľov, zadáva sa celé slovo s ohľadom na veľkosť písma:

1. Administrator
2. APE
3. Garant
4. Teacher
5. Pract
6. Scheduler
7. All

V triede deklarujeme konštruktor, v konštruktore deklarujeme jednotlivé premenné, vytvoríme prislúchajúci objekt z triedy modelu.

Príklad triedy typu Controller

Predkladáme príklad viac formulárovej triedy, telá funkcií sú vynechané, lebo nie sú podstatné .

```
<?php
class SuggestionsController extends ApplicationController
{
    protected $access = array('Admin');

    function __construct()
    {
        parent::__construct();
        Existuje príslušna trieda modelu v /model/suggestions/

        $this->suggestions = new Suggestions();
    }

    Existuje príslušný formulár /view/admin/suggestions/index.php

    /**
     * Pripravi data pre pohľad na zoznam pripomienok.
     */
    function index(){...}

    /**
     * Pripravi data na pohľad pre editáciu pripomienky.
```

```
* @param $id - id pripomienky
*/
function edit($id){...}
```

Existuje príslušný formulár /view/admin/suggestions/saveedited.php

```
/**
 * Upravi danu pripomienku - z pohľadu admina, cize upravi spatnu
vazbu
 * k pripomienke.
 * @param $id - id pripomienky
 */
function saveEdited($id){...}
```

Existuje príslušný formulár /view/admin/suggestions/delete.php

```
/**
 * Vymaze danu pripomienku.
 * @param $id - id pripomienky
 */
function delete($id){...}
```

?>

Príloha A – Ponuky

Profil tímu



Bc. Marcel Baláž

Absolvent bakalárskeho štúdia *FIIT STU* v odbore *Informatik*, ktoré ukončil úspešnou obhajobou bakalárskej práce na tému: *QEMU ako plugin pre webový prehliadač*.

V práci používal prostredie *Visual Studio*, jazyk *C++*, technológia *ActiveX*, *GUI* v *MFC*

V štúdiu pokračuje na rovnakej fakulte v odbore *Informačné systémy*.

S viac ako dvojročnou praxou v oblasti *automatizovaného testovania softvéru* pracuje s *UIAutomation frameworkom*, ovláda jazyk *Ironpython*, ktorý sa používa na testovanie.

Má základné skúsenosti s *Linux*, *MySQL*, *PHP*. Je vynikajúci v riešení technologických záležitostí.



Bc. Miroslav Beno

Absolvent bakalárskeho štúdia *FIIT STU* v odbore *Informatik*, ktoré ukončil úspešnou obhajobou bakalárskej práce na tému: *Podpora správneho zpisovania bibliografických odkazov*.

V štúdiu pokračuje v odbore *Informačné systémy*. Venuje sa predovšetkým *Návrhu interakcie človeka s počítačom* a *Programovaniu*.

Základné algoritmické znalosti získal programovaním v jazykoch *Pascal*, *Delphi* a počas štúdia nadobudol skúsenosti s jazykmi a technológiami *Java*, *C/C++*, *SQL*, a vývojovými a modelovacími prostrediami najmä od spoločnosti *IBM* - zúčastnil sa na seminároch *IBM Software Summer Camp*.



Bc. Alojz Gomola

Absolvent bakalárskeho štúdia *FIIT STU* v odbore *Informatik*, ktoré ukončil úspešnou obhajobou bakalárskej práce na tému: *Metódy určenia optimálnej štruktúry modelov s veľkými neuronovými sieťami*

V danej práci sa zaoberal *evolučným prístupom k prerezávaniu neuronovej siete*, tento aplikoval na zložitejšie modely tepelných procesov. Prácu spracoval v prostredí *Matlab*, niektoré časti v jazyku *C/C++*. Má záujem o modelovanie a simuláciu procesov, najmä v oblasti *hybridného modelovania*.

V štúdiu pokračuje v odbore *Softvérové inžinierstvo*.



Bc. Peter Korenek

Absolvent bakalárskeho štúdia FIIT STU v odbore Informatika, ktoré ukončil úspešnou obhajobou bakalárskej práce na tému: *Modelom riadená kompozícia služieb*

Promoval s vyznamenaním *Magnam Cum Laude*, v štúdiu pokračuje v odbore *Softvérové inžinierstvo*.

Má skúsenosti s integráciou služieb, generovaním kódu pre platformu *Apache Camel* z ML diagramov.

Ovláda na vysokej úrovni jazyk *Java*, najmä prostredie *NetBeans*, na dobrej úrovni ovláda *C/C#*.



Bc. Ján Kvak

Absolvent bakalárskeho štúdia FIIT STU v odbore *Informatica*, ktoré ukončil úspešnou obhajobou bakalárskej práce na tému: *Klasťové algoritmy pre paralelné prostredie*.

V práci dbal na efektívnosť kódu a multiprocessing, prácu vypracoval v jazyku *C/C++*, ktorý ovláda na veľmi vysokej úrovni.

V štúdiu pokračuje v odbore *Softvérové inžinierstvo*, už dva roky pracuje na vývoji *počítačových tomografů a röntgenů* pre *Technickú univerzitu Zvolen*, má skúsenosti s reálnymi projektmi väčšieho rozsahu.

Ďalej ovláda *HTML, PHP, MySQL, CSS, JavaScript* na dobrej úrovni.



Bc. Roman Meszároš

Absolvent bakalárskeho štúdia FIIT STU v odbore *Informatica*. V štúdiu pokračuje v odbore *Softvérové inžinierstvo*. Bakalársky projekt bol realizovaný v rámci medzinárodnej súťaže *Imagine Cup*.

S daným projektom sa zúčastnil *celosvetového finále vo Vrošve* (Poľsko).

Počas štúdia pracoval 2 roky ako vývojár s Oracle, PL/SQL Developerom, Power Builderom.

Odborné znalosti: *C, C#, .Net, Java, Oracle s PL/SQL Developerom, ProLog*.



Bc. Ján Kováč

Absolvent bakalárskeho štúdia na Fakulte prírodných vied univerzity Konštantína Filozofa v Nitre, ktoré ukončil bakalárskou prácou na tému: *Tvorba modelov pomocou produktu SIMULINK*.

Programovať začal vo svojom voľnom čase v období, keď študoval na strednej škole a to najprv v programovacom jazyku *Pascal* a neskôr v *Delphi*. Taktiež ovláda programovací jazyk *C#* a platformu *.NET*. Menšie skúsenosti má aj s databázami a jazykom *SQL*.

Ponuka na tému 19 - Model používateľa pre jeho identifikáciu

Motivácia

Identifikácia používateľa je súčasťou väčšiny webových portálov, rozsiahlych programových systémov i operačných systémov. Veľká časť týchto bezpečnostných opatrení je realizovaná formou zadávania prihlasovacieho mena a hesla. Tento spôsob vyhovoval dlhé roky, avšak už existujú viaceré techniky, ktoré umožnia dostať sa k týmto tajným údajom. Tiež sa po prihlásení totožnosť používateľa už neoveruje.

Začína sa hľadať ďalší spôsob, ako dodatočne overiť, či človek, ktorý sa prihlasuje do systému, je naozaj ten, za koho sa vydáva. Existujú spôsoby na identifikáciu osôb, založené na fyzikálnych princípoch, ako sú odtlačky prstov alebo skenovanie tváre. Tieto prístupy však vyžadujú rozličné technické prostriedky, čo môže byť finančne nákladné. Vhodnou alternatívou sú softvérové riešenia, ktoré umožňujú rozpoznať používateľa na základe jeho správania a biometrických charakteristík. Najnovšie výskumy dokazujú, že je možné rozpoznať používateľa na základe nameraných charakteristík, ako sú rýchlosť stlačenia klávesy, rýchlosť dvojkliku na myši, rýchlosť pohybu myšou, uhol tohto pohybu a iné. Tieto charakteristiky sú postačujúce pre rozpoznanie dvoch používateľov, ktorí zadávajú rovnaké prihlasovacie údaje.

Táto téma náš tím veľmi zaujala, je výzvou využiť naše doterajšie skúsenosti a poznatky. Naším cieľom je vytvoriť systém, ktorý uľahčuje identifikáciu používateľa a zvyšuje bezpečnosť tohto procesu. Sme presvedčení, že hĺbkový výskum tejto problematiky nám pomôže zdokonaľiť efektivitu systému tak, aby náš výsledný produkt mal výrazný prínos v tejto oblasti. Motiváciou pre nás je aj čiastočne výskumný charakter tejto témy, ktorá je novátorská, nakoľko zatiaľ existuje iba niekoľko prototypov. V našom tíme máme programátorov ovládajúcich väčšinu najpoužívanejších programovacích jazykov, schopných analytikov i návrhárov. Naše heslo je vytvárať softvér kvalitne, bezpečne a rýchlo.

Koncept riešenia

V tejto časti opíšeme, čo by náš produkt mal obsahovať, budeme sa venovať použitým technológiám a tiež rôznym ťažkostiam, ktoré by pri tvorbe produktu mohli nastať.

Použitie:

- ochrana koncového používateľa pred neautorizovaným prístupom,
- kontrola prítomnosti a identifikácia používateľov pri PC.

Funkcionálne požiadavky:

- zachytávanie biometrických údajov o používatelovi,
- vyhodnocovanie a identifikácia používateľa,
- notifikácie, blokovanie,
- sledovanie jednotlivých študentov pri rôznych počítačoch v miestnosti.

Sledované biometrické charakteristiky:

- rýchlosť dvojkliku myši, pohybu myši, rýchlosť pohybu myši voči prejdenej vzdialenosti
- dĺžka trvania stlačenia klávesy, rýchlosť písania
- chybovosť (klávesa backspace, delete), štýl písania bodiek, čiarok a medzier, dĺžky slov, používanie diakritiky, numerickej klávesnice.

Výstupom projektu je knižnica použiteľná na identifikáciu používateľa na základe biometrických charakteristík a tiež aplikácia, ktorá demonštruje jej použitie.

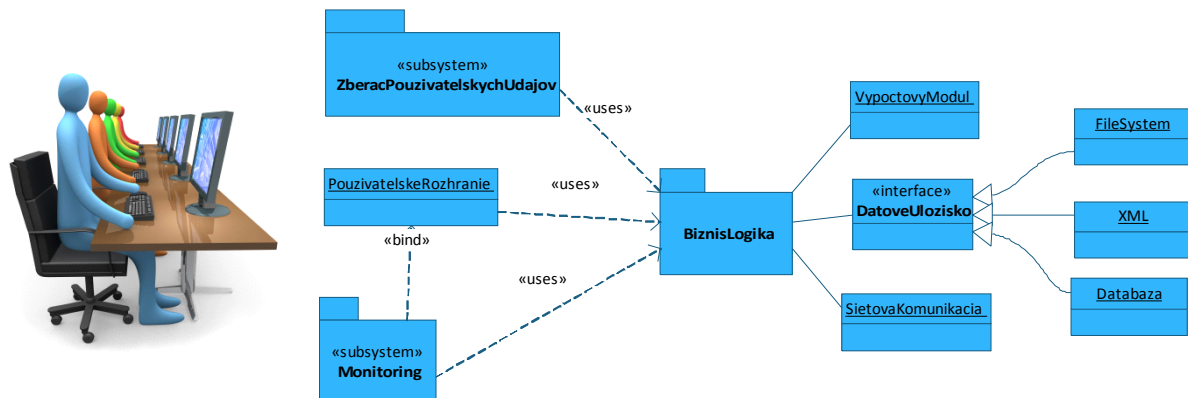
Aplikácia je použitá pri prihlasovaní sa a sleduje aj aktivitu počas behu operačného systému alebo inej aplikácie vo webovom prehliadači. Aplikácia vyvoláva aktivitu (upozornenie, zamknutie obrazovky), keď deteguje zmenu používateľa.

Na otestovanie aplikácie použijeme:

- používateľov, ktorých štatistiky, budú odosielané do našej databázy na vyhodnotenie,
- skupinu študentov, ktorí sa učia strojopis a ich štýl písania sa rýchlo vyvíja.

Na Obr. 1 sa nachádza schéma systému. Tento pozostáva z hlavného modulu, kde prebiehajú výpočty a samotná identifikácia (Výpočtový modul). Biznis logika ďalej zahŕňa úložisko dát a tiež sieťovú komunikáciu pre potrebu klient-server aplikácie.

Komunikácia s používateľom prebieha cez používateľské rozhranie, údaje o používateli sú získavané pomocou zberača používateľských údajov. Sledovanie prítomnosti používateľa a jeho porovnávanie s požadovaným používateľom je zabezpečované komponentom monitoring.



Obr. 1 Schéma komponentov systému

Na strane serveru používame IIS s webovými službami WCF, databáza je MS SQL alebo ORACLE. Výpočtový modul sa nachádza na serveri.

Klient používa .NET a identitu overuje cez volania webových služieb. Projekt je napísaný v jazyku C#, ktorý je najpoužívanejším v spojení s .NET a zároveň je výhodou jeho podobnosť s jazykmi C a Java, ktoré vyhovujú celému tímu.

Ponuka na tému 3 - Evolučný simulátor umelého života založený na heuristických pravidlách

Motivácia

Umelý život je zaujímavá časť umelej inteligencie. V dnešnej dobe sa čím ďalej tým viac uplatňuje evolučný princíp riešenia problémov. Umelý život patrí medzi otvorené témy vedeckého výskumu, zaujme svojim konceptom, ale aj možným prínosom pre spoločnosť. Pre pochopenie nášho komplexného sveta môžeme využiť svet umelý, ktorý je zjednodušeným modelom reálneho sveta.

Jedná sa o model reálneho sveta. Vystupujú v ňom zvieratá, ktoré majú definované správanie sadou produkčných pravidiel. Tieto zvieratá sa snažia prežiť a rozmnožiť v svete obmedzených zdrojov (zdroj energie –potrava). Zvieratá sa vyvíjajú a časom by sme mali dospieť k dokonalým jedincom, ktorí sú schopní v danom svete prežiť, populácia by sa mala ustáliť (svet dosiahne stabilný stav).

Experimentovanie s takýmto modelom je veľmi otvorená záležitosť, do daného sveta môžeme vniesť viac ako jeden druh zvierat, prípadne druh potravy. Príkladom môžu byť mäsožravce, ktoré získavajú energiu z bylinožravcov. Takisto potrava bylinožravcov (kapusta) a jej kvantita v svete sa môže meniť.

Takisto koncept energie zvieratá je veľmi otvorený a vidíme široké možnosti jeho využitia. Model nám môže pomôcť lepšie porozumieť vývoju v našom svete a budeme na ňom môcť sledovať rôzne udalosti.

