

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

Robocup 3D

The A Team

Dokumentácia k riadeniu

Vedúci tímu:

Ing. Marián Lekavý, PhD.

Členovia tímu:

Bc. Anton Balucha

Bc. Peter Borga

Bc. Tomáš Florek

Bc. Adam Pagáč

Bc. Eduard Pribula

Bc. Maros Unčík

Bc. Miroslav Vasil'

Kontakt: fiit-tim-04@googlegroups.com

Obsah

1	Úvod	1-1
2	Ponuka	2-1
2.1	Zloženie tímu	2-1
2.2	Motivácie	2-3
2.3	Koncepcie riešenia	2-5
2.4	Priorita tém	2-8
2.5	Rozvrhy členov tímu	2-9
3	Plán projektu	3-1
3.1	Termíny v zimnom semestri	3-1
3.2	Dlhodobý plán v zimnom semestri	3-2
3.3	Šprint č. 01	3-2
3.4	Šprint č. 02	3-3
4	Úlohy členov tímu	4-1
4.1	Úlohy jednotlivých členov tímu	4-1
4.2	Autorstvo jednotlivých častí dokumentácie	4-2
4.2.1	Dokumentácia k inžinierskemu dielu	4-2
4.2.2	Dokumentácia k riadeniu	4-5
5	Záznamy zo stretnutí	5-1
5.1	Zápisnica č. 01	5-1
5.2	Zápisnica č. 02	5-3
5.3	Zápisnica č. 03	5-6
5.4	Zápisnica č. 04	5-9
5.5	Zápisnica č. 05	5-14

6	Metodiky	6-1
6.1	Scrum	6-1
6.1.1	Role v metodike Scrum.....	6-1
6.1.2	Stretnutia počas vývoja softvéru.....	6-2
6.1.3	Proces metodiky Scrum.....	6-2
6.2	Štábna kultúra.....	6-3
6.2.1	Konvencia písania dokumentácia.....	6-3
6.2.2	Konvencia písania zdrojových kódov	6-6
6.3	Podporné prostriedky	6-13
7	Manažment projektu.....	7-1
7.1	Manažment verzií	7-1
7.2	Manažment dokumentácie	7-7
7.2.1	Dokumentácia k inžinierskemu dielu	7-7
7.2.2	Dokumentácia k riadeniu	7-8
7.2.3	Zápisnice.....	7-8
7.3	Manažment úloh.....	7-8
7.4	Manažment testovania.....	7-12
7.4.1	Tvorba testovacích tried a metód	7-12
7.4.2	Vytvorenie testu	7-14
7.4.3	Spustenie testu.....	7-16
7.5	Manažment zmien	7-18
8	Preberacie protokoly.....	8-1

1 Úvod

Tento dokument je vytvorený pre účel dokumentácie k riadeniu projektu Robocup 3D v rámci predmetu Tímový projekt v akademickom roku 2010/2011.

Dokument je štruktúrovaný po kapitolách a prehľad kapitol je nasledujúci. Druhá kapitola obsahuje ponuku vypracovanú za účelom pridelenia témy tímového projektu. V nasledujúcej tretej kapitole sú opísané dôležité termíny počas semestra, dlhodobé plány a plány jednotlivých šprintov. Kapitola štyri opisuje úlohy členov tímu a prácu na jednotlivých častiach dokumentácie k inžinierskemu dielu a dokumentácie k riadeniu. Zápisnice z jednotlivých stretnutí tímu sú zhrnuté v kapitole päť. Metodiky ako je napríklad štábna kultúra sú opísané v kapitole šesť. Manažment projektu a s ním súvisiace procesy sú opísané v siedmej kapitole. V poslednej ôsmej kapitole sú preberacie protokoly z jednotlivých kontrolných bodov.

2 Ponuka

Táto kapitola obsahuje ponuku k získaniu jednotlivých tímových projektov. Ponuka obsahuje prestavenie všetkých členov tímu The A Team, motivácie a koncepcie riešenia k jednotlivým tímovým projektom. Doplnená je o prioritu tém tímových projektov a rozvrhy všetkých členov tímu.

2.1 Zloženie tímu

Bc. Anton Balucha

Bakalárske štúdium som absolvoval na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Bakalársku prácu „Podpora získavania rodinných a sociálnych vzťahov“ som vypracoval pod vedením prof. Ing. Márie Bielikovej PhD., kde sme navrhli algoritmy pre vytváranie vzťahov a následne ich experimentálne overili.

Medzi moje programátorské znalosti patrí hlavne programovací jazyk Java, ktorý využívam v práci. Ďalej používam hlavne jazyky, ktoré súvisia s webom - HTML, XML, JavaScript. Databázové systémy používam MySQL a PostgreSQL.

Momentálne pracujem vo firme zaoberajúcej sa elektronickým vzdelávaním. Aktívne som sa zúčastnil vývoja troch informačných systémov. Toto moje pracovné zaradenie ďalej ovplyvnilo aj výber študijného programu Informačné systémy.

Výstupom tímového projektu by som chcel spraviť niečo, čo sa bude aktívne používať nebude to len zadanie pre zadanie samotné.

Bc. Peter Borga

Bakalárske štúdium som ukončil na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave obhájením práce s názvom „Automatizované generovanie metrík a dokumentácie zo zdrojových kódov“.

Počas štúdia som získal skúsenosti s programovacími jazykmi C, Java, SQL a Lua. Momentálne pokračujem v štúdiu na rovnakej fakulte v odbore Informačné systémy. Od tímového projektu očakávam prehĺbenie nadobudnutých vedomostí a ich prejavenie sa vo výslednom produkte.

Bc. Tomáš Florek

Bakalárske štúdium som absolvoval na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave a ukončil som ho bakalárskou prácou s názvom

„Odporúčanie vo veľkých grafoch“. Momentálne pokračujem v štúdiu na tejto fakulte a popri štúdiu pracujem na pozícii skriptéra.

Počas štúdia, v práci a voľnom čase som sa venoval hlavne týmto programovacím jazykom C, Java a Lua. Tiež som sa v súkromí venoval tvorbe web stránok, kde som pracoval hlavne s HTML, CSS a Flash. Taktiež mám základné skúsenosti s databázami hlavne MySQL a SQL. Skúsenosti mám taktiež s XML, ktoré som nadobudol počas práce na rôznych projektoch.

Bc. Adam Pagáč

Bakalárske štúdium na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave som ukončil prácou „Virtuálny album obrázkov“, ktorej výsledok bola standalone aplikácia implementovaná v jazyku Java. Už dva roky sa pohybujem v oblasti tvorby webových aplikácií. Mám skúsenosti prakticky s každou pozíciou pri tvorbe webového projektu, od programátora HTML (xHTML/CSS, Javascript), cez PHP vývojára (PHP/MySQL), až po projektového manažéra (3-5 členného tímu). Pracujem aj v oblasti predaja a marketingu webových služieb.

Bc. Eduard Pribula

Bakalárske štúdium som ukončil na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave obhájením bakalárskej práce „Robocup - vyššie schopnosti hráčov“ pod vedením Ing. Mariána Lekavého, PhD.

Medzi moje programátorské zručnosti patrí Java a C# na pokročilej úrovni, práca s databázami konkrétne PostgreSQL a MySQL. Taktiež má skúsenosti s HTML a XML.

Momentálne pracujem vo firme, zaoberajúcou sa dopravným dispečingom na pozícií zákazníckej podpory. Náplňou práce je dohľad nad databázou, testovanie novej funkcionality systému, kontakt so zákazníkom, vybavovanie požiadaviek a oprava chýb. Okrem toho sa zaoberám aj vývojom softvéru pre firmu, ktorý je implementovaný v jazyku C#. V minulosti som krátkodobo pracoval na pozícií testera softvéru pre firmu Orange.

Bc. Maros Unčík

Bakalárske štúdium som absolvoval na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave v odbore Informatika. Témou mojej bakalárskej práce bolo „Kolaboratívne poznámkovanie textu vo výučbovom systéme“ pod vedením prof. Ing. Marií Bielikovej, PhD., kde som zúžitkoval svoje doterajšie vedomosti z tvorby webových systémov a rozšíril ich o poznatky s programovacím jazykom Ruby. Zároveň som tak získal skúsenosti s prácou v tíme, keďže v rámci spomenutej bakalárskej práce som úzko spolupracoval s doktorandmi a diplomantmi.

