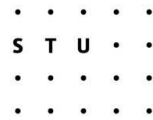




Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 812 19 Bratislava



Tímový projekt

Simulácia demonštrácie v meste

Dokumentácia k riadeniu projektu

Tím č. 9 - SimTeam

Bc. Jana Branišová

Bc. Adrián Kollár

Bc. Michal Kyžňanský

Bc. Miroslav Ort

Bc. Michal Ošvát

Bc. Filip Pakan

Vedúci tímového projektu: Ing. Peter Lacko, PhD.

Mailový kontakt na tím: tim2012@googlegroups.com

Akademický rok: 2012/2013

Obsah

1	Úvod.....	1:1
2	Ponuka.....	2:1
2.1	Členovia tímu	2:1
2.2	Motivácia	2:3
2.3	Koncepcia riešenia.....	2:3
2.4	Zoradenie všetkých tém podľa priority	2:4
2.5	Aktuálny rozvrh všetkých členov tímu.....	2:5
3	Plán projektu	3:1
3.1	Organizácia šprintov v zimnom semestri	3:1
3.2	Organizácia šprintov v zimnom semestri	3:1
4	Úlohy členov tímu a ich autorstvo	4:1
4.1	Dlhodobé úlohy	4:1
4.2	Autorstvo častí dokumentácie a inžinierskeho diela	4:3
5	Zápisnice zo stretnutí	5:1
5.1	Zápis zo stretnutia číslo 1	5:1
5.2	Zápis zo stretnutia číslo 2	5:3
5.3	Zápis zo stretnutia číslo 3	5:6
5.4	Zápis zo stretnutia číslo 4	5:9
5.5	Zápis zo stretnutia číslo 5	5:13
5.6	Zápis zo stretnutia číslo 6	5:17
5.7	Zápis zo stretnutia číslo 7	5:19
5.8	Zápis zo stretnutia číslo 8	5:22
5.9	Zápis zo stretnutia číslo 9	5:24
5.10	Zápis zo stretnutia číslo 10	5:26
5.11	Zápis zo stretnutia číslo 11	5:28
5.12	Zápis zo stretnutia číslo 12	5:30
5.13	Zápis zo stretnutia číslo 13	5:34
5.14	Zápis zo stretnutia číslo 14	5:36
5.15	Zápis zo stretnutia číslo 15	5:38
5.16	Zápis zo stretnutia číslo 16	5:40
5.17	Zápis zo stretnutia číslo 17	5:42
5.18	Zápis zo stretnutia číslo 18	5:44

5.19	Zápis zo stretnutia číslo 19	5:46
5.20	Zápis zo stretnutia číslo 20	5:48
5.21	Zápis zo stretnutia číslo 21	5:50
5.22	Zápis zo stretnutia číslo 22	5:52
6	Manažment kvality.....	6:1
6.1	Smerovanie celého projektu, otázka funkčnosti a splnenia požiadaviek.....	6:1
6.2	Explicitné zvyšovanie kvality zdrojového kódu (štábna kultúra, JUnit testy)	6:1
6.3	Štábna kultúra	6:2
6.4	Import štýlov štábnej kultúry do Netbeans.....	6:2
6.5	Pomenovanie tried	6:2
6.6	Pomenovanie atribútov a metód tried.....	6:3
6.7	Odsadenie vnorení a medzery.....	6:3
6.8	Zátvorky {}	6:4
6.9	Identifikátor this.*	6:5
6.10	Komentáre	6:5
6.11	Kompletná vizuálna stránka kódu	6:7
7	Manažment plánovania	7:1
7.1	Plány	7:1
7.2	Priebeh stretnutí tímového projektu	7:4
8	Manažment podpory vývoja.....	8:1
9	Manažment monitorovania	9:1
10	Manažment dokumentácie	10:1
11	Manažment rizík	11:1
11.1	Identifikované riziká.....	11:2
11.2	Vzniknuté riziká	11:5
12	Manažment komunikácie	12:1
12.1	Komunikačné nástroje	12:1
13	Metodiky používané pri vývoji	13:1
13.1	Metodika úloh.....	13:1
13.2	Metodika zberu požiadaviek.....	13:9
13.3	Metodika manažmentu chýb.....	13:16
13.4	Metodika údržby a štábnej kultúry	13:26
13.5	Metodika prehliadky zdrojového kódu.....	13:30

13.6	Metodika manažmentu verzí	13:36
14	Preberacie protokoly	14:1

1 Úvod

Tento dokument obsahuje dokumentáciu k riadeniu tímového projektu Simulácia demonštrácie v meste. Projekt vypracúva tím SimTeam v zložení: Bc. Jana Branišová, Bc. Martin Dušek, Bc. Adrián Kollár, Bc. Michal Kyžňanský, Bc. Michal Ošvát, Bc. Miroslav Ort, Bc. Filip Pakan, v rámci predmetu Tímový projekt. Vedúcim tímu je Ing. Peter Lacko, PhD.

2 Ponuka

2.1 Členovia tímu

Bc. Jana Branišová

Absolvovala bakalárske štúdium na FIIT STU v študijnom programe Informatika. Získala znalosti v programovacích jazykoch Assembler x86, Java, C, C++, HTML, CSS, JavaScript a z Qt frameworku, s ktorým sa oboznámila pri riešení bakalárskej práce s názvom Zhlukovanie v textových dokumentoch. V súčasnosti pracuje v tíme na multiplatformovom projekte pre spoločnosť zaoberajúcu sa vývojom mobilných a webových aplikácií na platforme iOS, Android, Windows 8.

Bc. Adrián Kollár

Prvý stupeň vysokoškolského štúdia absolvoval na FIIT STU v študijnom programe Informatika. Počas štúdia získal vedomosti ako aj praktické skúsenosti vo viacerých programovacích jazykoch, od nízkoúrovňových jazykov až po vysokoúrovňové. Ovláda programovacie jazyky Assembler x86, C, C++ a objektovo orientované jazyky Java a C#. Vypracoval a obhájil bakalársku prácu na tému "Dochádzkový systém". Pri práci na bakalárskom projekte získal skúsenosti so softvérovým používaním RFID čítačky čipových kariet.

Bc. Michal Kyžňanský

Absolvoval bakalárske štúdium na FIIT STU v študijnom programe Informatika. V rámci štúdia získal skúsenosti s programovacími jazykmi Asembler x86, C/C++, Java, C#. Popri štúdiu sa dlhodobo venuje webovým technológiám (PHP, XHTML, JavaScript - jQuery, CSS), návrhu / optimalizácii databáz (MySQL) a spracovaniu grafiky v prostredí Adobe Photoshop. Pracuje ako vývojár webových systémov vo svojej firme, kde súčasne analyzuje a rieši požiadavky zákazníkov. Vypracoval a obhájil bakalársku prácu na tému Simulácia davu, za ktorú dostal vyznamenanie. Pri tejto práci nadobudol skúsenosti s prepojením psychologických aspektov ľudského správania a jeho simulovaním programovacími technikami.

Bc. Miroslav Ort

V roku 2012 ukončil bakalárske štúdium na FIIT STU v študijnom programe Informatika. Počas štúdia získal skúsenosti s programovacími jazykmi Asembler x86, C/C++, C#, Java. Vypracoval a obhájil bakalársku prácu na tému Útoky na detekcie plagiátorstva v textoch. Pri tejto práci nadobudol znalosti a zručnosti s tvorbou experimentov nad stredne rozsiahlym softvérovým systémom. V súčasnosti pracuje ako vývojár v spoločnosti pôsobiacej na trhu medzipodnikových softvérových riešení, kde obohatil svoje teoretické vedomosti o praktické skúsenosti s programovaním v jazyku C/C++ a C# a taktiež sa bližšie oboznámil s prácou v tímoch.

Bc. Michal Ošvát

Ukončil bakalárske štúdium v študijnom programe Informatika na FIIT STU v Bratislave v roku 2012. Počas štúdia sa oboznámil s rôznymi programovacími jazykmi ako Asembler, C, C#, Java, ako aj s CASE nástrojmi a jazykom UML. Okrem toho má skúsenosti so spracovaním a editáciou zvukového záznamu a prácou s grafikou. Obhájená bakalárska práca má názov Viachlasný notový zápis, a bola implementovaná ako desktopová aplikácia v jazyku C#. Jej cieľom je spracovať polyfonické notové zápisy a na základe rôznych dopytov v nich vyhľadávať. Popri štúdiu pracuje ako tester softvéru v globálnej spoločnosti, ktorá poskytuje odborné služby v oblasti manažérskeho poradenstva, systémovej integrácie a outsourcingu.

Bc. Filip Pakan

Vysokoškolské štúdium prvého stupňa absolvoval na FIIT STU v Bratislave v študijnom programe Informatika. V rámci štúdia získal skúsenosti s programovacími jazykmi Asembler x86, C, C++, Java, Pascal, Lisp, Prolog. Záverečnú prácu vypracoval a obhájil na tému Automatické rozpoznávanie objektov inteligentnou kamerou. Počas tejto práce si osvojil základné princípy počítačového videnia a bližšie sa zameral najmä na algoritmy detekcie a rozpoznávania objektov. Pravidelne sa zúčastňuje súťaží v algoritmickom programovaní ACM ICPC a ProFIIT.

Bc. Martin Dušek

Jeho prvý úspech v IT bolo 2. miesto v celoslovenskom kole SOČ v informatike v téme WYSIWYG CMS. Profesionálne pracuje ako web developer PHP už od roku 2009, v roku 2011 zmenil zameranie na jazyk Java. Bakalárske štúdium skončil na FIIT STU v Bratislave. Vypracoval a obhájil bakalársku prácu "Podpora tvorby webových dotazníkov". Má veľké skúsenosti s vývojom veľkých webových projektov od analýzy až po support pre zákazníka.

2.2 Motivácia

Téma simulácie demonštrácie je zaujímavá, pretože predstavuje unikátne prepojenie ťažko uchopiteľných psychologických aspektov ľudského správania so snahou preniesť ich do sveta algoritmov a získať relevantné výstupy. Spracovanie témy ponúka skvelú príležitosť podieľať sa na vývoji prostriedkov, ktoré môžu pomôcť zachrániť ľudské životy v situáciách, ktoré sa opakujú z rôznych príčin po celom svete.

Demonštrácie a rôzne zoskupenia davov sú v dobe ekonomického a sociálneho napätia aktuálnou témou a neraz ľudia prichádzajú o život, prípadne sú zranení len kvôli nedostatočnej organizovanosti poriadkových síl, prípadne prítomnosti manipulátorov (buričov), ktorí chcú zneužiť zhromaždenie na vytvorenie chaosu a násilia. Dôvody zhromaždení môžu byť rôzne (nespokojnosť, hlad, nátlak...), ale emócie v dave ako strach a panika sú spoločným menovateľom všetkých takýchto situácií.

Daná téma predstavuje pre nás výzvu, ktorú by sme radi rozvinuli a prispeli tak k zdokonaleniu chápania hmlistých princípov, ktoré vedú ľudské správanie a poskytnú bezpečnostným zložkám podporné informácie pre lepšie zvládanie menovaných situácií. Vytvoriť verný model odrážajúci správanie demonštrantov a poriadkových síl môže rovnako pomôcť pri návrhu únikových východov pri stavbe štadiónov, budov a iných objektov, kde evakuácia je dôležitým prvkom plánovania.

Simulačné prostredie MASON používal člen nášho tímu a programovací jazyk Java ovládajú všetci členovia. Fakt, že sa jedná o simulovanie dvoch odlišných skupín ľudí, demonštrantov a poriadkových síl môže v jednom modeli zaujímavou ukázať rozdiely v cieľoch týchto skupín, pričom obe sú zmietané pocitmi a pohnútkami vyplývajúcimi zo stresových situácií a zážitkov.

Téma navyše poskytuje perspektívu získať spätnú väzbu priamo od poriadkových zložiek, a tak pracovať s reálnymi údajmi a názormi ľudí, ktorí boli priamo účastníkmi spomenutých situácií.

2.3 Konceptia riešenia

Pri riešení by sme sa chceli opierať o existujúce možnosti simulácie ľudského správania a začať kvalitnou a hĺbkovou analýzou psychologických modelov popisujúcich ľudské správanie, medzi ktoré patrí vyspelý model PECS, prípadne starší BDI. Ďalšou kľúčovou časťou pre správne pochopenie a vytvorenie modelu je využiť dostupné materiály a záznamy o priebehu demonštrácií a odvodiť mechanizmy zodpovedné za prenos emócií v rámci agentov (ľudí) v modelovanom priestore.

Kvalitná analýza pripraví pôdu pre identifikáciu cieľov a pohnútok oboch modelovaných skupín ľudí, pričom bude nasledovať postupné zjemňovanie a pridávanie črt správania a ich zapracovávanie do modelu. Netreba zanedbať aj zdanlivo triviálnu oblasť, ktorou je pohyb ľudí (agentov) a ich vzájomné vyhybanie sa jeden druhému. Každý človek (agent) bude disponovať určitou formou vnútornej reprezentácie sveta, ako ho vníma.

Navrhnuté budú dve skupiny ľudí, ktorí budú stáť proti sebe a budú skúmané parametre ako ich počet, rozmiestnenie, vlastnosti a iné. Postupné rozširovanie schopností agentov reagovať na podnety (oblečenie policajtov, vodné delá atď.) prinesie do modelu nové

úrovne prepracovanosti, čo by sa v konečnom dôsledku malo odraziť na vernosti simulácie ako celku. Dôraz bude kladený aj na jednoduché možnosti zmeny parametrov simulácie.

Jedným z cieľov simulácie bude zistenie optimálneho počtu poriadkových zložiek v závislosti od veľkosti zhromaždeného davu ľudí. Tento parameter je jedným z najvýznamnejších pre zvládnutie demonštrácie ľudí. Príliš málo poriadkových zložiek môže predstavovať riziko, keďže aj pokojná demonštrácia sa môže v priebehu okamihu radikálne zmeniť kvôli rozličným okolnostiam. Naopak príliš veľa poriadkových zložiek nie je vhodným riešením situácie, aj z ekonomického hľadiska a aj z psychologického, keďže veľký počet poriadkových zložiek môže nepriaznivo pôsobiť na zúčastnených ľudí.

Ďalším kľúčovým aspektom pre úspešné zvládnutie demonštrácie je rozmiestnenie poriadkových zložiek. Pri ich rozmiestnení je treba brať do úvahy konkrétne miesto konania demonštrácie. Z toho dôvodu bude simulácia vykonaná vo viacerých bežných mestských prostrediach ako sú námestia alebo ulice. Tieto prostredia je potrebné vymodelovať s rôznymi prekážkami, typickými pre mestské prostredie. V ideálnom prípade, bude v rámci projektu vytvorený aj náhodný generátor prostredí s preddefinovaným súborom prekážok. To zabezpečí dostatočnú variabilitu prostredia a umožní podrobnejšie preskúmanie problematických alebo kritických miest v prostredí konania demonštrácie.

2.4 Zoradenie všetkých tém podľa priority

1.	Simulácia demonštrácie v meste
2.	Odhaľovanie emocionálneho stavu používateľa
3.	Odporúčanie pre inteligentnú TV
4.	Inovatívna počítačová hra
5.	FIIT Kinect
6.	RoboCup - tretí rozmer
7.	Odhaľovanie a hodnotenie vzťahov v oblasti vedy a výskumu
8.	Offline Web

2.5 Aktuálny rozvrh všetkých členov tímu

Meno / Hodina		7	8	9	10	11	12	13	14	15	16	17	18	19	20
Pondelok	Branišová							Preferovaný termín			PROJ_I/ TP2_UISI_I				
	Kollár								PROJ_I/ TP2_UISI_I						
	Kyžňanský								PROJ_I/ TP2_UISI_I						
	Ort				EA_I				PROJ_I/ TP2_UISI_I						
	Ošvát								PROJ_I/ TP2_UISI_I						
	Pakan				EA_I				PROJ_I/ TP2_UISI_I						
Meno / Hodina		7	8	9	10	11	12	13	14	15	16	17	18	19	20
Útorok	Branišová			OZNAL_I				Preferovaný termín			AASS_I	AASS_I			
	Kollár			OZNAL_I					AASS_I	AASS_I					
	Kyžňanský			OZNAL_I					AASS_I	AASS_I					
	Ort			EA_I											
	Ošvát										AASS_I			AASS_I	
	Pakan	EA_I									AASS_I	AASS_I			
Meno / Hodina		7	8	9	10	11	12	13	14	15	16	17	18	19	20
Streda	Branišová														
	Kollár														
	Kyžňanský														
	Ort		DPRS_I						DPRS_I						
	Ošvát								PVID_I						
	Pakan														
Meno / Hodina		7	8	9	10	11	12	13	14	15	16	17	18	19	20
Štvrtok	Branišová		OZNAL_I												
	Kollár		OZNAL_I												
	Kyžňanský		OZNAL_I												
	Ort														
	Ošvát				PVID_I										
	Pakan														
Meno / Hodina		7	8	9	10	11	12	13	14	15	16	17	18	19	20
Piatok	Branišová														
	Kollár														
	Kyžňanský														
	Ort														
	Ošvát														
	Pakan														

3 Plán projektu

Vývoj nášho produktu realizujeme agilnou metódou SCRUM, teda máme priebeh projektu organizovaný do šprintov. Šprint trvá v našom projekte dva týždne, celkovo vychádzajú štyri šprinty v zimnom semestri a tri v letnom.

3.1 Organizácia šprintov v zimnom semestri

Týždeň semestra

1. prezentácia tém, odovzdanie ponúk
2. uchádzanie sa o témy, pridelenie tém
3. rozdelenie úloh, plán projektu, analýza problému
4. inicializácia riešenia projektu: vytvorenie Product Backlog, prvý dátový model, nastavenie prostredia pre vývoj a nasadzovanie, vytvorenie kostry projektu
5. práca na príbehoch šprintu č.1-1Z
6. ukončenie šprintu č.1-1Z
7. práca na príbehoch šprintu č.2-2Z
8. ukončenie šprintu č.2-2Z a odovzdanie dokumentácie k produktu - prvé dva šprinty, big picture a dokumentácie k riadeniu
9. práca na príbehoch šprintu č.3-3Z
10. ukončenie šprintu č.3-3Z
11. práca na príbehoch šprintu č.4-4Z
12. ukončenie šprintu č.4-4Z a odovzdanie

3.2 Organizácia šprintov v zimnom semestri

Týždeň semestra

1. práca na príbehoch šprintu č.5-1L
2. práca na príbehoch šprintu č.5-1L
3. ukončenie šprintu č.5-1L
4. práca na príbehoch šprintu č.6-2L
5. ukončenie šprintu č.6-2L
6. práca na príbehoch šprintu č.7-3L
7. ukončenie šprintu č.7-3L
8. práca na príbehoch šprintu č.8-4L
9. ukončenie šprintu č.8-4L odovzdanie produktu a dokumentácie k šprintom 5 až 8
10. práca na príbehoch šprintu č.9-5L
11. ukončenie šprintu č.9-5L
12. práca a ukončenie šprintu č.10-6L a odovzdanie celkového výsledku projektu (produkt, riadenie)

4 Úlohy členov tímu a ich autorstvo

4.1 Dlhodobé úlohy

- Bc. Miroslav Ort
 - Vedúci tímu
 - Riadi tím
 - Manažér komunikácie
 - Koordinuje diskusie
 - Určuje komunikačné nástroje
 - Organizuje stretnutia
- Bc. Jana Branišová
 - Manažér plánovania
 - Určuje priority
 - Dozerá nad plnením úloh
 - Navrhuje nasledujúce úlohy
 - Spravuje nástroj JIRA
- Bc. Adrián Kollár
 - Manažér monitorovania
 - Monitorovanie postupu prác
 - Monitorovanie zdrojových kódov
- Bc. Michal Kyžňanský
 - Manažér kvality
 - Kontrola smerovania celého projektu
 - Rozhoduje v otázkach funkčnosti
 - Explicitné zvyšovanie kvality zdrojového kódu
 - Správca webstránky
 - Aktualizácia webstránky

- Bc. Filip Pakan
 - Manažér rizík
 - Riadenie a sledovanie rizík
 - Identifikácia rizík
 - Analýza rizík
 - Plánovanie manažmentu rizík
- Bc. Michal Ošvát
 - Manažér podpory vývoja
 - Analyzuje a navrhuje použitie nástrojov na podporu vývoja
 - Spravuje SVN repozitár a webový server
- Bc. Martin Dušek
 - Manažér dokumentácie
 - Zodpovedný za kvalitu dokumentácie
 - Úloha udržovania konzistencie v dokumentácii
 - Kompletizácia dokumentácie od ostatných členov tímu

4.2 Autorstvo častí dokumentácie a inžinierskeho diela

4.2.1 Dokumentácia k inžinierskemu dielu

Integrátor dokumentácie: Bc. Miroslav Ort

- Bc. Miroslav Ort
 - Kapitola 2.10
 - Kapitola 2.11
 - Kapitola 2.12
 - Kapitola 3.7
 - Kapitola 4.3
 - Kapitola 5.3
 - Kapitola 6.2
 - Kapitola 7.5
 - Kapitola 8.1
 - Kapitola 9.3
- Bc. Jana Branišová
 - Kapitola 2.5
 - Kapitola 2.6
 - Kapitola 2.10
 - Kapitola 2.12
 - Kapitola 3.1
 - Kapitola 4.3
 - Kapitola 5.4
 - Kapitola 8.2
- Bc. Adrián Kollár
 - Kapitola 2.1
 - Kapitola 2.11
 - Kapitola 3.2
 - Kapitola 4.4
 - Kapitola 5.2
 - Kapitola 6.1
 - Kapitola 7.2
- Bc. Michal Kyžňanský
 - Kapitola 2.7

- Kapitola 2.8
- Kapitola 2.11
- Kapitola 3.3
- Kapitola 3.4
- Kapitola 4.1
- Kapitola 4.2
- Kapitola 6.3
- Kapitola 7.1
- Bc. Filip Pakan
 - Kapitola 2.3
 - Kapitola 2.4
 - Kapitola 2.11
 - Kapitola 4.4
 - Kapitola 5.1
 - Kapitola 5.2
 - Kapitola 7.3
 - Kapitola 9.1
 - Kapitola 9.2
 - Kapitola 10.1
- Bc. Michal Ošvát
 - Kapitola 2.9
 - Kapitola 3.5
 - Kapitola 3.6
 - Kapitola 4.3
 - Kapitola 5.4
 - Kapitola 7.4
- Bc. Martin Dušek

4.2.2 Dokumentácia riadenia projektu

Integrátor dokumentácie: Bc. Miroslav Ort

- Bc. Miroslav Ort
 - Kapitola 12
- Bc. Jana Branišová

- Kapitola 7
- Bc. Adrián Kollár
 - Kapitola 9
- Bc. Michal Kyžňanský
 - Kapitola 6
- Bc. Filip Pakan
 - Kapitola 11
- Bc. Michal Ošvát
 - Kapitola 8
- Bc. Martin Dušek
 - Kapitola 4
 - Kapitola 10

5 Zápisnice zo stretnutí

5.1 Zápis zo stretnutia číslo 1

Dátum: 8.10.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
Členovia tímu: Bc. Jana Branišová
Bc. Martin Dušek
Bc. Adrián Kollár
Bc. Michal Kyžňanský
Bc. Miroslav Ort
Bc. Michal Ošvát
Bc. Filip Pakan

Autor zápisu: Bc. Filip Pakan

5.1.1 Diskutované témy:

- Oboznámenie sa s projektom
- Priebeh práce na projekte
- Metrika pre určenie správnosti simulácie
- Rozdelenie tímu na dve skupiny
- Skupina zameraná na simuláciu demonštrantov
- Skupina zameraná na simuláciu poriadkových zložiek
- Projektový denník
- Utužovanie tímu

5.1.2 Prijaté rozhodnutia:

- Vyvíjať budeme agilne pomocou metodiky SCRUM
- Ako integrované vývojové prostredie sme sa rozhodli použiť NetBeans
- Každé vyvinuté úsilie a každú činnosť budeme dokumentovať
- Všetci si založíme projektový denník
- Súťaže TP Cup sa nezúčastníme
- Ako systém pre riadenie verzií sme si vybrali SVN

5.1.3 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
1.1	Vytvoriť šablóny na všetky druhy dokumentácií	Martin Dušek	15.10.2012	Pridelená
1.2	Nainštalovanie operačného systému Linux na virtuálny stroj	Michal Ošvát	15.10.2012	Pridelená
1.3	Nainštalovanie webového servera Apache	Michal Ošvát	15.10.2012	Pridelená
1.4	Nainštalovanie SVN servera	Michal Ošvát	15.10.2012	Pridelená
1.5	Vytvorenie používateľských účtov pre všetkých členov tímu na virtuálnom serveri	Michal Ošvát	15.10.2012	Pridelená
1.6	Vytvorenie webovej stránky tímu	Michal Kyžňanský	15.10.2012	Pridelená
1.7	Pozvať vedúceho tímu do Google Group	Jana Branišová	15.10.2012	Pridelená
1.8	Nastaviť Google Group pre prijímanie e-mailov z vonka	Jana Branišová	15.10.2012	Pridelená
1.9	Analyzovať možnosti importovania mapových podkladov z OpenStreetMap do knižnice MASON	Filip Pakan	15.10.2012	Pridelená
1.10	Knižnicu pre chodenie agentov prekompilovať vo Visual Studiu	Miroslav Ort	15.10.2012	Pridelená
1.11	Pripraviť plán projektu	Jana Branišová	15.10.2012	Pridelená
1.12	Analyzovať podporu SVN v NetBeans a napísať návod	Adrián Kollár	15.10.2012	Pridelená
1.13	Analyzovať softvér pre podporu riadenia projektov Redmine	Jana Branišová	15.10.2012	Pridelená

5.2 Zápis zo stretnutia číslo 2

Dátum: 15.10.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.