Daný projekt sa zameriava na vývoj simulátora takéhoto sveta, jeho logickej časti (back end) a grafickej vizualizácie sveta (front end). Projekt je zaujímavý najmä dôrazom na efektívnosť kódu, ktorú náš tím považuje za výzvu. Pre efektívne fungovanie takéhoto sveta považujeme za dôležité jeho veľkosť. Aby sme mohli dané výsledky zovšeobecniť, potrebujeme netriviálnu veľkosť sveta. Takisto za výzvu pokladáme paralelizmus (simuláciu viacerých svetov súčasne).

Koncept riešenia

Naša vízia riešenia predstavuje vytvorenie aplikácie, ktorá umožní daný svet simulovať, zobrazovať a ovplyvňovať jeho vývoj. Samozrejme predpokladáme, že náš systém bude *plne parametrizovateľný*.

Daný systém bude pozostávať z dvoch častí:

1. Používateľské rozhranie – zabezpečuje komunikáciu s používateľom, prezeranie sveta
2. Logickej časti – zabezpečuje simuláciu daného sveta/svetov

Logická časť, bude pracovať v dvoch režimoch:

1. Interaktívny – Spolupracuje s používateľským rozhraním. Má možnosť preberať od používateľského rozhrania pokyny, simulácia sveta je ovplyvniteľná z používateľského prostredia. Ako príklad uvádzame vloženie potravy do sveta, odobratie jedinca zo sveta a pod.
2. Pasívny – Nespolupracuje s používateľským rozhraním. Na začiatok dostane dávkový súbor, kde má určené parametre jednotlivých simulácií svetov (1-n svetov). Tento mód podporuje

paralelné spracovanie svetov, je veľmi rýchly, lebo nie je spomaľovaný používateľským prostredím. Výstupom takéhoto režimu je n. záznamov o simulácií.

Používateľské rozhranie by malo poskytovať nasledovné služby:

1. Simulácia umelého sveta:
 - a. **Parametrizácia simulácie** - nastaviť veľkosť sveta, harmonogram potravy, počítačový stav jedincov, podmienky zastavenia simulácie, typ výstupu simulácie
 - b. **Prehranie záznamu simulácie** - pozastavenie, pretočenie záznamu simulácie, sledovanie vybranej časti sveta, sledovanie vybraného jedinca a jeho správanie sa počas života.
 - c. **Živá simulácia** - živá simulácia sveta s možnosťami zásahu do daného sveta, možnosť nastavenia trvania dĺžky jedného kroku.
 - d. **Štatistika populácie/potravy**

2. Správu záznamov simulácie: Načítanie, spracovanie, export, príprava dávkových súborov.

Technická špecifikácia riešenia

Vzhľadom na požiadavku, aby bol projekt čo najviac otvorený, sme sa rozhodli použiť platformu C++/Java a vzhľadom na potrebu efektívneho výpočtu sa skôr prikláňame k riešeniu logickej časti v C++.

Pre moderný prístup k riešeniu GUI zvažujeme použiť nástroj *QT Creator* na vytvorenie používateľského rozhrania.

V C++/C#/Java sú implementované silné generátory náhodných čísel, čo pokladáme za dôležité.

Ponuka na tému 7 - Tréner mentálnych schopností

Motivácia

Trénovanie mentálnych schopností a zručností je dôležité pre väčšinu oblastí ľudského snaženia. Rozvíjaním svojich mentálnych schopností môžeme skvalitniť nielen našu prácu, ale celý náš život. Príroda má evolúciou vytvorený spôsob, ako sa organizmy, schopné učiť sa, učia. Týmto spôsobom je hra. Väčšiny cicavcov môžeme pozorovať, že všetky základné veci, potrebné pre život, sa naučia v mladom veku hrou. Človek sa však od tohto prirodzeného spôsobu odklonil a v procese učenia sa obmedzil na samotné podávanie hotových informácií, ktoré sa musia študenti jednoducho naučiť. Pritom väčšinu mladých ľudí intuitívne stále priťahujú skôr hry, lebo tie sú prirodzeným spôsobom, ako sa niečo naučiť. Pri mentálnych zručnostiach a schopnostiach to platí dvojnásobne.

Prvým dôležitým krokom pre rozvíjanie mentálnych schopností je nutné zistiť a konkretizovať, na ktoré schopnosti sa chceme zamerať, pretože je veľa schopností, ktoré vyžadujú rôzny prístup. Či už je to sústredenie, pamäť, obrazové vnímanie, matematická logika, rýchlosť reakcie, rozhodovanie pod tlakom alebo jazykové zručnosti. Každá z týchto oblastí vyžaduje rozličný prístup k jej rozvíjaniu.

Väčšinu z nich je možné rozvíjať pomocou počítačových hier, logických hádaniek či testov špeciálne určených na tento účel.

Väčšinou však nie je také ľahké získať prístup k takýmto hrám či testom. Veľké množstvo serverov s online hrami ponúka hry pod kolonkou logické, to však vôbec neznamená, že sú to hry rozvíjajúce logiku. Mnohokrát sú tieto hry jednotvárne a viac o mechanickom opakovaní niekoľkých úkonov, ako by nútili človeka rozmyšľať, či sústrediť sa. Zároveň sa dá pozorovať, že princíp hier sa opakuje, mení sa len príbeh, ktorým je hra spríjemnená.

Navyše väčšina dostupných hier je cudzokrajná, takže neposkytuje vysvetlivky či samotnú hru v slovenskom jazyku, čo môže mať najmä pri slovných hrách za následok ich nehrateľnosť pre našinca, ktorý daný jazyk neovláda.

Ďalším cieľom je vytvoriť celený a prehľadný systém hier, ktorý by dokázal rozvíjať jednotlivé schopnosti a zručnosti by bol nie len neoceniteľným pomocníkom pre školy a zamestnania, ale snáď aj príjemnou a užitočnou zábavou pre chvíle voľného času.

Koncept riešenia

Pri realizácii nášho riešenia sa v prvom rade zameriame na analýzu existujúcich riešení. Dôraz budeme kladieť na pochopenie modelov a procesov prebiehajúcich pri tréningu mentálneho mozgového aparátu. Táto tematika zasahuje aj do oblastí psychológie a neurológie, z ktorých budeme môcť čerpať.

Pri návrhu systému počítame s profilovaním jednotlivých používateľov na základe dosiahnutých výsledkov v rôznych typoch úloh. Tieto výsledky nám poskytnú významnú informáciu o oblastiach, ktoré je potrebné rozvíjať ďalej a budeme môcť odporučiť používateľovi nasledujúcu množinu úloh rozvíjajúcich práve tieto oblasti. Týmto vytvoríme personalizovaný tréningový program pre zdokonaľovanie individuálnych mentálnych schopností.

Dôležitou súčasťou koncepcie samotných tréningových úloh bude teda merateľnosť, čiže schopnosť vyhodnotiť výsledok v kontexte zdravého mentálneho vývoja. Vďaka tomu je zároveň možné motivovať tréningovaný subjekt k ďalšiemu progresu v rámci tréningového programu.

Čo sa týka konkrétnych úloh, určite budeme čerpať aj z už overených zdrojov, pričom prispôbíme implementáciu týchto úloh nášmu systému s ohľadom na už spomínanú merateľnosť a profilovanie používateľa. Dôraz bude kladený aj na pútavú audiovizuálnu prezentáciu úloh. Ďalšou ambíciou bude aj zapojenie kreativity používateľa, aby sa rozvíjali všetky oblasti mozgu a prípadne aj relaxačné aktivity, predstavujúce doplnok ku zvyčajne realizovaným funkciám podobných systémov.

Samozrejmosťou bude lokalizácia systému do slovenského jazyka, čo sa bude týkať nielen používateľského rozhrania, ale aj slovných úloh.

Pri podobnom type systému je možné uvažovať nad spracovaním aplikácie buď do podoby komplexného webového portálu s registráciou a štatistikami používateľov alebo do personalizovaného standalone programu prístupného napríklad v mobilných zariadeniach. Súčasťou analýzy pri tvorbe projektu bude zhodnotenie a výber z týchto dvoch smerovaní. Sme otvorení

viacerým rôznym platformám, frameworkom a jazykom, naše konkrétne skúsenosti sme vymenovali v tímovom profile.

Výsledok nášho snaženia podrobíme komplexnému testovaniu na vzorke používateľov so zdokumentovaním ich potenciálneho pokroku.

Príloha ponuky A - Zoradenie všetkých tém podľa priority

Poradové číslo témy	
Číslo témy	
1.	19. Model používateľa pre jeho identifikáciu
2.	3. Evolučný simulátor umelého života založený na heuristických pravidlách
3.	7. Tréner mentálnych schopností
4.	14. Dizajn s použitím obohatenej reality
5.	12. Správa študentských projektov na fakulte
6.	10. Portál pre časopis
7.	17. Simulated Car Racing Competition 2011
8.	5. Tvorba rozvrhov
9.	1. Objektové úložisko dát
10.	9. Prispôsobiteľný Widget
11.	13. Vyhľadávanie a sprístupnenie citácií
12.	4. Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení
13.	18. 3D grafická podpora vyhľadávania znalostí v dokumentoch
14.	2. Adaptívny proxy server
15.	11. RoboCup tretí rozmer
16.	6. Interaktívna vizualizácia grafových štruktúr v 3D priestore
17.	8. Virtuálna FIIT
18.	15. Crowdsourcing verejných dát
19.	16. Imagine Cup 2011: Game Design

Príloha ponuky B - Aktuálny rozvrh všetkých členov tímu

	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00
Pondelok														
Marcel Baláž						PDT								
Miroslav Beno						ZK								
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														
Utorok														
Marcel Baláž														
Miroslav Beno														
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														
Streda														
Marcel Baláž														
Miroslav Beno														
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														
Štvrtok														
Marcel Baláž														
Miroslav Beno														
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														
Piatok														
Marcel Baláž														
Miroslav Beno														
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														
Sobota														
Marcel Baláž														
Miroslav Beno														
Alojz Gomola														
Peter Korenek														
Ján Kvak														
Roman Meszároš														
Ján Kováč														

Legenda

Voľný čas	Prednáška	Nepovinné	Cvičenie	Povinné
-----------	-----------	-----------	----------	---------

Príloha B – Použité metodiky

Slovenská technická univerzita

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

Manažment úloh prostredníctvom nástroja dotProject

Metodika

Miroslav Beno

Študijný program: Informačné systémy

Predmet: Manažment projektov softvérových a informačných systémov

Cvičiaci: Ing. Michal Tvarožek

Ak. rok: 2010/11

1. Úvod

Účelom tejto metodiky je zvýšenie kvality v distribuovanom softvérovom projekte zadefinovaním jednotných pravidiel pre procesy súvisiace s manažmentom úloh. Metodika je určená na vykonávanie v rámci projektov podobného rozsahu a charakteru ako v predmete Tímový projekt na Fakulte informatiky a informačných technológií na Slovenskej technickej univerzite. Dokument je členený na dve časti podľa úrovne procesov, ktoré opisuje. Vyššia úroveň je obsiahnutá v kapitole 2, ktorá poskytuje komplexný pohľad na problematiku manažmentu úloh, pričom sú tu podrobne uvedené identifikované roly, zodpovednosti, a jednotlivé procesy spojené s manažmentom úloh, spolu s poradím ich vykonávania. Nižšiu úroveň predstavuje kapitola 3, ktorá definuje konkretizovaný postup vytvorenia, naplánovania a pridelenia úlohy v rozhraní nástroja dotProject, spolu s dodatočnými informáciami potrebnými na realizovanie tohto postupu.

1.1. Definícia pojmov v doméne dokumentu

- *Distribuovaný softvérový projekt* – jedná sa o projekt, na ktorého realizácii sa zúčastňuje niekoľkopočetný tím manažérov a vývojárov, pričom časti tejto realizácie prebiehajú paralelne, s rozdelenými zodpovednosťami medzi členmi tímu
- *Úloha* – dekomponovaná časť realizácie softvérového projektu v určenej miere granularity; problém z definovanými vstupmi, požadovanými výstupmi, ktorý treba vyriešiť súborom špecifikovaných činností
- *dotProject* - voľne dostupný podporný nástroj vo forme webovej aplikácie, zefektívňujúci manažment úloh v distribuovanom softvérovom projekte

1.2. Zoznam zdrojov

1. DotProject [online]. 2010 [cit. 2010-11-10]. Dostupné z WWW: <<http://www.dotproject.net/>>.
2. DotProject Version 2.0.2 Basic User Manual [online]. 2007-11-14 [cit. 2010-11-10]. Dostupné z WWW: <<http://sourceforge.net/projects/dotproject/files/dotProject%20.x%20Manuals/>>.
3. DotProject Documentation Site [online]. 2010 [cit. 2010-11-10]. Dostupné z WWW: <http://docs.dotproject.net/index.php?title=Main_Page>.

2. Proces manažmentu úloh

Manažment úloh sa zaoberá organizovaním a pridelením variabilného množstva času a ľudských zdrojov pre činnosti realizujúce softvérový projekt; medzi základné patria implementácia alebo vytváranie softvérových artefaktov. Pre tieto činnosti sa vyhodnocuje ich náročnosť a užitočnosť pre

celkový projekt. Na základe analýzy týchto údajov sa následne pristupuje k formulovaniu konkrétnych úloh a ich atribútov.

Manažment úloh sa opiera o prehľadnú evidenciu prebiehajúcich a plánovaných úloh, evidenciu zodpovednosti za ich vykonanie, a suplementárne pracuje so štatistikami o úspešnosti úloh a vyťažnosti členov tímu. Je úzko spätý s manažmentom plánovania v projekte, keďže formulované a pridelené úlohy s definovaným časovým horizontom predstavujú plán projektu s nízkou mierou granularity. Preto je nutné pri vykonávaní tejto metodiky, ktorá sa nezaobera stratégiou určovania časového rozsahu úloh, koordinovať postupy v súčinnosti s metodikou plánovania v projekte.

2.1.Roly

V tejto podkapitole sú vymenované a definované jednotlivé roly (Tab. 1) zúčastňujúce sa procesu manažmentu úloh spolu s príslušnými zodpovednosťami. V závislosti od veľkosti tímu je prípustné zastávanie viacerých rolí jednou fyzickou osobou. Vylúčené je ale zjednotenie rolí R3 a R4, aby sa predišlo zaujatosti a prípadnému konfliktu záujmov.

Tab. 1 Prehľad rolí zúčastňujúcich sa procesu manažmentu úloh.