V škole nadobudnuté poznatky z programovacieho jazyku Java, MySQL a z operačných systémov som v minulosti aj prakticky využil v práci na pozícii Java Developera na niekoľkých projektoch. Najväčším takýmto projektom bolo vytvorenie podpornej aplikácie pre magistrát mesta. Tu som nadobudol ďalšie skúsenosti so spravovaním a triedením dát, spravovaním hierarchie užívateľov a ich oprávnení. Okrem toho mám skúsenosti so značkovacími jazykmi HTML, XML, rovnako aj s CSS a JavaScriptom. V neposlednom rade na dobrej úrovni poznám grafické programy Corel PhotoPaint a Corel Draw. Verím, že môj záujem o túto oblasť a ochota jej venovať svoj voľný čas, je presne to, čo nám pomôže zrealizovať úspešný projekt.

Bc. Miroslav Vasiľ

Bakalárske štúdium som absolvoval na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave a úspešne ukončil štúdium obhájením bakalárskej práce „Vizualizácia tokov a zmien štruktúry grafov v čase“.

Medzi moje programátorské zručnosti patrí C# a Java na mierne pokročilej úrovni. Ďalej medzi moje zručnosti patrí SQL a práca sa databázami, konkrétne Oracle, a webové technológie HTML, XML a Java Script.

V minulosti som pracoval pre významné slovenské banky ohľadom prechodu na menu euro a bankové prostredie som nezmenil a v súčasnosti pracujem na projekte interného systému pre významnú slovenskú banku. Náplň práce má analyticko-testovacie zameranie a práca s databázami.

2.2 Motivácie

Vyhľadávanie a prístupnenie citácií

Práca s citáciami je neoddeliteľnou súčasťou akademického prostredia a študenti sú od začiatku štúdia vedený k používaniu citácií pri písaní každej odbornej práce. Práve správa citácií je v akademickej sfére jednou z kritických záležitostí a je potrebné, aby každá akademická autorita dokázala k svojim citáciám pristupovať. Vzhľadom k aktuálnemu tempu rastu počtu odbornej literatúry u nás a vo svete je nutné tento problém riešiť v rámci informačných systémov. Hoci systém, ktorý do určitej miery podporuje správu citácií na našej fakulte už existuje, je potrebné, aby pre túto oblasť existoval systém, ktorý je spoľahlivý, používateľsky prívetivý a ľahko použiteľný. Je pre nás preto výzva zapracovať požiadavky, ktoré sú požadované od nového systému a odstrániť nedostatky, ktoré zistíme pri dôkladnej analýze súčasného stavu.

Z povahy tohto projektu usudzujeme, že hlavné úsilie na projekte bude prebiehať nad databázami. Preto by sme radi v tomto projekte využili naše skúsenosti z oblasti databázových systémov a zároveň prácou na projekte naše vedomosti o nich posunuli

na vyššiu úroveň. Taktiež je pre nás veľmi motivujúce, že v tomto projekte vytvoríme vzájomnými silami niečo, čo sa bude aktívne používať a pomôže v rozvoji našej fakulty. Nebude to niečo, čo sa vytvorí a viac sa to nepoužije.

Každý člen nášho tímu, tímu číslo 4, vyniká v niektorej oblasti. Máme nielen teoretické vedomosti zo školy, ale v mnohých prípadoch praxou preverené skúsenosti, ktoré chceme v tímovom projekte uplatniť a ďalej prehĺbovať. Veríme, že náš tím má primerané rozdelenie vedomostí pre úspešné vyriešenie projektu a jeho úspech v TP Cupe. Z tohto dôvodu si myslíme, že táto téma by bola veľmi vhodná pre náš tímový projekt a je pre náš tím prioritná.

Správa študentských projektov na fakulte

Počas rokov strávených na našej fakulte sme mali niekoľko krát možnosť využiť systém YonBan. Tento informačný systém pre správu študentských projektov bol vytvorený pred ôsmimi rokmi, v dobe keď ešte neexistoval akademický informačný systém a bolo bežné vytvárať autonómne informačné systémy pre zabezpečenie nevyhnutných aktivít na našej fakulte. YonBan po väčšinu doby svojej prevádzky slúžil svojmu účelu dobre, no v súčasnosti je podľa nášho názoru jeho využitie v mnohých prípadoch nedostatočné. Preto sme už na začiatku identifikovali niekoľko vážnych nedostatkov. Tie pramenia najmä v autonómii tohto systému a chýbajúceho prepojenia s Akademickým informačným systémom. Naše doterajšie skúsenosti s odovzdávaním zadaní to len potvrdzujú a tento nevyhovujúci stav je zjavný hneď od začiatku použitia systému. Príkladom toho je, že študenti si musia vytvoriť nové konto, ktoré je použité len na účely v systéme YonBan, rovnako aj duplicitné odovzdávanie samotných študentských projektov do systému YonBan a Akademického informačného systému pokladáme za veľmi nešťastné riešenie.

Preto sme sa rozhodli využiť možnosť zmeniť súčasný nežiaduci stav a navrhnúť lepšie riešenie, ktoré by dokázalo zefektívniť, zrýchliť a sprehľadniť vykonávanie činností súvisiacich s odovzdávaním študentských projektov na našej fakulte. Tým nielen uľahčíme prácu študentom, ale najmä uľahčíme prácu pedagogickým pracovníkom fakulty. Máme záujem, aby výsledok nášho spoločného úsilia bol nielen používateľsky prívetivý, ale aj ľahko použiteľný. Chceme tak vytvoriť informačný systém, ktorý by bol reálne využiteľný a používal sa na fakulte minimálne rovnako dlhý čas ako súčasný systém YonBan.

V neposlednom rade by sme chceli v rámci projektu prehĺbiť naše profesionálne vedomosti. Všetci členovia tímu majú nielen školské skúsenosti z jednotlivých etáp vývoja, ale aj pracovné skúsenosti, ktoré môžu zúročiť pri práci na projekte. Myslíme si, že v našom tíme je primerané rozdelenie skúsenosti z rôznych oblastí vývoja softvéru a preto spĺňame všetky predpoklady na to, aby sme mohli vytvoriť softvérový výrobok v nadštandardnej kvalite a v stanovenom čase.

Prispôsobiteľný Widget

Informačný priestor, akým je web, obsahuje nespočetné množstvo informácií. Hoci je prístup k týmto informáciám voľný, bez veľkého rozmýšľania vieme povedať, že bez vhodnej kategorizácie a agregácie informácií sa k zaujímavým informáciám dostaneme len veľmi namáhavo. Práve riešenie takéhoto zaujímavého problému je pre nás motiváciou v pokuse o získanie tejto témy. Atraktivnosť a aktuálnosť problému je pre nás však len jedna z motivácií. Pri zvolení tejto témy nás motivuje samotné prostredie vypracovanie projektu - web, s ktorým sa väčšina členov nášho tímu chce zaoberať v budúcnosti, a s ktorým má aj najväčšie skúsenosti.

Z povahy projektu predpokladáme, že výsledok našej práce sa neuplatní len na akademickej pôde, ale pôjde o reálny softvérový produkt, ktorý budú využívať ľudia v praxi. Táto skutočnosť ako aj použitie moderných technológií v projekte ovplyvnila náš výber tejto témy ako jednej z tém, o ktoré sa uchádzame.

Na základe spomenutého máme aktívny záujem o danú tému a problematiku. Taktiež dúfame, že prácou na tomto projekte rozšírime naše vedomosti, znalosti a skúsenosti, nadobudnuté nielen počas štúdia na našej fakulte, ale aj prácou popri štúdiu. Veríme, že kombinácia našich schopností a odborných znalostí je dostatočná k vytvoreniu optimálneho riešenia.

2.3 Konceptie riešenia

Vyhľadávanie a sprístupnenie citácií

Vyhľadávanie a sprístupnenie citácií je problém, ktorý sa dá riešiť niekoľkými spôsobmi. V prvej analýze zadania sme vyhodnotili, že problém sa dá rozdeliť na niekoľko jednoduchších podproblémov. Následnou kompozíciou spomínaných podproblémov získame kompletne riešenie. Tieto podproblémy zahŕňajú:

- získavanie existujúcich zdrojov
- analýza získaných zdrojov a získanie záznamov
- mapovanie získaných záznamov
- prepojenie s akademickým informačným systémom
- sprístupnenie citácií pomocou rozhrania

Základom pre realizáciu takéhoto systému bude už existujúci portál pre publikačnú činnosť, ktorý bol vypracovaný v rámci minuloročného tímového projektu. Za ďalšie vhodné systémy, ktoré chceme analyzovať a získať tak požiadavky na vytváraný softvérový systém,

považujeme iné knižnično-informačné systémy pre správu citácií, akými sú napríklad systém Advanced Rapid Library, či LibQUAL+.