Členovia tímu: Bc. Jana Branišová
 Bc. Martin Dušek
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.2.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	Popis úlohy	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
1.1	Vytvoriť šablóny na všetky druhy dokumentácií	Martin Dušek	15.10.2012	Dokončená
1.2	Nainštalovanie operačného systému Linux na virtuálny stroj	Michal Ošvát	15.10.2012	Dokončená
1.3	Nainštalovanie webového servera Apache	Michal Ošvát	15.10.2012	Dokončená
1.4	Nainštalovanie SVN servera	Michal Ošvát	15.10.2012	Nedokončená
1.5	Vytvorenie používateľských účtov pre všetkých členov tímu na virtuálnom serveri	Michal Ošvát	15.10.2012	Dokončená
1.6	Vytvorenie webovej stránky tímu	Michal Kyžňanský	15.10.2012	Dokončená
1.7	Pozvať vedúceho tímu do Google Group	Jana Branišová	15.10.2012	Dokončená
1.8	Nastaviť Google Group pre prijímanie e-mailov z vonka	Jana Branišová	15.10.2012	Dokončená
1.9	Analyzovať možnosti importovania mapových podkladov z OpenStreetMap do knižnice MASON	Filip Pakan	15.10.2012	Dokončená

1.10	Knižnicu pre chodenie agentov prekompilovať vo Visual Studiu	Miroslav Ort	15.10.2012	Dokončená
1.11	Pripraviť plán projektu	Jana Branišová	15.10.2012	Dokončená
1.12	Analyzovať podporu SVN v NetBeans a napísať návod	Adrián Kollár	15.10.2012	Dokončená
1.13	Analyzovať softvér pre podporu riadenia projektov Redmine	Jana Branišová	15.10.2012	Dokončená

5.2.2 Diskutované témy:

- Priebeh práce na projekte
- Používanie softvérov na podporu riadenia projektov
- Systémy pre správu dokumentov
- Zdieľanie materiálov k téme projektu
- Tvorba backlog-u
- Projektový denník
- Intergrácia knižníc do projektu

5.2.3 Prijaté rozhodnutia:

- Budeme používať Apache Subversion systém na správu dokumentov
- Rozhodli sme sa používať systém na podporu riadenia projektov Jira
- GeoMason obohatíme o knižnicu OGR pre podporu GIS dátových formátov
- Na zabezpečenie výhýbania sa agentov budeme používať knižnicu RVO2

5.2.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
2.1	Vytvorenie prvého scenára pre triviálnych agentov	Michal Kyžňanský	22.10.2012	Pridelená
2.2	Integrácia knižníc MASON a RVO2	Michal Kyžňanský	22.10.2012	Pridelená
2.3	Import mapových podkladov do knižnice MASON	Filip Pakan	22.10.2012	Pridelená
2.4	Integrácia MASON, RVO2 s mapovými podkladmi	Michal Kyžňanský, Filip Pakan	22.10.2012	Pridelená
2.5	Analýza simulácie demonštrácií	Miroslav Ort, Jana Branišová	22.10.2012	Pridelená
2.6	Analýza metrík na určenie správnosti simulácie	Miroslav Ort, Jana Branišová	22.10.2012	Pridelená

2.7	Návrh scenárov	Miroslav Ort, Jana Branišová	22.10.2012	Pridelená
2.8	Vytvorenie backlog-u v JIRA-e	Jana Branišová	22.10.2012	Pridelená
2.9	Analýza poriadkových zložiek	Michal Ošvát	22.10.2012	Pridelená
2.10	Nainštalovanie SVN servera	Michal Ošvát	22.10.2012	Pridelená
2.11	Vyskúšať funkčnosť NetBeans s SVN serverom	Adrián Kollár, Michal Ošvát	22.10.2012	Pridelená
2.12	Analyzovať modely správania	Všetci	22.10.2012	Pridelená
2.13	Analyzovať šírenie emócií	Všetci	22.10.2012	Pridelená

5.3 Zápis zo stretnutia číslo 3

Dátum: 22.10.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Martin Dušek
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Jana Branišová

5.3.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
2.1	Vytvorenie prvého scenára pre triviálnych agentov	Michal Kyžňanský	22.10.2012	Dokončená
2.2	Integrácia knižníc MASON a RVO2	Michal Kyžňanský	22.10.2012	Znova pridelená
2.3	Import mapových podkladov do knižnice MASON	Filip Pakan	22.10.2012	Dokončená
2.4	Integrácia MASON, RVO2 s mapovými podkladmi	Michal Kyžňanský, Filip Pakan	22.10.2012	Znova pridelená
2.5	Analýza simulácie demonštrácií	Miroslav Ort, Jana Branišová	22.10.2012	Znova pridelená
2.6	Analýza metrík na určenie správnosti simulácie	Miroslav Ort, Jana Branišová	22.10.2012	Znova pridelená
2.7	Návrh scenárov	Miroslav Ort, Jana Branišová	22.10.2012	Znova pridelená
2.8	Vytvorenie backlog-u v JIRA-e	Jana Branišová	22.10.2012	Dokončená
2.9	Analýza poriadkových zložiek	Michal Ošvát	22.10.2012	Znova pridelená
2.10	Nainštalovanie SVN servera	Michal Ošvát, Adrián Kollár	22.10.2012	Dokončená

2.11	Vyskúšať funkčnosť NetBeans s SVN serverom	Adrián Kollár, Michal Ošvát	22.10.2012	Dokončená
2.12	Analyzovať modely správania	Všetci	22.10.2012	Znova pridelená
2.13	Analyzovať šírenie emócií	Všetci	22.10.2012	Znova pridelená

5.3.2 Diskutované témy:

- Priebeh práce na projekte
- Používanie softvérov na podporu riadenia projektov
- Šablóny a ich naplnenie
- Výsledky analýzy
- Intergrácia knižníc do projektu

5.3.3 Prijaté rozhodnutia:

- Použijeme GIS, ktorý dokáže stiahnuté OSM XML súbory s mapovými podkladmi prekonvertovať do ESRI shapefile súborov.
- Rozhodli sme sa, že navrhne scenáre pre simuláciu spočiatku jednoduché a potom k nim budeme postupne pridávať ďalšiu funkcionálnosť
- Rozhodli sme sa, že do projektovej dokumentácie umiestnime všetky ponuky.

5.3.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
3.1	Vytvorenie prvého scenára pre triviálnych agentov	Michal Kyžňanský	29.10.2012	Pridelená
3.2	Analyzovať ako spolupracuje NetBeans a SVN	Adrián Kollár	29.10.2012	Pridelená
3.3	Analyzovať spôsoby prezerania commitov	Adrián Kollár	29.10.2012	Pridelená
3.4	Vymyslieť user story na ďalší šprint a zapísať ich do Jiri	Všetci	29.10.2012	Pridelená
3.5	Inštalácia MASON do eclipse	Všetci	29.10.2012	Pridelená
2.2	Integrácia knižníc MASON a RVO2	Michal Kyžňanský	22.10.2012	Znova pridelená
2.4	Integrácia MASON, RVO2 s mapovými podkladmi	Michal Kyžňanský, Filip Pakan	29.10.2012	Znova pridelená
2.5	Analýza simulácie demonštrácií	Miroslav Ort, Jana Branišová	29.10.2012	Znova pridelená
2.6	Analýza metrík na určenie správnosti simulácie	Miroslav Ort, Jana Branišová	29.10.2012	Znova pridelená

2.7	Návrh scenárov	Miroslav Ort, Jana Branišová	29.10.2012	Znova pridelená
2.9	Analýza poriadkových zložiek	Michal Ošvát	29.10.2012	Znova pridelená
2.12	Analyzovať modely správania	Všetci	29.10.2012	Znova pridelená
2.13	Analyzovať šírenie emócií	Všetci	29.10.2012	Znova pridelená

5.4 Zápis zo stretnutia číslo 4

Dátum: 29.10.2012
Čas: 13:00
Dĺžka trvania: 180 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Martin Dušek
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Adrián Kollár

5.4.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
3.1	Vytvorenie prvého scenára pre triviálnych agentov	Michal Kyžňanský	29.10.2012	Dokončená
3.2	Analyzovať ako spolupracuje NetBeans a SVN	Adrián Kollár	29.10.2012	Dokončená
3.3	Analyzovať spôsoby prezerania commitov	Adrián Kollár	29.10.2012	Dokončená
3.4	Vymyslieť user story na ďalší šprint a zapísať ich do Jiri	Všetci	29.10.2012	Dokončená
3.5	Inštalácia MASON do Netbeans	Všetci	29.10.2012	Dokončená
2.2	Integrácia knižníc MASON a RVO2	Michal Kyžňanský	22.10.2012	Dokončená
2.4	Integrácia MASON, RVO2 s mapovými podkladmi	Michal Kyžňanský, Filip Pakan	29.10.2012	Znova pridelená
2.5	Analýza simulácie demonštrácií	Miroslav Ort, Jana Branišová	29.10.2012	Dokončená
2.6	Analýza metrík na určenie správnosti simulácie	Miroslav Ort, Jana Branišová	29.10.2012	Dokončená
2.7	Návrh scenárov	Miroslav Ort, Jana Branišová	29.10.2012	Dokončená
2.9	Analýza poriadkových zložiek	Michal Ošvát	29.10.2012	Dokončená

2.12	Analyzovať modely správania	Všetci	29.10.2012	Dokončená
2.13	Analyzovať šírenie emócií	Všetci	29.10.2012	Dokončená

5.4.2 Diskutované témy:

- Priebeh práce na projekte
- Výsledky analýzy
- Projektová dokumentácia
- Tvorba šprint backlogu
- Možnosti spolupráce s druhým tímom

5.4.3 Prijaté rozhodnutia:

- Všetci členovia budú implementovať druhý scenár aby sa každý zoznámil s prostredím Mason a knižnicou RVO
- Za účelom dohodnutia možnej spolupráce s iným tímom sa uskutoční osobné stretnutie členov tímov
- Z dôvodu zjednodušenia tvorby zápisníc a kontroly postupu prác na jednotlivých úlohách bude zmenená tabuľka pridelených úloh zápisnice. Jednotlivé úlohy budú odteraz exportované priamo zo systému Jira.

5.4.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
2.4	Vyextrahovanie máp z projektu SAV a integrácia spolu s RVO	Michal Kyžňanský	3.11.2012	Znova pridelená
4.1	Analýza tagov a branchov v SVN	Michal Ošvát	1.11.2012	Pridelená
4.2	Analýza fyzikálnych veličiny ako je tlaku pôsobiacich na človeka	Jana Branišová	2.11.2012	Pridelená
4.3	Analýza koordinácie poriadkových zložiek	Michal Ošvát	5.11.2012	Pridelená
4.4	Návod na rozbehanie SVN	Adrián Kollár	3.11.2012	Pridelená
4.5	Dokumentácia k integrácii RVO a Máp	Michal Kyžňanský	3.11.2012	Pridelená
4.6	Dokumentácia k riadeniu - Manažment kvality	Michal Kyžňanský	5.11.2012	Pridelená
4.7	Dokumentácia k riadeniu - Manažment rizík	Filip Pakan	5.11. 2012	Pridelená
4.8	Dokumentácia k riadeniu - Manažment rozvrhu a plánovania	Jana Branišová	5.11.2012	Pridelená
4.9	Dokumentácia k riadeniu - Manažment podpory vývoja	Michal Ošvát	5.11.2012	Pridelená
4.10	Dokumentácia k riadeniu - Monitorovanie projektu	Adrián Kollár	5.11. 2012	Pridelená
4.11	Dokumentácia k riadeniu - Manažment komunikácie	Miroslav Ort	5.11.2012	Pridelená
4.12	Dokumentácia k riadeniu - Manažment tvorby dokumentácie	Martin Dušek	5.11.2012	Pridelená
4.13	Dokumentácie k analýze v prvom šprinte	Martin Dušek	5.11.2012	Pridelená
4.14	Dokumentácia k výberu metrík na meranie/validovanie simulácie	Miroslav Ort	5.11. 2012	Pridelená
4.15	Realtime zmena parametrov RVO + zmena agentov	Michal Kyžňanský	7.11.2012	Pridelená
4.16	Implementácia scenára pre druhý šprint	Jana Branišová	5.11.2012	Pridelená
4.17	Implementácia scenára pre druhý šprint	Michal Ošvát	5.11.2012	Pridelená
4.18	Implementácia scenára pre druhý šprint	Adrián Kollár	5.11.2012	Pridelená
4.19	Implementácia scenára pre druhý šprint	Martin Dušek	5.11.2012	Pridelená
4.20	Implementácia scenára pre druhý šprint	Filip Pakan	5.11.2012	Pridelená

4.21	Implementácia scenára pre druhý šprint	Miroslav Ort	5.11.2012	Pridelená
4.22	Výber modelu správania	Peter Lacko	2.11.2012	Pridelená
4.23	Scenár pre ďalší šprint	Miroslav Ort	5.11.2012	Pridelená

5.5 Zápis zo stretnutia číslo 5

Dátum: 5.11.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Martin Dušek
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Ošvát

5.5.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
2.4	Vyextrahovanie máp z projektu SAV a integrácia spolu s RVO	Michal Kyžňanský	3.11.2012	Dokončená
4.1	Analýza tagov a branchov v SVN	Michal Ošvát	1.11.2012	Dokončená
4.2	Analýza fyzikálnych veličiny ako je tlaku pôsobiacich na človeka	Jana Branišová	2.11.2012	Znova pridelená
4.3	Analýza koordinácie poriadkových zložiek	Michal Ošvát	5.11.2012	Znova pridelená
4.4	Návod na rozbehanie SVN	Adrián Kollár	3.11.2012	Dokončená
4.5	Dokumentácia k integrácii RVO a Máp	Michal Kyžňanský	3.11.2012	Dokončená
4.6	Dokumentácia k riadeniu - Manažment kvality	Michal Kyžňanský	5.11.2012	Dokončená
4.7	Dokumentácia k riadeniu - Manažment rizík	Filip Pakan	5.11. 2012	Dokončená
4.8	Dokumentácia k riadeniu - Manažment rozvrhu a plánovania	Jana Branišová	5.11.2012	Dokončená
4.9	Dokumentácia k riadeniu - Manažment podpory vývoja	Michal Ošvát	5.11.2012	Dokončená
4.10	Dokumentácia k riadeniu - Monitorovanie projektu	Adrián Kollár	5.11. 2012	Dokončená

4.11	Dokumentácia k riadeniu - Manažment komunikácie	Miroslav Ort	5.11.2012	Dokončená
4.12	Dokumentácia k riadeniu - Manažment tvorby dokumentácie	Martin Dušek	5.11.2012	Dokončená
4.13	Dokumentácie k analýze v prvom šprinte	Martin Dušek	5.11.2012	Znova pridelená
4.14	Dokumentácia k výberu metrík na meranie/validovanie simulácie	Miroslav Ort	5.11. 2012	Dokončená
4.15	Realtime zmena parametrov RVO + zmena agentov	Michal Kyžňanský	7.11.2012	Dokončená
4.16	Implementácia scenára pre druhý šprint	Jana Branišová	5.11.2012	Znova pridelená
4.17	Implementácia scenára pre druhý šprint	Michal Ošvát	5.11.2012	Znova pridelená
4.18	Implementácia scenára pre druhý šprint	Adrián Kollár	5.11.2012	Znova pridelená
4.19	Implementácia scenára pre druhý šprint	Martin Dušek	5.11.2012	Znova pridelená
4.20	Implementácia scenára pre druhý šprint	Filip Pakan	5.11.2012	Znova pridelená
4.21	Implementácia scenára pre druhý šprint	Miroslav Ort	5.11.2012	Znova pridelená
4.22	Výber modelu správania	Peter Lacko	2.11.2012	Znova pridelená
4.23	Scenár pre ďalší šprint	Miroslav Ort	5.11.2012	Znova pridelená

5.5.2 Diskutované témy:

- Priebeh práce na projekte
- Funkčnosť základnej implementácie
- Pridelenie názvom stretnutiam
- Výsledky analýzy
- Projektová dokumentácia
- Možnosti spolupráce s druhým tímom

5.5.3 Prijaté rozhodnutia:

- Tím dohodne na nasledujúcom stretnutí na termíne „code review“, kde bude prebiehať diskusia ohľadom implementácie.
- Možná spolupráca s iným tímom prebehne po dohode vo forme osobného stretnutia a následnej vzájomnej diskusie.
- Ak bude členovi tímu pridelená úloha, ktorú je potrebné dokončiť skôr ako do nasledovného stretnutia, bude termín dokončenia po vzájomnej dohode určený už na danom stretnutí.

5.5.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
4.2	Analýza fyzikálnych veličiny ako je tlaku pôsobiacich na človeka	Jana Branišová	12.11.2012	Znova pridelená
4.3	Analýza koordinácie poriadkových zložiek	Michal Ošvát	12.11.2012	Znova pridelená
4.13	Dokumentácie k analýze v prvom šprinte	Martin Dušek	9.11.2012	Znova pridelená
4.16	Implementácia scenára pre druhý šprint	Jana Branišová	12.11.2012	Znova pridelená
4.17	Implementácia scenára pre druhý šprint	Michal Ošvát	12.11.2012	Znova pridelená
4.18	Implementácia scenára pre druhý šprint	Adrián Kollár	12.11.2012	Znova pridelená
4.19	Implementácia scenára pre druhý šprint	Martin Dušek	12.11.2012	Znova pridelená
4.20	Implementácia scenára pre druhý šprint	Filip Pakan	12.11.2012	Znova pridelená
4.21	Implementácia scenára pre druhý šprint	Miroslav Ort	12.11.2012	Znova pridelená
4.22	Výber modelu správania	Peter Lacko	12.11.2012	Znova pridelená

4.23	Scenár pre ďalší šprint	Miroslav Ort	12.11.2012	Znova pridelená
5.1	Analýza názvov historicky významných protestov	Miroslav Ort	12.11.2012	Pridelená
5.2	Finalizácia dokumentácie k produktu	Martin Dušek	9.11.2012	Pridelená
5.3	Vytvorenie dokumentácie k riadeniu	Martin Dušek	9.11.2012	Pridelená

5.6 Zápis zo stretnutia číslo 6

Dátum: 12.11.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Martin Dušek
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Kyžňanský

5.6.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
4.2	Analýza fyzikálnych veličiny ako je tlaku pôsobiacich na človeka	Jana Branišová	12.11.2012	Dokončená
4.3	Analýza koordinácie poriadkových zložiek	Michal Ošvát	12.11.2012	Dokončená
4.13	Dokumentácie k analýze v prvom šprinte	Martin Dušek	9.11.2012	Nesplnená
4.16	Implementácia scenára pre druhý šprint	Jana Branišová	12.11.2012	Dokončená
4.17	Implementácia scenára pre druhý šprint	Michal Ošvát	12.11.2012	Dokončená
4.18	Implementácia scenára pre druhý šprint	Adrián Kollár	12.11.2012	Dokončená
4.19	Implementácia scenára pre druhý šprint	Martin Dušek	12.11.2012	Nesplnená
4.20	Implementácia scenára pre druhý šprint	Filip Pakan	12.11.2012	Dokončená
4.21	Implementácia scenára pre druhý šprint	Miroslav Ort	12.11.2012	Dokončená
4.22	Výber modelu správania	Peter Lacko	12.11.2012	Dokončená
4.23	Scenár pre ďalší šprint	Miroslav Ort	12.11.2012	Dokončená

5.1	Analýza názvov historicky významných protestov	Miroslav Ort	12.11.2012	Dokončená
5.2	Finalizácia dokumentácie k produktu	Martin Dušek	9.11.2012	Nesplnená
5.3	Vytvorenie dokumentácie k riadeniu	Martin Dušek	9.11.2012	Nesplnená

5.6.2 Diskutované témy:

- Priebeh práce na projekte
- Opakovane nesplnené úlohy člena tímu (Martin Dušek)
- Úlohy na 3. šprint a ich rozdelenie
- Výsledky analýzy pôsobenia tlaku na človeka v dave
- Modely správania a ich možná implementácia
- Riziko absencie dvoch kolegov na stretnutí nasledujúci týždeň
- Projektová dokumentácia

5.6.3 Prijaté rozhodnutia:

- Tím sa dohodol na výbere modelu správania, ktorým bude PECS v kombinácii so SimPAN.
- Spolupráca s iným tímom bola slovne vymedzená na oblasť validácie simulácie v podobe výmeny informácií ohľadne benchmarku, pričom zdieľanie zdrojového kódu nie je akceptovaná a žiadaná forma spolupráce.
- Náš tím po konzultácii s Petrom Lackom opustil Martin Dušek na základe nedostatočného dokladovania výsledkov práce a nesplnenia viacerých pridelených úloh.

5.6.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.1	Refactoring a úprava kódu projektu	Michal Kyžňanský	26.11.2012	Pridelená
6.2	PECS objektový návrh a realizácia	Michal Kyžňanský	26.11.2012	Pridelená
6.3	PECS objektový návrh a realizácia	Miroslav Ort	26.11.2012	Pridelená
6.4	Analýza plánovania trasy	Filip Pakan	26.11.2012	Pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	26.11.2012	Pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	26.11.2012	Pridelená
6.7	Analýza pôsobenia tlaku + príklad/model	Michal Ošvát	26.11.2012	Pridelená

5.7 Zápis zo stretnutia číslo 7

Dátum: 19.11.2012
Čas: 13:00
Dĺžka trvania: 80 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Filip Pakan

5.7.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelení členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.1	Refactoring a úprava kódu projektu	Michal Kyžňanský	26.11.2012	Znova pridelená
6.2	PECS objektový návrh a realizácia	Michal Kyžňanský	26.11.2012	Dokončená
6.3	PECS objektový návrh a realizácia	Miroslav Ort	26.11.2012	Dokončená
6.4	Analýza plánovania trasy	Filip Pakan	26.11.2012	Znova pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	26.11.2012	Znova pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	26.11.2012	Znova pridelená
6.7	Analýza pôsobenia tlaku + príklad/model	Michal Ošvát	26.11.2012	Znova pridelená

5.7.2 Diskutované témy:

- Názvy šprintov
- Nahradenie odídeného člena tímu Martina Duška
- Priebeh projektu
- Integrovanie projektovej dokumentácie

5.7.3 Prijaté rozhodnutia:

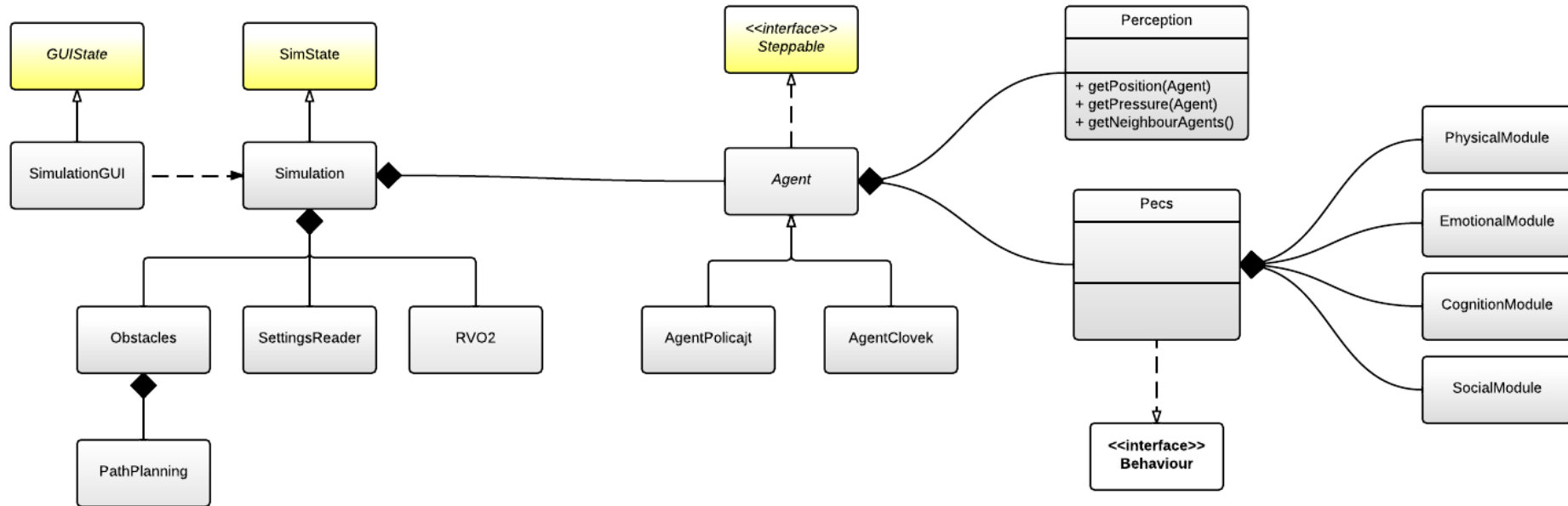
- Šprinty budeme pomenovávať podľa známych historických povstaní a demonštrácií
- Predošlé šprinty nebudeme retrospektívne pomenovávať
- Po odchode Martina Duška bude dokumentáciu integrovať Miroslav Ort za výdatnej podpory členov tímu
- Všetci budeme písať svoje texty dokumentácie do predpripravenej šablóny

5.7.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.1	Refactoring a úprava kódu projektu	Michal Kyžňanský	26.11.2012	Znova pridelená
6.4	Analýza plánovania trasy	Filip Pakan	26.11.2012	Znova pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	26.11.2012	Znova pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	26.11.2012	Znova pridelená
6.7	Analýza pôsobenia tlaku + príklad/model + prototyp	Michal Ošvát	26.11.2012	Znova pridelená
7.1	Analýza pôsobenia tlaku + príklad/model + prototyp	Miroslav Ort	26.11.2012	Pridelená
7.2	Využiť všetky možnosti systému JIRA a jeho pluginov pre tvorbu šprintov	Jana Branišová	26.11.2012	Pridelená

5.7.5 Přílohy

Objektový návrh UML:



5.8 Zápis zo stretnutia číslo 8

Dátum: 26.11.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.8.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.1	Refactoring a úprava kódu projektu	Michal Kyžňanský	26.11.2012	Dokončená
6.4	Analýza plánovania trasy	Filip Pakan	26.11.2012	Znova pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	26.11.2012	Znova pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	26.11.2012	Znova pridelená
6.7	Analýza pôsobenia tlaku + príklad/model + prototyp	Michal Ošvát	26.11.2012	Znova pridelená
7.1	Analýza pôsobenia tlaku + príklad/model + prototyp	Miroslav Ort	26.11.2012	Znova pridelená
7.2	Využiť všetky možnosti systému JIRA a jeho pluginov pre tvorbu šprintov	Jana Branišová	26.11.2012	Dokončená

5.8.2 Diskutované témy:

- Problémy s rozšírením GeoMASON

- Nutnosť ďalšej implementácie pri dodržaní jednotného scenára, v ktorom budú dva typy agentov, poriadkové zložky a demonštranti, pričom poriadkové zložky budú nehybne stáť a demonštranti sa budú snažiť cez ne dostať
- Analýza pôsobenia tlaku a síl na poriadkové zložky a demonštrantov
- Logovanie stavu agenta
- Metriky simulácie

5.8.3 Prijaté rozhodnutia:

- Nebudeme používať GeoMASON, nahradíme ho obrázkom mapy, ktorá bude zodpovedať lokalite demonštrácie
- Metriky simulácie budeme zvažovať neskôr
- Vytvoríme prvý scenár, v ktorom budeme uvažovať pôsobenie demonštrantov na poriadkové zložky silou
- Od štvrtého šprintu budeme v Jire používať možnosti pluginu GreenHopper, predošlé šprinty ostanú bezo zmeny, pretože na migráciu by bolo nutné zálohovať dosiaľ vytvorené úlohy a na zálohovanie nemáme v systéme právomoc

5.8.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.4	Analýza plánovania trasy	Filip Pakan	3.12.2012	Znova pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	3.12.2012	Znova pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	3.12.2012	Znova pridelená
6.7	Analýza pôsobenia tlaku + príklad/model + prototyp	Michal Ošvát	3.12.2012	Znova pridelená
7.1	Analýza pôsobenia tlaku + príklad/model + prototyp	Miroslav Ort	3.12.2012	Znova pridelená
8.1	Nahradiť GeoMASON	Michal Kyžňanský	1.12.2012	Pridelená
8.2	Vytvoriť metódu na logovanie stavu agenta	Adrián Kollár	3.12.2012	Pridelená
8.3	Vytvoriť metódu na logovanie stavu agenta	Filip Pakan	3.12.2012	Pridelená
8.4	Integrácia dokumentácie	Miroslav Ort	3.12.2012	Pridelená

5.9 Zápis zo stretnutia číslo 9

Dátum: 3.12.2012
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Jana Branišová

5.9.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.1	Refaktoring a úprava kódu projektu	Michal Kyžňanský	3.12.2012	Znovu pridelená
6.4	Analýza plánovania trasy	Filip Pakan	3.12.2012	Znovu pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	3.12.2012	Znovu pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	3.12.2012	Znovu pridelená
6.7	Analýza pôsobenia tlaku + príklad/model	Michal Ošvát	3.12.2012	Znovu pridelená
7.1	Analýza pôsobenia tlaku + príklad/model	Miroslav Ort	3.12.2012	Znovu pridelená
8.1	Modifikovať GeoMASON	Michal Kyžňanský	30.11.2012	Dokončená
8.2	Vytvoriť metódu na logovanie stavu agenta	Adrián Kollár	3.12.2012	Znovu pridelená
8.3	Vytvoriť metódu na logovanie stavu agenta	Filip Pakan	3.12.2012	Znovu pridelená
8.4	Integrácia dokumentácie	Miroslav Ort	3.12.2012	Znovu pridelená

5.9.2 Diskutované témy:

- Prezentácia riadenia projektu a vykonávania manažérskeho rolí
- Priebeh práce na projekte
- Implementácia pôsobenia tlaku a síl na poriadkovo zložky a demonštrantov
- Implementácia algoritmu pre plánovanie cesty agentov a akým spôsobom by sa dal spraviť viac efektívny
- Menej pamäťovo náročné logovanie vlastností simulácie a agentov

5.9.3 Prijaté rozhodnutia:

- Vo vetve projektu s nemodifikovaným Geomasonom sa nebude viac pokračovať. Implementácia sa upravila tak, aby bolo možné ukladať do vrstvy s mapou inštanciu agenta
- Rozhodli sme sa používať pri komitovaní zdrojového kódu v Netbeanse aj logovanie úlohy v Jire pomocou pluginu v Netbeanse
- Dohodli sme sa, že pre kontrolu štábnej kultúry budeme využívať možnosti SVN alebo SVN doplneného o plugin, ktorý to dokáže
- Systém Jira upravíme tak, že sa odošle každému upozornenie, že niekto vykonal v tomto systéme zmeny

5.9.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.4	Analýza plánovania trasy	Filip Pakan	10.12.2012	Pridelená
6.5	Analýza plánovania trasy	Adrián Kollár	10.12.2012	Pridelená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	10.12.2012	Pridelená
6.7	Analýza pôsobenia tlaku + príklad/model	Michal Ošvát	10.12.2012	Pridelená
7.1	Analýza pôsobenia tlaku + príklad/model	Miroslav Ort	10.12.2012	Pridelená
8.3	Zefektívniť metódu na logovanie stavu agenta	Filip Pakan	10.12.2012	Pridelená
8.4	Integrácia dokumentácie	Miroslav Ort	8.12.2012	Pridelená
9.1	Implementácia vlastností agentov aby mohli reagovať na pôsobenie tlaku	Michal Kyžňanský	10.12.2012	Pridelená
9.2	Posielanie upozornení o zmene systému Jira	Jana Branišová	10.12.2012	Pridelená

5.10 Zápis zo stretnutia číslo 10

Dátum: 10.12.2012
Čas: 13:00
Dĺžka trvania: 150 min.
Miestnosť: Softvérové štúdio D 004

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.

Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Adrián Kollár

5.10.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
6.4	Analýza plánovania trasy	Filip Pakan	10.12.2012	Dokončená
6.5	Analýza plánovania trasy	Adrián Kollár	10.12.2012	Dokončená
6.6	Analýza pôsobenia tlaku + príklad/model	Jana Branišová	10.12.2012	Dokončená
6.7	Analýza pôsobenia tlaku + príklad/model	Michal Ošvát	10.12.2012	Dokončená
7.1	Analýza pôsobenia tlaku + príklad/model	Miroslav Ort	10.12.2012	Dokončená
9.1	Kontrola štábnej kultúry pomocou SVN – ako na to	Michal Ošvát	10.12.2012	Dokončená
8.3	Zefektívniť metódu na logovanie stavu agenta	Filip Pakan	10.12.2012	Dokončená
8.4	Integrácia dokumentácie	Miroslav Ort	8.12.2012	Dokončená
9.1	Implementácia vlastností agentov aby mohli reagovať na pôsobenie tlaku	Michal Kyžňanský	10.12.2012	Dokončená
9.2	Posielanie upozornení o zmene systému Jira	Jana Branišová	10.12.2012	Dokončená

5.10.2 Diskutované témy:

- Priebeh práce na projekte
- Používateľská prezentácia projektu
- Integrácia a odovzdávanie dokumentácie
- Nedostatočné oprávnenia na zasielanie upozornení zo systému Jira
- Efektívnosť implementácie pôsobenia tlaku
- Problémy, ktoré vznikli implementáciou predpočítavania prázdnych bodov mapy

5.10.3 Prijaté rozhodnutia:

- Predpočítavaná mapa prázdnych bodov pre algoritmus plánovania trasy bude modifikovaná aby obsahovala vzdialenosť prázdnych bodov od budov
- Rozhodli sme sa pred komitovaním zdrojového kódu používať príkaz diff, aby sa predišlo nahratiu nechcených úprav kódu do SVN repozitára
- Algoritmus na zisťovanie a pôsobenie tlaku v dave je veľmi výpočtovo náročný, a preto bude buď zoptimalizovaný alebo nahradený iným efektívnejším algoritmom

5.11 Zápis zo stretnutia číslo 11

Dátum: 19.02.2013
Čas: 11:00
Dĺžka trvania: 80 min.
Miestnosť: Kancelária 4.35

Prítomní:

Vedúci tímu:	Ing. Peter Lacko, PhD.
Členovia tímu:	Bc. Jana Branišová Bc. Adrián Kollár Bc. Michal Kyžňanský Bc. Miroslav Ort Bc. Michal Ošvát Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.11.1 Diskutované témy:

- Aktualizácia web stránky a plánu na nej pre ďalší semester
- Zjednotenie backlogu
- Zjednotenie použitých technológií na vyhýbanie agentov s konkurenčným tímom
- Nutnosť ďalšej implementácie pri dodržaní jednotného scenára
- Formovanie poriadkových zložiek do útvarov
- Napĺňanie psychologického modelu PECS

5.11.2 Prijaté rozhodnutia:

- Pre backlog vytvoríme jedno miesto, kam umiestnime všetky doterajšie položky backlogu. Rozhodnutie bolo prijaté najmä z dôvodu prehľadnosti a jednoduchosti editácie backlogu. Výsledné miesto backlogu bude buď v systéme na podporu riadenia alebo samostatnom dokumente.
- Zdefinujeme konkrétny scenár, ktorý sa pokúsime nasimulovať
- Použijeme rovnakú technológiu zabezpečujúcu vyhýbanie agentov ako konkurenčný tím
- Zrefaktorizujeme súčasný zdrojový kód
- Naplníme model PECS údajmi z bakalárskej práce člena tímu a budeme ich ďalej vylepšovať, pričom sa budeme zameriavať na faktory ovplyvňujúce správanie človeka a šírenie emócií medzi osobami

5.11.3 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
11.1	Aktualizácia web stránky	Michal Kyžňanský	25.02.2013	Pridelená
11.2	Vytvorenie hrubého plánu na letný semester	Jana Branišová	25.02.2013	Pridelená
11.3	Analýza vytvorenia a pretransformovania všetkých súčasných položiek backlogu do Jiry	Jana Branišová	25.02.2013	Pridelená
11.4	Nájdanie videa simulácie s vhodným scenárom	Miroslav Ort	25.02.2013	Pridelená
11.5	RefaktORIZÁCIA zdrojového kódu	Michal Kyžňanský	25.02.2013	Pridelená
11.6	RefaktORIZÁCIA zdrojového kódu	Miroslav Ort	25.02.2013	Pridelená
11.7	Dokončenie implementácie pôsobenia tlakov v dave	Michal Ošvát	25.02.2013	Pridelená
11.8	Dokončenie implementácie plánovania trasy	Adrián Kollár	25.02.2013	Pridelená

5.12 Zápis zo stretnutia číslo 12

Dátum: 25.02.2013
Čas: 13:00
Dĺžka trvania: 180min
Miestnosť: Kancelária 4.35

Prítomní:

Vedúci tímu:	Ing. Peter Lacko, PhD.
Členovia tímu:	Bc. Jana Branišová Bc. Adrián Kollár Bc. Michal Kyžňanský Bc. Miroslav Ort Bc. Michal Ošvát Bc. Filip Pakan

Autor zápisu: Bc. Jana Branišová

5.12.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
11.1	Aktualizácia web stránky	Michal Kyžňanský	25.02.2013	Znova Pridelená
11.2	Vytvorenie hrubého plánu na letný semester	Jana Branišová	25.02.2013	Dokončená
11.3	Analýza vytvorenia a pretransformovania všetkých súčasných položiek backlogu do Jiry	Jana Branišová	25.02.2013	Dokončená
11.4	Nájdienie videa simulácie s vhodným scenárom	Miroslav Ort	25.02.2013	Dokončená
11.5	Refaktorizácia zdrojového kódu	Michal Kyžňanský	25.02.2013	Dokončená
11.6	Refaktorizácia zdrojového kódu	Miroslav Ort	25.02.2013	Znova Pridelená
11.7	Dokončenie implementácie pôsobenia tlakov v dave	Michal Ošvát	25.02.2013	Znova Pridelená
11.8	Dokončenie implementácie plánovania trasy	Adrián Kollár	25.02.2013	Dokončená

5.12.2 Diskutované témy:

- Návrh refaktorizácie zdrojového kódu prostredníctvom UML diagramov
- Roly v tíme
- Pôsobenie tlakov v dave
- Plánovanie trasy agenta
- Rozšírenie simulácie o element Božena

5.12.3 Prijaté rozhodnutia:

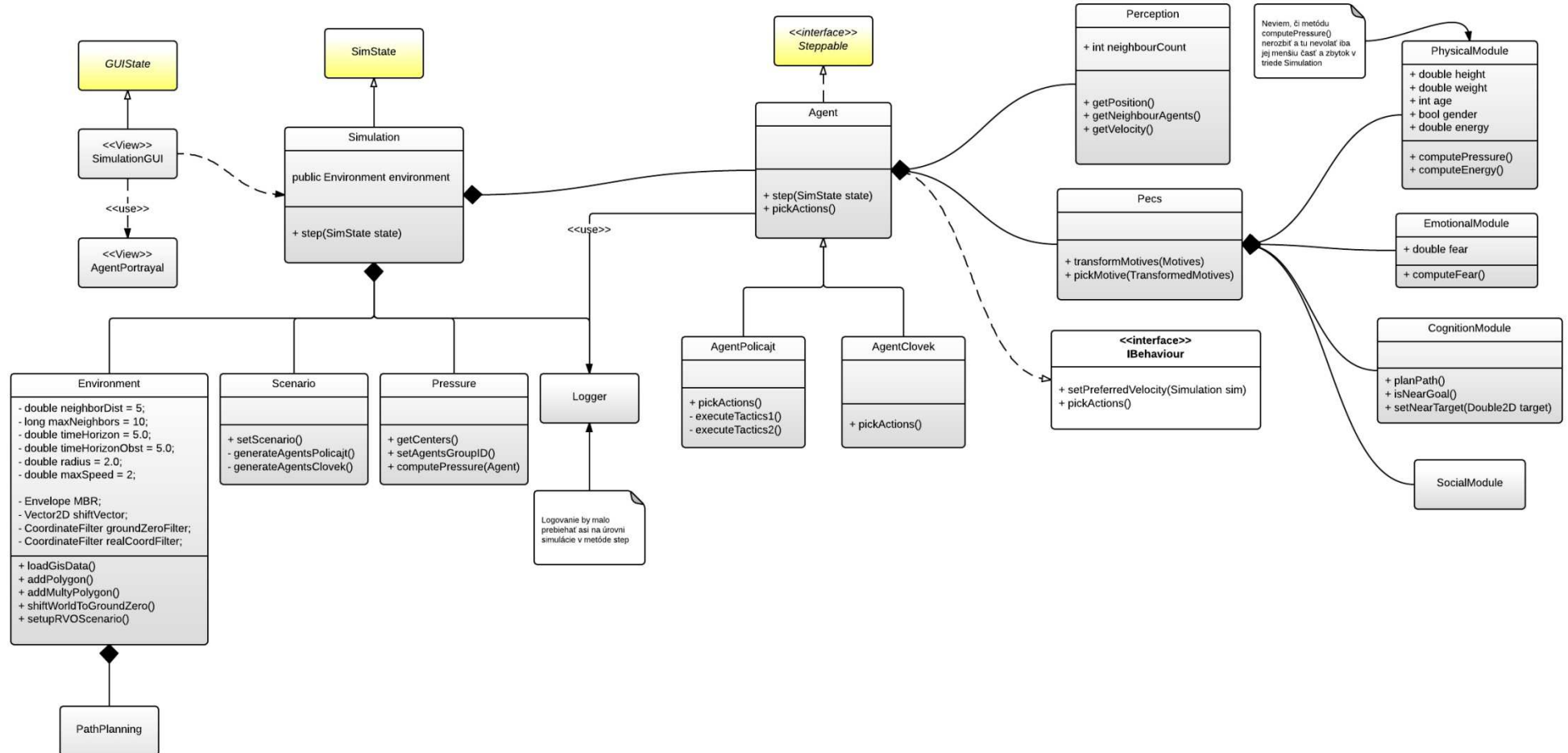
- Zdrojový kód sa zrefaktorizuje podľa navrhutej štruktúry
- V tíme sa zachová rola pre tvorbu dokumentácie, plánovania, štábnej kultúry a pridá sa nová rola tester, ktorá bude priradovať každý týždeň niekomu inému. Tento týždeň bol zvolený testerom Michal Ošvát

- Vytvorí sa jeden dokument, ktorý bude slúžiť na zapisovanie parametrov simulácie, ktoré bude tester testovať
- Algoritmus na vyhľadávanie skupín ľudí v dave sa vyjme z tela agenta a bude sa vykonávať len jeden krát za krok simulácie. Je nutné otestovať efektivitu algoritmu a popri tom ho optimalizovať alebo zmeniť

5.12.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
11.1	Aktualizácia web stránky	Michal Kyžňanský	4.03.2013	Pridelená
12.1	Implementácia scenára	Michal Kyžňanský	4.03.2013	Pridelená
12.2	Refaktorizácia algoritmu na vyhľadávanie skupín v dave	Filip Pakan	4.03.2013	Pridelená
12.3	Pridanie komponentu božena	Adrián Kollár	4.03.2013	Pridelená
11.6	Refaktorizácia zdrojového kódu	Miroslav Ort	4.03.2013	Pridelená
11.7	Dokončenie implementácie pôsobenia tlakov v dave	Michal Ošvát	4.03.2013	Pridelená

Prílohy: UML diagram štruktúry zdrojového kódu



5.13 Zápis zo stretnutia číslo 13

Dátum: 04.03.2013
Čas: 13:00
Dĺžka trvania: 120min
Miestnosť: Kancelária 4.35

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Ošvát

5.13.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
11.1	Aktualizácia web stránky	Michal Kyžňanský	4.03.2013	Dokončená
12.1	Implementácia scenára	Michal Kyžňanský	4.03.2013	Dokončená
12.2	Refaktorizácia algoritmu na vyhľadávanie skupín v dave	Filip Pakan	4.03.2013	Dokončená
12.3	Pridanie komponentu božena	Adrián Kollár	4.03.2013	Znova pridelená
11.6	Refaktorizácia zdrojového kódu	Miroslav Ort	4.03.2013	Dokončená
11.7	Dokončenie implementácie pôsobenia tlakov v dave	Michal Ošvát	4.03.2013	Dokončená

5.13.2 Diskutované témy:

- Výber jedného konkrétneho z viacerých možných scenárov simulácie
- Pôsobenie tlakov v dave

- Element božena a možnosti jeho použitia
- Možnosť vytvorenia policajných línií

5.13.3 Prijaté rozhodnutia:

- Pre finálny scenár sa použije časť mapy, kde sa cesta rozdeľuje, pričom v jednej časti sa použije element Božena, a v druhej bude na zastavenie agentov policajná línia
- Metóda pre pôsobenie tlakov bude modifikovaná zo 4 smerov pôsobenia univerzálne na pôsobenie v akomkoľvek rozsahu (360°)
- Algoritmus na vyhľadávanie skupín ľudí v dave bude zefektívnený pomocou komponentov súvislostí
- Agentom budú pridané parametre, a poriadkové zložky budú vytvárať línie

5.13.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.1	Pridanie parametrov do triedy Agent	Michal Kyžňanský	11.03.2013	Pridelená
13.2	Analýza otočenia bodov v dvojrozmernom priestore podľa ich centrálného vektora	Filip Pakan	09.03.2013	Pridelená
12.3	Pridanie komponentu božena	Adrián Kollár	11.03.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	11.03.2013	Pridelená
12.3	Pridanie komponentu božena	Jana Branišová	11.03.2013	Pridelená
13.4	Implementácia výpočtu pôsobenia tlaku pre všetky smery centrálného vektora	Michal Ošvát	11.03.2013	Pridelená

5.14 Zápis zo stretnutia číslo 14

Dátum: 11.03.2013
Čas: 13:00
Dĺžka trvania: 120min
Miestnosť: Kancelária 4.35

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Kyžňanský

5.14.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.1	Pridanie parametrov do triedy Agent	Michal Kyžňanský	11.03.2013	Dokončená
13.2	Analýza otočenia bodov v dvojrozmernom priestore podľa ich centrálného vektora	Filip Pakan	09.03.2013	Dokončená
12.3	Pridanie komponentu božena	Adrián Kollár	11.03.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	11.03.2013	Znova pridelená
12.3	Pridanie komponentu božena	Jana Branišová	11.03.2013	Znova pridelená
13.4	Implementácia výpočtu pôsobenia tlaku pre všetky smery centrálného vektora	Michal Ošvát	11.03.2013	Znova pridelená

5.14.2 Diskutované témy:

- Pôsobenie tlakov v dave
- Možnosti objektu Božena a interakcia v rámci prostredia RVO s agentmi
- Vykonané zmeny v nových verziách (výpočet strachu, energie, označenia skupiny agentov)

5.14.3 Prijaté rozhodnutia:

- Metóda pre pôsobenie tlakov bude modifikovaná zo 4 smerov pôsobenia univerzálne na pôsobenie v akomkoľvek rozsahu (360°)
- Objekt Božena bude v RVO väčší ako jeho vizuálne zobrazenie, kvôli lepšej reakcií agentov na jeho prítomnosť
- Taktika policajných zložiek bude realizovaná za pomoci grafovej štruktúry
- Použitie novej verzie RVO na pridávanie dynamických prekážok.

5.14.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Adrián Kollár	18.03.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	18.03.2013	Znova pridelená
12.3	Pridanie komponentu božena	Jana Branišová	18.03.2013	Znova pridelená
13.4	Implementácia výpočtu pôsobenia tlaku pre všetky smery centrálného vektora	Michal Ošvát	18.03.2013	Znova pridelená
14.1	Dokumentácia k 5. a 6. šprintu	Všetci	18.03.2013	Pridelená

5.15 Zápis zo stretnutia číslo 15

Dátum: 18.03.2013
Čas: 13:00
Dĺžka trvania: 120min
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Adrián Kollár

5.15.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Adrián Kollár	18.03.2013	Dokončená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	18.03.2013	Znova pridelená
12.3	Pridanie komponentu božena	Jana Branišová	18.03.2013	Znova pridelená
13.4	Implementácia výpočtu pôsobenia tlaku pre všetky smery centrálného vektora	Michal Ošvát	18.03.2013	Dokončená
14.1	Dokumentácia k 5. a 6. šprintu	Všetci	18.03.2013	Dokončená

5.15.2 Diskutované témy:

- Pohyb a formácie poriadkových zložiek
- Rotovanie objektu Božena na základe smeru pohybu
- Efektivita súčasného riešenia a možnosti optimalizácie

5.15.3 Prijaté rozhodnutia:

- Agentom bude vytvorený nový atribút reprezentujúci tlak pôsobiaci na agenta. Agenti budú prefarbení podľa aktuálne pôsobiaceho tlaku
- Objekt Božena sa bude pohybovať po stanovených waypointoch a rotovať podľa smeru pohybu
- Taktika policajných zložiek bude využívať mapu vzdialeností od budov na vytvorenie a pohyb formácií

5.15.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Jana Branišová	25.03.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	25.03.2013	Znova pridelená
15.1	Implementácia pohybu boženy	Adrián Kollár	25.03.2013	Pridelená
15.2	Pridanie atribútu reprezentujúceho tlak na agenta	Michal Ošvát	25.03.2013	Pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	25.03.2013	Pridelená
15.4	Implementácia scenára	Filip Pakan	25.03.2013	Pridelená

5.16 Zápis zo stretnutia číslo 16

Dátum: 25.03.2013
Čas: 13:00
Dĺžka trvania: 60 min
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Filip Pakan

5.16.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Jana Branišová	25.03.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	25.03.2013	Znova pridelená
15.1	Implementácia pohybu boženy	Adrián Kollár	25.03.2013	Znova pridelená
15.2	Pridanie atribútu reprezentujúceho tlak na agenta	Michal Ošvát	25.03.2013	Dokončená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	25.03.2013	Znova pridelená
15.4	Implementácia scenára	Filip Pakan	25.03.2013	Znova pridelená

5.16.2 Diskutované témy:

- Pohyb a formácie poriadkových zložiek
- Vplyv tlaku davu na policajné zložky
- Rotácia komponentu Božena

5.16.3 Prijaté rozhodnutia:

- Smer rotovania Boženy bude určený pomocou všeobecnej rovnice priamky
- Tlak davu bude ovplyvňovať policajné zložky
- Pri pohybe policajných zložiek sa bude spolu s nimi pohybovať aj dynamická prekážka

5.16.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Jana Branišová	01.04.2013	Znova pridelená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	01.04.2013	Znova pridelená
15.1	Implementácia pohybu boženy	Adrián Kollár	01.04.2013	Znova pridelená
16.1	Vplyv tlaku davu na policajné zložky	Michal Ošvát	01.04.2013	Pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	01.04.2013	Znova pridelená
15.4	Implementácia scenára	Filip Pakan	01.04.2013	Znova pridelená

5.17 Zápis zo stretnutia číslo 17

Dátum: 03.04.2013
Čas: 13:00
Dĺžka trvania: 40 min
Miestnosť: 4.35

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Jana Branišová

5.17.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu Božena	Jana Branišová	01.04.2013	Dokončená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	01.04.2013	Znova pridelená
15.1	Implementácia pohybu Boženy	Adrián Kollár	01.03.2013	Znova pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	01.04.2013	Znova pridelená
15.4	Implementácia scenára	Filip Pakan	01.04.2013	Dokončená
16.1	Vplyv tlaku davu na policajné zložky	Michal Ošvát	01.04.2013	Znova pridelená

5.17.2 Diskutované témy:

- Pohyb a formácie poriadkových zložiek
- Vplyv tlaku Boženy na agentov

5.17.3 Prijaté rozhodnutia:

- Pridanie pôsobenia tlaku do komponentu Božena
- Pri prelomení línie policajtov sa dynamický obstacle, ktorý drží demonštrantov, tiež rozdelí na dve časti

5.17.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	08.04.2013	Znova pridelená
15.1	Implementácia pohybu Boženy	Adrián Kollár	25.03.2013	Znova pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	08.04.2013	Znova pridelená
16.1	Vplyv tlaku davu na policajné zložky	Michal Ošvát	08.04.2013	Znova pridelená
17.1	Doplnenie pôsobenie tlakom komponentu Božena	Michal Ošvát	08.04.2013	Pridelená
17.2	Rozbitie línie policajtov	Miroslav Ort	08.04.2013	Pridelená
17.3	Rozbitie línie policajtov	Michal Ošvát	08.04.2013	Pridelená

5.18 Zápis zo stretnutia číslo 18

Dátum: 08.04.2013
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.18.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
12.3	Pridanie komponentu božena	Jana Branišová	08.04.2013	Dokončená
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	08.04.2013	Znova pridelená
15.1	Implementácia pohybu boženy	Adrián Kollár	08.04.2013	Dokončená
16.1	Vplyv tlaku davu na policajné zložky	Michal Ošvát	08.04.2013	Dokončená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	08.04.2013	Znova pridelená
15.4	Implementácia scenára	Filip Pakan	08.04.2013	Znova pridelená

5.18.2 Diskutované témy:

- Vytváranie dynamických prekážok v prostredí simulácie použitím knižnice RVO
- Príčiny nerešpektovania existencie dynamických prekážok agentami.
- Pôsobenie tlaku boženy
- Akcie boženy, policajtov a demonštrantov a interakcia medzi agentami
- Parametrizácia simulácie

- Tvorba scenárov
- Napĺňanie psychologického modelu PECS

5.18.3 Prijaté rozhodnutia:

- Namodelujeme vplyv vonkajšieho prostredia na psychologické a fyziologické parametre agentov
- Pridáme nové akcie demonštrantov na základe zmeny vonkajšieho prostredia
- Zistíme a upravíme policajné línie tak, aby cez ne neprechádzali demonštranti pri bežných situáciách
- Vytvoríme viacero scenárov
- Doplníme konfiguračný súbor o parametre vyplývajúce z existencie viacerých scenárov
- Doplníme model PECS o nové faktory ovplyvňujúce správanie človeka a šírenie emócií medzi agentami

5.18.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	15.04.2013	Znova pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	15.04.2013	Znova pridelená
15.4	Implementácia scenára	Filip Pakan	15.04.2013	Znova pridelená
18.1	Doplnenie konfiguračného súboru	Miroslav Ort	15.04.2013	Pridelená
18.2	Doplnenie konfiguračného súboru	Filip Pakan	15.04.2013	Pridelená
18.3	Doplnenie spôsobu ústupu pred boženou	Michal Ošvát	15.04.2013	Pridelená
18.4	Dynamické zväčšovanie boženy	Adrián Kollár	15.04.2013	Pridelená
18.5	Vytvoriť viac strojov božena	Adrián Kollár	15.04.2013	Pridelená
18.6	Pridanie správania agentom	Všetci	15.04.2013	Pridelená

5.19 Zápis zo stretnutia číslo 19

Dátum: 15.04.2013
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Kyžňanský

5.19.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.3	Návrh a implementácia policajných líníí a ich postupu	Miroslav Ort	15.04.2013	Znova pridelená
15.3	Implementácia motívov a aktivít	Michal Kyžňanský	15.04.2013	Dokončená
15.4	Implementácia scenára	Filip Pakan	15.04.2013	Dokončená
18.1	Doplnenie konfiguračného súboru	Miroslav Ort	15.04.2013	Dokončená
18.2	Doplnenie konfiguračného súboru	Filip Pakan	15.04.2013	Dokončená
18.3	Doplnenie spôsobu ústupu pred boženou	Michal Ošvát	15.04.2013	Dokončená
18.4	Dynamické zväčšovanie boženy	Adrián Kollár	15.04.2013	Dokončená
18.5	Vytvoriť viac strojov božena	Adrián Kollár	15.04.2013	Dokončená
18.6	Pridanie správania agentom	Všetci	15.04.2013	Znova pridelená

5.19.2 Diskutované témy:

- Nové motívy agentov, napr. agresivita
- Simulačný workshop
- IT SRC a plagát tímu
- Možnosti automatického rozťahovania objektu Božena v ulice
- Parametrizácia simulácie
- Napĺňanie psychologického modelu PECS
- Code review funkcionality

5.19.3 Prijaté rozhodnutia:

- Rozvrhnutie úloh pri práci na plagáte tímu na IT SRC
- Doplnenie modelu PECS o motív agresivity, motív mŕtvol v okolí agenta a ústup
- Doplnenie modelu PECS o aktivity vyplývajúce z motívov
- Pridáme nové akcie demonštrantov na základe zmeny vonkajšieho prostredia
- Zistíme a upravíme policajné línie tak, aby cez ne neprechádzali demonštranti pri bežných situáciách vrátane pridania taktiky
- Opravenie chyby v parsovaní konfigurácie

5.19.4 Úlohy do ďalšieho stretnutia:

ID úlohy	Popis úlohy	Pridelené členovi	Termín dokončenia	Stav
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	22.04.2013	Znova pridelená
19.1	Doplnenie motívov a aktivít	Michal Kyžňanský	22.04.2013	Pridelená
19.2	Opravenie parsovania konfiguračného súboru	Filip Pakan	22.04.2013	Pridelená
18.4	Dynamické zväčšovanie boženy	Adrián Kollár	22.04.2013	Znova pridelená
18.6	Pridanie správania agentom	Všetci	22.04.2013	Znovu pridelená

5.20 Zápis zo stretnutia číslo 20

Dátum: 22.04.2013
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.

Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.20.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
13.3	Návrh a implementácia policajných línií a ich postupu	Miroslav Ort	22.04.2013	Dokončená
19.1	Doplnenie motívov a aktivít	Michal Kyžňanský	22.04.2013	Znova pridelená
19.2	Opravenie parsovania konfiguračného súboru	Filip Pakan	22.04.2013	Dokončená
18.4	Dynamické zväčšovanie boženy	Adrián Kollár	22.04.2013	Dokončená
18.6	Pridanie správania agentom	Všetci	22.04.2013	Znovu pridelená

5.20.2 Diskutované témy:

- Prezentácia tímu na vedeckej študentskej konferencii IIT SRC 2013
- Záverečná prezentácia tímového projektu
- Prenos tlakov na policajné zložky
- Škálovanie boženy
- Postup pri tvorbe ďalších častí dokumentácie

5.20.3 Prijaté rozhodnutia:

- Do dokumentácie včelíme časť s dosiahnutými výsledkami, ktorá bude vo forme článku
- Vytvoríme viacero scenárov
- V scenároch budeme uvažovať aj prípad, kedy demonštranti odtlačia policajné zložky
- Súčasný produkt budeme ďalej stabilizovať a doplníme ho o faktory pôsobiace na demonštrantov a reakcie demonštrantov

5.20.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
18.6	Pridanie správania agentom	Všetci	29.04.2013	Znovu pridelená
19.1	Stabilizácia algoritmu na formovanie policajných línií	Miroslav Ort	29.04.2013	Pridelená
19.2	Stabilizácia algoritmu na formovanie policajných línií	Jana Branišová	29.04.2013	Pridelená
19.3	Tutoriál na pridávanie scenárov do konfiguračného súboru	Filip Pakan	29.04.2013	Pridelená
19.4	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Michal Ošvát	29.04.2013	Pridelená
19.5	Doplnenie motívov a aktivít	Michal Kyžňanský	29.04.2013	Znova pridelená
19.6	Doplnenie motívov a aktivít	Adrián Kollár	29.04.2013	Pridelená

5.21 Zápis zo stretnutia číslo 21

Dátum: 29.04.2013
Čas: 13:00
Dĺžka trvania: 120 min.
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.
 Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Michal Ošvát

5.21.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
18.6	Pridanie správania agentom	Všetci	06.05.2013	Znovu pridelená
19.1	Stabilizácia algoritmu na formovanie policajných líní	Miroslav Ort	06.05.2013	Znovu pridelená
19.2	Stabilizácia algoritmu na formovanie policajných líní	Jana Branišová	06.05.2013	Znovu pridelená
19.3	Tutoriál na pridávanie scenárov do konfiguračného súboru	Filip Pakan	29.04.2013	Dokončená
19.4	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Michal Ošvát	29.04.2013	Znovu pridelená
19.5	Doplnenie motívov a aktivít	Michal Kyžňanský	29.04.2013	Znova pridelená
19.6	Doplnenie motívov a aktivít	Adrián Kollár	29.04.2013	Znovu pridelená

5.21.2 Diskutované témy:

- Prezentácia projektu na vedeckej študentskej konferencii IIT SRC 2013
- Vplyv motívov na priebeh simulácie
- Záverečná prezentácia tímového projektu
- Prenos tlakov na policajné zložky príp. policajné línie
- Motív úteku od Boženy

5.21.3 Prijaté rozhodnutia:

- Simuláciu stabilizujeme a pripravíme rôzne scenáre pre otestovanie viacerých charakteristík a implementovaných prvkov (správanie agentov, zmeny hodnôt atribútov a pod.)
- Každý agent typu policajt bude mať rôzny atribút ktorý hovorí o hranici pri ktorej ho prostredníctvom pôsobenia tlaku demonštranti odtlačia
- Sfinalizovanie tvorenia policajných línií
- Zavedenie a odlíšenie 2 motívov: motív tlaku a motív božena, ktoré spolu nebudú priamo súvisieť

5.21.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
18.6	Pridanie správania agentom	Všetci	06.05.2013	Znovu pridelená
19.1	Stabilizácia algoritmu na formovanie policajných línií	Miroslav Ort	06.05.2013	Znovu pridelená
19.2	Stabilizácia algoritmu na formovanie policajných línií	Jana Branišová	06.05.2013	Znovu pridelená
19.4	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Michal Ošvát	06.05.2013	Znovu pridelená
19.5	Doplnenie motívov a aktivít	Michal Kyžňanský	06.05.2013	Znova pridelená
19.6	Doplnenie motívov a aktivít	Adrián Kollár	06.05.2013	Znovu pridelená
21.1	Pridanie nového parametra Boženy do všetkých scenárov	Filip Pakan	06.05.2013	Pridelená

5.22 Zápis zo stretnutia číslo 22

Dátum: 06.05.2013
Čas: 13:00
Dĺžka trvania: 60 min.
Miestnosť: 1.31a

Prítomní: Vedúci tímu: Ing. Peter Lacko, PhD.