Rola	Názov	Kapitola
R1	Vedúci projektový manažér	2.1.1
R2	Manažér vývoja	2.1.2
R3	Zadávateľ úloh	2.1.3
R4	Zodpovedný vykonávateľ úloh	2.1.4

2.1.1. R1: Vedúci projektový manažér

Projektový manažér zabezpečuje plánovanie a koordináciu tímu. V rámci procesu manažmentu úloh je zodpovedný za kontrolovanie vykonávania úloh a vyhodnocovanie štatistík a analýz o plnení úloh, na základe ktorého posudzuje výkonnosť a efektívnosť členov tímu. Taktiež verifikuje či úlohy zodpovedajú rámcovému plánu projektu ktorý vypracoval, a navrhuje ich reorganizáciu ak tomu tak nie je. Komunikuje priamo s manažérom vývoja a zodpovednými vykonávateľmi úloh pri ich hodnotení.

2.1.2. R2: Manažér vývoja

Manažér vývoja identifikuje a formuluje jednotlivé úlohy, ktorých splnenie realizuje stanovený rámcový plán projektu. Pri formulovaní jednotlivých úloh je zodpovedný za to, aby predstavovali ucelené samostatné celky s primeranou mierou granularity. Určuje podrobné atribúty úlohy technického charakteru.

2.1.3. R3: Zadávateľ úloh

Zadávateľ úloh analyzuje zoznam úloh vytvorených manažérom vývoja. Priradzuje úlohám odhadnutý časový rozsah a množstvo ľudských zdrojov. Musí si pri tom udržiavať prehľad o počte a charakteristikách jednotlivých členoch tímu. Vychádza aj z časového ohraničenia na určené súbory úloh. Na základe týchto vstupov prideliuje zodpovednosti konkrétnym vykonávateľom úloh na realizáciu s príslušným naplánovaním. Na starosti má evidenciu a kategorizáciu úloh tak, aby k nej mohli pristupovať všetky zainteresované osoby. Reaguje na spätnú väzbu od vykonávateľov úlohy, a reviduje už pridelené úlohy, pričom mení parametre časového rozsahu, alebo počet ľudských zdrojov. Táto pozícia si vyžaduje dobré analytické schopnosti a orientáciu v prostredí tímu.

2.1.4. R2: Zodpovedný vykonávateľ úloh

Jedná sa o člena tímu, ktorý je oboznámený o pridelení úlohy, za ktorej vykonanie je zodpovedný. Vychádza z presných informácií od zadávateľa úloh. Pracuje na splnení úlohy a počas jej plnenia eviduje odhadovaný progres. Poskytuje spätnú väzbu zadávateľovi úloh, ak zistí, že daná úloha vyžaduje viac času. Komunikuje aj s manažérom vývoja, ak zistí, že daná úloha vyžaduje dodatočnú špecifikáciu. Vykonávatelia úloh sú často združení v skupinách, v rámci ktorých spolupracujú na plnení úlohy.

2.2. Definícia procesu

V tejto podkapitole sú identifikované a opísané podprocesy (Tab. 2) tvoriace celkový proces manažmentu úloh.

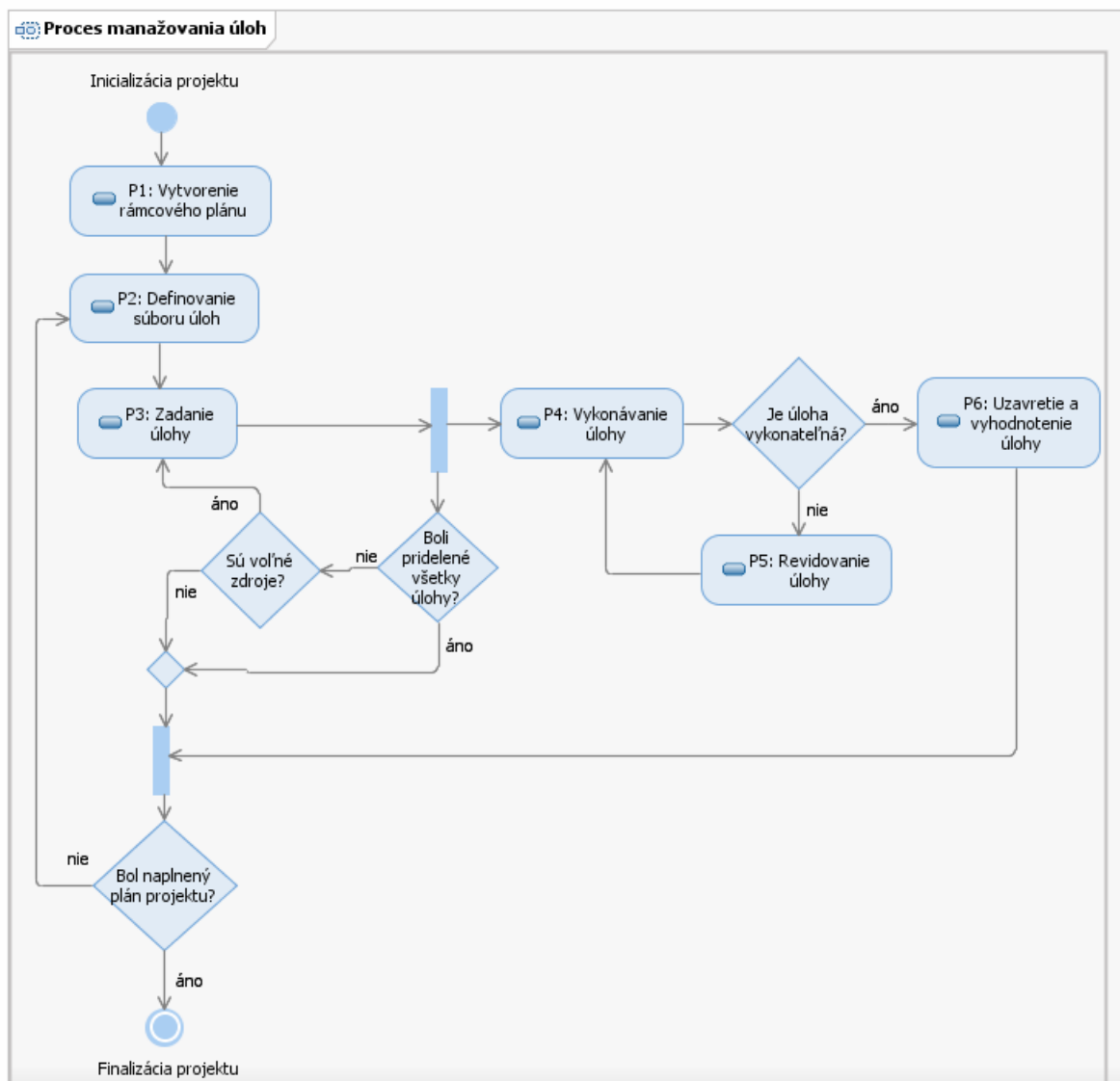
Tab. 2 Prehľad podprocesov manažmentu úloh.

Podproces	Názov	Kapitola
P1	Vytvorenie rámcového plánu	2.2.1
P2	Definovanie súboru úloh	2.2.2
P3	Zadanie úlohy	2.2.3
P4	Vykonávanie úlohy	2.2.4
P5	Revidovanie úlohy	2.2.5
P6	Uzavretie a vyhodnotenie úlohy	2.2.6

Manažment úloh sa aplikuje na všetky etapy celého životného cyklu softvéru. Preto môžeme abstrahovať od konkrétnych častí projektu a vyjadriť jeho postup prostredníctvom jednotlivých úloh, ktoré ho realizujú. Na Obr. 1 je zobrazený diagram aktivít (reprezentované sú šiestimi podprocesmi), ktorý ukazuje ich tok a nadväznosti v kontexte celého projektu. Po inicializácii a vytvorení rámcového plánu (P1) sú zostavené tri cykly podprocesov realizujúcich tento plán:

- Definuje sa súbor úloh (P2). Nové úlohy sa vytvárajú až po ukončení úloh z predošlého cyklu alebo uplynutí stanoveného termínu.
 - Úlohy sú zadané a pridelené vykonávateľom (P3). Pridelovanie prebieha pokým sú dostupné zdroje.
 - Úloha sa vykonáva (P4).
 - Úloha sa v prípade potreby reviduje (P5). Revízií môže byť viacero.
 - Úloha je ukončená a vyhodnotená (P6).

Po uzatvorení všetkých úloh a naplnení plánu sa projekt alebo jeho stanovená fáza ukončuje.



Obr. 1 Diagram aktivít (podprocesov) procesu manažmentu úloh v kontexte projektu.

2.2.1. P1: Vytvorenie rámcového plánu

Vstupy: Inicializácia projektu, špecifikácia projektu

Výstupy: Rámcový plán projektu

Zainteresované osoby: Vedúci projektový manažér

Opis: Vedúci projektový manažér vytvára rámcový plán projektu, v ktorom určí časový rozsah, ktoré časti projektu a v akom poradí sa budú realizovať, spolu s kontrolnými bodmi a prislúchajúcimi povinnosťami. Plán pokrýva dlhšie časové obdobia, pričom jednotka granularita je daná mesiacmi respektíve týždňami. Rámcový plán je potrebný ako základ na definovanie konkrétnych úloh.

2.2.2. P2: Definovanie súboru úloh

Vstupy: Časť rámcového plánu

Výstupy: Zoznam definovaných úloh

Zainteresované osoby: Manažér vývoja, vedúci projektový manažér

Opis: Manažér vývoja identifikuje konkrétne úlohy potrebné na splnenie určenej časti plánu. Následne ich formuluje v podobe textu obsahujúcom aj technické detaily spolu výstižným názvom úlohy. Ku každej úlohe identifikuje:

- špecifikované vstupy a výstupy úlohy
- kategorizácia úlohy
- priorita úlohy
- hierarchia úlohy
- závislosť úlohy
- termín vykonania úlohy:
 - do nasledujúceho stretnutia tímu alebo
 - špecifický časový úsek

Výsledkom podprocesu sú úlohy so zadanými metadátami technickej povahy z pohľadu vývoja projektu. Podrobný postup pre nástroj dotProject na vytvorenie jednej úlohy je opísaný v kapitole 3.2.1. Súbor úloh verifikuje vedúci projektový manažér.

2.2.3. P3: Zadanie úlohy

Vstupy: Zoznam definovaných úloh, informácie o členoch tímu

Výstupy: Zaevidované pridelené úlohy

Zainteresované osoby: Zadávateľ úloh, manažér vývoja

Opis: Podproces predstavuje jadro manažmentu úloh, zaoberá sa pridelovaním úloh na vykonanie.

Zadávateľ úloh analyzuje súbor úloh. Pre každú úlohu odhaduje:

- potrebný časový rozsah na vykonanie
- potrebné množstvo ľudských zdrojov

Následne priraduje k úlohe členov tímu na vykonanie, pričom berie do úvahy:

- Technickú špecializáciu
- Aktuálnu vyťaženosť
- Dostupnosť
- Doterajší výkon

Nejasnosti a odhady, ktorými si nie je istý, zadávateľ úloh konzultuje s manažérom vývoja. Výsledné údaje sumarizuje, eviduje a zverejní v podpornom nástroji. Podrobný postup pre nástroj dotProject je opísaný v kapitole 3.2.2.

2.2.4. P4: Vykonávanie úlohy

Vstupy: Pridelená špecifikovaná úloha

Výstupy: Spracovaná úloha

Zainteresované osoby: Zodpovedný vykonávateľ úloh

Opis: Zodpovedný člen tímu, alebo koordinovaná skupina členov, realizuje pridelenú úlohu podľa jej špecifikácie. Počas realizácie sa eviduje počet hodín vykonávania, a z neho priamo odvodený percentuálny progres.

2.2.5. P5: Revidovanie úlohy

Vstupy: Pridelená špecifikovaná úloha

Výstupy: Revidovaná úloha

Zainteresované osoby: Zodpovedný vykonávateľ úloh, zadávateľ úloh (R3), manažér vývoja (R2)

Opis: Počas, alebo pred realizáciou konkrétnej úlohy môže vykonávateľ identifikovať problémy, ktoré vedú k revidovaniu atribútov úlohy, alebo revidovaniu úlohy samotnej. Súhrn možných scenárov revízie je uvedený v Tab. 3.

Tab. 3 Prehľad revízií úloh

Problém	Reviduje	Akcia
Prekročenie časového rozsahu	R3	• Rozšírenie časového rozsahu a/alebo • Pridelenie ďalších ľudských zdrojov
Nedostatok ľudských zdrojov	R3	
Neschopnosť riešenia úlohy	R3	Pridelenie úlohy inému členovi
Odmietnutie úlohy	R3	
Nezmyselná / už realizovaná úloha	R2	Zrušenie úlohy
Nedostatočná / nezrozumiteľná technická špecifikácia úlohy	R2	Došpecifikovanie / modifikovanie úlohy

2.2.6. P6: Uzavretie a vyhodnotenie úlohy

Vstupy: Spracovaná úloha

Výstupy: Uzavretá alebo prenesená úloha, hodnotenie úlohy

Zainteresované osoby: Manažér vývoja, vedúci projektový manažér

Opis: Úloha sa po spracovaní alebo uplynutí termínu môže nachádzať v jednom zo stavov:

- S1: Zrušená – pri revidovaní úlohy prišlo k zisteniu, že jej vykonanie nie je žiaduce
- S2: Splnená – úloha bola úspešne vykonaná
- S3: Vypršaná – úloha nebola do daného termínu pridelená kvôli nedostatku zdrojov; neuzatvára sa, ale presúva sa do ďalšieho súboru úloh
- S4: Oneskorená – úloha je v štádiu rozpracovanosti; neuzatvára sa, ale presúva sa do ďalšieho súboru úloh
- S5: Neúspešná – úloha nebola úspešne vykonaná; neuzatvára sa, ale presúva sa do ďalšieho súboru úloh; manažér vývoja prehodnocuje špecifikáciu úlohy, jej dôležitosť; súčasný vykonávateľ je vyradený z množiny budúcich potenciálnych vykonávateľov úlohy

Vedúci projektový manažér hodnotí výsledky a spôsob spracovania úlohy, od jej špecifikácie až po realizáciu, pričom má možnosť vyvodiť dôsledky voči zodpovedným členom tímu. Podľa údajov o spracovaní konkrétnej úlohy vedie sumárnu evidenciu o úspešnosti a postupe práce na projekte.