Navrhované riešenie

Je množstvo technológií, ktoré je možné použiť pri riešení tohto projektu. Vzhľadom na naše skúsenosti sme sa zhodli na tvorbu aplikácií využiť platformu Java a nástroje, ktoré nám tento flexibilný jazyk pre vývoj softvérových systémov ponúka. Navrhujeme však aj iné riešenia, postavené na technológiách ExtJS a Caché, ktoré nám pri konzultácií témy boli odporučené.

Riešenie postavené na technológiách Java

Java, ako nástroj pre tvorbu softvérových systémov, poskytuje bohaté možnosti ich tvorby, ktoré sa odrážajú najmä v špeciálnych frameworkoch pre zabezpečenie backendu a frontendu. Naše riešenie postavené na technológiách Java bude reprezentované vytvorením portálu, ktorý bude prístupný prostredníctvom webu a bude sprístupňovať jednotlivé funkčné časti riešeného systému. Aplikáčna logika bude realizovaná na webovom serveri, kde pomocou platformy Java vieme takéto riešenie prepojiť takmer s ľubovoľným úložiskom dát, akými sú napríklad MySQL alebo PostgreSQL. Z ďalších progresívnych technológií, ktoré by sme chceli využiť pri realizácii riešenia spomeňme napríklad Hibernate, pre prácu s dátovým úložiskom, Spring pre jednoduchú konfiguráciu systému, JSP pre vytvorenie dynamického rozhrania, či JUnit pre testovanie samotného systému.

Caché a Ext JS

Prostredie pre vývoj webových aplikácií Ext JS a databázové prostredie Caché sú technológie vyžadované v zadaní a aj na konzultáciách so zadávateľom projektu nám bolo odporučené tieto technológie využiť. Aj napriek tomu, že nikto z vývojového tímu nemá skúsenosti s uvedenou technológiou, naše praktické skúsenosti s inými databázovými technológiami nám umožňujú navrhnúť a implementovať riešenie postavené aj na týchto technológiách.

Správa študentských projektov na fakulte

Koncepcia riešenia sa bude vyvíjať od novo požadovanej funkcionality informačného systému na správu študentských projektov. Podľa zadania je základná funkcionality nasledujúca:

- príprava zadaní projektov pedagógmi
- výber projektov študentmi
- prideľovanie oponentov projektom

- vypracovanie posudkov na projekty
- vyhľadavanie relevantných projektov

V ďalšom rade dokonale zanalyzujeme existujúci systém YonBan a v návrhu nového informačného systému využijeme vyhovujúce spôsoby riešenia problémov, prípadne odstránime drobné nedostatky. Problémové časti riešené nevyhovujúcim spôsobom nahradíme úplne novými spôsobmi riešenia.

Podľa nás za hlavné a známe nedostatky v súčasnom informačnom systéme YonBan považujeme neprehľadnosť resp. neintuitívnosť pri práci s týmto systémom. Ďalším problémom je celkovo zložitý proces odovzdávania projektov, pri ktorom sa využíva akademický informačný systém. Časť procesu prebieha v systéme YonBan a časť v akademickom informačnom systéme, avšak v oboch systémoch dochádza k nadbytočnej činnosti, ktorá už bola výkonná v druhom systéme.

Spomenuté problémy sa istým spôsobom prekrývajú, ale na každý z nich máme navrhnutý samostatný spôsob riešenia. Riešením prvého problému je navrhnutie intuitívnych používateľských rozhraní využívajúcich prvky priamej napovedy a sprievodcov. Používateľské rozhrania budú navrhnuté tak, aby užívateľ v každom momente vedel, čo práve robí, bez potreby využívania externej nápovedy napr. vo forme pdf súboru.

Riešení druhé problému je viacero. Jedným z nich je úplne oddelenie systému na správu študentských projektov od akademického informačného systému. To by znamenalo úplnú správu nad projektmi, čo by malo za následok pridanie ďalšej funkcionality ako je uchovávanie projektov, zálohovanie atď.

Ďalším riešením je rozdelenie procesu. To by znamenalo proces odovzdania projektu, myslí sa súboru, by zabezpečoval akademický informačný systém a samotná správa projektov by bola zabezpečovaná novým informačným systémom na správu študentských projektov. Prípadná potreba výmeny informácií medzi systémami by bola zabezpečená pomocou rozhrania medzi týmito systémami.

Samotnú implementáciu by sme radi realizovali pomocou programovacieho jazyka, s ktorým má väčšina tímu najväčšie skúsenosti a to je programovací jazyk Java. Na výber databázového systému nie sme striktně zameraní a sme schopní sa prispôbiť možnostiam.

Prispôsobiteľný Widget

Ako riešenie tohoto problému vidíme použitie web widgetu na šírenie takýchto informácií z centrálného úložiska. Toto riešenie zahŕňa vytvorenie prostredia na návrh widgetu. Toto prostredie by malo ponúkať možnosť voľby veľkosti a tvaru widgetu podľa požiadaviek web stránky, v ktorej bude umiestnený. Taktiež by malo umožniť vyber druhu informácií, ktoré bude navrhnutý widget zobrazovať z informácií zhromaždených v centrálnom úložisku.

A samozrejme personalizáciu spôsobu zobrazenia týchto informácií. Rovnako by toto prostredie malo umožniť výber algoritmu, ktorý bude dáta zobrazované vo widgete generovať z dát úložiska. Súčasťou tohoto prostredia budú samozrejme aj metódy, ktorých úlohou bude generovanie zdrojového kódu widgetu navrhnutého v tomto prostredí, ktorý bude môcť byť následne vložený do web stránky.

Druhou časťou riešenia bude umožniť centrálnemu úložisku získavať dáta z rôznych zdrojov a následne zabezpečiť ich spracovanie a distribúciu k príslušnému widgetu.

Tretia časť riešenia bude spočívať v návrhu nástroja, ktorého úlohou bude napĺňanie centrálného úložiska dátami, ktoré budú následne widgety celého systému môcť využívať.

Pri tvorbe systému bude hlavný dôraz kladený na jednoduchosť a intuitívnosť použitia koncovým používateľom a to či už v prípade prostredia na návrh widgetu alebo nástroja na napĺňanie centrálného úložiska dátami. Ďalšími hlavnými cieľami sú spoľahlivosť systému a jeho modulárnosť kvôli možnosti jednoduchého rozšírenia.

2.4 Priorita tém

1. Vyhľadávanie a prístupnenie citácií (Portál)
2. Správa študentských projektov na fakulte (Projekty)
3. Prispôsobiteľný Widget (Widget)
4. Tvorba rozvrhov (Rozvrhy)
5. Portál pre časopis (Časopis)
6. Model používateľa pre jeho identifikáciu (UserModel)
7. Tréner mentálnych schopností (Tréner)
8. 3D grafická podpora vyhľadávania znalostí v dokumentoch (3D-Znalosti)
9. Simulated Car Racing Competition 2011 (Car Racing)
10. Objektové úložisko dát (Úložisko)
11. Interaktívna vizualizácia grafových štruktúr v 3D priestore (Vizualizácia)
12. Evolučný simulátor umelého života založený na heuristických pravidlách (HERBAL)
13. Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení (Mobily)
14. Dizajn s použitím obohatenej reality (Dizajn)
15. Adaptívny proxy server (Proxy-plugins)
16. Crowdsourcing verejných dát (CrowdPublic)
17. RoboCup tretí rozmer (RoboCup 3D)
18. Virtuálna IIT (VFIIT)
19. Imagine Cup 2011: Game Design (ICup2011)