Členovia tímu: Bc. Jana Branišová
 Bc. Adrián Kollár
 Bc. Michal Kyžňanský
 Bc. Miroslav Ort
 Bc. Michal Ošvát
 Bc. Filip Pakan

Autor zápisu: Bc. Miroslav Ort

5.22.1 Vyhodnotenie úloh z predchádzajúceho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
18.6	Pridanie správania agentom	Všetci	06.05.2013	Dokončená
19.1	Stabilizácia algoritmu na formovanie policajných línií	Miroslav Ort	06.05.2013	Znovu pridelená
19.2	Stabilizácia algoritmu na formovanie policajných línií	Jana Branišová	06.05.2013	Znovu pridelená
19.4	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Michal Ošvát	06.05.2013	Znovu pridelená
19.5	Doplnenie motívov a aktivít	Michal Kyžňanský	06.05.2013	Znova pridelená
19.6	Doplnenie motívov a aktivít	Adrián Kollár	06.05.2013	Znovu pridelená
21.1	Pridanie nového parametra Boženy do všetkých scenárov	Filip Pakan	06.05.2013	Dokončená

5.22.2 Diskutované témy:

- Vplyv motívov na priebeh simulácie
- Záverečná prezentácia tímového projektu
- Záverečná dokumentácia produktu
- Prenos tlakov na policajné zložky a policajné línie

5.22.3 Prijaté rozhodnutia:

- Simuláciu stabilizujeme
- Každý agent typu policajt bude mať rôzny atribút ktorý hovorí o hranici, pri ktorej ho prostredníctvom pôsobenia tlaku demonštranti odtlačia
- Sfinalizovanie tvorenia policajných línií
- Agentom doplníme také motívy a ich implementujeme ich tak, aby sme boli schopní vygenerovať grafy, na ktorých budeme sledovať zmenu motívov počas behu simulácie

5.22.4 Úlohy do ďalšieho stretnutia:

<i>ID úlohy</i>	<i>Popis úlohy</i>	<i>Pridelené členovi</i>	<i>Termín dokončenia</i>	<i>Stav</i>
19.1	Stabilizácia algoritmu na formovanie policajných línií	Miroslav Ort	13.05.2013	Znovu pridelená
19.2	Stabilizácia algoritmu na formovanie policajných línií	Jana Branišová	13.05.2013	Znovu pridelená
19.4	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Michal Ošvát	13.05.2013	Znovu pridelená
19.5	Doplnenie motívov a aktivít	Michal Kyžňanský	13.05.2013	Znova pridelená
19.6	Doplnenie motívov a aktivít	Adrián Kollár	13.05.2013	Znovu pridelená
22.1	Tvorba scenárov na prenos tlakov medzi líniami policajtov	Filip Pakan	13.05.2013	Pridelená

6 Manažment kvality

Manažovanie kvality predstavuje dôležitú úlohu, ktorej výsledkom je zabezpečenie istej úrovne kvality na viacerých frontoch projektu spolu s iniciovaním procesov, ktorých úlohou je explicitne zvyšovať kvalitu v zvolených oblastiach práce na projekte. Manažment kvality na predmete tímový projekt v zvolenom projekte Simulácie demonštrácie pokrýva hlavne dve základne oblasti:

1. Smerovanie celého projektu, otázka funkčnosti a splnenia požiadaviek
2. Explicitné zvyšovanie kvality zdrojového kódu (štábna kultúra, JUnit testy)

6.1 Smerovanie celého projektu, otázka funkčnosti a splnenia požiadaviek

Manažér kvality spolu s product ownerom určuje smerovanie projektu v oblasti splnenia požiadaviek a funkčnosti jednotlivých častí samotného produktu. Na stretnutiach pravidelne participujem ako manažér kvality a dodávam tímu nápady, ktorých realizácia má priblížiť produkt čo najbližšie k splneniu požiadaviek, ktoré boli určené.

Manažér kvality tiež hodnotí doterajšie prístupy v tvorbe produktu a usmerňuje ich želaným smerom a po konzultácii s product ownerom má možnosť robiť zásahy do práce tímu, v čom mu je nápomocný vedúci tímu.

6.2 Explicitné zvyšovanie kvality zdrojového kódu (štábna kultúra, JUnit testy)

Explicitné zvyšovanie kvality je založené na aplikovaní procesov a úkonov, ktoré nepredstavujú žiadnu priamu pridanú hodnotu produktu, ale v dlhodobom meradle eliminujú nároky na čas práce všetkých členov tímu v pokročilej fáze implementácie.

Absolútnou nutnosťou pri akomkoľvek programovaní a najmä pri práci v tíme je nutnosť dodržiavať určitú **štábnu kultúru**, t.j. konvencie písania zdrojového kódu. Týmto tímová práca naberá na efektívite a odpadá nutnosť študovať kód niekoho iného, ktorý používa výrazne odlišné konvencie pri písaní zdrojového kódu.

Ďalším elementom, ktorý hrá dôležitú úlohu pri rozširujúcom sa projekte a práci v tíme je možnosť otestovať kritické a zložité komponenty systému ľahko a automaticky, bez zbytočnej konzumácie času a zaťažovaním ďalších členov tímu. Na tento účel budeme používať framework na vytváranie testov v prostredí Java - Netbeans tzv. JUnit. Na stretnutí musí byť zvolený človek, ktorý bude riešiť práve túto otázku, pretože manažér kvality je zodpovedný hlavne za iniciovanie tohto procesu.

Ako manažér kvality som zodpovedný za prehliadky kódu, resp. spisovanie zápisnice z nedostatkov ohľadom štábnej kultúry. Všetka táto práca by sa mohla zdať zbytočná, a že sa jedná o zbytočné otravovanie tímových kolegov, ale idea tohto šikanovania je v

odmene, ktorú prinesie čas a prejaví sa pri častých zmenách v kóde, prípadne jeho neskoršej analýze.

6.3 Štábna kultúra

Štábna kultúra pozostáva z niekoľkých pravidiel (viď Tabuľka 6.1), ktoré zaručia prehľadnosť, čitateľnosť a uniformitu kódu písaného jednotlivými členmi tímu. Je rozdelená do jednotlivých častí a následne vizuálne demonštrovaná (viď. kapitola 5.1.9.). Každý člen tímu si musí importovať do prostredia NetBeans definované formátovanie zdrojového kódu zo súboru *coding_style.xml*.

Tabuľka 6.1 Zoznam častí štábnej kultúry.

Názov časti	Kapitola
Import štýlov štábnej kultúry do Netbeans	6.4
Pomenovanie tried	6.5
Pomenovanie atribútov a metód tried	6.6
Odsadenie vnorení a medzery	6.7
Zátvorky { }	6.8
Identifikátor this.*	6.9
Komentáre	6.10
Kompletná vizuálna stránka kódu	6.11

6.4 Import štýlov štábnej kultúry do Netbeans

Je nutné otvoriť záložku *Tools* v hlavnej lište prostredia NetBeans a zvoliť možnosť *Options*. Nasleduje zvolenie záložky *Formatting* a stlačenie tlačidla *Import* v dolnej časti okna. Ďalej je nutné stlačením tlačidla *Browse...* vyvolať nové okno a zvoliť cestu k súboru *coding_style.xml* a potvrdiť voľbu tlačidlom *OK*. Po vykonaní spomenutých krokov bude prostredie automaticky generované metódy a určité ďalšie dopĺňanie kódu realizovať podľa nižšie uvedených štandardov. Importovanie štýlov kódovania (štábnej kultúry) predstavuje prvý krok k úspešnému dodržiavaniu predpísaného štýlu touto metodikou.

6.5 Pomenovanie tried

Každá trieda musí byť pomenovaná v súlade s praktikou Pascal Case (prísnejší variant Camel Case), t.j. začínať veľkým písmenom abecedy a v prípade, že obsahuje viac slov, prvé písmeno každého ďalšieho slova bude kapitál. **Názvy tried musia byť v angličtine.**

Príklad (správne)

```
public class VeryLongClass
{
    //telo triedy
}
```

Príklad (nesprávne)

```
public class class_name
{
    //telo triedy
}
```

6.6 Pomenovanie atribútov a metód tried

Atribúty a metódy triedy musia byť pomenované v súlade s praktikou Camel Case. Tento štýl pomenenia dodržiava rovnaké zásady ako Pascal Case, avšak prvé písmeno názvu môže byť malé aj veľké (rozhodnutie je na programátorovi). **Názvy atribútov a metód musia byť v angličtine.**

Príklad (správne)

```
...
public int attributeAge;
public int attributeHeight;

public void sumNumbers(int a, int b)
{
    //...
}

public void Output()
{
    //...
}
...
```

Príklad (nesprávne)

```
...
public int attribute_age;
public int attribute_long_height;

public void sum_numbers_and_output_results(int a, int b)
{
    //...
}
...
```

6.7 Odsadenie vnorení a medzery

Každé vnorenie v programe napr. (telo cyklu, telo podmienky, telo triedy a iné) musí byť odsadené od okraja predošlého kontextu o 1 tabulátor. Po importe štýlov v súbore

coding_style.xml bude 1 tabulátor veľkosti 4 medzery. Základný kontext nemá žiadne odsadenie, t.j. 0 znakov tabulátor. Takto je možné jednoducho sledovať viacero vnorených podmienok / vnorených konštrukcií a vidieť jednoznačne ich rozsah. Všetky priradenia a každý operátor musí byť zľava aj sprava odsadený medzerami. ďalej po každom znaku ",", ";" a ":" musí nasledovať medzera.

Príklad (správne)

```
...  
public int numberAgents = 45;  
public void processResult()  
{  
    int a;  
  
    this.numberAgents = (this.numberAgents - 2) * 4;  
  
    for(a = 0; a < 20; a++)  
    {  
        this.numberAgents++;  
    }  
}  
...
```

Príklad (nesprávne)

```
...  
public int numberAgents=45;  
public void processResult()  
{  
    int a;  
  
    this.numberAgents=(this.numberAgents-2)*4;  
  
    for(a=0;a<20;a++)  
    {  
        this.numberAgents++;  
    }  
}  
...
```

6.8 Zátvorky {}

Každé telo podmienky (*if*, resp. *else*) alebo cyklu (*for*, *while*, *foreach*), bude uzatvorené v zátvorkách *{}* a to aj v prípade, že toto telo je krátke a zmestilo by sa na jeden riadok. Bude tak zvýšená prehľadnosť a čitateľnosť kódu.

Príklad (správne)

```
...  
if(45 > a)  
{  
    System.out.println("TEST");  
}
```

```
...
```

Príklad (nesprávne)

```
...  
if(45 > a) System.out.println("TEST");  
...
```

6.9 Identifikátor *this*.*

Každému použitiu atribútu alebo metódy triedy v danej triede (v prípade, že nie je static resp. final) musí predchádzať identifikátor *this*, aby bolo jasné, že ide o metódu / atribút danej triedy.

Príklad (správne)

```
...  
public class Calculator  
{  
    public double numberAgents;  
  
    public void outputChange()  
    {  
        this.numberAgents = 45;  
        this.localMethod(this.numberAgents);  
    }  
}  
...
```

Príklad (nesprávne)

```
...  
public class Calculator  
{  
    public double numberAgents;  
  
    public void outputChange()  
    {  
        numberAgents = 45;  
        localMethod(this.numberAgents);  
    }  
}  
...
```

6.10 Komentáre

Každá metóda triedy musí obsahovať komentár, výnimku predstavujú metódy, ktoré iba čítajú resp. nastavujú hodnotu (setter, getter). Komentáre musia byť vo formáte

vhodnom pre spracovanie neskoršej dokumentácie, t.j. JavaDoc. **Všetky komentáře musia byť po slovensky.**

Každý komentár metódy bude obsahovať krátky popis funkčnosti metódy celou vetou (max. 1 riadok a 30 slov, bez diakritiky), opísané parametre a návratovú hodnotu. Parametre (*@param*) a návratová hodnota (*@return*) budú uvedené ako parametre tagov, ktoré využíva JavaDoc.

Každý komentár triedy bude obsahovať krátky opis jej významu a funkčnosti (max. 1 riadok a 30 slov, bez diakritiky). Komentár premennej je voliteľný a je na programátorovi aby posúdil, ktoré premenné majú netriviálny názov a nie je z nich jasné ich použitie. V takom prípade, je povinný danú premennú okomentovať štýlom uvedeným nižšie v rozsahu max. 1 riadok a 30 slov, bez diakritiky. Ďalšie komentovanie nie je povinné ale určite vítane, ak programátor usúdi, že jeho komentár pomôže pochopiť zložitejší blok zdrojového kódu. Akýkoľvek ďalší komentár však musí byť umiestnený nad zdrojovým kódom a nie napravo pomocou medzier / tabulátorov.

Príklad (správne)

```
/**
 * Trieda obsahujúca matematicke operacie.
 */
public class Calculator
{
    //Opis netrivialnej premennej.
    public int nonTrivialVariable;

    /**
     * Toto je metoda ktora spocitava dve cisla.
     *
     * @param a Prve cislo do suctu.
     * @param b Druhe cislo do suctu.
     * @return Vysledok scitania.
     */
    public int sumNumbers(int a, int b)
    {
        return a + b;
    }
}
```

Príklad (nesprávne)

```
//Trieda obsahujúca matematicke operacie
//Dokaze naozaj vela veci
public class Calculator
{
    public int nonTrivialVariable;           //opis premennej

    //Toto je metoda ktora spocitava dve cisla.
    public int sumNumbers(int a, int b)
    {
        return a + b;
    }
}
```

6.11 Kompletná vizuálna stránka kódu

Kompletné zhrnutie všetkých pravidiel štabnej kultúry v prehľadnom kúsku kódu.

```
package simTeam;

import edu.unc.cs.gamma.rvo2.RVOSimulator;
import edu.unc.cs.gamma.rvo2.Vector2;
import java.awt.Color;
import sim.engine.*;
import sim.field.continuous.*;
import sim.util.Double2D;

/**
 * Trieda obsahujúca riadenie simulácie a agentov.
 */
public class Simulation extends SimState
{
    //Opis netrivialnej premennej.
    public int nonTrivialVariable;
    public int totalAgents;

    /**
     * Toto je metoda ktora spočítava dve čísla.
     *
     * @param a Prvé číslo do súčtu.
     * @param b Druhé číslo do súčtu.
     * @return Výsledok sčítania.
     */
    public int sumNumbers(int a, int b)
    {
        return a + b;
    }

    /**
     * Metoda, ktorá upraví výsledok špeciálnym algoritmom.
     *
     * @param a Výsledok súčtu.
     */
    public void processSumResult(int a)
    {
        int i;

        for(i = 0; i < 10; i++)
        {
            a = a + (i * 4);
        }

        this.totalAgents = a;
    }
}
```

7 Manažment plánovania

Rozhodli sme sa vyvíjať metódou SCRUM. Šprint trvá štandardne mesiac. My sme si dĺžku šprintu upravili pre vývoj tohto projektu na dva týždne. Taktiež nepraktizujeme dayly SCRUM, čo je v preklade každodenné stretnutie tímu na pár minút.

Stretnutie mávame raz do týždňa v pondelok na pár hodín. Body stretnutia:

1. Zhrnutie doposiaľ vykonaných úloh

Prejdú sa všetky doposiaľ dosiahnuté míľniky

2. Kontrola vykonania úloh z minulého týždňa

Každý člen sa vyjadrí k jemu pridelenej úlohe, ako danú úlohu riešil, aké mal s ňou prípadné problémy a k akým výsledkom sa dopracoval. Tieto výsledky sú zhodnotené vedúcim tímu a celý tím o nich diskutuje. Tím sa potom dohodne na uzavretí alebo znovu otvorení už vyriešenej úlohy.

3. Naplánovanie úloh na ďalší týždeň

Z product backlogu sa vyberú používateľské scenáre na základe dohody celého tímu o ďalšom postupe. Každý používateľský scenár sa rozčlení na úlohy. Týmto úlohám je pridelená priorita a riešiteľ. Úlohy sa na základe priority rozplánujú na nasledujúci šprint. Každý riešiteľ sa vyjadrí k stanovenému času vyhotovenia jeho úlohy. Úlohy, ktorých výsledky nie sú vstupom do iných úloh a ich priorita je nízka môžu byť posunuté na neskôr.

Product backlog sa naplňa počas celého vývoja projektu a používateľské scenáre sú z neho čerpané na začiatku každého týždňa.

7.1 Plány

Pre projekt je potrebné stanoviť termíny a rozplánovať si ako sa bude postupovať. Z toho dôvodu sme sa venovali aj tvorbe plánu. V tejto kapitole je plán rozpísaný na viacero častí pre prehľadnosť.

Pre zimný semester sme si stanovili niekoľko cieľov, ktoré sa pokúsime dosiahnuť. Sú zobrazené v tabuľke (Tabuľka 7.1).

Tabuľka 7.1 Míľniky zimného semestra.

Dátum	Cieľ
15.10. 2012	<ul style="list-style-type: none"> • Nainštalovanie operačného systému Linux na virtuálnom stroji • Nainštalovanie webového servera Apache
22.10 2012	<ul style="list-style-type: none"> • Nainštalovanie SVN servera • Integrácia knižníc MASON a RVO2
29.10 2012	<ul style="list-style-type: none"> • Vytvorenie prvého scenára pre triviálnych agentov – agenti hýbu a vyhýbajú navzájom • Integrácia knižníc MASON a RVO2
5.11 2012	<ul style="list-style-type: none"> • Dva druhy agentov – demonštranti a poriadkové zložku
12.11 2012	<ul style="list-style-type: none"> • Odovzdanie dokumentácie k riadeniu • Odovzdanie dokumentácie k produktu
26.11 2012	<ul style="list-style-type: none"> • Pôsobenie síl na agentov
10.12 2012	<ul style="list-style-type: none"> • V dave sa šíria emócie z agenta na agenta, a ovplyvňuje to správanie davu aj jednotlivcov

Rozhodli sme sa vyvíjať metódou SCRUM. Šprint je základná časová jednotka v metóde SCRUM. My sme si dĺžku šprintu stanovili pre vývoj tohto projektu na dva týždne. V tabuľke (Tabuľka 7.2) sú rozplánované jednotlivé šprinty.

Tabuľka 7.2 Plán šprintov na zimný semester.

Názov šprintu	Obdobie	Kľúčové činnosti
1. Šprint	5. – 6. týždeň (15.10 – 29.10 2012)	<ul style="list-style-type: none"> • Príprava vývojového prostredia • Zabezpečenie verziovania • Integrácia knižníc pre implementáciu • Analýza poriadkových zložiek a demonštrácií • Implementovanie prvého scenára – agenti sa pohybujú a navzájom sa vyhýbajú • Testovanie prvého scenára • Vyhotovenie dokumentácie k prvému šprintu
2. Šprint	7. – 8. týždeň (29.10 – 12.11 2012)	<ul style="list-style-type: none"> • Analyzovanie psychologických faktorov agentov • Analyzovanie realtime zmeny parametrov RVO a agentov • Implementácia druhého scenára – dva druhy agentov • Testovanie scenára • Vyhotovenie dokumentácie k druhému šprintu. • Vyhotovenie dokumentácie k riadeniu
3. Šprint	9. – 10. týždeň (12.11 – 26.11 2012)	<ul style="list-style-type: none"> • Analyzovanie pôsobenia síl v dave • Navrhnutie, ktorý model síl sa vyberie • Implementácia tretieho scenára - pôsobenia sily na agentov • Testovanie tretieho scenára • Dokumentácia k tretiemu šprintu
4. Šprint	11. – 12. týždeň (26.11 – 10.12 2012)	<ul style="list-style-type: none"> • Analýza šírenia emócií • Návrh modelu správania sa • Návrh vhodného modelu šírenia emócií • Otestovanie ako sa správa simulácie pri rôzne nastavených parametroch • Dokumentácia k štvrtému šprintu
5. Šprint	13. týždeň (10.12 – 14.12 2012)	<ul style="list-style-type: none"> • Dokumentácia k piatemu šprintu • Dokončenie dokumentácie k riadeniu

Počas zimného semestra budeme preukazovať vynaložené úsilie tak, že budeme odovzdávať priebežne dokumentácia k produktu a k riadeniu. Taktiež budeme naše výsledky predstavovať ostatným prostredníctvom prezentácie. Preto sme si vytvorili plán tvorby výstupov. V tabuľke (Tabuľka 7.3) sú termíny odovzdávania dokumentácie a vybraných častí prototypu.

Tabuľka 7.3 Plán tvorby výstupov.

Termín	Úloha	Začiatok práce	Odhad (v dňoch)
11.10 2012	<ul style="list-style-type: none"> • Odovzdanie dokumentácie prvých dvoch šprintov • Odovzdanie dokumentácie k riadeniu 	5.11 2012	5
14.12 2012	<ul style="list-style-type: none"> • Odovzdanie dokumentácie za zimný semester • Odovzdanie prototypu vybraných častí projektu • Odovzdanie dokumentácie k riadeniu • Prezentácia projektu 	8.12 2012	5

7.2 Priebeh stretnutí tímového projektu

Stretnutie mávame raz do týždňa v pondelok na pár hodín. Body stretnutia:

1. Zhrnutie doposiaľ vykonaných úloh

Prejdú sa všetky doposiaľ dosiahnuté míľniky

2. Kontrola vykonania úloh z minulého týždňa

Každý člen sa vyjadrí k jemu pridelenej úlohe, ako danú úlohu riešil, aké mal s ňou prípadné problémy a k akým výsledkom sa dopracoval. Tieto výsledky sú zhodnotené vedúcim tímu a celý tím o nich diskutuje. Tím sa potom dohodne na uzavretí alebo znovu otvorení už vyriešenej úlohy.

3. Naplánovanie úloh na ďalší týždeň

Z product backlogu sa vyberú používateľské scenáre na základe dohody celého tímu o ďalšom postupe. Každý používateľský scenár sa rozčlení na úlohy. Týmto úlohám je pridelená priorita a riešiteľ. Úlohy sa na základe priority rozplánujú na nasledujúci šprint. Každý riešiteľ sa vyjadrí k stanovenému času vyhotovenia jeho úlohy. Úlohy, ktorých výsledky nie sú vstupom do iných úloh a ich priorita je nízka môžu byť posunuté na neskôr.

8 Manažment podpory vývoja

Náš tím sa po vzájomnej dohode rozhodol pre vývoj v prostredí s názvom NetBeans, keďže projekt bude implementovaný v jazyku Java a uvedené prostredie je po viacerých stránkach vhodnou voľou pre náš projekt. V projekte budeme používať verziovací softvér SVN, ktorý bude na našom linuxovom serveri (s distribúciou operačného systému Fedora v.16), kde je umiestnená taktiež aj naša webová stránka s aktuálnymi informáciami o tímovom projekte.

Každý člen tímu bude prispievať do projektu pod svojím používateľským kontom, aby sme vedeli rozlíšiť kto, kedy, a čo upravil alebo doplnil v zdrojovom kóde.

Komunikácie je taktiež dôležitá, a dohodli sme sa na komunikovaní prostredníctvom skupiny v Google Groups, kde sme vytvorili účet pod názvom „tim2012“ (kontakt: tim2012@googlegroups.com).

Pri vývoji je veľkou výhodou použitie nástroja na zaznamenávanie úloh a chýb, pričom náš tím používa systém JIRA (dostupný na jira.fiit.stuba.sk), kde máme vytvorené konto špeciálne za účelom nášho projektu. V nástroji sú zadané úlohy, dá sa v ňom sledovať celý vývoj, rozdelenie úloh a ich plnenie, a nástroj poskytuje mnoho ďalších informácií. JIRA priamo podporuje agilný vývoj softvéru.

9 Manažment monitorovania

Rozhodli sme sa, že v rámci tímového projektu bude monitorovanie spočívať v nasledujúcich úlohách:

1. Monitorovanie postupu prác
2. Monitorovanie zdrojových kódov

Monitorovanie postupu prác na projekte bude vykonávané v systéme Jira. Systém Jira je podporným nástrojom vývoja, ktorý dokáže podrobne zaznamenať stav riešenia pridelených úloh členom tímu. Po vyriešení úlohy budú následne výsledky skontrolované a prekonzultované na tímovom stretnutí, ktoré sa koná v týždenných intervaloch.

Monitorovanie zdrojových kódov je kľúčové pri práci vo viacčlennom tíme vývojárov. Je potrebné dohliadať na dodržanie požadovanej kvality zdrojových kódov, ako aj komentárov a štábnej kultúry tímu. V opačnom prípade by sa po určitom čase v zdrojových kódoch ľudia prestali orientovať, čo by viedlo priam ku katastrofálnym výsledkom samotného projektu. Z toho dôvodu budú vykonávané prehliadky zdrojových kódov. Prehliadky sa budú konať po vytvorení uceleného modulu. Autor daného modulu bude zodpovedný za otestovanie modulu, zdokumentovanie modulu a naplánovanie prehliadky. Pri prehliadke sa tím oboznámi s daným modulom a spôsobom použitia. Rovnako tím skontroluje spomínanú kvalitu zdrojových kódov, komentárov a štábnu kultúru. Prípadné zistené nedostatky bude potrebné odstrániť.

10 Manažment dokumentácie

Dokumentácia sa tvorí v našom projekte pomocou nástroja Google Drive, kam členovia tímu nahrávajú svoje hotové dokumentácie a manažér dokumentácie potom tieto dokumentácie spracuje. Google drive – nástroj na ukladanie súborov a ich synchronizáciu obsahuje štruktúru adresárov pre jednotlivé šprinty, aby manažér dokumentácie vedel, ku ktorému šprintu odovzdaná dokumentácia patrí.