3. Manažment a evidencia úloh v nástroji dotProject

3.1. Konfigurácia procesu v dotProject

Proces manažmentu úloh v dotProject predpokladá predchádzajúce vytvorenie entít projektu so zadanými metadátami:

- Spoločnosť realizujúca projekt

- Projekt a priradenú zodpovednosť
- Kontá členov tímu s kontaktmi a rolami

Pri procese manažmentu úloh v dotProject treba dbať ďalej aj na problematiku opísanú v nasledujúcich podkapitolách.

3.1.1. Prispôbitelnosť nástroja dotProject

DotProject je koncipovaný ako modulárny a prispôbitelný nástroj, podporujúci individuálne požiadavky lokálneho projektu. V Tab. 4 sú zhrnuté jednotlivé aspekty, ktoré ovplyvňujú rozhranie a funkcionality nástroja, a uvedené odporúčané nastavenia, ktoré predpokladajú aj postupy uvedené v tejto metodike. Pre konfiguráciu nastavení, ktoré nie sú dostupné z rozhrania dotProject treba kontaktovať administrátora, alebo osobu zodpovednú za nasadenie nástroja.

Tab. 4 Prehľad prispôbitelných aspektov dotProject.

Prispôbitelný aspekt	Ovplyvňuje	Spôsob prispôbenia	Odporúčané nastavenie
Štýl rozhrania	Vizuálnu reprezentáciu a rozmiestnenie prvkov rozhrania	Upravenie nastavení používateľa	<i>Default Clean Style</i>
Moduly	Pridáva aplikácie ponúkajúce prídavnú funkcionality	Upravenie systémovej administrácie, položka <i>Modules</i>	Aktívne: <i>Companies, Projects, Tasks, Calendar, Files, Contacts, Forums, Tickets, User Administration, System administration</i>
Prístupové práva	Možnosť modifikovať atribúty projektov a úloh	Upravenie používateľskej administrácie, položka <i>User Type</i>	Podľa Tab. 5
Lokalizácia rozhrania	Texty v rozhraní	– Upravenie nastavení používateľa – Doinštalovanie jazykového balíčka	Anglický jazyk
Záložky	Zobrazenie formulárov organizovaných v záložkách	– Upravenie nastavení používateľa – Prepnutie prístupné pri každom formulári	

3.1.2. Roly a typy používateľov

Tab. 5 zobrazuje mapovanie identifikovaných rolí v procese manažovania úloh na typy používateľov, ktoré podporuje nástroj dotProject.

Tab. 5 Mapovanie rolí na typ používateľov dotProject.

Rola	Typ používateľa
R1: Vedúci projektový manažér	Director
R2: Manažér vývoja	Manager
R3: Zadávateľ úloh	Supervisor
R4: Zodpovedný vykonávateľ úloh	Employee

3.2. Formulácia a zadanie úlohy v dotProject

Nasledujúci postup predstavuje implementáciu rámcových podprocesov P2 (Definovanie súboru úloh – kapitola 2.2.2.) a P3 (Zadanie úlohy – kapitola 2.2.3.) do špecifických podmienok prostredia dotProject. DotProject neposkytuje oddelené rozhranie pre tieto dva podprocesy. Najskôr sa realizuje podproces P2, pričom manažér vývoja definuje iba vybrané atribúty úloh. Následne sa uskutoční pridelenie a naplánovanie úloh (P3) zadáním ďalších príslušných atribútov úloh.

3.2.1. Vytvorenie úlohy

Vytvorenie úlohy vyvolá manažér vývoja kliknutím na tlačidlo *New Task* v rozhraní projektu (*View Project*). Atribúty úlohy sa zadávajú prostredníctvom formulára, ktorého štruktúru možno vidieť na obrázku 2. Nasleduje opis jednotlivých atribútov:

Názov (*Task name*) – Musí byť jasný, predmetný (obsahuje podstatné meno) a charakterizujúci požadovanú činnosť (obsahuje sloveso). Voči ďalším úlohám musí byť názov vymedzujúci.

Stav úlohy (*Status*) – Pri zadávaní úlohy sa vždy nastaví na aktívny.

Postup (*Progress*) – Pri naplánovanej úlohe sa vždy nastaví na 0%.

Priorita (*Priority*) – Určí sa na základe nasledujúcich kritérií:

- Vysoká (*high*)
 - úlohy týkajúce sa integrálnych súčastí systému ovplyvňujúce veľké množstvo funkcionality
 - úlohy vyžadujúce zvýšené množstvo pozornosti zo strany manažéra vývoja a vedúceho projektového manažéra
- Nízka (*low*)
 - úlohy realizujúce suplementárnu funkcionality systému
 - úlohy nie priamo vyplývajúce so špecifikácie projektu
- Normálna (*normal*) – všetky ostatné úlohy

Cieľom nie je rovnomerné rozdelenie priorít úloh – väčšina úloh má mať priradenú normálnu prioritu, ostatné priority sú používané striedmo.

Míľnik (*Milestone*) – Úloha bez časového rozsahu predstavujúca kontrolný bod projektu. Vyplýva z rámcového plánu projektu. Spravidla je označený vysokou prioritou a má definované závislosti od iných štandardných úloh.

Majiteľ (*Task Owner*) – Nastaví sa na zadávateľa úloh (R3), ktorý bude ďalej s úlohou v nástroji pracovať.

Typ úlohy (*Task type*) – Podľa nasledujúcich kritérií:

- *Administrative* – vytváranie podpornej dokumentácie

- *Operative* – implementácia
- *Unknown* – ostatné úlohy

Prístup (*Access*) – Nastaví sa na vždy na *protected* – prístup k úlohám majú všetci zainteresovaní na projekte.

Rozpočet (*Target Budget*) – Neuvádza sa.

Hierarchia úlohy (*Task Parent*) – Ak je úloha čiastkovou úlohou s vysokou granularitou, priradí sa k nej príslušná nadradená úloha (predpokladá jej predošlé vytvorenie). Používa sa maximálne jednoúrovňová hierarchia.

Opis úlohy (*Description*) – Musí byť dostatočne podrobný a jasný, so špecifikovanými vstupmi a požadovanými výstupmi úlohy.

Dátum ukončenia (*Finish Date*) – Termín ukončenia úlohy:

- do nasledujúceho stretnutia tímu alebo
- špecifický časový úsek

Výber prebieha kliknutím na ikonu kalendára a vyznačením dňa v novootvorenom okne. Je možné zadať v rámci dátumu aj konkrétny čas termínu.

Závislosti úlohy (*Dependencies*) – Ak je vykonanie úlohy závislé na splnení inej úlohy (alebo viacerých úloh), priradí sa jej jednoduchá závislosť (nedefinuje sa typ závislosti – implicitne je daná závislosť koniec-štart). Priradenie sa vykoná vybratím príslušných úloh z ľavej tabuľky podformulára závislosti a kliknutím na tlačidlo „>“. Ostatné ovládacie prvky ostávajú v štandardnom nastavení.

Formulár sa odošle na spracovanie tlačidlom *Save* v pravom dolnom rohu.

3.2.2. Pridelenie a naplánovanie úlohy

Po analýze súboru úloh pristupuje zadávateľ úloh k prideleniu a naplánovaniu úloh. V rozhraní projektu treba vybrať *Organize tasks* a následne konkrétnu úlohu. Formulár sa vyvolá kliknutím na *Edit task*. Nasleduje opis vyplňaných atribútov:

Dátum začatia (*Start Date*) – Naplánovaný čas začiatku vykonávania úlohy. Výber prebieha kliknutím na ikonu kalendára a vyznačením dňa v novootvorenom okne. Je možné zadať v rámci dátumu aj konkrétny čas termínu.

Doba trvania úlohy (*Expected Duration*) – Zadá sa odhadnutý čas pridelený na vykonanie úlohy v hodinách. Jednotky dní sa nepoužívajú.

Ľudské zdroje (*Human Resources*) – Pridelenie zodpovedných vykonávateľov úlohy sa vykoná vyznačením mien v ľavej tabuľke podformuláru a kliknutím na tlačidlo „>“. V prípade viacerých zodpovedných vykonávateľov sa necháva podiel 100% všetkým zúčastneným, iba vo výnimočných situáciách je možné definovať odlišný podiel na určenie hierarchie zodpovednosti (vedúci v rámci skupiny vykonávateľov má priradené 100%).

Notifikácia e-mailom – Zodpovední vykonávatelia budú automaticky notifikovaní e-mailovou správou (zaškrtnutie tlačidla *Notify Assignees of Task by Email*). Súčasťou správy, ak je to potrebné, je aj dodatočný text od zadávateľa zadaný v poli *Additional Email Comments*.

Formulár sa odošle na spracovanie tlačidlom *Save* v pravom dolnom rohu.

Schedule of Pain dotProject.net
FREE SOFTWARE

Companies | Projects | Tasks | Calendar | Files | Contacts | Forums | Tickets | User Admin | System Admin - New Item -

Welcome Miroslav Beno Help | My Info | **Todo** | Today | Logout

Add Task

tasks list : [view this project](#)

Project: System na podporu tvorby rozvrhov

Task Name *	Status <input type="text" value="Active"/>	Priority * <input type="text" value="normal"/>
<input type="text"/>	Progress <input type="text" value="0"/> %	Milestone? <input type="checkbox"/>

* indicates required field

Details

Task Owner: <input type="text" value="Beno, Miroslav"/> Access: <input type="text" value="Public"/> Web Address: <input type="text"/> Task Parent: <input type="text" value="None"/>	Task Type: <input type="text" value="Unknown"/> <input type="button" value="Select contacts..."/> Target Budget: € <input type="text"/>
Description: <div style="border: 1px solid #ccc; height: 100px;"></div>	

Dates

Start Date: <input type="text" value="19.11.2010"/> <input type="button" value="calendar"/>	<input type="text" value="11"/> : <input type="text" value="15"/>
Finish Date: <input type="text"/> <input type="button" value="calendar"/>	<input type="text" value="17"/> : <input type="text" value="00"/>
Expected Duration: <input type="text" value="1"/> hours	Daily Working Hours: 8.0
Calculate: <input type="button" value="Duration"/> <input type="button" value="Finish Date"/>	Working Days: Mon, Tue, Wed, Thu, Fri

Dependencies

Dependency Tracking

On
 Off
 Dynamic Task
 Do not track this task

All Tasks: <div style="border: 1px solid #ccc; padding: 2px; min-height: 100px;"> Vytvorenie webu timu Studium minuloročnej dokumentácie Nasadenie lokálnych snapshotov Analýza podporných nástrojov Nasadenie systému v SS2 Testovanie minuloročného výstupu Analýza alternatívneho riešenia Udržba aktualnosti webu Analýza študentskej časti Analýza učiteľskej časti </div>	Task Dependencies: <div style="border: 1px solid #ccc; height: 100px;"></div>
--	---

Set task start date based on dependency

Human Resources

Human Resources: <div style="border: 1px solid #ccc; padding: 2px; min-height: 100px;"> Baláž, Marcel Beno, Miroslav Gomola, Alojz Guest, Guest Korenek, Peter Kováč, Ján Kvak, Ján Mészáros, Roman </div>	Assigned to Task: <div style="border: 1px solid #ccc; padding: 2px; min-height: 100px;"> Beno, Miroslav [100%] </div>
<input type="button" value=">"/> <input type="text" value="100%"/> <input type="button" value="<"/>	
Additional Email Comments: <div style="border: 1px solid #ccc; height: 100px;"></div>	

Notify Assignees of Task by Email

* indicates required field

Obr. 2 Formulár na pridávanie úlohy v nástroji dotProject.

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Metodika manažmentu verzií zdrojového kódu v programe GIT

Ján Kvak

Študijný program: Softvérové inžinierstvo
Predmet: Manažment projektov softvérových a informačných systémov
Školský rok: 2010/2011

1. Úvod

Účelom tejto metodiky je poskytnúť návod jednotlivým členom tímu vyvíjajúceho softvérový produkt, najmä projektovému manažérovi, programátorom, a manažerom kvality, pomocou ktorého by dokázali efektívne pracovať so systémami na kontrolu verzií. Konkrétne táto metodika opisuje prácu so systémom na kontrolu verzií GIT počas celého životného cyklu softvérového projektu so širším opisom niektorých konkrétnych postupov.

2. Slovník použitých pojmov

- *Systém na kontrolu verzií* – systém umožňujúci efektívne manažovať zmeny v životnom cykle softvérového produktu, konkrétne zmeny v zdrojovom kóde, dokumentácií, dátach a akýchkoľvek iných používaných súboroch.
- *GIT* – voľne dostupný, open-source, distribuovaný systém pre kontrolu verzií vyvinutý Linusom Torvaldsom.
- *commit* – operácia, ktorou používateľ dáva najavo systému na kontrolu verzií, že jeho zmeny sú finálne a chce ich sprístupniť ostatným používateľom.
- *staging area* – prostredie úrovni. Pre vloženie úboru (*commit*) je dôležité súbor pridať do prostredia úrovni. GIT tým umožňuje pridávanie väčšieho množstva malých zmien súboru radšej ako pridanie veľkej zmeny naraz.

3. Súvisiace metodiky

[1] Metodika inštalácie systému na kontrolu verzií GIT

[2] Metodika vytvorenia a konfigurácie úložiska

[3] Metodika sledovania histórie zmien

[4] Metodika práce v úložisku

4. Manažment verzií

Používateľské roly v manažmente verzií

V procese tvorby softvérového produktu vystupujú rôzni používatelia. Podľa úloh, ktoré vykonávajú ich môžeme rozdeliť do niekoľkých používateľských rolí.

Rola	Zodpovednosť za
Projektový manažér	<ul style="list-style-type: none">• celý projekt• rozdelenie úloh a činností• plánovanie a organizáciu
Manažér kvality	<ul style="list-style-type: none">• kvalitu projektu• kvalitu dokumentácie• odstraňovanie chýb
Dokumentarista	<ul style="list-style-type: none">• zdokumentovanie projektu• organizácia príspevkov
Programátor	<ul style="list-style-type: none">• tvorba zdrojového kódu• oprava chýb• zmeny existujúcich programov
Tester	<ul style="list-style-type: none">• kontrola výstupu• odhaľovanie chýb
Integrátor pomocného nástroja	<ul style="list-style-type: none">• inštalácia systému na kontrolu verzií• konfigurácia systému• údržba systému• správa systému

Procesy

Pri práci so systémami na kontrolu verzií sa dá identifikovať niekoľko procesov.