2.5 Rozvrhy členov tímu

		7 8	8 9	9 10	10 11	11 12	12 13	13 14	14 15	15 16	16 17	17 18	18 19	19 20	20 21
Pon	AB		VI	VI	VI	VI					TP	TP	VIS	VIS	
	PB								ML II	ML II	TP	TP	VIS	VIS	
	TF								SU	SU	TP	TP	VIS	VIS	
	AP								ML II	ML II	TP	TP	VIS	VIS	
	EP										TP	TP	VIS	VIS	
	MU		VI	VI	VI	VI			zkgra	zkgra	TP	TP	VIS	VIS	
	MV						ooans	ooans	zkgra	zkgra	TP	TP	VIS	VIS	
		7 8	8 9	9 10	10 11	11 12	12 13	13 14	14 15	15 16	16 17	17 18	18 19	19 20	20 21
Utr	AB									MSI	MSI	MSI	MSI	MSI	MSI
	PB					Mark	Mark			MSI	MSI	MSI	MSI	MSI	MSI
	TF					NS	NS			MSI	MSI	MSI	MSI	MSI	MSI
	AP									MSI	MSI	MSI	MSI	MSI	MSI
	EP	Kod	Kod							MSI	MSI	MSI	MSI	MSI	MSI
	MU									MSI	MSI	MSI	MSI	MSI	MSI
	MV									MSI	MSI	MSI	MSI	MSI	MSI
		7 8	8 9	9 10	10 11	11 12	12 13	13 14	14 15	15 16	16 17	17 18	18 19	19 20	20 21
Str	AB														
	PB	Mark	Mark						ML II	ML II	ML II				
	TF			NS	NS										
	AP								ML II	ML II	ML II				
	EP										KOD	KOD			
	MU					PeWe	PeWe								
	MV														
		7 8	8 9	9 10	10 11	11 12	12 13	13 14	14 15	15 16	16 17	17 18	18 19	19 20	20 21
Štv	AB								PDT	PDT	AIS	AIS	AIS		
	PB								PDT	PDT	AIS	AIS	AIS		

	TF					SU	SU			ASS	ASS	ASS			
	AP								PDT	PDT	AIS	AIS	AIS		
	EP									ASS	ASS	ASS	PDT	PDT	
	MU							zkgra	zkgra	ASS	ASS	ASS			
	MV							zkgra	zkgra	ASS	ASS	ASS		ooans	ooans
		7 8	8 9	9 10	10 11	11 12	12 13	13 14	14 15	15 16	16 17	17 18	18 19	19 20	20 21
Pia	AB			PDT	PDT										
	PB			PDT	PDT	PDT	PDT	PDT	PDT						
	TF														
	AP			PDT	PDT										
	EP			PDT	PDT										
	MU														
	MV														

Tabuľka 2-1 – Rozvrh tímu

Prednášky

Cvičenia

Nevyhovujúce termíny

Vyhovujúce termíny

3 Plán projektu

Tímový projekt Robocup 3D je vyvíjaný pomocou metodiky vývoja SCRUM. Táto metodika pristupuje k plánovaniu projektu pomocou šprintov. Plánovanie prebieha vždy na začiatku šprintu, ktorého dĺžka je v tíme The A Team je dva týždne. Plánovanie jednotlivých šprintov prebieha z dlhodobého plánovania tímového projektu.

3.1 Termíny v zimnom semestri

Dátum	Popis
07.10.2010	Začiatok šprintu č. 01
21.10.2010	Začiatok šprintu č. 02, Vyhodnotenie šprintu č. 01
04.11.2010	Začiatok šprintu č. 03, Vyhodnotenie šprintu č. 02
04.11.2010	Kontrolný bod č. 01 - Odovzdanie dokumentácie prvých dvoch šprintov
18.11.2010	Začiatok šprintu č. 04, Vyhodnotenie šprintu č. 03
02.12.2010	Začiatok šprintu č. 05, Vyhodnotenie šprintu č. 04
09.12.2010	Vyhodnotenie šprintu č. 05
14.12.2010	Kontrolný bod č. 02 - Odovzdanie dokumentácie prvých piatich šprintov a opisu vytvoreného prototypu
15. - 21.12.2010	Prezentácia prototypu

Tabuľka 3-1 – Termíny v zimnom semestri

3.2 Dlhodobý plán v zimnom semestri

ID	Úloha	Zadané	Plánované ukončenie	Ukončené
1	Zanalyzovať existujúcich hráčov a vybrať hráča, na ktorého nadviažeme	07.10.2010	21.10.2010	21.10.2010
2	Analýza zdrojových kódov, vytvorenie testov a návrh vylepšení	07.10.2010	21.10.2010	21.10.2010
3	Analýza zdrojových kódov, vytvorenie testov a návrh vylepšení	21.10.2010	04.11.2010	04.11.2010
4	Implementácia návrhov s cieľom vytvorenia stabilnej spodnej vrstvy	21.10.2010	09.12.2010	
5	Vytvorenie základných pohybov hráča	21.10.2010	Letný semester	Letný semester

Tabuľka 3-2 – Dlhodobý plán v zimnom semestri

3.3 Šprint č. 01

ID	Úloha	Zadané	Plánované ukončenie	Ukončené
01.01	Vytvoriť stránku tímu	07.10.2010	21.10.2010	10.10.2010
01.02	Nástroje pre podporu projektu	07.10.2010	21.10.2010	Presunuté do ďalšieho šprintu

01.03	Vytvoriť základnú kostru dokumentácie	07.10.2010	21.10.2010	10.10.2010
01.04	Analýza fakultných hráčov	07.10.2010	21.10.2010	20.10.2010
01.05	Analýza zahraničných hráčov	07.10.2010	21.10.2010	19.10.2010
01.06	Zanalyzovať fungovanie systému a pravidiel RoboCup-u	07.10.2010	21.10.2010	20.10.2010
01.07	Inštalácia serveru, editoru pohybov a hráča(každý člen tímu iného)	07.10.2010	21.10.2010	14.10.2010
01.08	Vyskúšanie si vytvorenie a editáciu pohybu hráča	07.10.2010	21.10.2010	14.10.2010
01.09	Výber hráča, z ktorého tím bude vychádzať	07.10.2010	21.10.2010	21.10.2010

Tabuľka 3-3 – Backlog šprintu č. 01

3.4 Šprint č. 02

ID	Úloha	Zadané	Plánované ukončenie	Ukončené
02.01	Vytvoriť základ pre dokumentáciu riadenia projektu	21.10.2010	04.11.2010	30.10.2010
02.02	Zriadenie repozitára pre manažment verzií	21.10.2010	28.10.2010	25.10.2010
02.03	Nainštalovanie klienta manažmentu verzií a vykonanie skúšobného commitu	21.10.2010	28.10.2010	27.10.2010

02.04	Oboznámenie sa so zdrojovými kódmi vybraného hráča, jeho kompilácia a spustenie	21.10.2010	28.10.2010	27.10.2010
02.05	Analýza kódu hráča - časť zodpovedná za prijímanie správ - napísanie unit testov a prípadný refaktoring	21.10.2010	04.11.2010	03.11.2010
02.06	Analýza kódu hráča - časť zodpovednú za odosielanie správ –napísanie unit testov a prípadný refaktoring	21.10.2010	04.11.2010	03.11.2010
02.07	Analýza kódu hráča - časť zodpovednú fyziku - napísanie unit testov a prípadný refaktoring	21.10.2010	04.11.2010	03.11.2010
02.08	Analýza kódu hráča - časť zodpovedná za riadiacu logiku - napísanie unit testov a prípadný refaktoring	21.10.2010	04.11.2010	03.11.2010
02.09	Celková analýza kódu, zamyslieť sa ako kód rozšíriť – vytvoriť hrubý návrh	21.10.2010	04.11.2010	03.11.2010
02.10	Nástroje pre podporu projektu	21.10.2010	24.10.2010	24.10.2010

Tabuľka 3-4 – Backlog šprintu č. 02

4 Úlohy členov tímu

Táto kapitola opisuje úlohy členov tímu a autorstvo jednotlivých častí dokumentácie k inžinierskemu dielu a dokumentácie k riadeniu.