Úlohy manažéra dokumentácie po prijatí dokumentácie od člena tímu:

- korekcia pravopisu a grafiky
- vytvorenie referencie na každý zdroj obsiahnutý v dokumente
- preklad z anglického jazyka
- integrácia popisov tabuliek a obrázkov
- posielanie správ o nedostatkoch odovzdaných dokumentácii ostatným členom tímu

Ak sú všetky dokumenty spojené ho jedného, manažér dokumentácie sa postará o finalizáciu dokumentov, čo zahŕňa:

- spísanie autorstva jednotlivých kapitol
- finálnu korekciu dokumentu

11 Manažment rizík

Pod pojmom riziko rozumieme možnosť utrpieť stratu, poškodenie alebo znevýhodnenie. Motiváciou pre manažment rizík je predchádzanie nepredvídaným udalostiam. Základné procesy manažmentu rizík sú nasledovné:

- Riadenie a sledovanie rizík
 - identifikácia rizík
 - analýza rizík
 - plánovanie manažmentu rizík

Každému riziku je potrebné určiť pravdepodobnosť s akou nastane. Uvažovať budeme tri úrovne pravdepodobnosti:

- Nízka
- Stredná
- Vysoká

Každému riziku je potrebné určiť rozsah škôd, ak by riziko nastalo. Uvažovať budeme tri úrovne dopadu na projekt:

- Malý
- Stredný
- Veľký

Závažnosť rizika sa vypočíta ako súčin pravdepodobnosti, že nastane a dopadu na projekt, ak nastane. Mieru rizika možno vizualizovať nasledovnou maticou:

Pravdepodobnosť Dopad	Nízka	Stredná	Vysoká
Malý	Zanedbateľná	Znesiteľná	Stredná
Stredný	Znesiteľná	Stredná	Vysoká
Veľký	Stredná	Vysoká	Neznesiteľná

11.1 Identifikované riziká

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 1	Popis rizika Odchod člena tímu	Identifikované 8.10.2012	
Pravdepodobnosť Stredná	Kontext Člen tímu nebude spolupracovať, plniť zadané úlohy a prispievať k celkovému úsiliu v tímovom projekte. Výsledky jeho práce budú minimálne a neuspokojivé.		
Dopad Stredný			
Časové ohraňenie Dlhodobé	Autor Filip Pakan	Dôsledok Prerozdelenie úloh	Zodpovedný Miroslav Ort
Kritérium spustenia Nezáujem o projekt, zlé vnútro tímové vzťahy, nesplnenie pridelených úloh			
Stratégia vyhnutia sa riziku Zúčastňovanie sa tímových stretnutí, častá komunikácia, časté preukazovanie výsledkov			
Plán nápravy Utužovanie tímu, častejšie diskutovanie o projekte, rozdelenie si úloh odídeného člena			

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 2	Popis rizika Zlyhá integrácia externých modulov	Identifikované 4.11.2012	
Pravdepodobnosť Stredná	Kontext Knižnice tretích strán sa nepodarí nainštalovať a správne nakonfigurovať. Knižnice nebudú navzájom spolupracovať, prípadne nebudú poskytovať požadovanú funkcionálnosť.		
Dopad Veľký			
Časové ohraňenie Krátkodobé	Autor Filip Pakan	Dôsledok Nedodržanie rozvrhu	Zodpovedný Michal Kyžňanský
Kritérium spustenia Zlý výber knižníc, neprečítanie dokumentácie ku knižniciam			
Stratégia vyhnutia sa riziku Podrobné oboznámenie sa s knižnicami, naštudovanie si ich možností			
Plán nápravy Zapojenie do riešenia problému viacerých členov tímu, vyhľadanie alternatívnych knižníc			

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 3	Popis rizika Nesplnenie naplánovaných úloh v šprinte		Identifikované 26.11.2012
Pravdepodobnosť Stredná	Kontext Nepodarí sa splniť všetky úlohy naplánované pre daný šprint, prípadne samotné úlohy sa splnia, ale nebudú zdokumentované na konci šprintu.		
Dopad Stredný			
Časové ohraňenie Dlhodobé	Autor Filip Pakan	Dôsledok Nedodržanie rozvrhu	Zodpovedný Miroslav Ort
Kritérium spustenia Zlé rozloženie času venovaného na tímový projekt, riešenie úloh na poslednú chvíľu			
Stratégia vyhnutia sa riziku Začať úlohy riešiť hneď po rozdelení, identifikovať problémy a komunikovať ich v tíme			
Plán nápravy Okamžité doriešenie nesplnených úloh, prípadne ich zdokumentovanie v daný deň			

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 4	Popis rizika Neprehľadnosť zdrojového kódu		Identifikované 26.11.2012
Pravdepodobnosť Vysoká	Kontext Zdrojový kód programu sa stane časom neprehľadný. Pri dopĺňaní novej funkcionality programátor nebude vedieť, kam začať písať nový kód. Existujúcemu kódu nebude rozumieť.		
Dopad Stredný			
Časové ohraňenie Dlhodobé	Autor Filip Pakan	Dôsledok Nedostatočná kvalita	Zodpovedný Michal Kyžňanský
Kritérium spustenia Nepoužívanie komentárov, nedodržanie štábnej kultúry, nepoužívanie UML diagramu tried			
Stratégia vyhnutia sa riziku Vytvorenie modelu systému, komentovanie zdrojového kódu, dodržanie štábnej kultúry			
Plán nápravy Zorganizovať tímový code review, nový kód pravidelne komunikovať v tíme			

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 5	Popis rizika Nedostatočné monitorovanie projektu		Identifikované 26.11.2012
Pravdepodobnosť Nízka	Kontext Nedostatočným monitorovaním dôjde k strate sledovateľnosti v projekte. Tím nebude vedieť, kto čím prispel k celkovému úsiliu v tímovom projekte.		
Dopad Stredný			
Časové ohraňenie Dlhodobé	Autor Filip Pakan	Dôsledok Nerovnomerné rozloženie úsilia	Zodpovedný Adrián Kollár
Kritérium spustenia Nepoužívanie systému riadenia úloh a monitorovania projektu			
Stratégia vyhnutia sa riziku Zaznamenávanie vykonaných úloh do systému JIRA a ich komunikovanie v tíme			
Plán nápravy Tímové stretnutie, komunikovanie vykonaných úloh			

Informácia o riziku			
Názov projektu Simulácia demonštrácie			
ID 6	Popis rizika Neidentifikovanie závažného rizika		Identifikované 26.11.2012
Pravdepodobnosť Stredná	Kontext Manažér rizík neidentifikuje závažné riziko v tímovom projekte. Ak takéto riziko nastane, tím to nezistí a nedokáže na neho včas zareagovať efektívnym spôsobom.		
Dopad Veľký			
Časové ohraňenie Dlhodobé	Autor Filip Pakan	Dôsledok Ohrozenie projektu	Zodpovedný Filip Pakan
Kritérium spustenia Nedostatočný manažment rizík, nepravidelné identifikovanie nových rizík			
Stratégia vyhnutia sa riziku Zainteresovanie celého tímu do identifikácie rizík			
Plán nápravy Celý tím vynaloží všetko svoje úsilie pre zmiernenie dopadu závažného rizika			

11.2 Vzniknuté riziká

Nasleduje zoznam rizík, ktoré počas projektu nastali a spôsob ich riešenia.

11.2.1 ID 1: Odchod člena z tímu

Kto: Bc. Martin Dušek

Kedy: 10.11.2012 15:28

Prečo: neuspokojivé výsledky, minimálne množstvo úsilia, neplnenie pridelených úloh načas

Spustenie: Bc. Martin Dušek nedodal dokumentáciu k inžinierskemu dielu do 10.11.2012 8:00

Podrobnosti: Vedúci tímu Ing. Peter Lacko, PhD. oznámil ukončenie spolupráce s Bc. Martinom Dušekom prostredníctvom tímového e-mailu. Dôvodom boli nesplnené požiadavky a minimum úsilia z jeho strany od začiatku semestra. Spustením bolo nedodanie dokumentácie k inžinierskemu dielu načas a v požadovanej kvalite. Podľa pripraveného plánu si jeho úlohu rozdelíme v tíme, pričom o finálne pospájanie dokumentácie sa postará Bc. Miroslav Ort.

12 Manažment komunikácie

Vývoj stredne veľkého projektu si vyžaduje koordináciu všetkých členov tímu. Na tento účel je dôležité, aby všetci členovia tímu vzájomne komunikovali. Počas vzájomnej komunikácie si vedľa členovia tímu zadeliť úlohy a získajú prehľad o riešenom projekte na určitej úrovni abstrakcie. Najvhodnejšia forma komunikácie umožňujúca koordináciu ľudí je osobná komunikácia. Osobne sme komunikovali najmä počas tímových stretnutí. Častokrát sme sa stretávali v škole aj mimo stretnutí a diskutovali problematiku tímového projektu. Keďže si povaha práce a frekvencia osobných stretnutí nebola na dostatočnej miere, v tíme sme používali rôzne komunikačné prostriedky na zdieľanie informácií a vzájomné dorozumievanie. Komunikačné nástroje nám výrazne pomohli riešiť vzniknuté nejasnosti a prispeli k vyššej informovanosti jednotlivcov o projekte a tým aj k lepším výsledkom.

12.1 Komunikačné nástroje

Pri práci v tíme na projekte simulácie demonštrácie sme používali tieto komunikačné nástroje:

- Email
- Google Talk
- Google Groups
- Google Drive
- Google Calendar
- Jira

Email

Tvorí väčšinu našej komunikácie. Pomocou emailu si posielame aj rozsiahlejšie texty a dopĺňujúce vysvetlenia. V tíme sa asynchronita tohto druhu komunikácie prejavuje v minimálnej miere, mailovú schránku si každý pravidelne sledujeme.

Google Talk

Instant messenger, ktorý nám cez prostredie internetového prehliadača umožňuje komunikovať. Komunikácia prebieha medzi dvomi členmi tímu. Je rýchla, synchronna a umožňuje riešiť nedorozumenia v reálnom čase.

Google Groups

Predstavuje spôsob, ako zapájať do diskusií členov tímu. Pomocou služby sa dá vyjadrovať k témam súvisiacim s tímovým projektom. Súčasťou skupiny diskutujúcich je aj náš vedúci Ing. Peter Lacko, PhD., ktorý môže diskusiu usmerniť správnym smerom a zároveň je informovaný o celom dianí v tíme. Väčšina vyjadrení, ktoré sa týkajú tímového projektu sa uverejňuje na Google Groups, čo nám umožňuje byť neustále informovaní o názoroch členov tímu. Ďalšou výhodou služby Google Groups je spoločná tímová adresa, z ktorej sa všetky správy preposielajú na mailové adresy členov tímu.

Google Drive

Všetky materiály, ktoré potrebujeme k riešeniu projektu uverejňujeme na Google Drive. Zároveň sem umiestňujeme dokumenty, na ktorých pracuje naraz viacero tímových členov.

Google Calendar

Online kalendár slúži na zdieľanie dôležitých termínov. Uverejňujeme všetky termíny týkajúce sa tímového projektu.

Jira

Nástroj slúžiaci na manažovanie projektu. Môžeme ho tiež považovať za prostriedok komunikácie. Každý člen tímu vidí svoje úlohy a ostatní presne vedia, aké úlohy sa v danom šprinte riešia a kým. Nástroj slúži aj na komunikáciu s okolím. Jira umožňuje tvorbu výkazov o prácach a šprintoch.

13 Metodiky používané pri vývoji

13.1 Metodika úloh

Použité pojmy

- SCRUM – metodika vývoja projektu
- Jira – podporný webový nástroj na manažovanie úloh
- Backlog – zoznam požiadaviek, pretransformovaných do používateľských scenárov
- Šprint – základná časová jednotka vývoja v SCRUM-e
- User story – používateľský scenár
- Sub-task – pod úloha

Role a zodpovednosti

Nasledujúca tabuľka (Tabuľka 13.1) obsahuje zoznam rolí, ktoré v jednotlivých procesoch vystupujú.

Tabuľka 13.1 Tabuľka rolí a zodpovedností.

Role	Zodpovednosť
Vedúci tímu	- rozhodnutie o uzavretí úlohy
Riešiteľ	- vyriešenie úlohy - zalogovanie času stráveného na riešení úlohy do Jiri - prenastavenie stavu úloh
Kontrolór	- kontrola správnosti vykonania úloh - prednesenie zhodnotenia vyriešenej úlohy vedúcemu tímu

Proces manažmentu úloh

V tabuľke (Tabuľka 13.2) sú zoradené procesy, z ktorých sa skladá manažment úloh.

Tabuľka 13.2 Procesy manažmentu úloh.

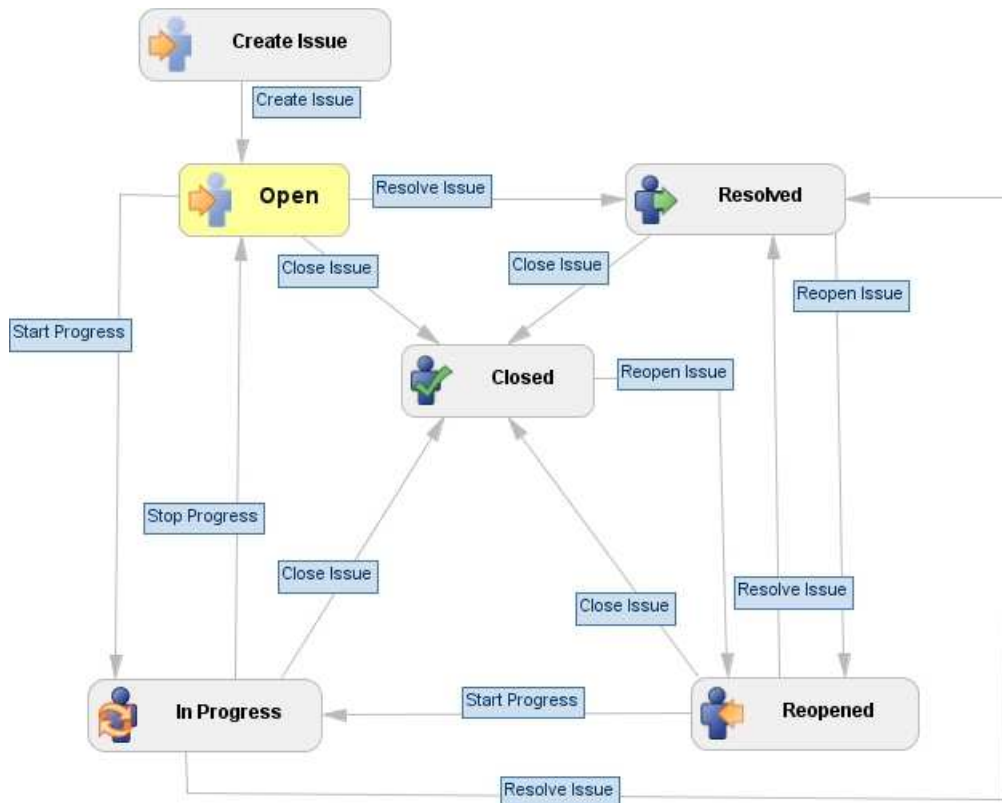
Krok	Názov
1	Proces plánovania úloh
2	Proces zaznamenávania úloh do podporného nástroja Jira a pridelenia úloh riešiteľom
3	Proces začatia riešenia úlohy
4	Proces ukončenia práce na úlohe
5	Proces zaznamenania času stráveného na úlohe
6	Proces kontroly vykonaných úloh
7	Proces zatvorenia úlohy
8	Proces znovu otvorenia úlohy

Životný cyklus úlohy

Úloha môže nadobudnúť tieto stavy:

- Open – úloha v tomto stave ešte nie je riešená, čaká až sa jej začne riešiť
- In progress – úloha je riešená riešiteľom
- Resolved – úloha je vyriešená
- Reopened – Úloha čaká na opravenie alebo doriešenie riešiteľom
- Closed – úloha je uzavretá, ďalej sa už nebude riešiť

Stavy, ktorými prechádza úloha počas svojej existencie, sú zobrazené na obrázku (Obr. 13:1).



Obr. 13:1 Životný cyklus úlohy (Prevzaté zo systému Jira).

13.1.1 Procesy manažmentu úloh

13.1.1.1 Proces plánovania úloh

Vstup: Požiadavky zákazníka.

Výstup: Úlohy vytvorené na základe vybraných používateľských príbehov

Zodpovednosť: Zodpovedným v tomto procese je celý tím

Na základe požiadaviek zákazníka sa vytvoria user stories, ktorými je naplnený product backlog. Počas stretnutia tímu sa vyberajú používateľské scenáre (User story) z product baglogu, podľa toho aký pomer náklady a veľkosť príspevku k úspešnosti produktu je pri vyberanej úlohe. Vybrané používateľské scenáre sa rozplánujú na konkrétne úlohy, ktoré budú riešené počas nasledujúceho šprintu.

13.1.1.2 Proces zaznamenávania úloh do podporného nástroja Jira a pridelenia úloh riešiteľom

Vstup: Vytvorená úlohy

Výstup: Pridelená úloha riešiteľovi zaznamenaná v systéme Jira s nastaveným stavom na otvorený (Open)

Zodpovednosť: Manažér plánovania

V Jire sa vytvoria úlohy a sú im k pridelený riešitelia podľa metodiky SIM_1219, aby mali všetci prehľad, kto na čom robí.

13.1.1.3 Proces začatia riešenia úlohy

Vstup:	Úloha v stave otvorená (Opened)
Výstup:	Úloha v stave riešená (In progress)
Zodpovednosť:	Riešiteľ danej úlohy

Po tom ako sa začne na úlohe pracovať, musí sa jej stav v Jire zmeniť z otvorená na riešená. Pri výbere úloh, ktoré sa začnú riešiť sa postupuje od úloh s najvyššou prioritou po úlohu z najnižšou prioritou. Viac o prioritách sa je napísané v metodike SIM_1219.

13.1.1.4 Proces ukončenia práce na úlohe

Vstup:	Úloha v stave riešená (In progress)
Výstup:	Úloha v stave vyriešená (Resolved) a čakajúca na skontrolovanie
Zodpovednosť:	Riešiteľ danej úlohy

Keď je úloha vypracovaná tak, že spĺňa všetky body zadania, jej stav je nutné zmeniť v Jire na vyriešená a tím oznámi kontrolórovi, že treba preveriť správnosť vyriešenia úlohy.

13.1.1.5 Proces zaznamenania času stráveného na úlohe

Vstup:	Úloha so stavom Vyriešená (Resolved)
Výstup:	Čas strávený na úlohe zalogovaný v systéme
Zodpovednosť:	Riešiteľ danej úlohy

Po zmene stavu úlohy na vyriešená je nutné zalogovať, koľko trvalo vyriešenie danej úlohy a napísať krátky opis ako bola úloha riešená. Tento proces opisuje metodika SIML_8567.

13.1.1.6 Proces kontroly vykonaných úloh

Vstup:	Úloha so stavom Vyriešená (Resolved)
Výstup:	Rozhodnutie o uzavretí úlohy
Zodpovednosť:	Kontrolór zvolený pre overenie správnosti vyriešenia danej úlohy a vedúci tímu

Vyriešená úloha je skontrolovaná členom tímu na tento úkon určitým a výsledky sú zhodnotené vedúcim tímu. Ak riešenie neobsahuje chyby a spĺňa všetky požiadavky, tak je vedúcim tímu vydané rozhodnutie o jej uzatvorení a naopak je vydané rozhodnutie o jej znovuo tvorení.

13.1.1.7 Proces zatvorenia úlohy

Vstup:	Rozhodnutie o zatvorení úlohy
Výstup:	Úloha v stave zatvorená (Closed)
Zodpovednosť:	Riešiteľ danej úlohy

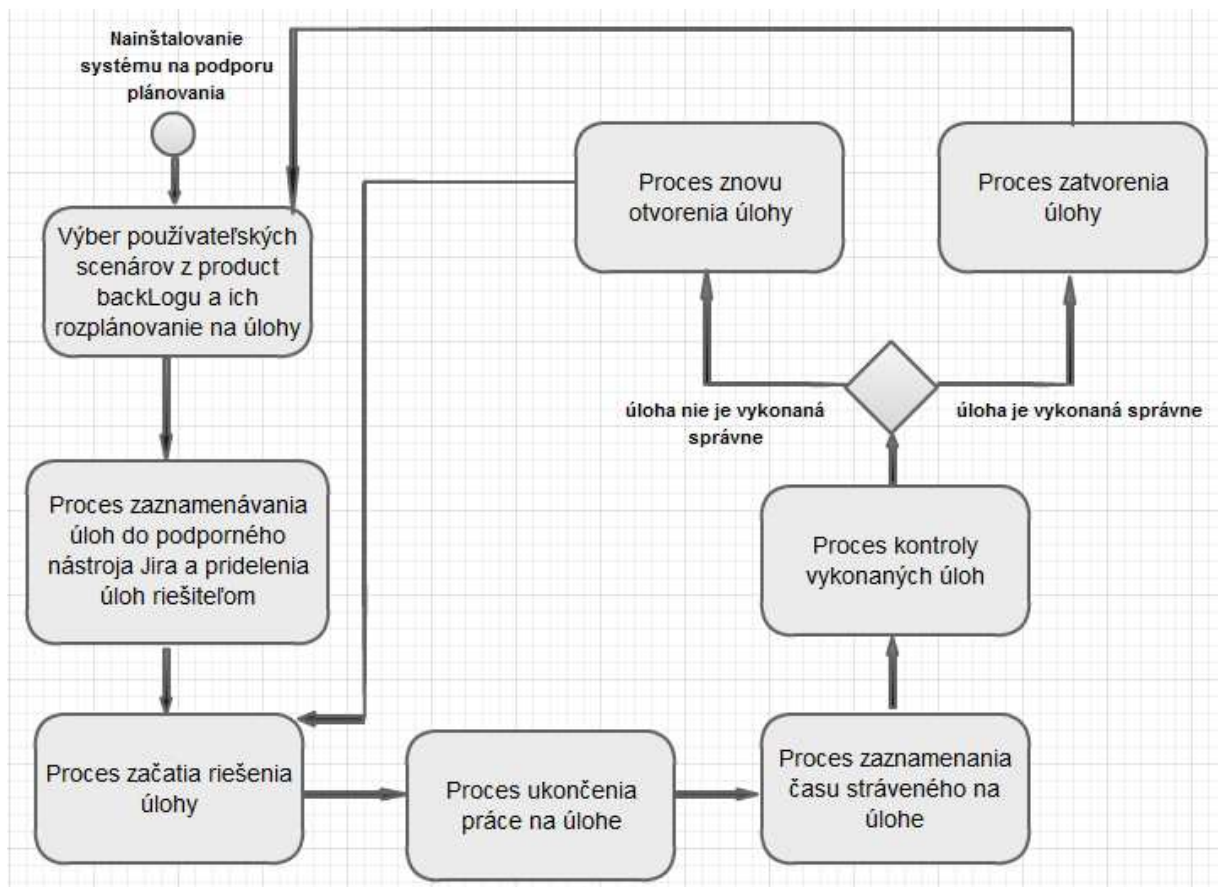
Po rozhodnutí, že úloha bude uzavretá, je potrebné do Jiri zaznamenať, že na úlohe sa úspešne ukončila práca. To sa vykoná zmenou stavu úlohy na zatvorená.

13.1.1.8 Proces znovu otvorenia úlohy

Vstup: Rozhodnutie o novu otvorení úlohy
 Výstup: Úloha v stave otvorená (Open)
 Zodpovednosť: Riešiteľ danej úlohy

Po rozhodnutí, že úloha bude znovu otvorená, je potrebné do Jiri zaznamenať, že na úlohe sa musí ešte pracovať. To sa vykoná zmenou stavu úlohy na otvorená.

Ako procesy za sebou nasledujú je zobrazené na obrázku (Obr. 13:2).



Obr. 13:2 Procesy.

13.1.2 Proces vytvárania úloh v systéme Jira

V tejto metodike sa píše akým spôsobom sa vyplní formulár pre vytvorenie úlohy a pod úlohy v systéme Jira.

Názov projektu - project

Člen tímu, môže súčasne pracovať aj na inom projekte v inom tíme. Z tohto dôvodu sa môže stať, že vidí v kolónke Projekt viac možností pre zaradenie úlohy k projektu. Je potrebné zvoliť ten názov projektu, pre ktorý ide úlohu zakladať.

Druhy úloh v Jire - Issue type

- User story
- Epic

User story je používateľský scenár, do ktorého sa vytvárajú pod úlohy (sub-task).

Epic reprezentuje Product backlog v Jire. Tento typ úlohy sa vytvára len raz a slúži na zálohovanie nápadov k ďalšiemu šprintu všetkých členov.

Názov úlohy – Summary

Názov sa skladá z podstatného mena a predmetu. Podstatné meno zaraďuje úlohu do jednej z týchto z oblastí vymenovaných v tabuľke (Tabuľka 13.3). Predmet má bližšie špecifikovať čoho sa daná úloha týka. Prvé písmeno je veľké a ostatné malé, pokiaľ názov neobsahuje vlastné mono. Napríklad, keď sa vytvára úloha pre oblasť analýzy a analyzovať sa bude simulácia demonštrácií, tak sa zvolí názov „Analýza simulácie demonštrácií“.

Tabuľka 13.3 Názov úlohy.

Podstatné meno	Predmet (čoho? / k čomu ?)
Analýza	
Návrh	
Implementácia	
Testovanie	
Podpora	
Dokumentácia	

Priorita úlohy – Priority

Priorita sa nastavuje podľa chronologického zaradenia úlohy v pláne a na základe dôležitosti jej vyriešenia pre úspech projektu. Jira poskytuje tieto možnosti nastavenia priorít (Tabuľka 13.4):

Tabuľka 13.4 Priority úloh.

Trivial	Úloha, na ktorej vyriešení nezávisí úspech projektu. Rieši sa, keď zostane čas.
Minor	Úloha, ktorá je dôležitá pre úspešné vyriešenie projektu, ale je možné jej vyriešenie odložiť na neskôr.
Major	Úloha, ktorá je dôležitá pre úspešné vyriešenie projektu a nie je možné jej vyriešenie odložiť na neskôr.
Critical	Úloha, ktorej vyriešenie je najdôležitejšie pre napredovanie v projekte, ale jej nevykonanie nebrzdí riešenie ďalších úloh.
Blocker	Úloha, ktorej vykonanie brzdí riešenie ďalších úloh. Jej výsledok tvorí vstup pre inú úlohu.

Dátum ukončenia práce na úlohe – Due date

Tento dátum sa nastavuje podľa toho ako určuje plán, ktorý bol schválený na tímovom stretnutí. Dátum sa vyberá kliknutím na obrázok kalendára a vybratím si konkrétneho mesiaca a dňa.

Komponenty - Componenty

Komponenty v Jire predstavujú podoblasti projektu. Vytvárať ich môže jedine administrátor podľa metodiky s názvom SimComp.

- Analýza
- Návrh
- Implementácia
- Testovanie
- Dokumentácia
- Technická podpora

Pri vyváraní úlohy sa budú úlohy zaraďovať do týchto kategórií z hľadiska lepšej filtrácie úloh. Podľa toho ako bolo dohodnuté na stretnutí sa zaraďí úloha do kategórie, teda podľa toho sa vyberie komponent.

Verzie – Affect version/s a Fix version/s

Verzie predstavujú iterácie v projekte. Verzie má tiež právo vytvoriť iba administrátor podľa metodiky SimVerzie.

Budú sa členiť na jednotlivé šprinty a ich názov bude tiež od nich odvodený. Je to z toho dôvodu, pretože po každom šprinte sa vytvorí prototyp produktu, ktorý tvorí vlastne novú verziu projektu. Názov verzie tvorí:

Číslovka	Šprint
-----------------	---------------

Treba do kolonky začať písať názov verzie a automaticky sa používateľovy vyselektujú všetky rovnako začínajúce názvy verzií. Z nich je potrebné vybrať jednu verziu, teda šprint, v ktorom má byť úloha vyriešená

Príklad: *Prvý šprint*

Jedinou výnimkou bude verzia s názvom Product backlog, ktorá sa vytvorí na začiatku vývoja. Túto verziu si môže nastaviť len úloha typu Epic, ktorá bola opísaná v metodike SIM_1219 v časti 1.1.2.

Riešiteľ - Assignee

Následne je úloha pridelená tomu členovi tímu, ktorí má na starosti jej vyriešenie alebo tento typ úloh manažuje. Do tohto políčka je potrebné napísať prihlasovacie meno riešiteľa, pod ktorým je v jire zaregistrovaný.

Napríklad: xbranisovaj1

Zadávateľ úlohy - Reporter

Toto políčko býva štandardne predvyplnené menom používateľa, ktorý úlohu vytvára. Ak je potrebné zadávateľa pozmeniť, do tohto políčka sa napíše jeho prihlasovacie meno.

Prostredie - Environmet

Toto políčko sa vyplňa len ak sa jedná o úlohu so zaradením do komponentu Implementácia. Vyplní sa názvom prostedia, v ktorom sa úloha, resp. časť projektu bude implementovať. Príklad: *Eclipse*

Proces vytvorenie úlohy sa dokončí kliknutím na tlačidlo Create.

Vytvorenie sub-tasku v Jire

Sub-task sa vytvorí keď bude treba úlohu prideliť viacerým riešiteľom. Sub-task sa v Jire vytvára veľmi podobne ako úloha. V texte nižšie sú opísané len rozdiely vo vytváraní.

Druh sub-tasku

Prvý rozdiel je v nastavovaní druhu sub-tasku. Sub-task má dva druhy a to, Technical Task a obyčajný sub-task. Typ sub-tasku sa odvoduje od nastavenia komponentu používateľského scenára, do ktorého je pod úloha vytváraná.

Typ úlohy

Komponent

<i>Technical task</i>	- Implementácia
	- Testovanie
	- Technická podpora
<i>Subtask</i>	- analýzu
	- dokumentáciu
	- návrh

Názov Sub-tasku – Summary

Názov tvorí podstatné meno, ktoré kopíruje názov úlohy, v ktorej sa subtask vytvoril a vlastné meno, ktorým je meno riešiteľa, ktorému je sub-task priradený.

Podstatné meno	-	Vlastné meno
-----------------------	---	---------------------

Príklad: Analýza simulácie demonštrácie - Jozef Mrkvička

Verzia sub-tasku a Komponent sub-tasku

Verzia a komponent sa nastavujú rovnako ako ich má nastavené úloha, v ktorej sa sub-task vytvára.