Poradie	Proces	Kapitola
1.	Inštalácia systému na kontrolu verzií	4.2.1
2.	Vytvorenie a inicializácia úložiska	4.2.2
3.	Vykonanie zmien v projekte	4.2.3
4.	Sledovanie projektu	4.2.4

4.2.1 Inštalácia systému na kontrolu verzií

Podnety: začiatok práce so systémom na kontrolu verzií

Vstupy: inštalračný balík

Výstupy: systém inštalovaný na lokálnej úrovni

Zodpovedný: integrátor pomocného nástroja

Aby bolo možné so systémom na kontrolu verzií pracovať, je dôležitá jeho prvá inštalácia na lokálnej úrovni. Vykoná ju integrátor pomocného nástroja a je podrobne opísaná v metodike [1].

4.2.2 Vytvorenie a inicializácia úložiska

Podnety: umožnenie zdieľania práce medzi vývojármi

Vstupy: inštalovaný systém na kontrolu verzií na lokálnom počítači

Výstupy: vytvorené úložisko

Zodpovedný: integrátor pomocného nástroja

Pred začatím práce v systéme na kontrolu verzií je nutnosťou vytvorenie úložiska. Distribuované systémy na kontrolu verzií síce umožňujú, aby pracovník robil všetky zmeny lokálne a aj bez pripojenia na internet, ale pre väčší tím je nutné, aby existovalo jedno centrálné úložisko. Jeho vytvorenie je zodpovednosťou integrátora pomocného nástroja, ktorý musel najprv nainštalovať na lokálnom počítači systém na kontrolu verzií (jeho inštaláciou sa zaoberá metodika [1]). V niektorých prípadoch je potrebné pre použitie úložiska inicializovať ho prvou verziou projektu. Podrobne sa týmto procesom zaoberá metodika [2].

4.2.3 Vykonanie zmien v projekte

Podnety: inicializácia úložiska prvou verziou projektu

Vstupy: inštalovaný systém na kontrolu verzií, vytvorené úložisko

Výstupy: úložisko inicializované a pripravené na prácu vývojárov

Zodpovedný: programátor, dokumentarista

Keď je úložisko inicializované, môžu do neho vývojári, dokumentaristi a všetci ostatní členovia tímu pristupovať. Úložisko sa používa na verziovanie zdrojového kódu, dokumentácie a všetkých ostatných súborov, používaných v procese tvorby softvérového produktu. Táto metodika sa zaoberá postupom, ako vývojári pristupujú do centrálného úložiska, vytvárajú si jeho lokálne kópie, vykonávajú zmeny a vkladajú zmeny do centrálného úložiska.

4.2.4 Sledovanie projektu

Podnety: manažment zmien, manažment kvality, projektový manažment, testovanie
Vstupy: inštalovaný systém na kontrolu verzií, úložisko obsahujúce projekt
Výstupy: informácie o stave projektu, informácie o kvalite, odchytenie chýb
Zodpovedný: projektový manažér, manažér kvality, tester

Vzhľadom na fakt, že samotný vývoj celého projektu prebieha v systéme na kontrolu verzií, je tento dôležitým zdrojom informácií pre projektového manažéra a manažéra kvality. Je zdrojom metrík, pomocou ktorých môže projektový manažér sledovať postup projektu, príspevky jednotlivých vývojárov a plnenie projektového plánu. Systém na kontrolu verzií poskytuje prístup k celej histórii zmien všetkých súborov a informácie o príspevkoch jednotlivých vývojárov či dokumentaristov.

5. Vykonanie zmien v projekte pomocou systému na kontrolu verzií GIT

Táto kapitola sa zaoberá procesom vykonávania zmien v súboroch softvérového projektu v systéme na kontrolu verzií GIT. Systém na kontrolu verzií spravuje všetky súbory súvisiace s projektom, ktoré vytvárajú rôzni používatelia, táto časť sa bude však venovať konkrétne manažmentu verzií zdrojového kódu programátorom.

5.1 Systém na kontrolu verzií GIT

Git je voľne dostupný, open-source, distribuovaný systém pre kontrolu verzií vyvinutý Linusom Torvaldsom. Lokálne vetvy používané jednotlivými vývojármi spájajú rovnakým spôsobom ako vo vnútri vetvy. Na prístup k jednotlivým úložiskám využíva buď účinný Git protokol alebo obyčajný HTTP protokol. Systém Git podporuje nelineárny vývoj a efektívne pracuje s veľkými projektmi.

5.2 Kroky procesu vykonávania zmien v projekte

Kroky tohto procesu je možné vykonať dvoma spôsobmi. Buď použitím príkazov v prostredí GIT Bash alebo v grafickom prostredí GIT GUI. Popíšeme si obidva spôsoby.

V tomto procese sa dajú identifikovať tieto kroky:

Poradie	Názov kroku	Kapitola
1.	Vytvorenie lokálneho úložiska	5.2.1
2.	Klonovanie úložiska	5.2.2
3.	Pridanie zmenených súborov	5.2.3
4.	Kontrola rozdielov oproti poslednej verzií	5.2.4
5.	Vloženie novej verzie do úložiska	5.2.5

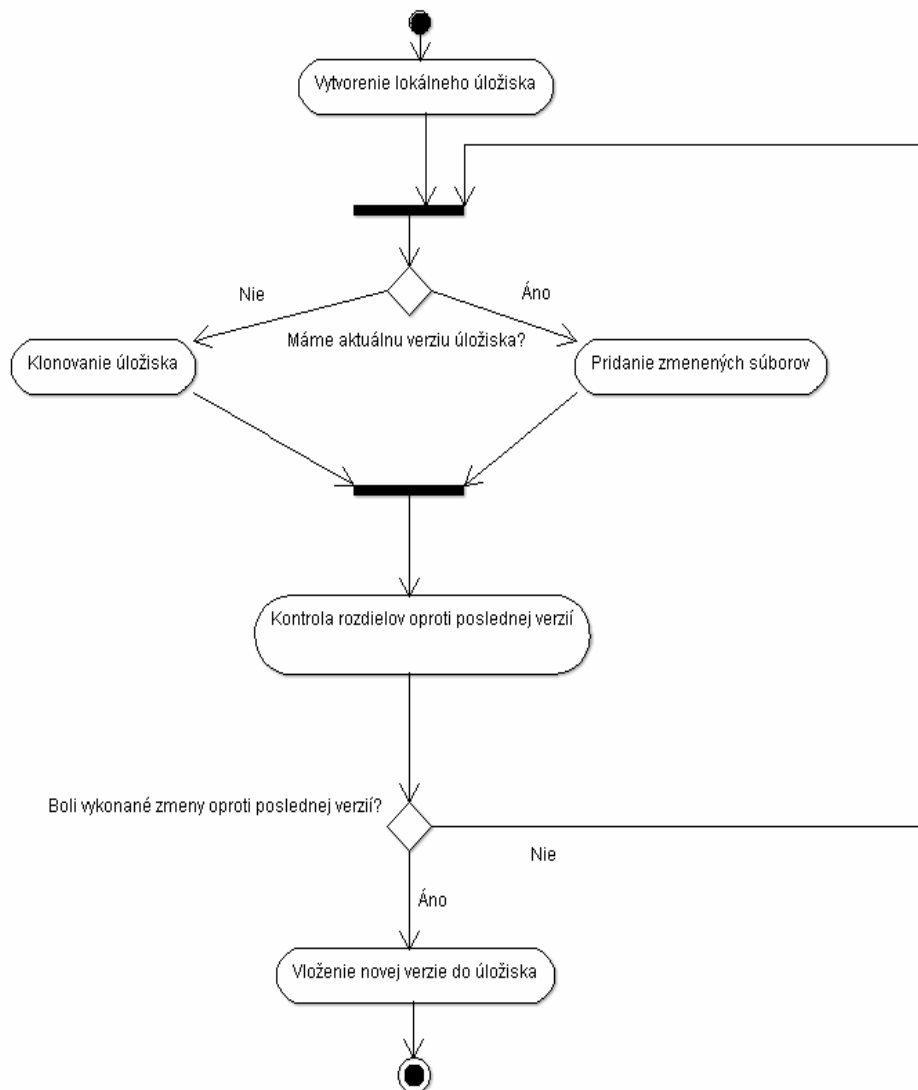


Diagram č. 1.: Postupnosť akcií procesu vykonávania zmien

5.2.1 Vytvorenie lokálneho úložiska

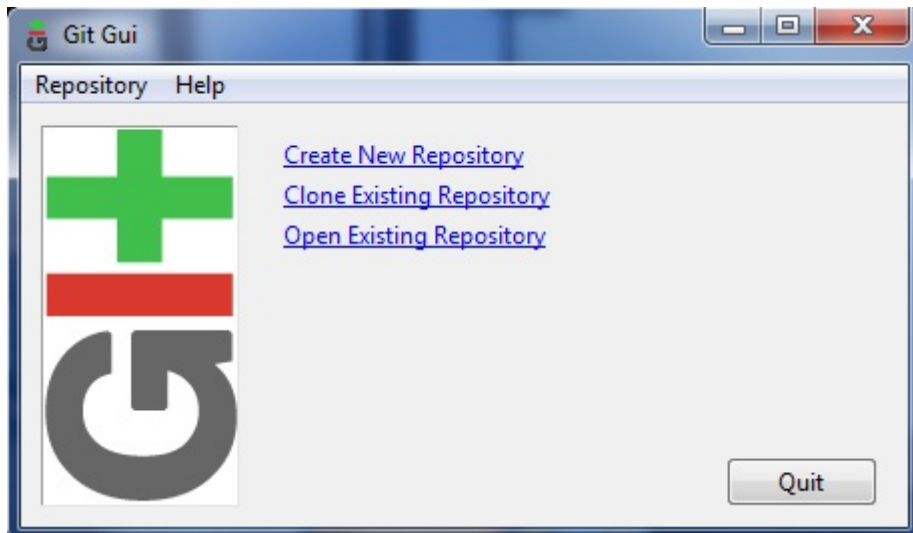
Pre vykonanie akejkoľvek činnosti v programe GIT je nutné mať vytvorené nejaké úložisko. Ak začíname pracovať na projekte, bez toho aby sme mali nejaké centrálné úložisko, je potrebné vytvoriť si ho lokálne. Lokálne úložisko je možné vytvoriť z akéhokoľvek adresára.

Vytvorenie lokálneho úložiska v GIT Bash:

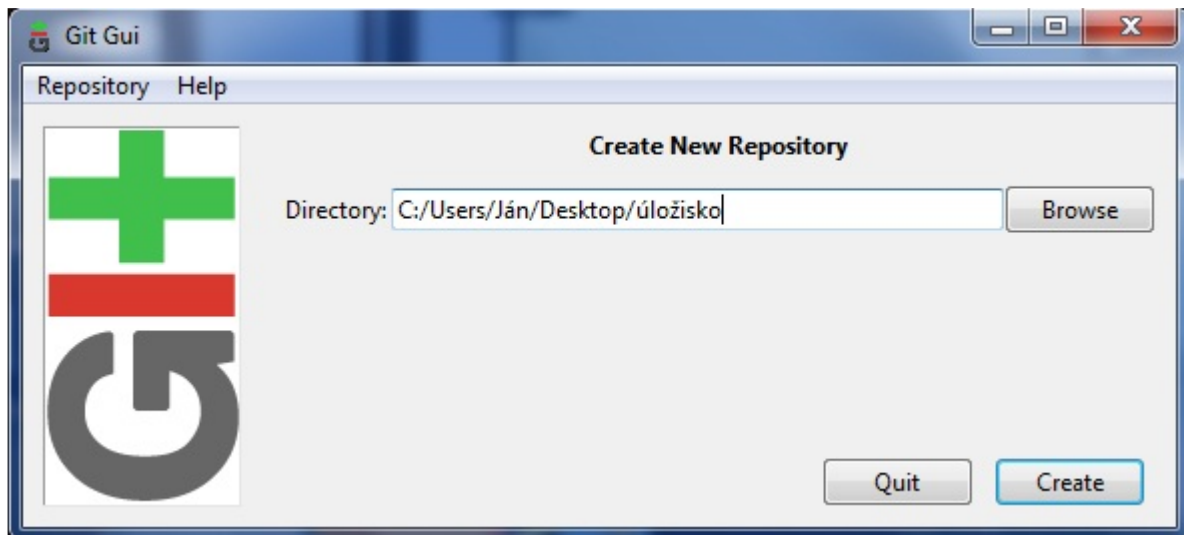
1. V štart menu vyberte program GIT Bash a spustite ho.
2. Pomocou príkazu `cd <cesta k adresáru>` sa prepnete do adresára, z ktorého chcete vytvoriť úložisko
3. Príkazom `git init` inicializujte tento adresár na nové lokálne úložisko.

Vytvorenie lokálneho úložiska v GIT GUI:

1. V štart menu vyberte program GIT GUI a spustite ho.
2. Z ponúknutého menu vyberte možnosť *Create New Repository* (obrázok č.1.)
3. Vložte cestu k vybranému adresáru a potvrdte tlačítkom *Create* (obrázok č.2.)



Obrázok č. 1.:Menu v GIT GUI



Obrázok č.2: Cesta k adresáru

5.2.2 Klonovanie úložiska

Ak predpokladáme, že je v centrálnom úložisku vytvorený projekt a programátor na ňom má pracovať, potrebuje získať jeho aktuálnu kópiu. Tú získa klonovaním úložiska. Pomienkou, ktorá je nutná pre vstup do tohto procesu, je pridanie SSH kľúča integrátorom do zoznamu kľúčov v úložisku. Kľúč umožňuje programátorovi vstup do úložiska. Program GIT mu takýto kľúč vygeneruje. Podrobný postup obsahuje metodika [2].