4.1 Úlohy jednotlivých členov tímu

Úloha	Člen tímu
Vedúci tímu	Bc. Miroslav Vasiľ
Zástupca vedúceho tímu	Bc. Tomáš Florek
Manažér plánovania	Bc. Adam Pagáč
Manažér vývoja	Bc. Eduard Pribula
Manažér kvality	Bc. Maroš Unčík
Manažér podporných činností	Bc. Peter Borga
Scrum master	Bc. Anton Balucha
Správca dokumentácie	Bc. Miroslav Vasiľ
Správca stránky tímu	Bc. Adam Pagáč

Tabuľka 4-1 – Úlohy jednotlivých členov tímu

4.2 Autorstvo jednotlivých častí dokumentácie

Táto sekcia opisuje autorstvo jednotlivých častí dokumentácie k inžinierskemu dielu a dokumentácie k riadeniu.

4.2.1 Dokumentácia k inžinierskemu dielu

Časť	Člen tímu
1.1 Účel dokumentu	Bc. Miroslav Vasiľ
1.2 Cieľ dokumentu	Bc. Maroš Unčík
2.1.1.1 Nástroj pre podporu projektu	Bc. Maroš Unčík
2.1.1.2 Analýza fakultných hráčov	Bc. Tomáš Florek Bc. Peter Borga
2.1.1.3 Analýza zahraničných hráčov	Bc. Maroš Unčík Bc. Miroslav Vasiľ
2.1.1.4 Analýza fungovania systému a pravidiel Robocup	Bc. Eduard Pribula
2.1.2.1 Vytvorenie stránky tímu	Bc. Adam Pagáč
2.1.2.2 Nástroj pre podporu projektu	Bc. Maroš Unčík
2.1.2.3 Vytvorenie dokumentácie	Bc. Miroslav Vasiľ
2.1.2.4 Výber hráča k ďalšiemu vývoju	Bc. Miroslav Vasiľ
2.1.3.1 Vytvorenie stránky tímu	Bc. Adam Pagáč
2.1.3.2 Nástroj pre podporu projektu	Bc. Maroš Unčík

2.1.3.3 Vytvorenie dokumentácie	Bc. Miroslav Vasiľ
2.1.3.4 Inštalácia serveru, editora pohybov a hráča	Bc. Peter Borga Bc. Miroslav Vasiľ
2.1.3.5 Vytvorenie a editácia pohybov hráča	Bc. Miroslav Vasiľ
2.1.4.1 Vytvorenie stránky tímu	Bc. Adam Pagáč
2.1.4.2 Nástroj pre podporu projektu	Bc. Maroš Unčík
2.1.4.3 Vytvorenie dokumentácie	Bc. Miroslav Vasiľ
2.2.1.1 Analýza prevzatých kódov JIM	Bc. Maroš Unčík
2.2.1.2 Analýza najdôležitejších tried	Bc. Tomáš Florek
2.2.1.3 Analýza miest pre rozšírenie	Bc. Maroš Unčík
2.2.1.4 Analýza testovania	Bc. Maroš Unčík
2.2.1.5 Manažment verzií	Bc. Tomáš Florek Bc. Maroš Unčík
2.2.2.1 Návrh v analýze zdrojových kódov	Bc. Maroš Unčík
2.2.2.2 Návrh testovanie	Bc. Maroš Unčík
2.2.2.3 Návrh manažment verzií	Bc. Adam Pagáč
2.2.2.4 Vytvorenie dokumentácie k riadeniu	Bc. Miroslav Vasiľ
2.2.2.5 Aktualizácia webovej stránky	Bc. Adam Pagáč
2.2.3.1 Implementácia úpravy zdrojových kódov	Bc. Peter Borga Bc. Maroš Unčík
2.2.3.2 Implementácia testovania	Bc. Peter Borga

	Bc. Maroš Unčík
2.2.3.3 Implementácia manažmentu verzií	Bc. Tomáš Florek Bc. Maroš Unčík
2.2.3.4 Vytvorenie dokumentácie k riadeniu	Bc. Miroslav Vasiľ
2.2.3.5 Aktualizácia webovej stránky	Bc. Adam Pagáč
2.2.4.1 Testovanie úpravy zdrojových kódov	Bc. Maroš Unčík
2.2.4.2 Testovanie správnosti testov	Bc. Maroš Unčík
2.2.4.3 Testovanie manažmentu verzií	Bc. Peter Borga Bc. Tomáš Florek
2.2.4.4 Vytvorenie dokumentácie k riadeniu	Bc. Miroslav Vasiľ
2.2.4.5 Testovanie aktualizácie webovej stránky	Bc. Adam Pagáč
Integrácie jednotlivých častí dokumentácie	Bc. Miroslav Vasiľ
Obsahová úprava dokumentácie	Bc. Maroš Unčík Bc. Miroslav Vasiľ
Formálna úprava dokumentácie	Bc. Miroslav Vasiľ

Tabuľka 4-2 – Autorstvo jednotlivých častí dokumentácie k inžinierskemu dielu

4.2.2 Dokumentácia k riadeniu

Časť	Člen tímu
1.Úvod	Bc. Miroslav Vasiľ
2.1 Zloženie tímu	Všetci
2.2 Motivácie	Všetci
2.3 Koncepcie riešenia	Všetci
2.4 Priorita tém	Všetci
2.5 Rozvrhy členov tímu	Všetci
3.1 Termíny v zimnom semestri	Bc. Miroslav Vasiľ
3.2 Dlhodobý plán v zimnom semestri	Bc. Adam Pagáč
3.3 Šprint č. 01	Bc. Miroslav Vasiľ
3.4 Šprint č. 02	Bc. Miroslav Vasiľ
4.1 Úlohy jednotlivých členov tímu	Bc. Miroslav Vasiľ
4.2 Autorstvo jednotlivých časti dokumentácie	Bc. Miroslav Vasiľ
5.1 Zápisnica č. 01	Bc. Tomáš Florek
5.2 Zápisnica č. 02	Bc. Tomáš Florek
5.3 Zápisnica č. 03	Bc. Maroš Unčík
5.4 Zápisnica č. 04	Bc. Maroš Unčík
5.5 Zápisnica č. 05	Bc. Miroslav Vasiľ

6.1 Scrum	Bc. Anton Balucha
6.2 Štábna kultúra	Bc. Peter Borga
6.3 Podporné prostriedky	Bc. Miroslav Vasiľ
7.1 Manažment verzií	Bc. Maroš Unčík
7.2 Manažment dokumentácie	Bc. Miroslav Vasiľ
7.3 Manažment úloh	Bc. Maroš Unčík
7.4 Manažment testovania	Bc. Maroš Unčík
7.5. Manažment zmien	Bc. Miroslav Vasiľ
8. Preberacie protokoly	Bc. Miroslav Vasiľ
Integrácie jednotlivých častí dokumentácie	Bc. Miroslav Vasiľ
Obsahová úprava dokumentácie	Bc. Miroslav Vasiľ
Formálna úprava dokumentácie	Bc. Miroslav Vasiľ

Tabuľka 4-3 - Autorstvo jednotlivých častí dokumentácie k riadeniu

5 Záznamy zo stretnutí

Táto kapitola obsahuje zápisnice z jednotlivých stretnutí tímu.

5.1 Zápisnica č. 01

Téma	Úvodné stretnutie
Miesto stretnutia	Pracovňa (D210)
Dátum stretnutia	30.09.2010
Čas stretnutia	8:00 – 10:00
Prítomní	Ing. Marián Lekavý, PhD. Bc. Anton Balucha Bc. Peter Borga Bc. Tomáš Florek Bc. Adam Pagáč Bc. Eduard Pribula Bc. Maroš Unčík Bc. Miroslav Vasiľ
Zapisovateľ	Bc. Tomáš Florek
Overil	Bc. Maroš Unčík, Bc. Miroslav Vasiľ

Priebeh

1. Na začiatku stretnutia nás Ing. Marián Lekavý, PhD. krátko oboznámil s projektom, jeho predstavami a očakávaniami - vylepšenie pohybov hráča a implementácia samotnej logiky hry.
2. Následne sme prešli k formálnym záležitostiam. Stretnutia budú prebiehať v softvérom štúdiu každý štvrtok o 8:00. Predpokladaná dĺžka každého stretnutia sú tri hodiny.
3. Vzájomnou diskusiou sme identifikovali úlohy, ktoré je potrebné vykonať pred zahájením prvého šprintu. Úlohy sú orientované na oboznámenie sa s RoboCupom a preto ich vykonávajú všetci členovia tímu.