13.2 Metodika zberu požiadaviek

13.2.1 Úvod

13.2.1.1 Účel a rozsah dokumentu

Dokument obsahuje metodiku číslo 2012/1538 opisuje postupy spojené so zberom požiadaviek. Metodika je určená všetkým **interným** zamestnancom firmy MojaFirma s.r.o.

Dokument je určený pre stanovenie jednotného postupu pri zbere požiadaviek zákazníkov a ich zahrnutie do obchodných procesov firmy.

13.2.1.2 Prehľad dokumentu

Dokument je členený na tri časti.

V prvej časti sa nachádza popis účelu dokumentu, slovník pojmov, použité skratky a grafická notácia. Prvá časť dokumentu je tvorená kapitolou 1.

V druhej časti je popísaný proces zberu požiadaviek a podmienky pre začatie procesu. Túto časť pokrývajú kapitola 2.

Tretia časť dokumentu popisuje tvorbu user-story v informačnom systéme na podporu vývoja. Táto časť je pokrytá kapitolou 3.

13.2.1.3 Slovník pojmov

User story je požiadavka na časť funkcionality softvéru.

Team product owner je člen vývojového tímu, ktorý zastupuje rolu zákazníka v tíme.

Šprint je obdobie dvoch týždňov, počas ktorých vývojový tím vyvíja vopred dohodnutú funkcionality softvéru.

Business impact je vplyv zmeny softvéru na prácu zákazníka. Zmena softvéru predstavuje implementáciu novej funkcionality alebo zmenu pôvodnej.

Vývojový tím je tím ľudí, ktorý je zložený z odborníkov na každú oblasť vývoja softvéru. Ich úloha je vývoj a testovanie funkcionality softvéru. Ich výsledky priamo podliehajú vlastníčkovi produktu (team product owner).

13.2.1.4 Skratky






US: User story

TPO: Team product owner

13.2.1.5 Použitá notácia

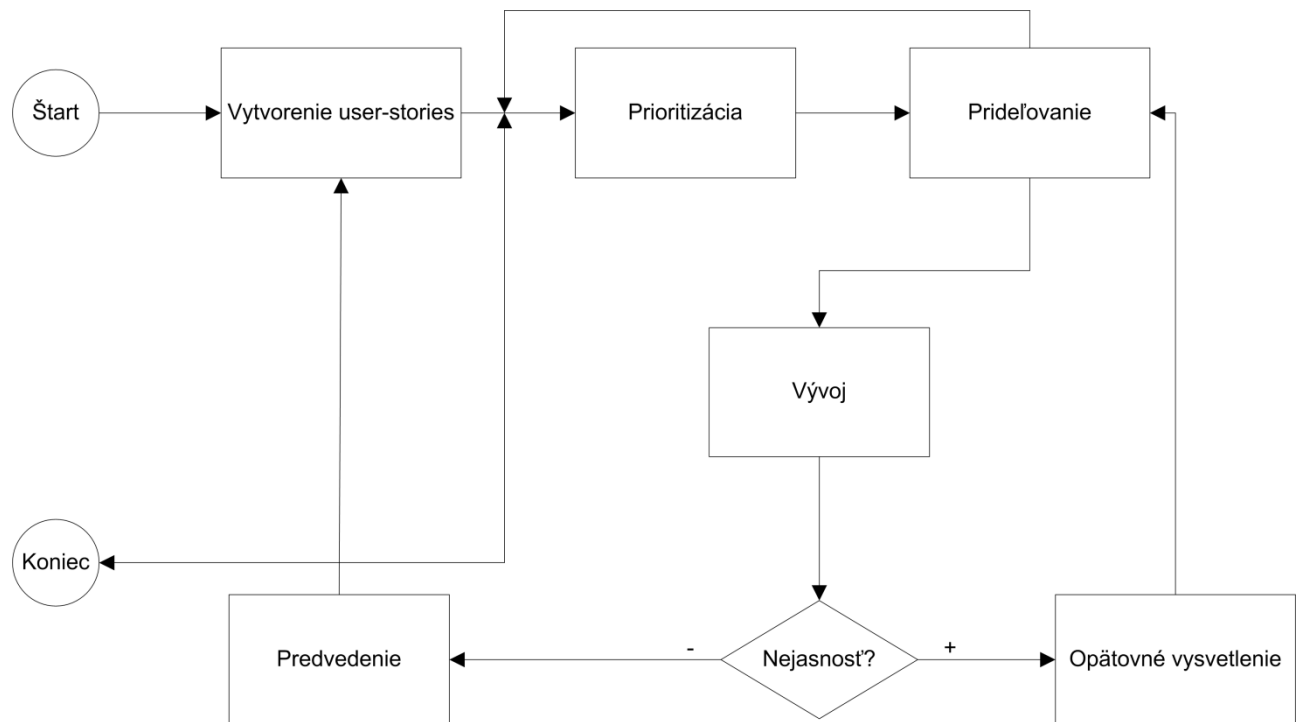
V dokumente je použitá notácia diagramov toku.

Diagram toku:

Inicializácia procesu: zobrazuje počiatočný bod toku procesu. Je zobrazený ako krúžok doplnený textom.	
Proces: je činnosť vykonávaná človekom alebo strojom. Predstavuje proces na úrovni skupiny aktivít vykonávaných za určitým účelom.	
Rozhodnutie: je logické rozhodnutie. Z hľadiska procesu nie je chápané ako samostatný krok ale iba ako informácia o ďalšom kroku procesu v závislosti od rozhodnutia. Znárodňuje sa kosoštvorcom.	
Tok: je prechod medzi dvomi procesmi. Znárodňuje sa ako šípka.	
Ukončenie procesu: ukončuje celý diagram toku. Znárodňuje sa krúžkom doplneným textom.	

13.2.2 Zber požiadaviek

Zber požiadaviek je proces, pri ktorom zákazník určuje, ktoré prvky má preňho vytvorený softvér obsahovať. Tento proces prebieha počas celého vývoja softvéru. Zber požiadaviek začína okamihom podpísania kontraktu na dodávku softvéru. Náležitosti kontraktu ako aj procesu vyjednávania o kontrakte sa nachádzajú v metodike číslo 5487 z roku 2010 ([metodika_2010/5487](#)). Proces zberu požiadaviek je znázornený na obrázku nižšie (Obr. 13:3).



Obr. 13:3 Proces zberu požiadaviek.

Fázy procesu zberu požiadaviek sú popísané v nasledujúcich podkapitolách metodiky.

13.2.2.1 Štart procesu

Vstup: Potreba zákazníka vlastniť softvér.

Výstup: Premyslená funkcionálna softvéru.

Zodpovednosť: Zákazník.

Opis: Proces zberu požiadaviek začína okamihom, kedy zákazník nadefinuje, aký softvér požaduje.

13.2.2.2 Vytvorenie user stories

Vstup: Premyslená funkcionálna softvéru.

Výstup: User stories.

Zodpovednosť: Team product owner.

Opis: User story chápeme ako jednu alebo niekoľko viet v jazyku klienta, ktoré zachytávajú to, čo sa používateľ so systémom robí. User stories definujú vlastnosti a funkcie vyvíjaného informačného systému. Sú súčasťou akceptačných testov pri odovzdávaní informačného systému zákazníkovi. Presný popis tvorby a obsahu user-stories je popísaný v metodike číslo 1638 z roku 2012 ([metodika 2012/1638](#)).

13.2.2.3 Prioritizácia

Vstup: User stories.

Výstup: User stories s prioritou.

Zodpovednosť: Team product owner.

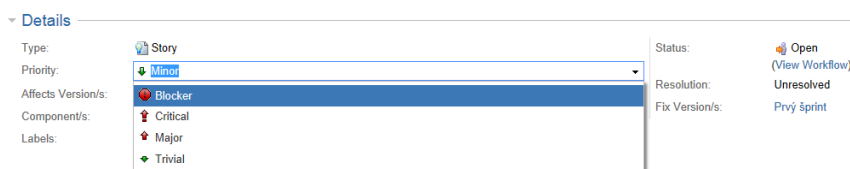
Opis: Prioritizácia je proces, počas ktorého zákazník ohodnotí všetky dosiaľ nadefinované user stories podľa toho, aký majú business impact. Prioritizáciu vykonáva zákazník spolu s TPO v systéme Jira. Prítomnosť TPO je nevyhnutná, lebo je administrátor system Jira a môže meniť priority user stories. Prioritizácia sa vykonáva

buď počas vzniku US alebo následne po vzniku. Zákazník spolu s TPO zaznačí do políčka *Priority* prioritu, s ktorou chce, aby na danej user story vývojový tím pracoval (Obr. 13:4). Zákazník nemôže definovať vlastnú úroveň priority pre user story, musí určiť jednu z možností priority:

- Blocker,
- Critical,
- Major,
- Minor,
- Trivial.

Priorita user story sa interpretuje:

- Priorita *Blocker* znamená, že zákazník nutne potrebuje naimplementovať a otestovať user story už v najbližšom šprinte. Bez tejto funkcionality nemá vývoj ďalších pre zákazníka zmysel.
- Priorita *Critical* znamená, že zákazník nutne potrebuje naimplementovať a otestovať user story už v najbližšom šprinte. Nie je však nutné ju spraviť na to, aby zákazník nemohol používať iné naimplementované user stories.
- Priorita *Major* znamená dôležitú user story, ktorú zákazník potrebuje. Nie je nevyhnutné ju naimplementovať a otestovať v rámci jedného šprintu.
- Priorita *Minor* znamená, že daná US nie je pre zákazníka nevyhnutná a jej preloženie na ďalší šprint nepredstavuje pre zákazníka riziko, tj. business impact je nízky.
- Priorita *Trivial* znamená, že zákazník nevie rozhodnúť, či user story má byť naimplementovaná v nasledujúcom šprinte. Zákazník jej nedodanie nepovažuje za riziko, ale jej implementácia by bola pre zákazníka prínosom. Takto prioritizovaná user story bude predmetom debaty v ďalšom šprinte.



Obr. 13:4 Priority user story.

V procese prioritizácie sa zákazník vyjadrí ku všetkým, tj. aj už prioritizovaným, user stories z dôvodu možnej reprioritizácie ešte nenaimplementovaných user stories.

13.2.2.4 Pridelovanie

Vstup: User stories s prioritou.

Výstup: Časť vytvorených user stories s prioritou je pridelená na vývoj.

Zodpovednosť: Team product owner.

Opis: Pridelovanie je process, v ktorom sa user story prideli na analýzu a vývoj konkrétnemu vývojárovi. Pridelovanie sa vykonáva až v momente, kedy sú všetky user stories oprioritizované zákazníkom, tj. po procese prioritizácie. Pridelenie vykonáva Team product owner v spolupráci s vývojovým tímom. TPO prideliuje user stories podľa priority zostupne. TPO prideli user stories s najvyššou prioritou vývojárom, ktorí v danej dobe nepracujú na inej US. Vývojár musí s pridelením ústne súhlasiť a musí mu byť vysvetlené, akú funkcionality US predstavuje. V prípade, že nesúhlasí musí uviesť dôvod, ktorý TPO uzná za relevantný. TPO prideli US vývojárovi tak, že v políčku

Assignee klikne v systéme Jira na meno vývojára pri danej US. Vývojár v tom okamihu zaškrtnie *Start progress*, aby TPO vedel, že na US sa pracuje.

13.2.2.5 Nejasnosť

Vstup: User story s prioritou pridelená na vývoj.

Výstup: User story s prioritou opätovne pridelená na vývoj.

Zodpovednosť: Vývojár, Team product owner.

Opis: Nejasnosť je definovaná ako nemožnosť implementácie user story z dôvodu neporozumenia jej účelu. Takáto US je prevedená do stavu *Open* kliknutím na *Stop Progress* v systéme Jira. Vývojár zároveň do komentára dopíše v čom bola nejasnosť. Tvorbou komentárov sa zaoberá samostatná metodika číslo 2584 z roku 2009 ([metodika 2584/2009](#)). Ak sa US dostane do stavu *Open* počas šprintu, TPO má povinnosť opätovne vysvetliť US vývojárovi. Opätovné vysvetlenie prebieha s súlade s procesom *Opätové vysvetlenie*.

13.2.2.6 Opätovné vysvetlenie

Vstup: User story v stave *Open* počas šprintu s komentárom s nejasnosťou.

Výstup: User story s prioritou opätovne pridelená na vývoj v stave *In Progress*.

Zodpovednosť: Vývojár, Team product owner.

Opis: Opätovné vysvetlenie je proces, ktorý prebieha tesne v okamihu, ak vznikla vývojárovi nejasnosť pri vývoji US. TPO vysvetlí zaznamenané nejasnosti a pridelí US v súlade s procesom pridelovania user stories. Opätovné pridelenie prebieha podľa procesu *Pridelovanie* s tým rozdielom, že vývojár dostane tú US, v ktorej mal nejasnosti bez ohľadu na prioritu danej US.

13.2.2.7 Vývoj

Vstup: Pridelené user stories s prioritami.

Výstup: Naimplementované user stories.

Zodpovednosť: Vývojári.

Opis: Vývoj je proces, ktorý prebieha počas celého šprintu. Jeho primárnou úlohou je naimplementovať všetku funkcionality pokrývajúcu user stories naplánované do daného šprintu. Ak počas procesu príde k nejasnostiam požiadaviek zákazníka, tak sa zaznačia v súlade s procesom *Nejasnosť*. Samotným procesom vývoja sa zaoberá metodika číslo 2684 z roku 2009 ([metodika 2684/2009](#)).

13.2.2.8 Predvedenie

Vstup: Naimplementované user stories.

Výstup: Ohodnotené user stories.

Zodpovednosť: Vývojári, Team product owner.

Opis: Predvedenie je proces, ktorý má dve fázy a prebieha po skončení šprintu. V prvej fáze vývojári ukážu naimplementované user stories TPO, ktorý ich zhodnotí. Ak TPO nie je spokojný, user story nie je ukončená a nechá sa opätovne prioritizovať zákazníkovi. V druhej fáze TPO predvedie naimplementovanú funkcionality zákazníkovi, ktorý ju ohodnotí. Následne zákazník dodá nové požiadavky na funkcionality softvéru v súlade s procesom *Vytvorenie user stories*.

13.2.2.9 Koniec

Vstup: Naimplementované user stories.

Výstup: Funkčný softvér pripravený pre zákazníka.

Zodpovednosť: Team product owner.

Opis: Koniec procesu zberu požiadaviek nastáva v okamihu, kedy sú všetky user stories s prioritou vyššou ako *Minor* naimplementované.

13.2.3 Vytvorenie user story

Vytvorenie user story tvorí základ pre spracovanie požiadavky zákazníka na funkcionality softvéru. Všetka funkcionality je spísaná formou user stories. User story obahuje jednu alebo niekoľko viet v jazyku používateľa (v tomto prípade nerozlišujeme medzi pojmi zákazník a používateľ).

User story sa skladá z troch častí:

- rola,
- čo chce robiť,
- účel.

Rola predstavuje typ používateľa systému s nastavenými právomocami.

Čo chce robiť používateľ v systéme, tj. akú funkcionality požaduje.

Účel práce, ktorú používateľ vykonal definujeme ako výsledok softvéru na akcie používateľa.

Ukážka user story je tabuľke nižšie (Tabuľka 13.5).

Tabuľka 13.5 Ukážka user story.

User Story	Ako ...
	Manažér predaja
	By som chcel ...
	Nastaviť zľavu pre skupinu subjektov
	Nato, aby ...
	V ďalšom účtovacom období započítala zľava do ceny

User stories vytvára zákazník za prítomnosti TPO. Za formuláciu US je zodpovedný výhradne zákazník. TPO sa môže zákazníka pýtať na účel konkrétnej US a môže si ju dať vysvetliť. Team product owner zaznamenáva user stories do systému Jira. Vyvíjaný softvér je v Jire vedený ako projekt. TPO v projekte vytvorí novú user story stlačením tlačidla *Story* s piktogramom v pravom hornom rohu obrazovky vedľa popisku *Create*. TPO na novom formulári ako prvé klikne na tlačidlo *Configure Fields* (Obr. 13:5). Klikne na *Custom* a vyberie nasledujúce položky:

- Affects Version/s
- Attachment
- Description
- Components,

- Due Date
- Priority

Ostatné možnosti voľby ponechá nezaškrtnuté. V procese tvorby US nie sú pre nás relevantné. Následne sa TPO venuje vyplňaniu ostatných položiek v okne podľa želania zákazníka (**Error! Reference source not found.**). Polia sa vyplňajú nasledovne:

Project – preddefinovaný názov projektu. Tento predstavuje interný názov súboru všetkých prác potrebných na dodanie požadovaného softvéru zákazníkovi. TPO ponechá.

Issue Type – preddefinované na *Story*. TPO ponechá.

Summary – názov user story. Predstavuje stručný ale výstižný názov pre US. TPO dopíše. Názov user story tvorí minimálne 1 a maximálne 10 slov. Zodpovedajú požiadavke zákazníka z user story z časti **By som chcel**. Sloveso z uvedenej časti user story je spodstatnené (Tabuľka 13.6). Popis je uvedený v slovenskom alebo anglickom jazyku.

Tabuľka 13.6 Názov user story.

By som chcel	Názov user story
Nastaviť zľavu pre skupinu subjektov	Nastavenie zľavy pre skupinu subjektov

Priority – priorita user story. TPO ponechá na preddefinovanej hodnote Major, alebo zmení podľa želania zákazníka. K zmene môže dôjsť aj po vytvorení US v súlade s metodikou 1538 z roku 2012 ([metodika 2012/1538](#)).

Due Date – dátum, do ktorého sa má US naimplementovať a otestovať. TPO nastaví v súlade s požiadavkami zákazníka v tvare deň/3_písmenková_skratka_mesiac/posledné_dvojčíslenie_roku, napr.: 11/Nov/2012. Dvojpísmenková skratka mesiaca je uvedená v anglickom jazyku.

Component/s – názov jedného alebo viacerých komponentov softvéru, ktorých sa US týka. TPO vyberie zo zoznamu všetky ovplyvnené komponenty.

Affects Version/s – zaznačenie verzí produktu, ktoré sú ovplyvnené danou US. TPO vyberie jednu alebo viacero zo zoznamu.

Description – slovný popis US v štandardnom tvare (Tabuľka 13.5). Popis je krátky (do 100 slov) a musí byť napísaný v slovenskom alebo anglickom jazyku. Jazyk popisu sa musí zhodovať s jazykom použitým v časti *Summary*. Vyplní TPO podľa slov zákazníka. Popis predstavuje celú user story so všetkými jej časťami (Tabuľka 13.5) zapísanú do jednej vety.

Attachment – pripojenie jedného alebo viacerých súborov vo formáte *png* tak, aby spoločná veľkosť súborov nepresiahla 10 MB. Súbor slúži na spresnenie požiadaviek alebo vizuálnu ukážku funkcionality. Názov obrázku predstavuje 1 až 5 slov oddelených podčiarkovníkom („_“). Súbor je formátu *png*. TPO môže alebo nemusí uviesť a robí tak so súhlasom zákazníka.

Všetky vyššie definované polia musia byť vyplnené alebo musia mať preddefinované hodnoty okrem poľa *Attachment*. Vytvorenie user story TPO potvrdí stlačením tlačidla *Create* (Obr. 13:5).

Obr. 13:5 Vytvorenie novej user story.

13.3 Metodika manažmentu chýb

13.3.1 Úvod

Cieľom tejto metodiky je definovanie procesov pri manažmente chýb za účelom zvýšenia kvality produktu. Pod chybou sa v tomto dokumente označuje softvérová chyba, ktorá spôsobuje nesprávne alebo neočakávané správanie softvérového systému. Ďalej sa v tejto metodike uvádza detailný opis zaevidovania chyby do systému pre sledovanie projektu. Metodika predpokladá používanie softvérového systému JIRA. Táto metodika je určená pre malé tímy (do 10 ľudí) vyvíjajúce softvér, avšak základné procesy možno aplikovať aj pre stredne veľké tímy, či organizácie.

13.3.2 Súvisiace metodiky

- Metodika pre pridelovanie úloh
- Metodika pre stanovenie priorít úloh
- Metodika pre vykonávanie testov

13.3.3 Vymedzenie pojmov

JIRA

- podporný systém pre riadenie projektu, zaznamenávanie úloh a evidovanie chýb

Bug

- termín označujúci softvérovú chybu

Stavový diagram

- graf definujúci všetky možné stavy entity a prechody medzi stavmi

13.3.4 Klasifikácia chýb

K chybám môže dôjsť v každej etape vývoja softvéru. Podľa toho v ktorej etape došlo k chybe, vzniknuté chyby sa klasifikujú nasledovne:

13.3.4.1 Analytické

- vznikli vo fáze analýzy nepochopením požiadaviek zákazníka
- majú za následok nesprávnu funkcionálnu architektúru systému (systém nerobí, to čo by mal robiť)
- je dôležité ich včas identifikovať, inak je ich odstránenie najnákladnejšie

13.3.4.2 Návrhové

- vznikli vo fáze návrhu nesprávnym premietnutím požiadaviek zákazníka do architektúry systému
- majú za následok nesprávnu architektúru systému (systém robí to, čo má robiť, ale nerobí to tak, ako by to mal robiť)
- po začatí implementácie systému sa zvyšujú náklady na odstránenie

13.3.4.3 Implementačné

- vznikli vo fáze implementácie pochybením alebo nepozornosťou programátora
- objavujú sa v každom softvérovom systéme
- nedá sa im celkom vyhnúť
- majú za následok neočakávané správanie systému
- ich odstránenie je spravidla najmenej nákladné

13.3.4.4 Testovacie

- vznikli vo fáze testovania nesprávnym otestovaním systému
- majú za následok dodanie systému s chybami
- ich odstránenie je kľúčové pre dodanie odladeného systému a zabezpečenie kvality produktu

13.3.5 Roly a zodpovednosti

Na začiatku procesu manažovania chýb stojí človek, ktorý chybu v systéme objaví. Jeho zodpovednosťou je o chybe informovať. Ak tak neurobí, tak nie je čo manažovať. Manažment chýb predpokladá na vstupne objavenú a nahlásenú chybu.

Úlohou testera je potvrdiť relevantnosť chyby a zaevidovať ju do systému. Projektový manažér následne pridelí chybe prioritu, termín opravy a osobu, ktorá ju bude riešiť. Po

opravení chyby tester overí opravu a buď chybu uzavrie alebo ju vráti späť na doriešenie.

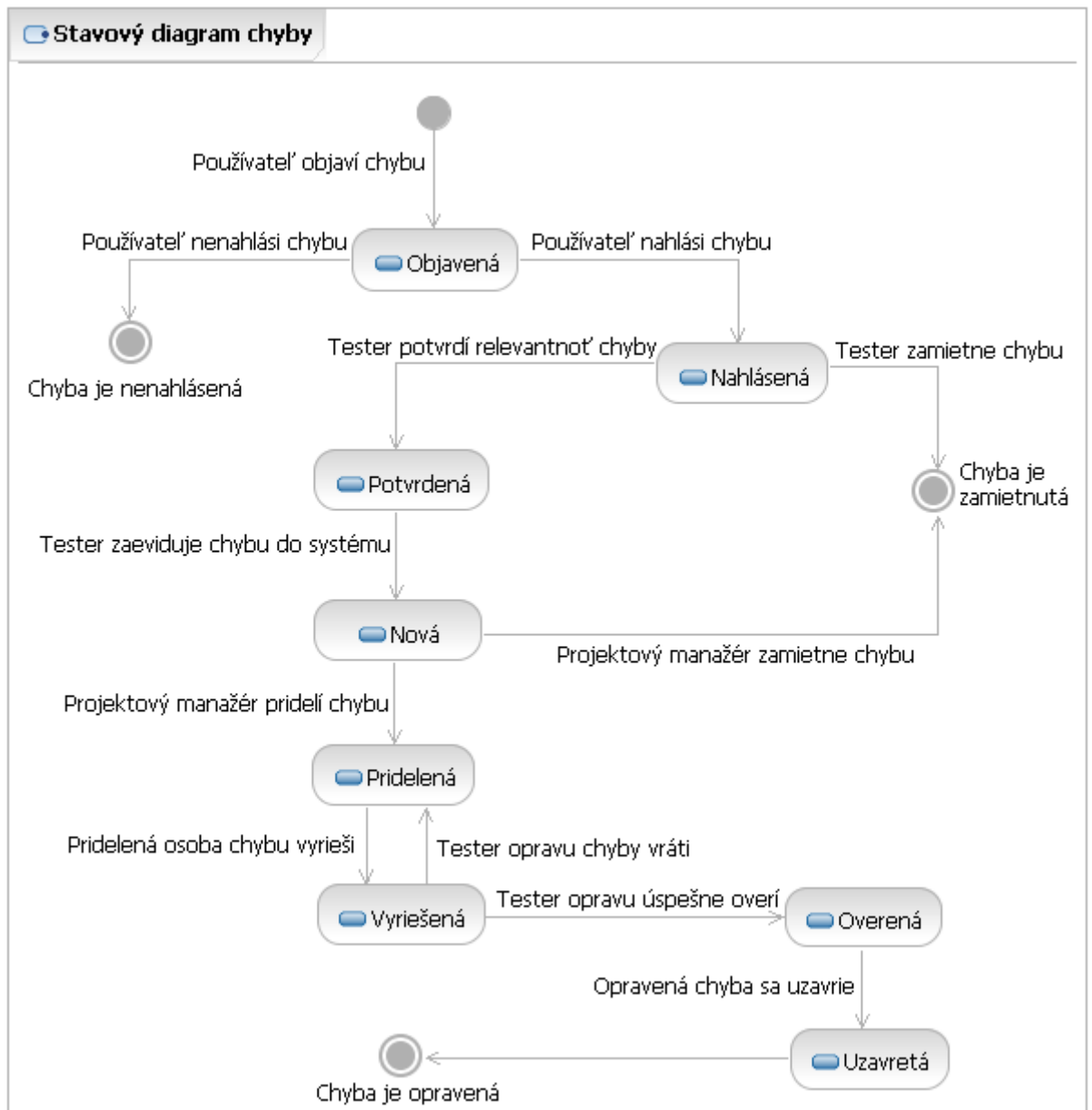
Zodpovednosti všetkých účastníkov manažmentu chýb sú uvedené v Tabuľka 13.7.

Tabuľka 13.7 Roly a zodpovednosti účastníkov manažmentu chýb.

Rola	Zodpovednosť
Nálezca chyby	<ul style="list-style-type: none"> ▪ Nahlásiť chybu
Projektový manažér	<ul style="list-style-type: none"> ▪ Stanovenie termínu opravy chyby ▪ Pridelenie opravy chyby
Analytik	<ul style="list-style-type: none"> ▪ Oprava analytickej chyby
Návrhár	<ul style="list-style-type: none"> ▪ Oprava chyby v návrhu systému
Programátor	<ul style="list-style-type: none"> ▪ Oprava implementačnej chyby
Tester	<ul style="list-style-type: none"> ▪ Potvrdenie relevantnosti chyby ▪ Zaevidovanie chyby do systému ▪ Overenie opravy

13.3.6 Životný cyklus chyby

Softvérová chyba prechádza životným cyklom, ktorý je definovaný stavmi, v ktorých sa chyba môže nachádzať. Prechody medzi stavmi sú determinované procesmi, ktoré chybu spracovávajú. Stavový diagram chyby je znázornený na Obr. 13:6.



Obr. 13:6 Životný cyklus chyby.

13.3.7 Identifikované procesy

Manažovanie chýb je komplexná činnosť, ktorá vyžaduje zainteresovanie viacerých účastníkov. Každý jeden účastník prispieva svojimi činnosťami k hladkému a bezchybnému priebehu manažovania chýb. Nasleduje zoznam identifikovaných procesov súvisiacich s manažmentom chýb.

13.3.7.1 Nahlásenie chyby

Aby bolo možné chybu opraviť, je nevyhnutné, aby nálezca chyby chybu nahlásil. V prípade, že sa jedná o používateľa kontaktuje oddelenie podpory. Ak ide o testera, ktorý objavil chybu v systéme pri testovaní, zaeviduje chybu do systému.

Vstupné podmienky:

- V systéme bola objavená chyba

Výstupné podmienky:

- Chyba bola nahlásená na príslušných miestach

Účastníci:

- Nálezca chyby

13.3.7.2 Potvrdenie chyby

Každú nahlásenú chybu od používateľov treba potvrdiť, či sa skutočne jedná o chybu alebo o vlastnosť systému. Tester musí mať k dispozícii analýzu a špecifikáciu požiadaviek, aby vedel určiť, či skutočne ide o chybu. Pokiaľ vyhodnotí správanie systému ako korektné, môže chybu zamietnuť.