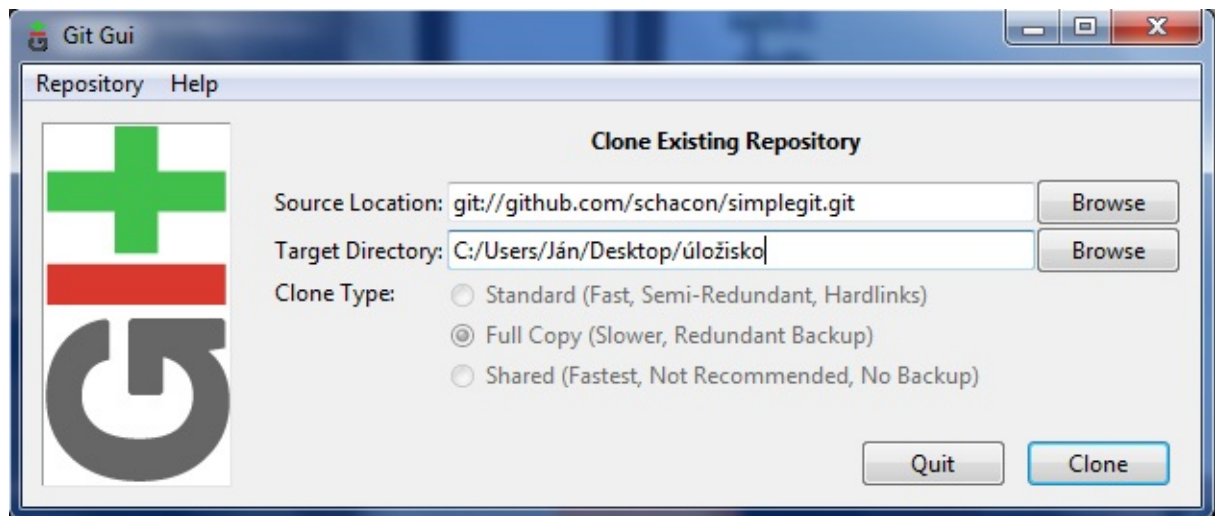
Klonovanie úložiska v GIT Bash:

1. V štart menu vyberte program GIT Bash a spustite ho.
2. Použitím príkazu `git clone <url>` naklonujte úložisko. Po úspešnom vykonaní príkazu získate informatívny výpis obsahujúci základné informácie o súboroch stiahnutých z úložiska.

Klonovanie úložiska v GIT GUI:

1. V štart menu vyberte program GIT GUI a spustite ho.
2. Z ponúknutého menu vyberte možnosť *Clone Existing Repository* (obrázok č.1.)

3. Do ponúknutého formulára vložte adresu úložiska a cestu k adresáru, kam ho chcete uložiť (obrázok č. 3). Tento adresár je nový, to znamená, že doteraz nesmel existovať, inak program skončí s chybou.



Obrázok č.3.: Cesta k úložisku a cieľový adresár

5.2.3 Pridanie zmenených súborov

Program GIT pracuje s konceptom prostredia úrovni (*staging area*). Preto ak chceme do úložiska pridať zmenený súbor, potrebujeme najprv zmeny pridať do prostredia úrovni. Týmto spôsobom je možné rovnako pridať nový súbor alebo existujúci súbor zmenený oproti pôvodnej verzii.

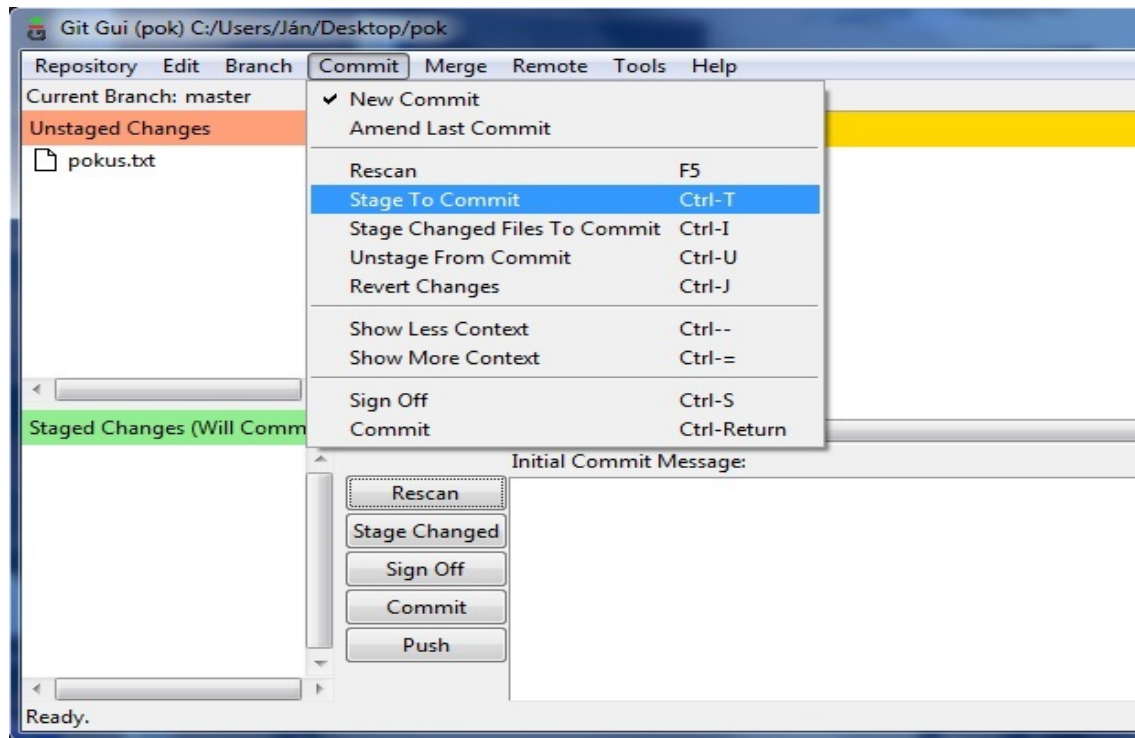
Pridanie zmeneného súboru v GIT Bash:

1. Pomocou príkazu `cd <cesta k adresáru>` sa prepnete k požadovanému adresáru.
2. Spustením príkazu `git status -s` dostanete výpis, na ktorých súboroch boli vykonané zmeny, ktoré ešte neboli pridané.
3. Príkazom `git add <názov súboru>` pridajte súbor do prostredia úrovni.

Ak chcete pridať všetky súbory v danom adresári, v ktorých boli vykonané zmeny alebo boli nanovo pridané, môžete príkaz spustiť v tvare `git add .` kde bodka symbolizuje aktuálny adresár.

Pridanie zmeneného súboru v GIT GUI:

1. Súbor, ktorý chcete pridávať sa musí nachádzať v adresári, ktorý je našim úložiskom.
2. V štart menu vyberte program GIT GUI a spustite ho.
3. V časti *Unstaged Changes* sa objaví zmenený súbor. Označte ho kliknutím naň.
4. Vybratím položky z menu *Commit->Stage to commit* tento súbor pridajte do prostredia úrovni. (Obrázok č.5). Po správnom vykonaní sa objaví v časti *Staged changes*.



Obrázok č.5.: Pridanie zmeneného súboru v GIT GUI

5.2.4 Kontrola rozdielov oproti poslednej verzii

V programe GIT je možné skontrolovať, ktoré súbory boli zmenené a aké zmeny boli v súbore vykonané oproti jeho predchádzajúcej verzii. Samozrejme predpokladáme, že máme nainštalovaný GIT a vytvorené neprázdne úložisko.

Kontrola rozdielov v GIT Bash:

1. V štart menu vyberte program GIT Bash a spustíte ho. Vykonajte predchádzajúce kroky procesu.
2. Spustíte príkaz diff zadaním `git diff` (bez argumentov). Ten zobrazí všetky zmeny vykonané v súborech oproti poslednej verzii, riadok po riadku v unifikovanom formáte. (obrázok č.6.)

```

$ git diff
diff --git a/hello.rb b/hello.rb
index d62ac43..8d15d50 100644
--- a/hello.rb
+++ b/hello.rb
@@ -1,7 +1,7 @@
 class HelloWorld

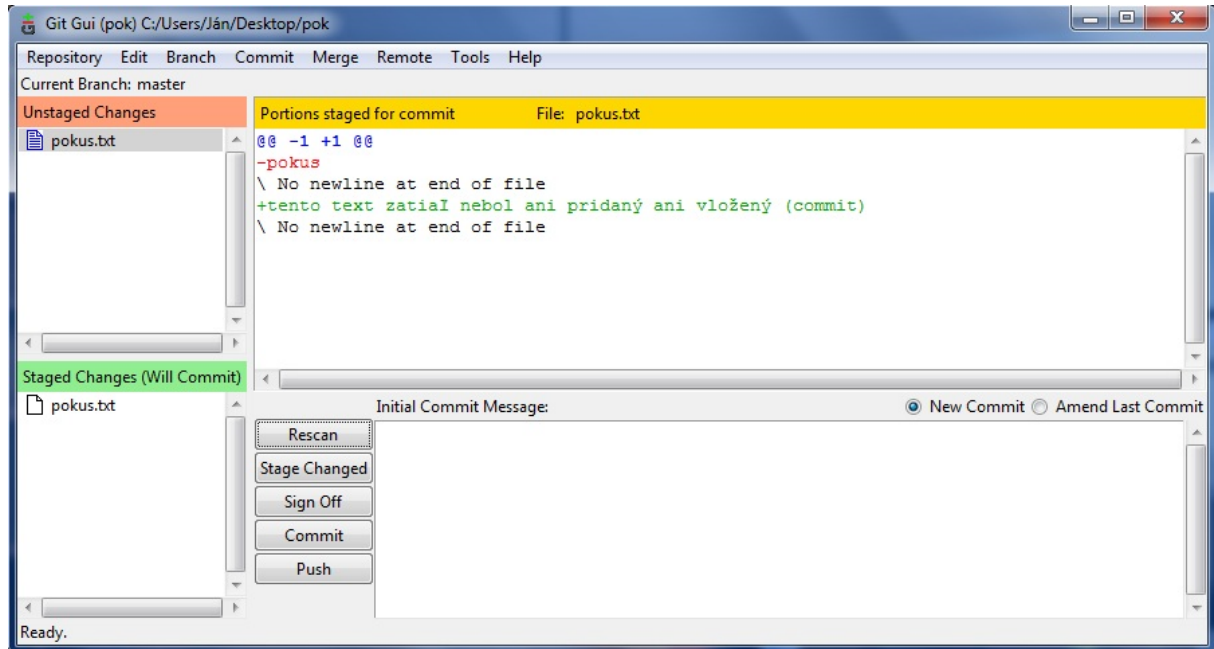
   def self.hello
-   puts "hello world"
+   puts "hola mundo"
   end

 end
  
```

Obrázok č.6.: Výstup príkazu `diff`

Kontrola rozdielov v GIT GUI:

Grafické užívateľské prostredie poskytuje vo svojich formulároch prehľad o súboroch, s ktorými pracuje, bez nutnosti používať príkazy. Časť *Unstaged changes* obsahuje zmenené, ale zatiaľ nepridané súbory. Časť *Staged changes* obsahuje zmeny, ktoré už boli pridané do prostredia úrovni. Hlavná časť zobrazuje konkrétne zmeny v súboroch. (obrázok č.7.)



Obrázok č.7.: Informácie o spracovávaných súboroch

5.2.5 Vloženie novej verzie do úložiska

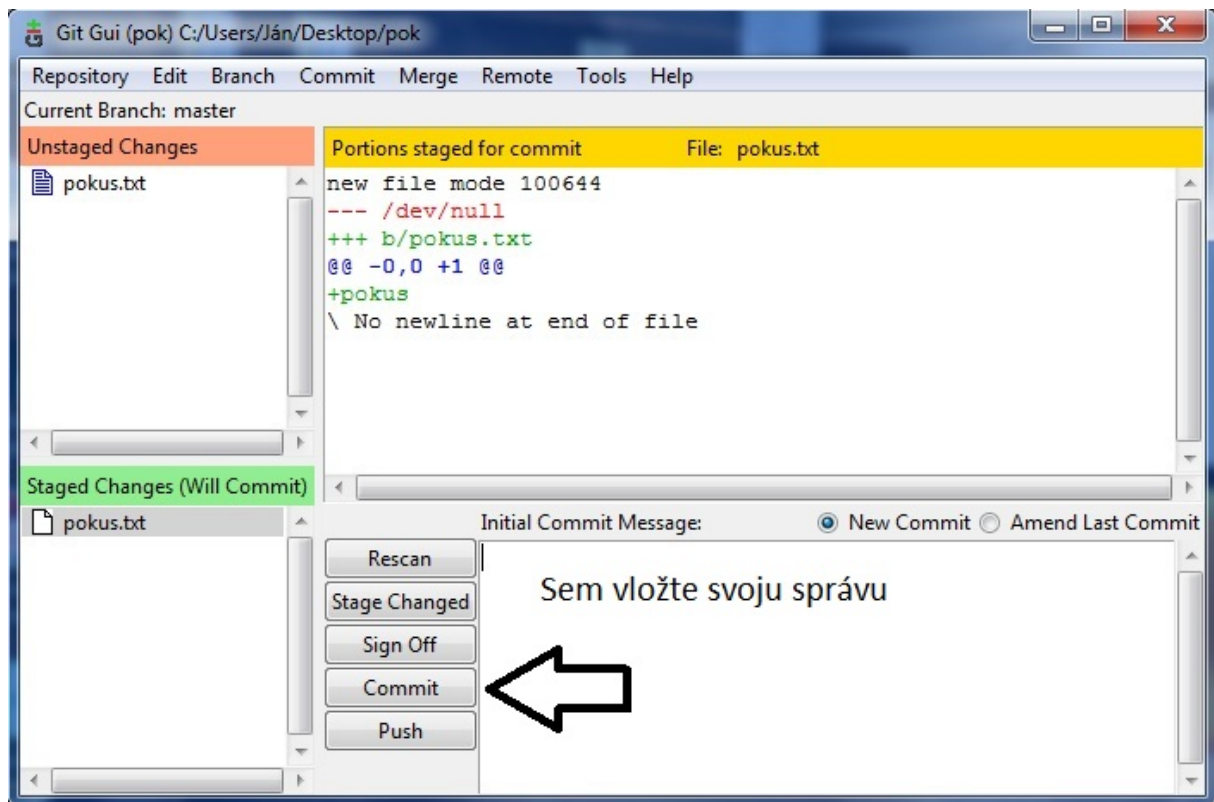
Po kontrole zmien v súboroch tieto zmenené súbory vložte do úložiska.

Vloženie nových súborov pomocou GIT BASH:

1. V Štart menu vyberte GIT BASH a otvorte ho. Vykonajte všetky predchádzajúce kroky procesu.
2. Pridajte vykonané zmeny. Vykonáte to príkazom `git commit -m <správa>`.

Vloženie nových súborov pomocou GIT GUI:

1. V Štart menu vyberte GIT GUI a spustite ho. Vykonajte všetky predošlé kroky procesu.
2. Na označené miesto na obrázku č.8. napíšte správu pre commit.
3. Stlačte tlačítko commit označené šípkou (obrázok č. 8.).



Obrázok č.8.: Vkladanie zmien pomocou commit.