4. Ing. Marián Lekavý, PhD. nás upozornil, že je potrebné čo najskôr si stanoviť ciele projektu.

Úlohy

ID	Riešiteľ	Popis	Dátum zadania	Termín
00.01	Všetci	Určiť vedúceho a jeho zástupcu	30.09.2010	07.09.2010
00.02	Všetci	Naštudovať prácu tímov z predchádzajúcich rokov	30.09.2010	07.09.2010
00.03	Všetci	Naštudovať metódu SCRUM	30.09.2010	07.09.2010
00.04	Všetci	Určiť človeka na kontrolu termínov	30.09.2010	07.09.2010
00.05	Všetci	Vytvoriť plagát tímu	30.09.2010	07.09.2010
00.06	Všetci	Vymyslieť názov tímu	30.09.2010	07.09.2010

5.2 Zápisnica č. 02

Téma	Prezentácia minuloročného tímu. Identifikácia a rozdelenie úloh v šprinte č. 01
Miesto stretnutia	Softvérové štúdio (D07)
Dátum stretnutia	07.10.2010
Čas stretnutia	8:00 – 11:00
Prítomní	Ing. Marián Lekavý, PhD. Bc. Anton Balucha Bc. Peter Borga Bc. Tomáš Florek Bc. Adam Pagáč Bc. Eduard Pribula Bc. Maroš Unčík Bc. Miroslav Vasiľ
Zapisovateľ	Bc. Tomáš Florek
Overil	Bc. Maroš Unčík, Bc. Miroslav Vasiľ

Priebeh

1. Na začiatku stretnutia nám boli prezentované výsledky tímových projektov RoboCup 3D. Prednášateľ bol Bc. Ivanom Hujsi, člen minuloročného tímu Critical Error. Hlavné body prezentácie boli nasledovné:
 - Opis prostredia a modelu hráča
 - Ukážka práce s editorom pohybov
 - Ukážka fungovania hráča
2. Bc. Ivan Hujsi nám poskytol prezentáciu spolu s hráčom implementovaným v jazyku Java.
3. V skratke sme si pripomenuli metódu SCRUM. Zhodli sme sa na dvojtyždňových šprintoch. Dohodli sme sa, že každý týždeň reportujeme na stretnutiach naše úlohy a zároveň, že úloha je splnená, ak je preukázateľné, že je splnená.
4. Rozdelili sme si roly v tíme.
 - Vedúci tímu Bc. Miroslav Vasiľ
 - Zástupca vedúceho tímu: Bc. Tomáš Florek
 - Scrum master Bc. Anton Balucha

- Vedúci vývoja Bc. Eduard Pribula
- Manažér plánovania Bc. Adam Pagáč
- Manažér kvality Bc. Maroš Unčík
- Manažér podporných činností Bc. Peter Borga
- Správa dokumentácie Bc. Miroslav Vasiľ
- Správa web stránky Bc. Adam Pagáč

- Identifikovali sme vzájomnou diskusiou úlohy, ktoré je potrebné vykonať v prvom šprinte.
- Pomocou plánovacieho pokru sme odhadli časovú náročnosť úloh prvého šprintu.
- Pridelili sme si zodpovednosť za vykonanie úloh prvého šprintu.

Bilancia splnenia úloh z prvého stretnutia

ID	Popis úlohy	Zodpovedný	Termín	Aktuálny stav
	Poznámky			
00.01	Určiť vedúceho a jeho zástupcu	Všetci	07.10.2010	Splnená
	- za vedúceho bol určený Bc. Miroslav Vasiľ			
00.02	Naštudovať prácu tímov z predchádzajúcich rokov	Všetci	07.10.2010	Splnená
	- každý člen tímu sa oboznámil so základnými informáciami o RoboCupe, ako aj o výsledkoch z minulých ročníkov			
00.03	Naštudovať metódu SCRUM	Všetci	07.10.2010	Splnená
00.04	Určiť človeka na kontrolu termínov	Všetci	07.10.2010	Splnená
	- dohľad nad termínmi zabezpečuje Bc. Maroš Unčík			
00.05	Vytvoriť plagát tímu	Všetci	07.10.2010	Splnená
	- plagát vytvoril Bc. Adam Pagáč			
00.06	Vymyslieť názov tímu	Všetci	21.10.2010	Splnená
	- názov tímu je The A Team			

Sprint BackLog

ID	Riešiteľ	Popis	Odhadovaná časová náročnosť	Dátum zadania	Termín
01.01	Bc. Adam Pagáč	Vytvoriť stránku tímu	5h	07.10.2010	21.10.2010
01.02	Bc. Anton Balucha	Nástroje pre podporu projektu	5h	07.10.2010	21.10.2010
01.03	Bc. Miroslav Vasiľ	Vytvoriť základnú kostru dokumentácie	2h	07.10.2010	21.10.2010
01.04	Bc. Peter Borga Bc. Tomáš Florek	Analýza fakultných hráčov	13h	07.10.2010	21.10.2010
01.05	Bc. Maroš Unčík Bc. Miroslav Vasiľ	Analýza zahraničných hráčov	13h	07.10.2010	21.10.2010
01.06	Bc. Eduard Pribula	Zanalyzovať fungovanie systému a pravidiel RoboCup-u	5h	07.10.2010	21.10.2010
01.07	Všetci	Inštalácia serveru, editoru pohybov a hráča(každý člen tímu iného)	20h	07.10.2010	21.10.2010
01.08	Všetci	Vyskúšanie si vytvorenie a editáciu pohybu hráča	7h	07.10.2010	21.10.2010
01.09	Všetci	Výber hráča, z ktorého tím bude vychádzať	8h	07.10.2010	21.10.2010

5.3 Zápisnica č. 03

Téma	Kontrola priebehu šprintu č. 01
Miesto stretnutia	Softvérové štúdio (D07)
Dátum stretnutia	14.10.2010
Čas stretnutia	8:00 – 10:40
Prítomní	Ing. Marián Lekavý, PhD. Bc. Anton Balucha Bc. Peter Borga Bc. Tomáš Florek Bc. Adam Pagáč Bc. Eduard Pribula Bc. Maroš Unčík Bc. Miroslav Vasiľ
Zapisovateľ	Bc. Maroš Unčík
Overil	Bc. Maroš Unčík, Bc. Miroslav Vasiľ

Priebeh

1. Na začiatku stretnutia sme si postupne predstavili úlohy splnené za posledný týždeň – každý povedal to, čo spravil, prípadne s čím mal problém.
2. Riešili sme vzájomne problémy, ktoré vznikli v súvislosti s riešením úloh:

- a. Problém: Bc. Peter Borga mal problémy s editorom pohybov – vyhadzoval výnimku pri pripojení hráča na server pod Windows 7

Riešenie: inštalácia Visual Studia a následné rekompilácia projektu, pravdepodobne mu v počítači chýba knižnica pre debugovanie v C#

- b. Problém: Bc. Miroslav Vasiľ mal problém s rozbeháním hráča implementovaného v Jave

Riešenie: problém je pravdepodobne v argumentoch, ktoré program hráča potrebuje pri spustení – bude kontaktovať Bc. Ivana Hujsi (člen tímu, ktorý hráča vytvoril) ako postupovať

c. Problém: Bc. Anton Balucha mal problémy s inštaláciou JIRA

Riešenie: rozhodli sme nestrácať čas s hľadaním riešenia a skúsiť inštalovať iný nástroj pre manažment projektu - dotProject

3. Ing. Marián Lekavý, PhD. nás upozornil na potrebu písania príbehu v dokumentácií.
4. Ing. Marián Lekavý, PhD. nás uvedomil o tom, že na budúcom stretnutí predvážame výsledky našej práce.