Vstupné podmienky:

- Chyba bola nahlásená

Výstupné podmienky:

- Relevantné chyby sú potvrdené
- Nerelevantné chyby sú zamietnuté

Účastníci:

- Tester

13.3.7.3 Zaevidovanie chyby

Každú potvrdenú chybu musí tester zaznamenať do systému pre manažment chýb. Do systému uvedie záznam o chybe, ktorý obsahuje informácie o tom, aký dopad môže mať chyba na funkčnosť celého softvéru, aká kombinácia vstupov a používateľských interakcií vedie k chybe a zhodnotenie chyby.

Vstupné podmienky:

- Chyba bola potvrdená ako relevantná

Výstupné podmienky:

- Chyba je zaznamenaná v systéme pre manažment chýb spolu s jej popisom

Účastníci:

- Tester

13.3.7.4 Pridelenie chyby

Projektový manažér vyberie zo systému novú, zaevidovanú chybu. Určí jej prioritu podľa rozsahu škôd, ktoré môže napáchať. Pri stanovovaní priority mu asistuje zhodnotenie testera uvedené v zázname o chybe. Pokiaľ si určením priority nebude istý, má právo si zavolať kompetentných ľudí.

Na základe priority, ktorú chybe priradil, stanoví termín opravy chyby. Následne podľa kategórie chyby, prideli chybu konkrétnemu vývojárovi.

V špeciálnych prípadoch, ak by oprava chyby vyžadovala nepomerne veľa úsilia a zdrojov vzhľadom k rozsahu škôd, ktoré chyba spôsobí, môže sa projektový manažér rozhodnúť zamietnuť opravu chyby.

Vstupné podmienky:

- Nová chyba bola zaevidovaná do systému

Výstupné podmienky:

- Chyba má určenú priorita
- Chyba má stanovený termín opravy
- Chyba je pridelená konkrétnemu vývojárovi
- Chyba je zamietnutá, ak ju projektový manažér odmietne riešiť

Účastníci:

- Projektový manažér

13.3.7.5 Vyriešenie chyby

Vývojár si pridelenú úlohu vyberie zo systému a prečíta si jej záznam. Z toho zistí, pri akých vstupoch sa chyba prejaví. Ďalej vie, do kedy ju musí opraviť a tiež závažnosť chyby na celkový chod systému. Po vyriešení chyby musí zadať do systému report o oprave chyby, kde zhodnotí, čo všetko v systéme upravil, aby chyba už viac nenastala.

Vstupné podmienky:

- Chyba bola pridelená vývojárovi

Výstupné podmienky:

- Chyba je opravená
- Report o oprave je zadaný v systéme

Účastníci:

- Analytik
- Návrhár
- Programátor
- Tester

13.3.7.6 Overenie opravy chyby

Oprava chyby sa musí overiť, či prebehla úspešne. Pokiaľ tester zistí, že chyba bola opravená a systém sa správa korektne, môže chybu v systéme uzavrieť. Inak musí chybu vrátiť späť vývojárovi, nech svoju opravu chyby dovŕši do úspešného konca.

Vstupné podmienky:

- Chyba bola opravená
- Záznam o oprave bol uložený do systému

Výstupné podmienky:

- Ak bola oprava úspešná, chyba je uzavretá
- Ak bola oprava neúspešná, chyba je vrátená vývojárovi

Účastníci:

- Tester

13.3.8 Podrobne opísaný proces zaevidovania chyby

Táto časť metodiky detailne definuje a opisuje všetky kroky procesu zaevidovania chyby do systému pre manažment chýb JIRA. Správne nastavený proces zaevidovania chýb je pre výslednú kvalitu produktu nesmierne dôležitý. Jasne, zrozumiteľne a jednoznačne opísané chybové správanie systému podporuje rýchlejšie odstránenie chyby a poskytnutie kvalitného softvéru zákazníkovi.

Proces zaevidovania chyby predpokladá na vstupe potvrdenú relevantnosť chyby. Všetky nahlásené chyby, ktoré neboli relevantné, boli v predchádzajúcom procese zamietnuté. Na výstupe tohto procesu musí byť v systéme JIRA chyba zaznamenaná a zrozumiteľne opísaná.

13.3.8.1 Vyhľadanie chyby v systéme

Pre každú nahlásenú a potvrdenú chybu musí člen tímu zistiť, či už rovnaká chyba nie je v systéme evidovaná. Vyhľadanie chyby v systéme JIRA pozostáva z nasledujúcich krokov:

1. Autor záznamu o chybe sa prihlási do systému JIRA so svojim používateľským menom a heslom.
2. Autor záznamu si zvolí v menu záložku *Issues* a v rozbalenom menu položku *Search for Issues*. Vľavo sa mu zobrazí panel na vyhľadávanie znázornený na Obr. 13:7.

The image shows a search interface for JIRA issues. At the top is a 'Query' text input field with a yellow question mark icon to its right. Below the input are four checkboxes: 'Summary' (checked), 'Description' (checked), 'Comments' (unchecked), and 'Environment' (unchecked). Underneath is a 'Project' dropdown menu with 'All projects' and 'SimTEAM' (highlighted in blue). Below that is an 'Issue Type' dropdown menu with 'Any', 'Standard Issue Types', 'Bug' (highlighted in blue), 'New Feature', and 'Task'. A yellow question mark icon is to the right of the 'Issue Type' dropdown. At the bottom is a 'Search' button.

Obr. 13:7 Panel určený na vyhľadávanie chýb.

3. Do poľa *Query* zadá výraz na vyhľadanie. Následne si zvolí, či chce prehľadávať v názvoch zaevidovaných chýb, v ich popise, v komentároch alebo aj v prostredí, ktorého sa chyba týka. Vzhľadom na to, že rovnakú chybu do systému v minulosti

nemusel zadávať on, môže byť taká istá chyba v systéme opísaná odlišne. Preto autor záznamu o chybe musí prehľadávať minimálne názov chyby a jej popis.

4. V ďalšom kroku si vyberie projekt, ktorého sa chyba týka. V časti *Issue Type* zvolí autor ako typ hodnotu *Bug*.
5. Kliknutím na tlačidlo *Search* sa spustí vyhľadávanie. Napravo sa zobrazia výsledky hľadania. Ak hľadaná chyba v systéme neexistuje, autor pokračuje procesom *Vytvorenie záznamu o chybe*. Inak autor pokračuje procesom *Doplnenie informácií o chybe*.

13.3.8.2 Vytvorenie záznamu o chybe

Tento podproces pozostáva zo samotného evidovania chyby do systému JIRA. Tvorí ho v prevažnej miere vyplnenie formuláru o novej chybe. Je v záujme celého tímu, aby zadané informácie boli dostatočne detailné a zrozumiteľné pre rýchle odhalenie príčiny chyby.

Proces vytvorenia záznamu o chybe sa skladá z nasledujúcich krokov:

1. Autor záznamu o chybe sa prihlási do systému JIRA so svojim používateľským menom a heslom.
2. Zvolí si v menu záložku *Issues* a v rozbalenom menu položku *Create Issue*.
3. Zobrazí sa formulár pre zadanie informácií o novej chybe uvedený na Obr. 13:8 s ukázkou vyplnených polí.

Create Issue
 Configure Fields ▾

Project * SimTEAM ▾

Issue Type * Bug ▾

Summary *

Priority Major ▾

Due Date

Component/s Implementácia × ▾
Start typing to get a list of possible matches or press down to select.

Affects Version/s Druhý šprint × ▾
Start typing to get a list of possible matches or press down to select.

Fix Version/s Druhý šprint × ▾
Start typing to get a list of possible matches or press down to select.

Assignee Filip Pakan ▾

Environment

For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).

Description

Original Estimate (eg. 3w 4d 12h)
The original estimate of how much work is involved in resolving this issue.

Remaining Estimate (eg. 3w 4d 12h)
An estimate of how much work remains until this issue will be resolved.

Attachment log.txt

The maximum file upload size is 10,00 MB.

Labels ▾
Begin typing to find and create labels or press down to select a suggested label.

Create another

Obr. 13:8 Formulár pre pridanie novej chyby.

4. Autor vyplní polia podľa Tabuľka 13.8.

Tabuľka 13.8 Návod na vyplnenie polí pri evidovaní novej chyby.

Názov poľa	Vysvetlenie
Project	<ul style="list-style-type: none"> ▪ Názov projektu, ktorého sa chyba týka
Issue type	<ul style="list-style-type: none"> ▪ Typ zadávanej úlohy ▪ Pre softvérové chyby sa musí nastaviť <i>Bug</i>
Summary	<ul style="list-style-type: none"> ▪ Stručný ale výstižný opis chyby (max. 10 slov) ▪ Musí vyjadrovať hlavnú podstatu chyby (čo nefunguje)

	<ul style="list-style-type: none"> ▪ Slúži ako názov záznamu v systéme JIRA
Priority	<ul style="list-style-type: none"> ▪ Určuje prioritu chyby ▪ Prípustné hodnoty sú Trivial, Minor, Major, Critical a Blocker ▪ Stanoví sa na základe početnosti výskytu chyby a dopadu na bezproblémové používanie produktu ▪ Podrobnejšie informácie o určovaní priorít sú uvedené v <i>Metodike pre stanovenie priorít chýb</i>
Due date	<ul style="list-style-type: none"> ▪ Termín, do kedy musí byť chyba opravená
Component/s	<ul style="list-style-type: none"> ▪ Určuje, do akej súčasti projektu chyba patrí ▪ Možné hodnoty sú Analýza, Návrh, Implementácia, Testovanie, Dokumentácia
Affects Version/s	<ul style="list-style-type: none"> ▪ Určuje ktoré verzie produktu sú postihnuté
Fix Version/s	<ul style="list-style-type: none"> ▪ Určuje, ktorú verziu produktu treba opraviť
Assignee	<ul style="list-style-type: none"> ▪ Prideľuje osobu na vyriešenie tejto chyby ▪ Zadá sa meno člena tímu, ktorého nedôsledná práca spôsobila vznik tejto chyby
Environment	<ul style="list-style-type: none"> ▪ Popis prostredia, na ktorom sa chyba prejavuje ▪ Treba uviesť operačný systém, hardvérovú špecifikáciu, verzie používaných knižníc
Description	<ul style="list-style-type: none"> ▪ Musí sa uviesť sa čo najdetailnejší opis chyby (min. 30 slov) ▪ Treba sa vyjadriť k nasledovným bodom: <ul style="list-style-type: none"> ▫ Čo presne zlyhá, aká funkcionálna nefunguje ▫ Pri akých vstupoch a interakciách používateľa dôjde k chybe ▫ Čo systém zobrazí pri vzniku chyby ▫ Ako systém zareaguje na vzniknutú chybu
Original Estimate	<ul style="list-style-type: none"> ▪ Časový odhad trvania opravy chyby ▪ Odhad musí vychádzať z predošlých skúseností
Remaining Estimate	<ul style="list-style-type: none"> ▪ Pri evidovaní chyby do systému autor záznamu toto pole nevyplňa ▪ Toto pole vyplňa až pridelená osoba pri riešení tejto chyby
Attachment	<ul style="list-style-type: none"> ▪ Priložené súbory, ktoré môžu pomôcť zanalyzovať a opraviť chybu ▪ Napríklad výpis z logu alebo screenshot aplikácie v chybovom stave

13.3.8.3 Doplnenie informácií o chybe

Tento proces sa vykonáva v prípade, že nahlásená chyba už je evidovaná v systéme a zároveň sú k dispozícii nové poznatky o príčinách a dôsledkoch chyby. V takom prípade sa musia tieto užitočné informácie doplniť k existujúcej chybe do systému. Proces pozostáva z nasledujúcich krokov:

1. Člen tímu vyhledá chybu v systéme podľa postupu uvedeného v kapitole 13.3.8.1.
2. Klikne na nájdenú chybu a v pravom hornom rohu zvolí možnosť *Edit*.
3. Systém JIRA zobrazí rovnaký formulár ako je uvedený v kapitole 13.3.8.2. Člen tímu zadá nové informácie do formulára podľa tabuľky uvedenej v 4. kroku procesu *Vytvorenie záznamu o chybe* a formulár uloží.

13.4 Metodika údržby a štábnej kultúry

13.4.1 Úvod

Úlohou tejto metodiky je poskytnúť presné opisy procesov (obr. č. 1 a tabuľka č.1) a postupov, ktoré zahŕňajú manažment údržby softvérového projektu v prostredí NetBeans. Jej aplikovanie a vysvetlenie je prispôsobené konkrétnemu projektu, ktorým je Simulácia demonštrácie.

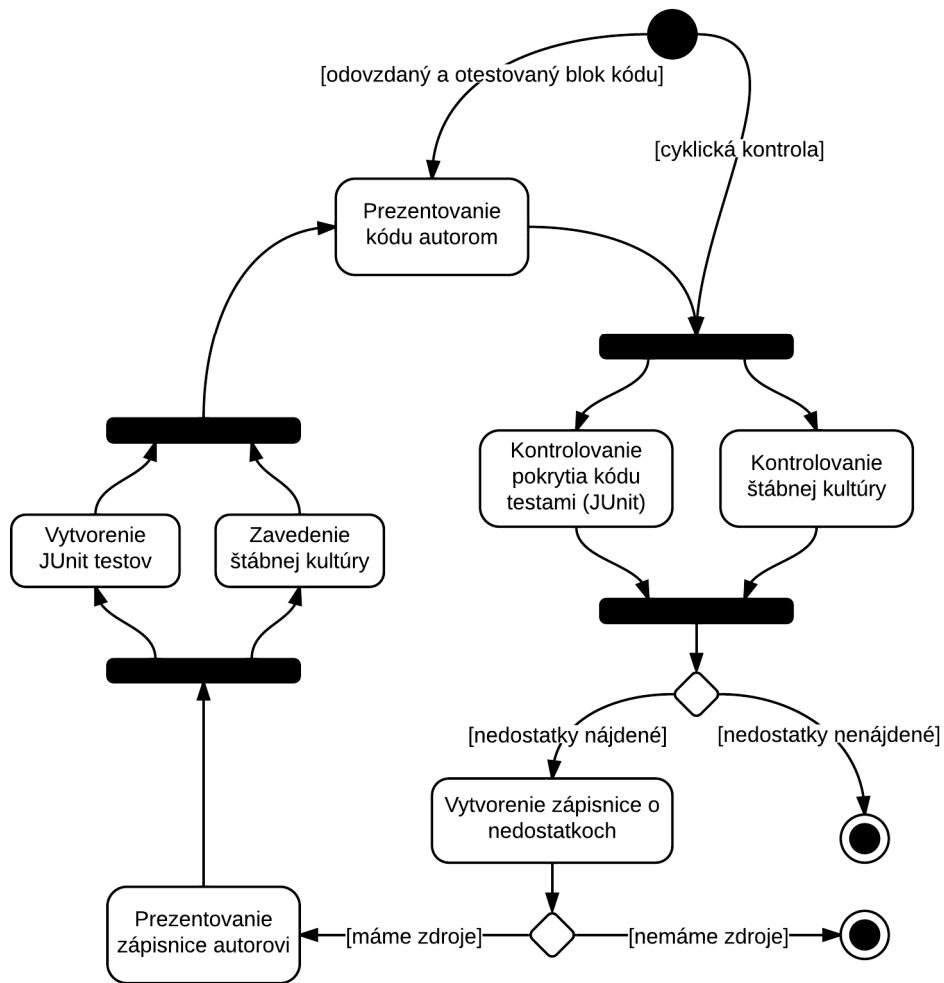
13.4.2 Použité pojmy

- **Netbeans** – integrované vývojové prostredie pre jazyk Java od Oracle.
- **Camel Case** – konvencia pomenovávaní, nepoužíva znak "_" a viacslovné názvy spája priamo, pričom každé slovo začína veľkým písmenom. Výnimkou je prvé (možnosť voľby).
- **Pascal Case** – konvencia pomenovávaní, ktorá predstavuje prísnejší variant konvencie Camel Case, t.j. aj prvé písmeno prvého slova názvu je kapitál.
- **JavaDoc** – nástroj na generovanie dokumentácie v jazyku Java, ktorý vyžaduje presné formátovanie komentárov, z ktorých je následne vygenerovaná dokumentácia vo formáte HTML.
- **Tag** – všeobecne značka (značkovacie jazyky HTML, XML). Používame ju v kontexte vyplnenia JavaDoc tagov.

13.4.3 Roly a zodpovednosti

Rola	Zodpovednosť
Autor bloku kódu	<ul style="list-style-type: none"> • Prezentácia bloku kódu členom tímu. • Odstránenie zistených nedostatkov na základe zápisnice (štábna kultúra a JUnit testy).
Manažér kvality	<ul style="list-style-type: none"> • Iniciuje cyklickú kontrolu. • Kontroluje odchýlky v kóde od štábnej kultúry. • Spisuje zápisnicu o nedostatkoch v oblasti štábnej kultúry. • Prezentuje a objasňuje nájdené nedostatky autorovi bloku kódu
Člen tímu zodpovedný za testovanie	<ul style="list-style-type: none"> • Explicitne vytvára JUnit testy. • Kontroluje kód na nedostatočne pokryté časti JUnit testami. • Spisuje zápisnicu o nedostatkoch v oblasti pokrytia kódu JUnit testami. • Prezentuje a objasňuje nájdené nedostatky autorovi bloku kódu.

13.4.4 Proces údržby projektu



Obr. 13:9 Diagram aktivít / procesov.

Tabuľka 13.9 Prehľad procesov metodiky.

Názov procesu	Kapitola
Prezentovanie kódu autorom	13.4.4.1
Kontrolovanie pokrytia kódu testami (JUnit)	13.4.4.2
Kontrolovanie štabnej kultúry	13.4.4.3
Vytvorenie zápisnice o nedostatkoch	13.4.4.4
Prezentovanie zápisnice autorovi	13.4.4.5
Vytvorenie JUnit testov	13.4.4.6
Zavedenie štabnej kultúry	13.4.5.1

13.4.4.1 Prezentácia kódu autorom

Vstup: Funkčná časť zdrojového kódu odobrená testovaním.

Výstup: Ucelená a stručná informácia o funkčnosti a konštrukciách použitých v skúmanom bloku kódu.

Zodpovedný: *programátor (autor kódu), všetci členovia tímu.*

Autor ucelenej funkcionality (metódy, triedy atď.) musí prezentovať svoju prácu heslovite na stretnutí, prípadne v textovej podobe (max. 1/2 A4), objasniť význam kľúčových premenných a konštrukcií použitých v danej časti. Odpovedá tiež na otázky ohľadne nejasností jeho prezentácie.

13.4.4.2 Kontrolovanie pokrytia kódu testami JUnit

Vstup: *Funkčná časť zdrojového kódu odobrená testovaním, ktorá bola posunutá manažérom kvality na ďalšiu kontrolu.*

Výstup: *Zápis o metódach resp. častiach zdrojového kódu, ktorý nie je pokrytý vôbec alebo nedostatočne testami JUnit.*

Zodpovedný: *člen tímu zodpovedný za testovanie*

Člen tímu zodpovedný za testovanie musí prejsť metódy, časti kódu zodpovedné za kalkulácie ako aj iné netriviálne procesy a identifikovať ich testovateľnosť. Ak v takýchto prípadoch nenájde vhodné resp. žiadne JUnit testy od autora bloku kódu, uvedie to do zápisu spolu s návrhom možných testov (rozsah 1 - 3 vety na každý nedostatok).

13.4.4.3 Kontrolovanie štábnej kultúry

Vstup: *Funkčná časť zdrojového kódu odobrená testovaním, ktorá bola posunutá manažérom kvality na ďalšiu kontrolu.*

Výstup: *Zápis o častiach zdrojového kódu, ktoré nevyhovujú štábnej kultúre a nedostatkom, ktoré obsahuje.*

Zodpovedný: *manažér kvality*

Manažér kvality prezrie zdrojový kód, prípadne využije výstup z metodiky 2910/2012 - Manažment prehliadok zdrojových kódov a zaznamená heslovite všetky odchýlky od štábnej kultúry, ktorú predpisuje táto metodika.

13.4.4.4 Vytvorenie zápisnice o nedostatkoch

Vstup: *Výstupy (zápisy) z procesov Kontrolovanie štábnej kultúry a Kontrolovanie pokrytia kódu JUnit testami.*

Výstup: *Finálna zápisnica popisujúca nedostatky ohľadne pokrytia kódu JUnit testami a nedostatky v dodržiavaní štábnej kultúry spolu s návrhmi na opravu (netriviálne prípady).*

Zodpovedný: *člen tímu zodpovedný za testovanie, manažér kvality*

Manažér kvality spolu s členom tímu zodpovedným za testovanie prejdú spolu zápisy z predošlých dvoch procesov a rozhodnú, ktoré zo zaznamenaných nedostatkov budú opravené a zanesené do finálnej zápisnice. K netriviálnym prípadom musí každý napísať aj pomôcku ako treba k oprave pristupovať.

13.4.4.5 Prezentovanie zápisnice autorovi

Vstup: Finálna zápisnica popisujúca nedostatky ohľadne pokrytia kódu JUnit testami a odchýlkami od štábnej kultúry spolu s návrhmi na ich odstránenie.

Výstup: Zrozumiteľné osvetlenie jednotlivých bodov zápisnice na stretnutí, prípadne textovo v maximálnom rozsahu 1 A4. Zodpovední musia odpovedať na všetky nejasnosti ohľadne navrhnutých spôsobov riešenia a musia byť presvedčení, že autor kódu porozumel požiadavkám na opravu.

Zodpovedný: člen tímu zodpovedný za testovanie, manažér kvality, autor kódu

Člen tímu zodpovedný za testovanie spolu s manažérom kvality musia v prehľadnej a zrozumiteľnej forme informovať o záveroch kontroly kódu jeho autora. Vysvetliť mu aké chyby urobil a taktiež musia zdôvodniť nutnosť nápravy spolu s objasnením spôsobov, ktoré sami navrhli. Musia tak učiniť najneskôr do 48 hodín od ukončenia kontroly a tento úkon vykonať ústne na stretnutí, prípadne v textovej forme, ktorá nepresiahne dĺžku 1 A4 textu. Na ukončenie tohto procesu si musia byť obaja manažéri istí, že autor kódu rozumie aké aktivity sú od neho požadované.

13.4.4.6 Vytvorenie JUnit testov

Vstup: Funkčná časť zdrojového kódu odobrená testovaním, finálna zápisnica vytvorená manažérom kvality a členom tímu zodpovedným za testovanie na základe analýzy kódu.

Výstup: Zdrojový kód obsahujúci implementované JUnit testy v požadovanom rozsahu zápisnice.

Zodpovedný: autor skúmaného bloku kódu

Autor skúmaného bloku kódu musí na základe zápisnice (sekcia JUnit testy) implementovať testovacie mechanizmy v rozsahu zápisnice (v prípade dodatočných pokynov presne podľa nich).

13.4.5 Podrobný opis krokov

13.4.5.1 Zavedenie štábnej kultúry

Vstup: Funkčná časť zdrojového kódu odobrená testovaním, ktorá bola posunutá manažérom kvality na ďalšiu kontrolu.

Výstup: Zdrojový kód naformátovaný podľa zadaných konvencií a pravidiel.

Zodpovedný: programátor, manažér kvality

Zdrojový kód je testerom a ostatnými členmi tímu odobrený, následne je autor kódu požiadaný manažérom kvality o zosúladienie s pravidlami popisujúcimi štábnu kultúru v tejto metodike.

13.4.5.2 Úvod k štábnej kultúre

Štábná kultúra je popísaná v manažmente kvality (kapitola 6).

13.5 Metodika prehliadky zdrojového kódu

13.5.1 Úvod

Účelom metodiky prehliadky zdrojového kódu je oboznámenie s presnými procesmi a postupmi vykonávanými v rámci prehliadky zdrojových kódov. Obsahuje kompletný priebeh procesu prehliadky zdrojových kódov, ktorý je podrobne znázornený na obrázku č. 1. Jednotlivé procesy majú presne definované roly a ich povinnosťami. Uvedené metodické pokyny boli prispôsobené pre použitie v projekte simulácia demonštrácie.

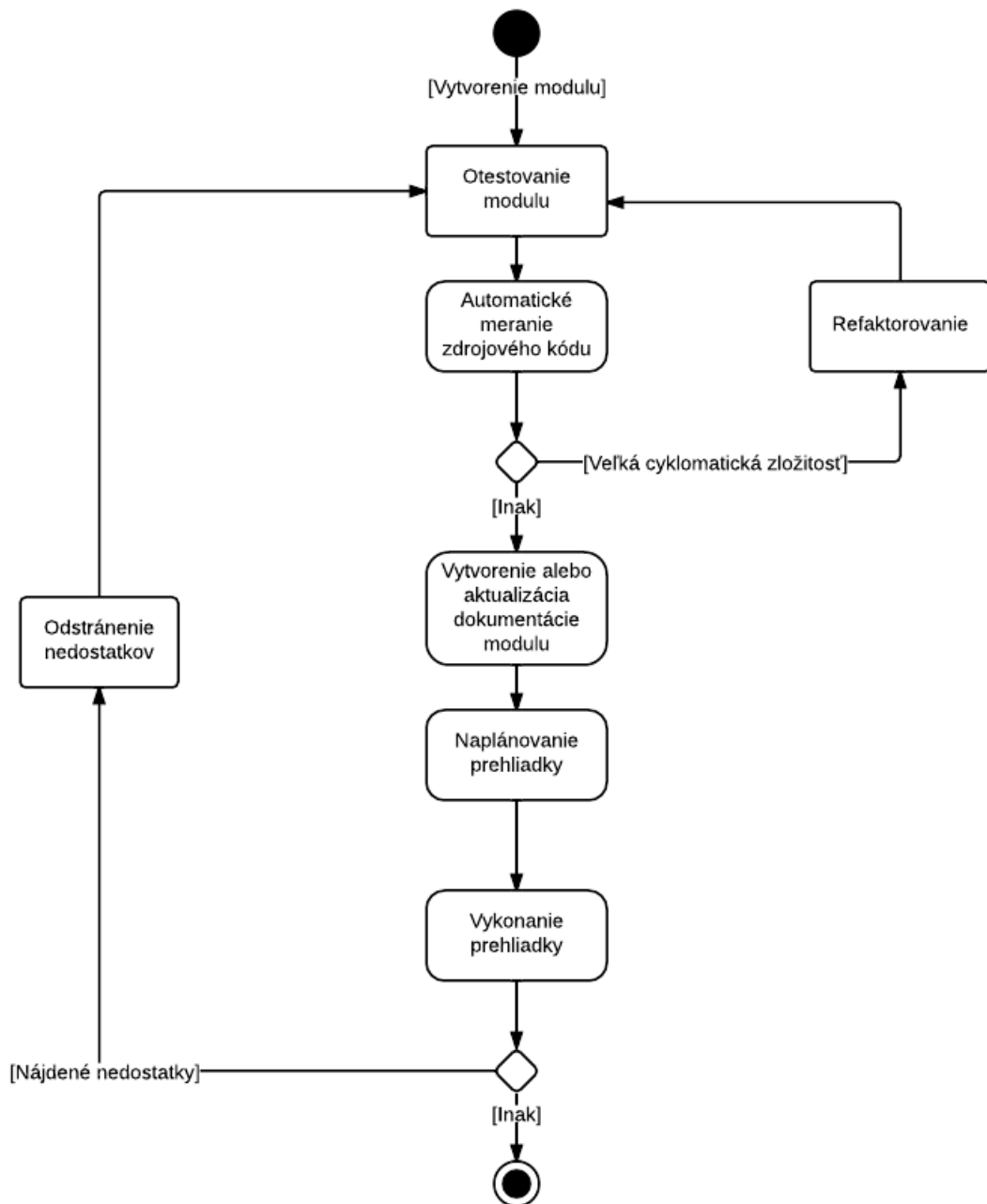
13.5.2 Použité pojmy

- Scrum master – osoba zodpovedná za odstraňovanie prekážok v tímoch vyvíjajúcich Scrumom
- Cyklomatická zložitosť – softvérová metrika na meranie zložitosti zdrojových kódov na základe množstva vetvení v programe
- Javadoc – systém na generovanie dokumentácie z komentárov jazyka Java
- Refaktorovanie – zmena štruktúry zdrojového kódu pri zachovaní funkčnosti
- CyVis – nástroj na meranie a vizualizáciu zložitosti zdrojových kódov

13.5.3 Roly a zodpovednosti

Rola	Zodpovednosť
Autor zdrojového kódu	<ul style="list-style-type: none"> • Otestovanie modulu • Meranie modulu • Dokumentáciu modulu • Odstránenie prípadných nedostatkov
Moderátor	<ul style="list-style-type: none"> • Vedenie a priebeh stretnutia prehliadky • Vytvorenie zápisnice o stretnutí
Člen tímu zodpovedný za posúdenie zdrojového kódu	<ul style="list-style-type: none"> • Kontrola štábnej kultúry • Kontrola limitu cyklomatickej zložitosti • Kontrola pokrytia testami • Kontrola dokumentácie • Kontrola komentárov
Scrum master	<ul style="list-style-type: none"> • Určenie termínu prehliadky • Určenie moderátora a členov na posúdenie kódu

13.5.4 Proces prehliadky zdrojových kódov



Obr. 13:10 Diagram procesu prehliadky zdrojového kódu.