5.2.6 Vloženie lokálneho úložiska do centrálneho

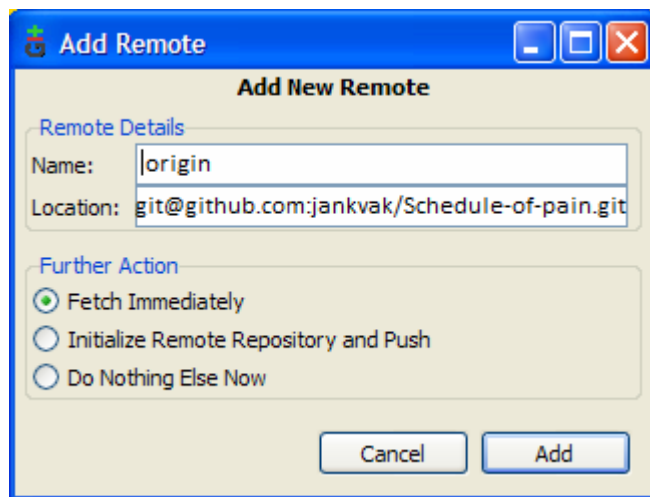
Na sprístupnenie zmien ostatným spolupracovníkom je nutné vlastné úložisko vložiť do centrálneho.

Vloženie lokálneho úložiska do centrálneho v GIT GUI

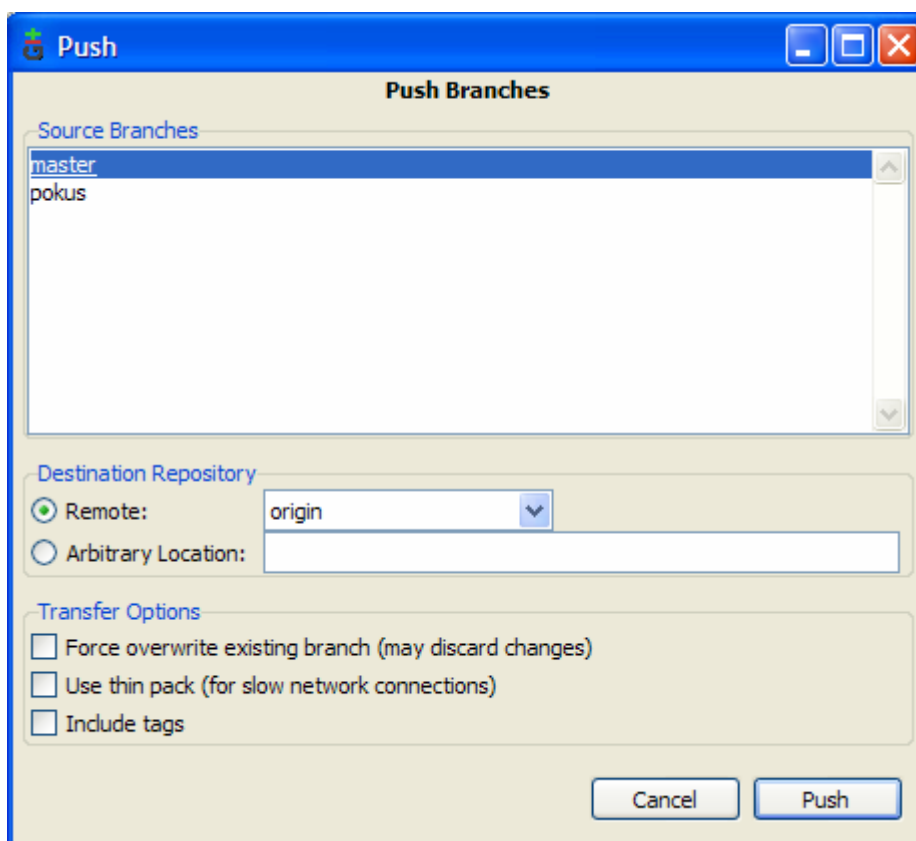
1. V Štart menu vyberte GIT BASH a otvorte ho. Vykonajte všetky predchádzajúce kroky procesu.
2. Spustením príkazu `git remote add origin <cesta>` pridajte vzdialený server s názvom *origin*. Cesta k tomuto serveru je v podobe [git@cesta/ulozisko.git](mailto:git@cesta.ulozisko.git) napríklad git@github.com:jankvak/Schedule-of-pain.git.
3. Spustením príkazu `git push <názov vzdialeného servera> <názov vetvy>` vložíme naše úložisko do centrálneho. V našom prípade bude názvom úložiska *origin* a názvom vetvy *master*.

Vloženie lokálneho úložiska pomocou GIT GUI

1. V Štart menu vyberte GIT GUI a spustite ho.
2. Vyberte Remote->Add... Vyplňte názov, v našom prípade *origin* a cestu (napr. git@github.com:jankvak/Schedule-of-pain.git) (Obrázok č.9.).
3. Vykonajte všetky predošlé kroky procesu.
4. Vyberte tlačítko *push*.
5. Vyberte vetvu a vzdialené úložisko (*origin*) a stlačte *push* (Obrázok č. 10.).



Obrázok č.9.: Pridanie vzdialeného úložiska



Obrázok č. 10.: Vloženie vetvy do vzdialeného úložiska

Záver

Táto metodika poskytuje ucelený návod pre základnú prácu so systémami na kontrolu verzií a konkrétne s programom GIT. Môže byť cennou pomocou pre vývojárov, ktorí v tomto programe začínajú pracovať. V jednoduchých krokoch sú tu opísané najdôležitejšie činnosti vykonávané v systéme na kontrolu verzií GIT. Nasledujúce činnosti potrebné pre pokročilejšiu prácu so systémom sú opísané v metodike [4].

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Evidencia chýb pomocou nástroja Bugzilla M2010/11/1

Metodika

Peter Korenek

Študijný program: Softvérové inžinierstvo
Predmet: Manažment projektov softvérových a informačných systémov
Školský rok: 2010/2011

1 Úvod

Metodika sa zaoberá evidenciou chýb v softvérových projektoch pomocou nástroja Bugzilla [6]. Je určená pre zákazníkov, ktorí používajú naše produkty, ale hlavne pre členov tímu na evidenciu chýb vo vyvíjaných produktoch. V kapitole 2 sú zobrazené jednotlivé procesy tak, ako za sebou nasledujú. V kapitole 3 sú opísané roly, ktoré sa zúčastňujú procesu evidencie chýb. Kapitola 4 podrobne opisuje procesy, ktoré je potrebné dodržiavať pri správe chýb softvérových projektoch. Kapitola 5 definuje, ako vyplniť a odoslať správu o chybe.

Pojmy

V tabuľke č.1 sú vysvetlené pojmy, ktoré sú použité v tejto metodike.

Tabuľka 1. Pojmy použité v texte.

Report	správa o výskyte chyby v projekte, ktorá opisuje prejavy chyby i podmienky, za akých chyba nastala
Chyba	nezrovnalosť medzi požadovanou a aktuálnou funkcionalitou produktu
Bugzilla	voľne dostupný nástroj na evidenciu chýb v projektoch
Patch	softvér malého rozsahu, ktorý opravuje určitú chybu, ktorá vznikla v produkte.
Login	prihlásenie k používateľskému účtu pomocou mena a hesla
Keyword	klúčové slovo, ktoré vystihuje podstatu riešeného problému
Spracovanie chýb	proces evidencie chýb, za účelom vytvorenia optimálneho plánu ich opravy
Správa	dokument obsahujúci detailné informácie o vykonanej práci
ScheduleBug	systém na správu chýb dostupný na adrese http://labss2.fiit.stuba.sk/TeamProject/2010/team07is-si/bugzilla

Nadväzujúce metodiky

- [1] Metodika na správu zmien v prostredí PHP M2010/10/2
- [2] Metodika na testovanie súčastí systému v prostredí PHP M2010/10/4
- [3] Metodika na inštaláciu a konfiguráciu nástroja Bugzilla M2010/10/6
- [4] Metodika na dokumentovanie spôsobu opravy chyby M2010/10/7
- [5] Metodika na dokumentovanie príčin neúspešnej opravy chyby M2010/10/10

Súvisiace zdroje

- [6] Bugzilla home page, <http://www.bugzilla.org/>
- [7] Bugzilla online documentation, <http://www.bugzilla.org/docs/4.0/en/html/>

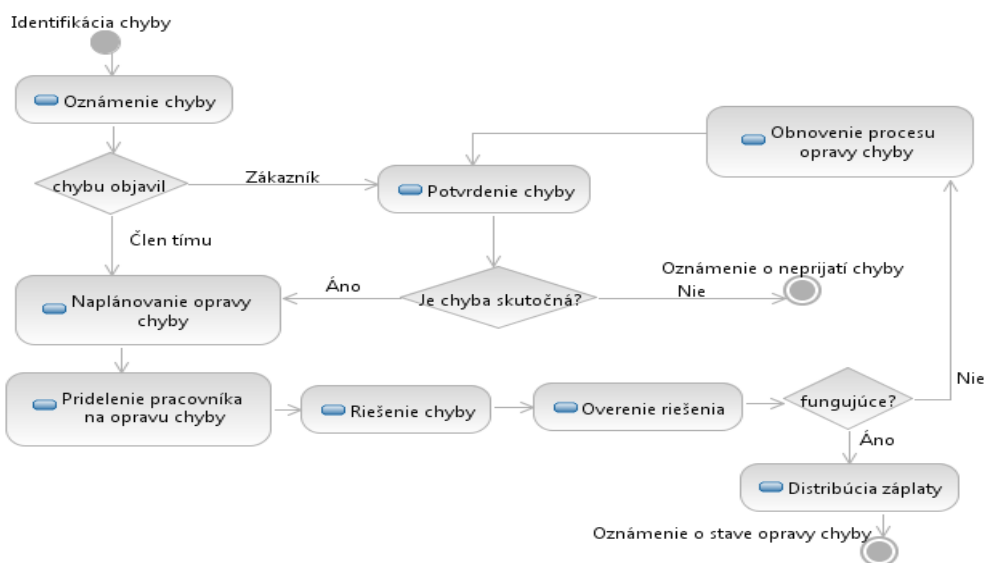
2 Procesy správy chýb

Správa chýb pozostáva z procesov, ktoré sú uvedené v tabuľke 2. Ku každému procesu sú uvedené kapitoly, v ktorých sa nachádza jeho podrobný opis. Vzájomné prepojenie procesov, ktoré je znázornené na obrázku č.1, môžeme opísať nasledujúcim postupom. Na začiatku procesu správy chýb oznamovateľ identifikuje a oznámi výskyt chyby v produkte. Podľa toho, či je chybu oznámil zákazník alebo člen tímu, sa vykoná kontrola existencie chyby v produkte. Ak chyba existuje, jej oprava sa zaradi do plánu opráv a pridelia sa k nej pracovníci. V opačnom prípade je oznamovateľ upozornený o odmietnutí chyby. Potvrdená chyba sa následne rieši a validuje. V prípade neúspešnej opravy začína proces odznova od potvrdenia chyby. V prípade úspešnej opravy sa patch distribuuje a oznámi sa úspešná kontrola oznamovateľovi.

Základným predpokladom pre úspešnú správu chýb je správna inštalácia a konfigurácia nástroja Bugzilla, podľa metodiky [3] . Tieto procesy vykonáva manažér.

Tabuľka 2. Procesy vykonávané pri správe chýb.

Poradie	Názov procesu	Kapitola
1.	Oznámenie chyby	4.1
2.	Potvrdenie chyby	4.2
3.	Naplánovanie opravy chyby	4.3
4.	Pridelenie pracovníka na opravu chyby	4.4
5.	Riešenie chyby	4.5
6.	Overenie riešenia	4.6
7.	Distribúcia záplaty	4.7
8.	Oznámenie o úspešnej oprave chyby	4.8
9.	Obnovenie procesu opravy chyby	4.9
10.	Oznámenie o neprijatí chyby na opravu	4.10



Obrázok 1. Vzájomné prepojenie procesov.

3 Roly a zodpovednosti pri správe chýb

Tabuľka č.3 opisuje roly, ktoré sa zúčastňujú správy chýb. Ku každej role je priradená zodpovednosť.

Tabuľka 3. Roly a zodpovednosti pri správe chýb.

Rola	Zodpovednosť
Vývojár	<ul style="list-style-type: none">• Je zodpovedný za identifikáciu chýb v projektoch a vytváranie reportov o týchto chybách• Vytvára záplaty, ktoré opravujú identifikované chyby v systéme• Distribuuje vytvorené záplaty zákazníkom
Tester	<ul style="list-style-type: none">• Overuje, či sa oznámená chyba naozaj vyskytuje v produkte pomocou testovania systému• Overuje riešenie chyby, ktoré bolo vytvorené vývojárom
Zákazník	<ul style="list-style-type: none">• Nahlasuje chyby, ktoré sa stali pri prevádzke produktu
Vedúci projektu	<ul style="list-style-type: none">• Prideluje úlohy, ktoré sa týkajú opravy chýb pracovníkom• Je zodpovedný za kvalitu výsledného produktu• Oboznamuje zákazníkov o stave opravy chyby• Kontroluje stav opravy chyby
Manažér plánovania	<ul style="list-style-type: none">• Vytvára odhady nákladov a trvania opravy chýb• Vytvára plány opravy chýb
Manažér podporných prostriedkov	<ul style="list-style-type: none">• Vykonáva inštaláciu nástroja Bugzilla• Konfiguruje nástroj Bugzilla<ul style="list-style-type: none">○ prispôsobuje vzhľad formulárov○ určuje právomoci v systéme○ spravuje používateľské kontá

4 Opis procesov správy chýb

V tejto kapitole sú opísané procesy, ktoré sa vykonávajú pri bežnom životnom cykle chýb [7].

4.1 Oznámenie chyby

Vstup: identifikovaná chyba pri prevádzke systému

Výstup: report o výskyte chyby

Určené pre: každý člen tímu, zákazník

Zákazník a člen vývojového tímu oznámia každý výskyt chyby v produkte pomocou nástroja Bugzilla. Do pripraveného formuláru zadá osoba, ktorá odhalila chybu všetky požadované údaje, ktoré sú potrebné pre úspešné odstránenie chyby podľa kapitoly 5.

4.2 Potvrdenie chyby

Vstup: prijatý report o chybe od zákazníka

Výstup: protokol o vykonaných testoch

Zodpovedný: tester

Tester po prijatí reportu o výskyte chyby od zákazníka rozhodne o prijatí alebo zamietnutí chyby. Ako prvé overí, či sa jedná o skutočnú chybu v systéme, falošnú chybu alebo rozšírenie. V prípade rozšírenia postupuje podľa metodiky pre zmeny [1]. Každé rozhodnutie o prijatí alebo odmietnutí chyby musí zdokumentovať formou testov. Testovanie produktu sa uskutočňuje za rovnakých podmienok, aké sú opísané v správe o chybe, v zmysle metodiky o testovaní [2]. Výsledky testov zdokumentuje vo forme protokolu o vykonaných testoch, v zmysle metodiky [2] a priloží ho ako prílohu reportu. Ak bola chyba potvrdená, nastaví stav chyby na *confirmed*, teda potvrdená v opačnom prípade nastaví stav chyby na odmietnutá, teda *refused* a pokračuje sa procesom 4.10.