Bilancia splnenia úloh z šprintu č. 01

ID	Popis úlohy	Zodpovedný	Termín	Aktuálny stav
	Poznámky			
01.01	Vytvoriť stránku tímu	Bc. Adam Pagáč	21.10.2010	Splnená
	- stránka bola vložená na server labss, naplnená obsahom a predvedená			
01.02	Nástroje pre podporu projektu	Bc. Anton Balucha	21.10.2010	Rozpracovaná
	- neúspech s inštaláciou JIRA, prechádzame na nástroj dotProject			
01.03	Vytvoriť základnú kostru dokumentácie	Bc. Miroslav Vasil'	21.10.2010	Splnená
	- kostra dokumentácie nám bola zaslaná prostredníctvom mailu			
01.04	Analýza fakultných hráčov	Bc. Peter Borga Bc. Tomáš Florek	21.10.2010	Rozpracovaná
01.05	Analýza zahraničných hráčov	Bc. Miroslav Vasil' Bc. Maroš Unčík	21.10.2010	Rozpracovaná
01.06	Zanalyzovať fungovanie systému a pravidiel RoboCup-u	Bc. Eduard Pribula	21.10.2010	Rozpracovaná
01.07	Nainštalovať si na vlastnom počítači server a rozbehať hráča,	Všetci	21.10.2010	Splnená

	skúsiť editovať pohyby			
	- táto úloha sa podarila splniť všetkým okrem Bc. Petra Borgu, ktorý mal problém s nástrojom na editovanie pohybov, na stretnutí sme problém odstránili			
01.08	Vyskúšanie si vytvorenie a editáciu pohybu hráča	Všetci	21.10.2010	Splnená
01.09	Výber hráča, z ktorého tím bude vychádzať	Všetci	21.10.2010	Rozpracovaná

5.4 Zápisnica č. 04

Téma	Zhodnotenie šprintu č. 01 a identifikácia a rozdelenie úloh v šprinte č. 02
Miesto stretnutia	Softvérové štúdio (D07)
Dátum stretnutia	21.10.2010
Čas stretnutia	8:00 – 11:10
Prítomní	Ing. Marián Lekavý, PhD. Bc. Anton Balucha Bc. Peter Borga Bc. Tomáš Florek Bc. Adam Pagáč Bc. Eduard Pribula Bc. Maroš Unčík Bc. Miroslav Vasiľ
Zapisovateľ	Bc. Maroš Unčík
Overil	Bc. Maroš Unčík, Bc. Miroslav Vasiľ

Priebeh

1. Na začiatku stretnutia sme začali bilanciou práce každého člena tímu za posledný týždeň. Každý z členov postupne vymenoval a opísal úlohy, ktoré vykonával, prípadne spomenul s čím mal problém. Problém sa vyskytol iba so splnením úlohy č. 01.02 – nainštalovanie podporného nástroja pre manažment projektu. Hoci sa Bc. Antonovi Baluchovi, podaril nainštalovať jeden systém pre manažment projektu, s týmto systémom nie sme spokojní, a preto sa táto úloha presúva do ďalšieho šprintu. Úloha má v druhom šprinte vysokú prioritu.
2. Ing. Marián Lekavý, PhD. nám prezrel výsledky práce. Najprv skontroloval webovú stránku, s ktorou bol spokojný. Následne skontroloval projektovú dokumentáciu spolu s výstupom jednotlivých analýz. Túto dokumentáciu mu máme poslať na prečítanie. Pri prezeraní dokumentácie nás upozornil na niektoré formálne nezrovnalosti, ktoré by sme mali v dokumentácií opraviť, pre jej lepšiu kvalitu:

- číslovanie stránok v prílohe je podľa písmen abecedy, je to neštandardné riešenie a odporučil nám prehodnotiť toto číslovanie
 - každá kapitola musí obsahovať aspoň minimálne množstvo textu
 - jednotlivé kapitoly by mali byť číslované (aj podľa úrovni)
 - treba upraviť odkazy na použitú literatúru, ktoré sú momentálne v nesprávnom formáte
3. Ing. Marián Lekavý, PhD. nás upozornil na potrebu písania dokumentácie k riadeniu projektu. Hoci viaceré informácie k projektovému riadeniu máme k dispozícii v písomnej podobe, zatiaľ sme nedali tieto informácie do jednotného dokumentu. V nasledujúcom šprinte vykonáme nápravu.
 4. Stručne sme zhodnotili výsledky prvého šprintu. Prácu členov tímu sme v uplynulom týždni mali vyjadriť percentuálne. Toto hodnotenie máme zaslať Ing. Mariánovi Lekavému, PhD. mailom, rovnako ako uviesť v projektovej dokumentácii. Neskôr sme sa zhodli na takomto rozdelení:

• Anton Balucha	12%
• Eduard Pribula	12%
• Maroš Unčík	20%
• Miroslav Vasil'	20%
• Adam Pagáč	12%
• Peter Borga	12%
• Tomáš Florek	12%
 5. Identifikovali a diskutovali sme úlohy, ktoré je potrebné vykonať v nasledujúcom šprinte. Tieto úlohy sú zapísané v Sprint Backlog.
 6. Ing. Marián Lekavý, PhD. nás usmernil, aby sme s ním prekonzultovali prostredníctvom mailu dôležité rozhodnutia do konca prebiehajúceho týždňa, najmä tie, ktoré sa týkajú výberu hráča. Preto mu musíme prostredníctvom mailu zaslať naše rozhodnutie o výbere hráča, v ktorom budeme pokračovať v tímovom projekte.
 7. Urobili sme časový odhad jednotlivých úloh pomocou plánovacieho pokru.
 8. Pre nové úlohy sme si určili zodpovednosť a termín ich splnenia.
 9. Diskutovali sme výber hráča. Za hráča, v ktorom budeme pokračovať, sme vybrali hráča implementovaného v jazyku Java.
 10. Diskutovali sme nástroj pre manažment verzií. Za tento nástroj sme zvolili Mercurial.

Bilancia splnenia úloh z šprintu č. 01

ID	Popis úlohy	Zodpovedný	Termín	Aktuálny stav
	Poznámky			
01.01	Vytvoriť stránku tímu	Bc. Adam Pagáč	21.10.2010	Splnená
	- stránka bola vložená na server labss, naplnená obsahom a predvedená			
01.02	Nástroje pre podporu projektu	Bc. Anton Balucha	21.10.2010	Presunutá do ďalšieho šprintu
	- neúspech s inštaláciou JIRA, prechádzame na nástroj dotProject – ten sa rovnako nepodaril nainštalovať, a preto sa táto úloha presúva do ďalšieho šprintu			
01.03	Vytvoriť základnú kostru dokumentácie	Bc. Miroslav Vasiľ	21.10.2010	Splnená
	- kostra dokumentácie nám bola zaslaná prostredníctvom mailu			
01.04	Analýza fakultných hráčov	Bc. Peter Borgia Bc. Tomáš Florek	21.10.2010	Splnená
	- analýza fakultných hráčov bola pridaná do projektovej dokumentácie			
01.05	Analýza zahraničných hráčov	Bc. Miroslav Vasiľ Bc. Maroš Unčík	21.10.2010	Splnená
	- analýza zahraničných hráčov bola pridaná do projektovej dokumentácie			
01.06	Zanalyzovať fungovanie systému a pravidiel RoboCup-u	Bc. Eduard Pribula	21.10.2010	Splnená
	- analýza fungovania systému bola pridaná do projektovej dokumentácie			

01.07	Nainštalovať si na vlastnom počítači server a rozbehať hráča, skúsiť editovať pohyby	Všetci	21.10.2010	Splnená
01.08	Vyskúšanie si vytvorenie a editáciu pohybu hráča	Všetci	21.10.2010	Splnená
01.09	Výber hráča, z ktorého tím bude vychádzať	Všetci	21.10.2010	Splnená
	-za hráča, z ktorého bude náš tím vychádzať bol vybraný hráč postavený na technológií JAVA			

Sprint BackLog

ID	Riešiteľ	Popis	Odhadovaná časová náročnosť	Dátum zadania	Termín
02.01	Bc. Miroslav Vasiľ	Vytvoriť základ pre dokumentáciu riadenia projektu	5h	21.10.2010	04.11.2010
02.02	Bc. Adam Pagáč	Zriadenie repozitára pre manažment verzií	5h	21.10.2010	28.10.2010
02.03	Všetci	Nainštalovanie klienta manažmentu verzií a vykonanie skúšobného commitu	7h	21.10.2010	28.10.2010
02.04	Všetci	Oboznámenie sa so zdrojovými kódmi vybraného hráča, jeho kompilácia a spustenie	20h	21.10.2010	28.10.2010
02.05	Bc. Peter Borga	Analýza kódu hráča - časť zodpovedná za	20h	21.10.2010	04.11.2010