Tabuľka 13.10 Prehľad procesov metodiky.

Proces	Kapitola metodiky
Otestovanie modulu	13.5.4.1
Automatické meranie zdrojového kódu	13.5.4.2
Refaktorovanie	13.5.4.3
Vytvorenie alebo aktualizácia dokumentácie modulu	13.5.4.4
Naplánovanie prehliadky	13.5.4.5
Vykonanie prehliadky	13.5.4.6
Odstránenie nedostatkov	13.5.4.7

13.5.4.1 Otestovanie modulu

Vstup: Zdrojový kód modulu, testy JUnit daného modulu

Výstup: Dokument s vyhodnotením testov

Zodpovedný: Autor zdrojového kódu

Autor zdrojového kódu vykoná všetky príslušné testy daného modulu. Výsledky vykonaných testov zdokumentuje.

13.5.4.2 Automatické meranie zdrojového kódu

Vstup: Zdrojový kód vytvoreného modulu

Výstup: Dokument

Zodpovedný: Autor zdrojového kódu

Po skompletizovaní zdrojového kódu modulu, vykoná autor tohto zdrojového kódu automatické meranie. Metodické pokyny na vykonanie automatického merania sú podrobnejšie opísané v časti 5. Podrobný opis automatického merania.

13.5.4.3 Refaktorovanie

Vstup: Zdrojový kód vytvoreného modulu

Výstup: Refaktorovaný zdrojový kód modulu

Zodpovedný: Autor zdrojového kódu

Proces refaktoringu je iniciovaný prekročením limitu stanovenej cyklomatickej zložitosti. Konkrétne pravidlá a postupy refaktoringu sú podrobne uvedené v metodike 2710/2012.

13.5.4.4 Vytvorenie alebo aktualizácia dokumentácie modulu

Vstup: Zdrojový kód vytvoreného modulu

Výstup: Aktuálna dokumentácia modulu

Zodpovedný: Autor zdrojového kódu

Dokumentácia modulu bude obsahovať zoznam tried usporiadaný podľa abecedy spolu s číslom revízie daného modulu z verziovacieho systému. Každá trieda bude obsahovať zoznam jej metód spolu s cyklomatickou zložitou. V prípade odstránenie nedostatkov po vykonaní prehliadky je potrebné dokumentáciu aktualizovať podľa uskutočnených zmien.

13.5.4.5 Naplánovanie prehliadky

Vstup: Požiadavka na naplánovanie prehliadky

Výstup: Naplánovaná prehliadka, určený moderátor a členovia na posúdenie kódu

Zodpovedný: Scrum master

Scrum master zvolí členov na posúdenie kódu v rámci tímu všetkých, ktorí budú daný modul následne používať. Moderátor bude zvolený podľa vlastného uváženia scrum mastera. Následne sa prehliadka vykoná pred najbližším tímovým stretnutím. Členovia, ktorí sa majú zúčastniť prehliadky zdrojového kódu, budú notifikovaní emailom.

Metodické pokyny na vykonanie naplánovania prehliadky sú podrobnejšie opísané v časti 13.5.6. Podrobný opis naplánovania prehliadky.

13.5.4.6 Vykonanie prehliadky

Vstup: Zdrojový kód vytvoreného modulu, dokumentácia modulu

Výstup: Vytvorený záznam o prehliadke

Zodpovedný: Moderátor

Prehliadka zdrojového kódu pozostáva z troch fáz:

1. Prezentácia a objasnenie zdrojového kódu prítomným členom
2. Kontrola zdrojového kódu
3. Vytvorenie záznamu o vykonanej prehliadke, ktorý bude obsahovať prípadné nedostatky

V prvej fáze autor zdrojového kódu vysvetlí prítomným členom základnú funkčnosť a použitie daného modulu. Stručne popíše jednotlivé triedy a metódy moduly a ich základné parametre. Po krátkej prezentácii by mali byť prítomní členovia schopní bez problémov používať daný modul v rámci projektu. Moderátor je zodpovedný za riadenie obsahu prehliadky a tvorbu dokumentácie záznamu prehliadky.

Fáza kontroly zdrojového kódu najprv skontroluje cyklomatickú zložitosť jednotlivých metód z dokumentácie modulu. Cyklomatická zložitosť metódy nesmie prekročiť hranicu 12. Táto hodnota vychádza z článku, kde je uvedená zložitosť metódy uvedená ako vysoká. V prípade, že nejaká metóda prekročí tento limit, zadokumentuje sa príslušný nedostatok a autor zdrojového kódu bude musieť danú metódu refaktorovať podľa metodiky 2710/2012. Následne sa pristúpi ku kontrole štábnej kultúry podľa metodiky 21245/2012. V prípade zistených nedostatkov budú tieto nedostatky zadokumentované. Nakoniec sa pristúpi ku kontrole komentárov Javadoc jednotlivých tried a metód podľa metodiky 21245/2012. Pri zistených odchýlkach sa tieto nedostatky zadokumentujú.

V tretej fáze sa finalizuje dokumentácia o prehliadke. V prípade zistenia prekročenia limitu cyklomatickej zložitosti autor zdrojového kódu musí zistené nedostatky odstrániť. V prípade menej závažných nedostatkov v prípade štábnej kultúry alebo komentárov sa jednotlivé komentáre podľa možností buď odstránia na mieste alebo sa hlasovaním členov zodpovedných za posúdenie zdrojového kódu určí, či je potrebné vykonať opravu zistených nedostatkov.

13.5.4.7 Odstránenie nedostatkov

Vstup: Zdrojový kód vytvoreného modulu, záznam o prehliadke

Výstup: Opravený zdrojový kód

Zodpovedný: Autor zdrojového kódu

Autor odstráni všetky zistené nedostatky, ktoré boli zistené počas prehliadky zdrojového kódu a boli zdokumentované v zázname o prehliadke.

13.5.5 Podrobný opis automatického merania

Automatické meranie bude uskutočnené v systéme CyVis. Systém CyVis je jednoduchý nástroj na meranie a vizualizáciu zložitosti zdrojových kódov jazyka Java. Systém CyVis je konfigurovateľný na limity cyklomatickej zložitosti. Tieto limity následne systém zohľadní pri vizualizácii zložitosti zdrojových kódov. Vizualizácia pomôže autorovi zdrojového kódu veľmi rýchlo a jednoducho odhaliť problematické metódy, ktoré nespĺňajú kritérium cyklomatickej zložitosti. Po správnom nastavení limit ich autor uvidí v systéme CyVis zobrazené červenou farbou. Zdokumentovať bude potrebné okrem cyklomatickej zložitosti jednotlivých metód aj počet riadkov zdrojového kódu jednotlivých metód.

13.5.5.1 Inštalácia a spustenie

1. Systém prevezmete na nasledujúcej adrese:
<http://cyvis.sourceforge.net/download.html>
2. Po prevzatí systému je potrebné vykonať jeho extrakciu na ľubovoľné miesto v súborovom systéme
3. Po otvorení adresára do ktorého bol v predchádzajúcom kroku systém extrahovaný, systém spustíte kliknutím na súbor *cyvis-0.9.jar*

13.5.5.2 Nastavenie

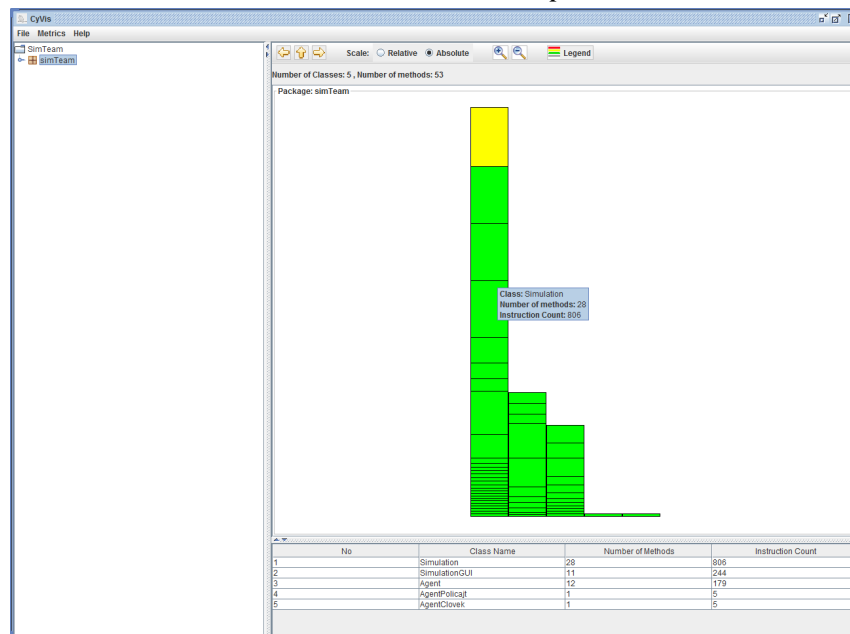
1. V hornom menu programu kliknite na nápis *Metrics*. Po kliknutí sa zobrazí menu týkajúce sa metrík.
2. V zobrazenom menu zvolíte *Preferences*. Po kliknutí sa zobrazí okno nastavení.
3. V okne nastavení v časti *Report Directory* zvolíte *Project Directory*.
4. Časť *Metrics* nastavíte na nasledovné hodnoty:

a. High Complexity	12
b. Moderate Complexity	6
c. Low Complexity	0
5. Kliknete na tlačidlo *Save*.

13.5.5.3 Meranie

1. V hornom menu programu kliknite na nápis *File*. Po kliknutí sa zobrazí menu so zoznamom základných operácií.
2. V zobrazenom menu zvolíte *New Project*.
3. V novom okne do kolónky *Project Name* napíšete názov projektu. Názov projektu sa bude zhodovať s názvom modulu, ktorý plánujete merať.
4. *Project Directory* umiestnite na ľubovoľné miesto v súborovom systéme, avšak adresár s projektom pomenujete rovnako ako názov modulu.
5. V časti *Files* kliknete na tlačidlo *Add*.
6. V otvorenom okne sa premiestnite do adresára so zdrojovými kódmi modulu, ktorý chcete merať. Jednotlivé zdrojové súbory s príponou *.class* označíte so stlačeným tlačidlom CTRL a kliknutím ľavým tlačidlom myši na príslušný súbor. Po označení všetkých zdrojových súborov modulu, kliknete na tlačidlo *Add*.
7. Po pridaní všetkých súborov modulu kliknite na tlačidlo *Finish*.

8. Systém CyVis následne automaticky odmeria a vizualizuje zložitosť daného modulu, ako je zobrazené na obrázku č. 2. Jednotlivé metódy sú farebne odlišené podľa cyklomatickej zložitosti. Zelené metódy majú malú cyklomatickú zložitosť, žlté metódy strednú zložitosť a červené metódy veľkú cyklomatickú zložitosť.
9. Ak systém CyVis zobrazí nejaké červené metódy, autor bude zdrojový kód refaktorovať za účelom dosiahnutia menšej cyklomatickej zložitosti, podľa metodiky 5610/2012. Po refaktorovaní vykoná znovu proces automatického merania.
10. V hornom menu kliknite na nápis *Metrics*.
11. V zobrazenom menu zvolíte *Generate TXT Report*.



Obr. 13:11 CyVis vizualizácia.

13.5.5.4 Archivácia výsledkov merania

1. Po vykonaní merania príslušný adresár projektu merania skomprimujete do súboru s názvom modulu a príponou *zip*. Komprimáciu je možné vykonať priamo v systéme Windows, po kliknutí pravým tlačidlom na adresár a zvolení *Send to, Compressed (zipped) folder*.
2. Uvedený komprimovaný súbor skopírujete na tímový *Google Drive*, do adresáru *merania*.

13.5.6 Podrobný opis naplánovania prehliadky

Pred vykonaním samotnej prehliadky je potrebné ju naplánovať. Proces naplánovania zahŕňa zaslanie samotnej požiadavky na prehliadku scrum masterovi, určenie moderátora a členov tímu zodpovedných za posúdenie kódu a určenie termínu.

13.5.6.1 Požiadavka

Po vykonaní automatického merania a dokumentácie modulu, pošle autor zdrojového kódu e-mailom požiadavku o vykonanie prehliadky scrum masterovi. Požiadavka bude

ako predmet správy obsahovať [Review request] a meno modulu. V prílohe e-mailu sa bude nachádzať dokumentáciu modulu určeného na prehliadku.

13.5.6.2 Naplánovanie

1. Po obdržaní požiadavky určí scrum master moderátora prehliadky. Moderátora zvolí podľa vlastného uváženia tak, aby sa jednotliví členovia v roli moderátora striedali.
2. Následne zvolí členov tímu, ktorí budú posudzovať daný zdrojový kód. Na posúdenie zdrojového kódu vyberie scrum master všetkých členov, ktorí by mohli daný modul používať. Jeden člen tímu môže na prehliadke zastávať aj rolu moderátora aj môže posudzovať zdrojový kód.
3. Ako termín prehliadky zdrojového kódu scrum master zvolí najbližšie tímové stretnutie. Prehliadka sa vykoná pred týmto stretnutím, pričom presný čas prehliadky zvolí scrum master podľa vlastného posúdenia komplexnosti daného modulu.
4. Po určení príslušných členov tímu, termínu a času vykonania prehliadky, odošle scrum master všetkým členom tímu e-mail s uvedenými informáciami o prehliadke. E-mail bude obsahovať ako prílohu dokumentáciu modulu.
5. Termín vykonania prehliadky zdrojového kódu následne scrum master zaznamená do zdieľaného *Google kalendára*.

13.6 Metodika manažmentu verzií

13.6.1 Úvod

Dokument upravuje riadenie verzií v spoločnosti CODE-IT-WELL a.s., ktorej pracovníci sú povinní riadiť sa ním pri príslušných pracovných činnostiach. Metodika sa vzťahuje na verziovací systém SVN (z angl. Subversion), ktorý je v našej spoločnosti používaný pri vývoji softvéru. Projektív manažéri môžu prostredníctvom tohto systému sledovať samotný proces vývoja, a takýto systém slúži okrem verziovania aj na zálohovanie dát.

13.6.2 Slovník pojmov a skratiek

SVN	Skratka od “subversion”, systém pre verziovanie softvéru.
IDE	I ntegrated D evelopment E nvironment- prostredie na vývoj softvéru.
Repozitár	Centrálne úložisko dát, ktoré sa nachádza na serveri.
Commit	Aktualizácia jedného alebo viacerých súborov nachádzajúcich sa v repozitári.
Revert	Opačný proces ako commit, jedná sa o krok späť, väčšinou nutné po chybnom príkaze commit.
Rollback	Proces podobný ako revert, avšak keď sa jedná o krok späť v rámci celej verzie repozitára, a nielen jedného súboru.
Log	Textové záznamy, ktoré uchovávajú každú zmenu vykonanú v SVN.

13.6.3 Roly a zodpovednosti

Ako vyplýva z tabuľky (Tabuľka 13.11), za celý SVN systém je zodpovedný manažér podpory vývoja, pričom vývoj sleduje projektový manažér. Manažér podpory vývoja má teda na starosti procesy od inštalácie, konfigurácie, až po spravovanie SVN. Ostatní členovia tímu pracujú s verziovacím systémom na základe postupov uvedených v tejto metodike, a používajú SVN (kap. 13.6.5) hlavne pri vývoji, odovzdávajú nové verzie zdrojových kódov, prípadne môžu sami vykonať operáciu revert(kap. 13.6.5.4), ak je to potrebné. Dokument opisuje dve časti: všeobecný opis procesov a používania verziovacieho systému, a konkrétny opis činností pri používaní SVN.

Tabuľka 13.11 Roly a zodpovednosti.

Rola	Zodpovednosť
Projektový manažér	Sledovanie vývoja softvéru
Manažér podpory vývoja	Inštalácia a spravovanie SVN
Vývojári	Úpravy zdrojového kódu a následné zdieľanie pomocou centrálného repozitára v SVN

13.6.4 Manažment verzií

Identifikované procesy sa vykonávajú v rámci celého cyklu softvérového vývoja. Pri každom procese je uvedené, čo je vstupom do procesu, čo má byť výstupom procesu, a taktiež kto je zodpovedný za daný proces.

13.6.4.1 Inštalácia SVN

Vstup: Server s nainštalovaným operačným systémom (Linux)

Výstup: SVN je nainštalované

Zodpovedný: Manažér podpory vývoja

Proces inštalácie SVN na server je nevyhnutnou podmienkou pre jeho používanie, keďže SVN bude poskytovať centrálny repozitár pre všetkých vývojárov, ktorý sa k nemu budú pripájať. Inštaláciu je potrebné vykonať ako prvý krok, t.j. pred začiatkom samotného vývoja softvéru. Ak nie je splnená vstupná podmienka- nie je nainštalovaný a správne nakonfigurovaný samotný server, je potrebné postupovať podľa metodiky č. 3/G-2011 o inštalácii a konfigurácii servera.

13.6.4.2 Vytvorenie prístupov pre používateľov

Vstup: Proces 13.6.4.1

Výstup: Všetci používatelia majú vytvorené kontá (meno+heslo)

Zodpovedný: Manažér podpory vývoja

Proces vytvorenia prístupov je možný až po inštalácii SVN, a je taktiež nevyhnutný. Systém SVN potom v logoch zaznamenáva nielen čo sa vykonalo, ale na základe prístupov zaznamenáva aj kto danú zmenu vykonal, čo je vhodné pri monitorovaní projektu.

Manažér podpory vývoja vytvorí všetkým používateľom kontá. Následne dodá všetkým používateľom ich prístupové údaje a vyzve ich k zmene hesla. Zmena hesla je povinnou súčasťou tohto procesu.

13.6.4.3 Spravovanie SVN

Vstup: Proces 13.6.4.1 a 13.6.4.2

Výstup: vykonanie potrebných zmien (13.6.4.3.1, 13.6.4.3.2, 13.6.4.3.3)

Zodpovedný: Manažér podpory vývoja

Po správnej inštalácii je dôležitý proces spravovania SVN. Dohľad nad celým verziovacím systémom má manažér podpory vývoja. Spravovanie predstavuje viaceré podprocesy:

13.6.4.3.1 Konfigurácia repozitára

Konfigurácia predstavuje vytvorenie štruktúry adresárov v SVN: branch, tag, trunk. Po nakonfigurovaní musí byť možné verziovací systém začať používať.

13.6.4.3.2 Záloha repozitára

Záloha repozitára predstavuje uchovanie aktuálnej verzie repozitára, pričom ak vývoj softvéru pokračuje, ja potrebné vytvorenie novej vetvy(branch) pre vývoj softvéru.

13.6.4.3.3 Odstránenie repozitára

Ak bol výsledný softvér dodaný zákazníkovi, a nie je potreba uchovávať verzie softvéru, tak podľa Autorského zákona č. 383/1997 Z. z. v znení novely č. 234/2000 Z. z. je nutné odstrániť celý repozitár aj so zálohami na serveri ako aj všetky jeho lokálne kópie.

13.6.5 Používanie SVN v prostredí NETBEANS

Metodika v nasledujúcej časti usmerňuje používanie v konkrétnom IDE prostredí, ktoré v našej spoločnosti používame na vývoj softvéru, a tým je NETBEANS. Nasledovné procesy vykonávajú všetci členovia tímu, ktorí sa priamo podieľajú na vývoji upravovaním, dopĺňaním a kontrolou zdrojových kódov.

13.6.5.1 Checkout

Vstup: Správne nakonfigurované SVN ako aj pripojenie k nemu

Výstup: Vytvorenie lokálnej kópie aktuálneho repozitára z SVN

Zodpovedný: Každý člen vývojového tímu

Checkout je potrebné vykonať ak ešte používateľ nemá stiahnutú lokálnu kópiu projektu z SVN repozitára. Checkout je potrebné vykonať, pokiaľ ešte nemáme vytvorenú lokálnu kópiu repozitára na disku. Checkout sa vykonáva v prostredí Netbeans podľa nasledovného postupu:

- 1.) Spustenie prostredia Netbeans.
- 2.) Kliknutie na Team > Subversion > Checkout.
- 3.) Repository URL - zadať presnú adresu, ktorá sa týka **aktuálneho projektu**.
- 4.) Do políček "User" a "Password" je potrebné napísať pridelené prihlasovacie meno a heslo, ktoré boli vytvorené a pridelené v procese 13.6.4.2.
- 5.) Kliknutie na tlačidlo Next.
- 6.) V nasledovnom okne je potrebné upraviť **iba** "Local folder" - adresár do ktorého sa nakopíruje repozitár.
- 7.) Kliknutie na tlačidlo Finish spustí sťahovanie.

- 8.) Po stiahnutí sa Netbeans opýta či chce používateľ otvoriť projekt– zvoliť možnosť áno.

13.6.5.2 Update

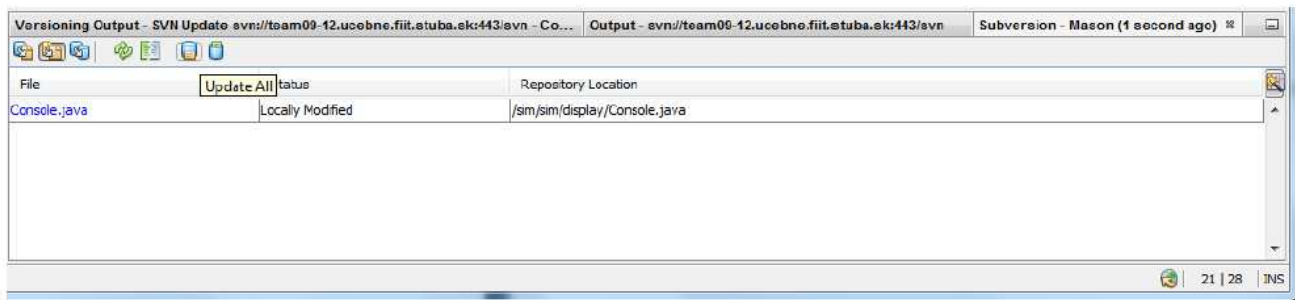
Vstup: Proces 13.6.5.1

Výstup: Aktualizácia lokálneho repozitára

Zodpovedný: Každý člen vývojového tímu

Update súborov aktualizuje lokálnu kópiu projektu podľa najnovšej verzie repozitára na serveri. Je nutné ho vykonať pred začatím práce na projekte aby sme mali aktuálne údaje, a taktiež pred vykonaním príkazu commit. Update vykonané zmeny v lokálnej kópii zachováva a aktualizuje len časti kódu ktoré neboli modifikované. V prípade, že do SVN repozitára bola uložená novšia verzia presne tej istej časti kódu s ktorou ste pracovali, je následne nutné vykonať proces riešenie kolízií- *resolve conflicts* (kap. 5.4). Update je možné vykonať nasledovne:

- 1.) Označiť príslušný projekt kliknutím
- 2.) Zvoliť v hornom menu Team > Show changes
- 3.) V dolnej časti prostredia sa zobrazí okno v ktorom je potrebné zvoliť “Update all”, ako je uvedené na Obr. 13:12.



Obr. 13:12 Update.

13.6.5.3 Commit

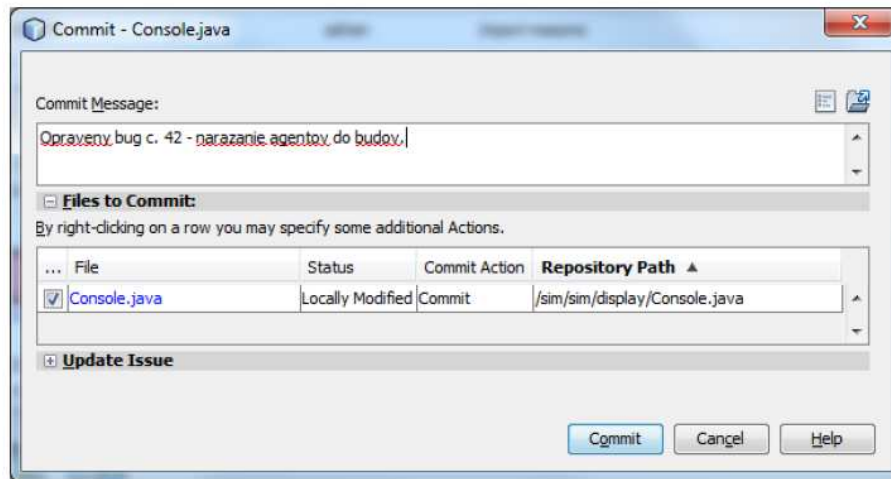
Vstup: Proces 13.6.5.1

Výstup: Aktualizácia lokálneho repozitára

Zodpovedný: Každý člen vývojového tímu

Príkaz commit uloží zmeny vykonané v lokálnej kópii do SVN repozitára na serveri. Je ho teda nutné použiť po dokončení a aj otestovaní nejakej ucelenej funkcionality. Commit je vykonávaný podľa nasledovného postupu:

- 1.) Zvoliť v hornom menu Team > Show changes
- 2.) V dolnej časti prostredia sa zobrazia všetky modifikované súbory
- 3.) Zvoliť “Commit All”.
- 4.) V otvorenom okne do časti Commit Message napísať správu ktorá bude obsahovať popis vykonaných úprav v rozsahu max. 12 slov (viď Obr. 13:13), pričom musí byť z popisu evidentné, **aká zmena bola vykonaná v zdrojovom kóde**.
- 5.) V časti Files to Commit zvoliť požadované súbory na commit.
- 6.) Kliknutie na tlačidlo Commit.



Obr. 13:13 Commit.

13.6.5.4 Riešenie kolízií

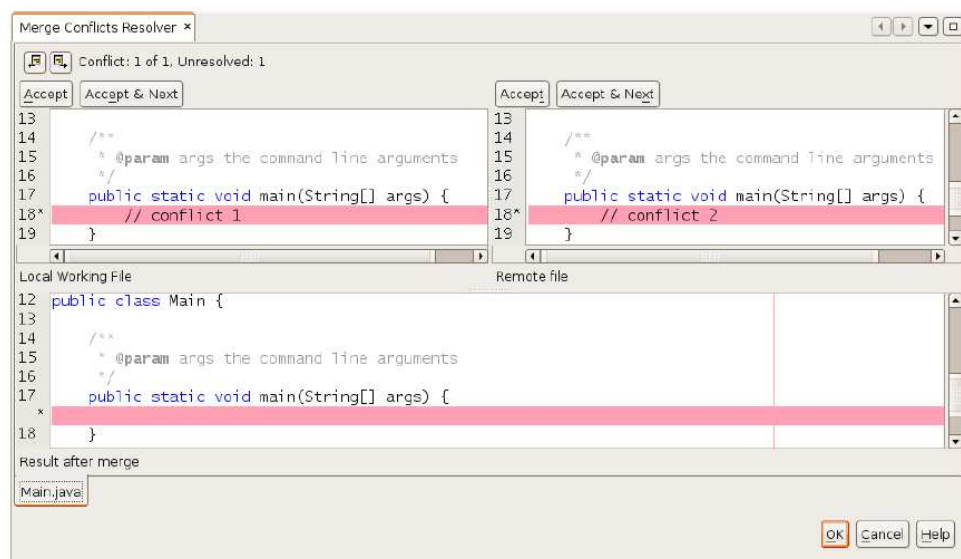
Vstup: Pri práci s SVN došlo ku kolízii

Výstup: Vyriešenie kolízie

Zodpovedný: Každý člen tímu podieľajúci sa na vývoji

Riešenie konfliktov je potrebné vykonať ak sa počas príkazov update alebo commit vyskytnú konflikty, ktoré nie je možné automaticky vyriešiť. Postup riešenia pri vyskytnutí sa konfliktu je nasledovný:

- 1.) Kliknutie pravým tlačidlom myši na príslušný súbor.
- 2.) Zvoliť Subversion > Resolve conflicts.
- 3.) Otvorí sa okno (viď Obr. 13:14) na vyriešenie konfliktov.
- 4.) Ak je problém možné vyriešiť priamo, kliknúť podľa možnosti na Accept (vľavo alebo vpravo), inak kontaktovať manažéra podpory vývoja (podľa metodiky Komunikácia tímu)



Obr. 13:14 Riešenie kolízií.

14 Preberacie protokoly

Preberací protokol

Tímový projekt 2012/2013

Tím č. 9 - SimTeam

Predmet odovzdávania:

Dokumentácia riadenia

- finálna verzia po dvoch semestroch
- rozsah: 131 strán

Projektová dokumentácia

- finálna verzia po dvoch semestroch
- rozsah: 126 strán

Vedúci tímového projektu: Ing. Peter Lacko, PhD.

Podpisom potvrdzuje prevzatie vyššie uvedených častí dokumentácie

V Bratislave

.....

Dátum

.....

Podpis