4.3 Naplánovanie opravy chyby

Vstup: report o chybe

Podmienky: stav chyb musí byť nastavený na New alebo Confirmed

Výstup: aktualizovaný plán opráv

Zodpovedný: manažér plánovania

Po prijatí reportu vytvorí manažér plánovania odhad času a nákladov, ktoré budú potrebné na opravu chyby. Na základe týchto informácií zaradí opravu chyby do plánu opráv. Aktualizovaný plán priloží ako prílohu do detailov reportu o chybe.

4.4 Pridelenie pracovníka na opravu chyby

Vstup: report o chybe doplnený o odhady

Výstup: aktualizácia mena pracovníka zodpovedného za odstránenie chyby v reporte

Zodpovedný: vedúci projektu

Podľa odhadov náročnosti opravy a vytvoreného plánu priradí vedúci projektu pracovníkov na opravu chyby. Štandardne sa prideli ku každej chybe jeden pracovník. Ak sa jedná o závažnú chybu, ktorá sa týka viacerých komponentov produktu, počet pracovníkov sa zvýši úmerne k odhadnutej časovej náročnosti. Ku každej chybe sa priraduje pracovník, ktorý robil na komponente, v ktorom sa vyskytla chyba. V reporte o chybe zapíše vedúci projektu meno prideleného pracovníka do políčka *assigned to*.

4.5 Riešenie chyby

Vstup: report o chybe

Výstup: patch opravujúci chybu, správa o riešení chyby

Zodpovedný: vývojár

Vývojár sa na základe detailov, ktoré sa nachádzajú v reporte o chybe, pokúsi opraviť odhalenú chybu. Na začiatku opravy zmení stav chyby v reporte na *In progress*, teda riešený. Po oprave chyby zapíše do detailov reportu spôsob opravy chyby, a to maximálne dvoma

vetami. Vývojár vytvorí dokument obsahujúci podrobný opis spôsobu opravy chyby v zmysle metodiky [4]. Priloží ho do prílohy reportu spolu s vytvoreným patchom, ktorý opravuje chybu. Upraví tiež stav riešenia chyby na *resolved*, teda vyriešený.

4.6 Overenie riešenia

Vstup: report o chybe, dokument opisujúci spôsob riešenia chyby, vytvorený patch

Výstup: protokol o testovaní riešenia

Zodpovedný: tester

Po zmene stavu chyby na *resolved* tester overí pomocou testov, ktoré odhalili chybu, či sa vývojárovi podarilo túto chybu odstrániť. Pri testovaní postupuje podľa metodiky o testovaní [2]. Protokol s výsledkami testov pripojí k reportu o chybe. V prípade, že chyba bola odstránená, označí stav chyby ako *verified*, teda overený. V opačnom prípade zmení stav riešenia chyby na *failed*, teda neúspešný.

4.7 Distribúcia záplaty

Vstup: report o chybe, protokol o testovaní riešenia

Výstup: distribúcia záplaty

Zodpovedný: vývojár

Ak bol stav riešenia chyby označený za overený, vývojár umiestni vytvorený patch na verejne dostupné miesto, odkiaľ sa bude distribuovať zákazníkom.

4.8 Oznámenie o úspešnej oprave chyby

Vstup: report o chybe, protokol o testovaní riešenia

Výstup: správa o úspešnej oprave chyby, aktualizácia stránky produktu.

Zodpovedný: vedúci projektu

Po uverejnení patchu na dostupnom serveri odošle vedúci projektu správu o úspešnej oprave chyby všetkým členom tímu, ktorí sa podieľali na oprave. Správu odošle aj osobe, ktorá identifikovala chybu. Zároveň vypíše aj oznam na webovú stránku produktu.

4.9 Obnovenie procesu opravy chyby

Vstup: dokument opisujúci spôsob riešenia chyby, protokol o testovaní riešenia

Výstup: vytvorenie záznamu o neúspešnej oprave chyby

Zodpovedný: vedúci projektu

Vedúci projektu po zmene stavu chyby na neúspešný, vytvorí záznam o neúspešnej oprave podľa metodiky [5] a označí stav chyby za potvrdený. Proces opravy chyby pokračuje bodom 4.3.

4.10 Oznámenie o neprijatí chyby na opravu

Vstup: report o chybe, protokol o testovaní riešenia

Výstup: upovedomenie oznamovateľa o odmietnutí chyby

Podmienka: stav chyby bol nastavený na refused

Zodpovedný: vedúci projektu

Po zmene stavu chyby na refused oznámi vedúci projektu oznamovateľovi, že jeho chyba bola odmietnutá a uvedie dôvod odmietnutia. Forma komunikácie závisí od stupňa závažnosti chyby podľa toho, ako ju oznamovateľ oznámil v reporte. Kritické chyby sa oznamujú telefonicky a menej závažné pomocou emailu.

5 Vytvorenie nového reportu o chybe

V rámci procesu oznámenia chyby uvedenom v kapitole 4.1 musí oznamovateľ chyby vykonať prihlásenie sa do systému a vytvorenie nového reportu o chybe. Pre správne vykonanie týchto krokov, musí oznamovateľ postupovať podľa postupov definovaných v kapitolách 5.1 a 5.2 .

5.1 Prihlásenie do systému

- Vložte URL adresu <http://labss2.fiit.stuba.sk/TeamProject/2010/team07is-si/bugzilla> do internetového prehliadača
- Ak máte vytvorené konto v systéme ScheduleBug
 - Stlačte login v hornej časti obrazovky
 - Zadajte prihlasovacie meno a heslo, ktoré ste zadali pri registrácii v systéme ScheduleBug. Všetci študenti a zamestnanci, ktorí majú vytvorené konto v systéme AIS, majú vopred vytvorené konto. Ak máte konto v systéme AIS, použite prihlasovacie údaje z AIS
 - Stlačte tlačidlo login
- Ak nemáte vytvorené konto
 - Stlačte Open a New Account
 - Zadajte vašu emailovú adresu
 - Potvrďte tlačidlom Send
 - Pokračujte s vytvorením konta podľa pokynov z prijatej emailovej správy
 - Pokračujte podľa bodu b

5.2 Pridanie chyby do systému

- 1) Po prihlásení do systému si zvolte produkt, v ktorom sa vyskytla chyba.
- 2) Vyplňte základné informácie o chybe vo formulári, ktorý sa Vám zobrazil. Povinné údaje sú znázornené na obrázku č.2 červenou farbou.
 - a) V políčku Component vyberte komponent systému, v ktorom sa vyskytla chyba.
 - b) V políčku Version zvolte verziu softvéru. Verziu svojho softvéru nájdete zobrazenú v našich produktoch vždy v ľavom dolnom rohu aplikácie.
 - c) Určte stupeň kritickosti:
 - Blocker – pre chyby, ktoré blokujú ďalšiu činnosť v produkte
 - Critical – pre chyby, ktoré spôsobili pád produktu, ale ešte umožňujú vykonávať štandardné operácie
 - Major – pre chyby, ktoré nepôsobili pád produktu, avšak vážne ovplyvnili funkciu ostatných komponentov.

- Normal – pre chyby, ktoré spôsobili, že aplikácia vykonala nesprávnu operáciu, ktorá neohrozuje chod produktu.
 - Minor – chyby, ktoré neohrozujú aplikáciu a sú skôr kozmetické – chyby v názvoch tlačidiel, nekorektné zobrazenie diakritiky, nefungujúce alternatívne tlačidlo.
 - Rozšírenie – nápady, ktoré pomôžu vylepšiť produkt.
- d) Vyberte z ponuky Hardware platformu, na ktorej sa vyskytla chyba.
- e) Vyberte z ponuky Operačný systém OS, na ktorom sa vyskytla chyba. Na výber sú nasledovné možnosti:
- Windows XP
 - Windows Vista
 - Windows 7
 - Linux
 - Other
- f) Určte stupeň priority, ktorý má oprava chyby:
- P1 – oprava má najvyššiu prioritu, oprava by mala byť vykonaná v priebehu najbližších hodín, pretože ohrozuje chod spoločnosti.
 - P2 – oprava má vysokú prioritu, je potrebné ju vykonať v priebehu najbližšieho dňa
 - P3 – oprava má strednú prioritu, je potrebné ju vykonať v priebehu jedného týždňa
 - P4 – oprava má nízku prioritu, je potrebné ju vykonať v priebehu mesiaca
 - P5 – oprava má najnižšiu prioritu, je potrebné ju zapracovať v priebehu najbližších mesiacov
- g) Zvoľte status chyby – ak ste zákazník, vyberte Unconfirmed, ak ste člen tímu vyberte New.
- h) Pridelte k chybe zodpovednú osobu - vyplňte pole assigned to:
- Ak ste zákazník, ponechajte túto hodnotu na prednastavenej
 - Ak ste člen tímu, máte dve možnosti.
 - (1) Priradte úlohu konkrétnej osobe, ak ju je na základe typu chyby viete určiť.
 - (2) Ponechajte prednastavenú hodnotu.
- i) Vyplňte pole Summary so stručným opisom chyby, nie dlhším ako jedna veta.
- 3) Vyplňte podrobné informácie o chybe. Tieto nepovinné údaje sú zobrazené na obrázku č.2 modrou farbou.
- a) Ak sa jedná o webový produkt, do textového pola s názvom URL vpíšte URL adresu produktu, na ktorej sa vyskytla chyba. Adresu nájdete v adresovom poli vo vašom internetovom prehliadači.
- b) Vyplňte pole Time when bug happened, s údajmi o čase, kedy nastala chyba. Dátum uvádzajte vo formáte RRRR-MM-DD HH:MM. V prípade, že si nepamätáte čas, uveďte aspoň dátum.
- c) Vyplňte pole Description so širším opisom chyby:
- Prvý odsek musí obsahovať postupnosť krokov, ktoré viedli k chybe, oddelené pomlčkami.
 - Druhý odsek musí obsahovať opis následkov chyby.

- Zvyšok opisu vyplňte ľubovoľným textom. Je vhodné ak ešte napíšete, s akým dátami ste pracovali, či sa táto chyba už niekedy vyskytla, prípadne aký druh pomoci v tejto situácii ste očakávali od systému.
- d) Ak bola chyba zdokumentovaná formou obrázku alebo iného dokumentu, vložte tento dokument pomocou tlačidla Attach File.
- Postup pre zachytenie chyby formou obrázku je nasledovný. Zobrazte si okno aplikácie, v ktorom nastala chyba. Stlačte klávesy Ctrl+PrtSc. Otvorte si editor obrázkov (napríklad Skicár) a pomocou voľby - úpravy – prilepiť vložte obrázok do editora. Obrázok si uložte ako jpg súbor. Po stlačení tlačidla Attach File si vyhľadajte uložený súbor v zobrazenom dialógu a stlačte tlačidlo Open.
- e) Do pola Keywords vložte aspoň jedno kľúčové slovo z množiny {failure, login, functionality, GUI}. Kľúčové slová oddel'te čiarkou. Ak žiadne kľúčové slovo z tejto množiny nevystihuje váš druh chyby, kontaktujte administrátora prostredníctvom emailu – pkorenek@gmail.com s požiadavkou na vytvorenie nového kľúčového slova. Ako predmet správy uveďte „New keyword request“.
- f) Ostatné polia, ktoré neboli spomenuté v tejto metodike ponechajte nevyplnené.
- 4) Odošlite vyplnený formulár pomocou tlačidla Submit bug.

Product: ScheduleOfPain Reporter: pkorenek@gmail.com

a * Component: Administrácia
 Databáza
 Študentská časť
 Učiteľská časť

Component Description
 Interaktívne prostredie pre špecifikáciu požiadaviek na rozvrh učiteľov

b Version: v 0.1

c Severity: blocker

d Hardware: PC

e OS: Windows 7

f Priority: P1

We've made a guess at your operating system and platform. Please check them and make any corrections if necessary.

g Initial State: UNCONFIRMED

h Assign To: pkorenek@gmail.com

Estimated Hours:

Deadline: (YYYY-MM-DD)

a URL: https://labss2.fiiit.stuba.sk/TeamProject/2010

b Time when bug happend: 2010-11-07 22:23

i * Summary: Po stlačení tlačidla rezervuj cvičenie mi zmizli všetky žetóny

c Description: login - kliknutie v menu na voIbu rezervovať cvičenie - výber cvičenia na predmet
 MSI v utorok o 17 - rezervuj
 zmizli všetky žetóny a nedovolilo rezervovať iné cvičenie
 stále mám nepriradené cvičenie

d Attachment:

e Keywords: failure (optional)

Depends on:

Blocks:

4

Obrázok 2. Formulár pre vytvorenie nového report o chybe.

Na obrázku č.2 sú vyznačené červenou farbou povinné polia, ktoré treba vyplniť pri každom oznamovaní chyby. Modrou farbou sú označené polia, ktoré môžu vyplniť členovia tímu, pokiaľ poznajú príslušné informácie. Označenie polí vo formulári zodpovedá označeniu v kapitole 5.2.

Príloha C – Preberacie protokoly

Preberací protokol

Odvzdávajúci subjekt: Schedule of Pain (tím č.7)

Preberajúci subjekt: Ing. Miroslav Galbavý (vedúci tímu)

Predmety prebratia:

- odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrhu riešenia
- odovzdanie dokumentácie k riadeniu projektu

Poznámky: dokumenty boli odovzdané v tlačenej podobe

.....

podpis zástupcu odovzdávajúceho subjektu

.....

podpis zástupcu preberajúceho subjektu

V Bratislave, dňa 3.11.2010.

Preberací protokol

Odozdávajúci subjekt: Schedule of Pain (tím č.7)

Preberajúci subjekt: Ing. Miroslav Galbavý (vedúci tímu)

Predmety prebratia:

- odovzdanie finálnej systémovej dokumentácie za 1. semester
- odovzdanie dokumentácie k riadeniu projektu
- odovzdanie prototypu systému

Poznámky: dokumenty boli odovzdané v tlačenej podobe, prototyp systému na CD médiu

.....

podpis zástupcu odozdávajúceho subjektu

.....

podpis zástupcu preberajúceho subjektu

V Bratislave, dňa 13.12.2010.