	Bc. Eduard Pribula	prijímanie správ - napísanie unit testov a prípadný refaktoring			
02.06	Bc. Anton Balucha Bc. Adam Pagáč	Analýza kódu hráča - časť zodpovednú za odosielanie správ – napísanie unit testov a prípadný refaktoring	20h	21.10.2010	04.11.2010
02.07	Bc. Tomáš Florek Bc. Miroslav Vasiľ Bc. Maroš Unčík	Analýza kódu hráča - časť zodpovednú fyziku - napísanie unit testov a prípadný refaktoring	20h	21.10.2010	04.11.2010
02.08	Bc. Tomáš Florek Bc. Miroslav Vasiľ Bc. Maroš Unčík	Analýza kódu hráča - časť zodpovedná za riadiacu logiku - napísanie unit testov a prípadný refaktoring	20h	21.10.2010	04.11.2010
02.09	Bc. Anton Balucha Bc. Maroš Unčík	Celková analýza kódu, zamyslieť sa ako kód rozšíriť – vytvoriť hrubý návrh	13h	21.10.2010	04.11.2010
02.10	Bc. Anton Balucha	Nástroje pre podporu projektu	5h	21.10.2010	24.10.2010

5.5 Zápisnica č. 05

Téma	Kontrola priebehu šprintu č. 02
Miesto stretnutia	Softvérové štúdio (D07)
Dátum stretnutia	28.10.2010
Čas stretnutia	8:00 – 10:40
Prítomní	Ing. Marián Lekavý, PhD. Bc. Anton Balucha Bc. Peter Borga Bc. Tomáš Florek Bc. Adam Pagáč Bc. Eduard Pribula Bc. Maroš Unčík Bc. Miroslav Vasiľ
Zapisovateľ	Bc. Miroslav Vasiľ
Overil	Bc. Maroš Unčík

Priebeh

1. Na začiatku stretnutia sme si postupne predstavili úlohy splnené za posledný týždeň – každý povedal to, čo spravil, prípadne s čím mal problém.
2. Úloha č. 02.10 nebolo splnená. Ako alternatívne riešenie sme vybrali systém JTrack, ktorý sa podarilo nainštalovať. Hoci tento systém neposkytuje také množstvo nástrojov ako systémy, ktoré sme sa snažili nainštalovať (Jira, dotProject), zhodli sme sa na tom, že nám bude v zimnom semestri postačovať.
3. Ing. Marián Lekavý, PhD. nám vysvetlil niektoré časti v dokumentácii k riadeniu.

Bilancia splnenia úloh z šprintu č. 02

ID	Popis úlohy	Zodpovedný	Termín	Aktuálny stav
	Poznámky			
02.01	Vytvoriť základ pre dokumentáciu riadenia projektu	Bc. Miroslav Vasiľ	04.11.2010	Rozpracovaná
02.02	Zriadenie repozitára pre manažment verzií	Bc. Adam Pagáč	28.10.2010	Splnená
	- repozitár bol zriadený a manažment verzií bol zvolený Mercurial			
02.03	Nainštalovanie klienta manažmentu verzií a vykonanie skúšobného commitu	Všetci	28.10.2010	Splnená
	- klient bol nainštalovaný každým členom tímu a vykonaný skúšobný commit			
02.04	Oboznámenie sa so zdrojovými kódmi vybraného hráča, jeho kompilácia a spustenie	Všetci	28.10.2010	Splnená
02.05	Analýza kódu hráča - časť zodpovedná za prijímanie správ - napísanie unit testov a prípadný refaktoring	Bc. Peter Borgia Bc. Eduard Pribula	04.11.2010	Rozpracovaná
02.06	Analýza kódu hráča - časť zodpovednú za odosielanie správ –napísanie unit testov a prípadný refaktoring	Bc. Anton Balucha Bc. Adam Pagáč	04.11.2010	Rozpracovaná
02.07	Analýza kódu hráča - časť zodpovednú fyziku - napísanie unit testov a prípadný refaktoring	Bc. Tomáš Florek Bc. Maroš Unčík Bc. Miroslav Vasiľ	04.11.2010	Rozpracovaná

02.08	Analýza kódu hráča - časť zodpovedná za riadiacu logiku - napísanie unit testov a prípadný refaktoring	Bc. Tomáš Florek Bc. Maroš Unčík Bc. Miroslav Vasiľ	04.11.2010	Rozpracovaná
02.09	Celková analýza kódu, zamyslieť sa ako kód rozšíriť – vytvoriť hrubý návrh	Bc. Anton Balucha Bc. Maroš Unčík	04.11.2010	Rozpracovaná
02.10	Nástroje pre podporu projektu	Bc. Anton Balucha	24.10.2010	Nesplnená

6 Metodiky

Táto kapitola obsahuje opis metodík využívaných pri vývoji tímového projektu.

6.1 Scrum

Scrum je iteratívna a inkrementálna metodika pre riadenie projektu využívaná v agilnom riadení vývoja softvéru.

6.1.1 Role v metodike Scrum

V metodike Scrum poznáme dva typy rolí – takzvané Pig roles a Chicken roles.

Pig roles predstavujú ľudí priamo zapojených do vývoja softvéru. Do tejto kategórie patria nasledujúce role.

- **scrum master** – nie je to líder tímu. Jeho hlavnou úlohou je odstraňovať rôzne prekážky medzi tímom a nepriaznivým stavom tak, aby boli úspešne splnené úlohy šprintu. Zabezpečuje beh scrum procesu tak ako je jeho použitie definované v tíme. Týmto svojím správaním chráni tím a udržuje ho v sústredení na prácu.
- **tím** – má zodpovednosť za dodanie produktu. Obyčajne je tím tvorený skupinou 5 až 9 ľuďmi s rôznymi skúsenosťami, ktorý robia zadanú prácu.
- **product owner** – predstavuje zákazníka. Pomáha k tomu, aby scrum tím pracoval so správnymi vecami a správnym spôsobom z business pohľadu. Taktiež má na starosti písanie user stories – zoznam úloh zoradených podľa priority a umiestnených do product backlogu.

Chicken roles predstavujú ľudí, ktorí sa nepriamo zapájajú do vývoja softvéru. Nie sú priamou súčasťou scrum procesu, no i tak musia byť braní do úvahy. Sú to totiž ľudia, pre ktorých je softvér vytváraný.

- **stakeholders, customers** – ľudia, vďaka ktorým existuje projekt, a pre ktorých je projekt vytvorený. Sú priamo zapojení do tvorby softvéru iba v počas posudzovania a zhodnocovania každého šprintu.
- **managers** – ľudia, ktorí zabezpečia prostredie pre vývoj softvéru.

6.1.2 Stretnutia počas vývoja softvéru

Počas vývoja softvéru metodikou Scrum sa tím pravidelne stretáva. Existuje niekoľko typov stretnutí.

- **daily scrum, daily standup** – stretnutia vykonávané každý deň počas šprintu za účelom získania aktuálneho statusu. Každý člen tímu na tomto stretnutí odpovie na tri otázky:
 - čo som spravil včera?
 - čo plánujem spraviť dnes?
 - vyskytli sa nejaké problémy, ktoré by mi bránili od dosiahnutia cieľa?

V prípade, že nastanú nejaké problémy je na scrum masterovi, aby odstránil zdroje týchto problémov.

- **stretnutie za účelom plánovania** – predstavuje stretnutie, na ktorom sa hovorí, aká práca má byť vykonaná. Pripravuje sa sprint backlog. Vytvárajú sa časové odhady koľko sa bude daná práca vykonávať, koľko času zaberie, aby členovia tímu úspešne dokončili danú úlohu. Toto stretnutie sa nevykonáva každý deň.
- **stretnutia za účelom zhodnotenia** – vyhodnocuje sa práca, ktorá bola ukončená, a ktorá prípadne ešte nebola ukončená. Prezentujú sa ukončené výsledky práce stakeholderom na zhodnotenie. Toto stretnutie sa taktiež nevykonáva každý deň, ale vo väčších časových intervaloch.
- **retrospektívne zhodnotenie** – každý člen tímu zhodnotí predchádzajúci šprint. Týmto zhodnotením a vzatím poučenia zo stretnutia sa proces vývoja softvéru zlepšuje. Tu každý člen odpovedá na dve otázky:
 - čo išlo a fungovalo dobre počas šprintu?
 - čo môže byť zlepšené počas nasledujúceho šprintu?

6.1.3 Proces metodiky Scrum

Product owner píše user stories (opis funkcionalít a požiadaviek podľa istých pravidiel toho, čo používateľ chce dosiahnuť), ktoré sú umiestňované do product backlogu (zoznam požiadaviek zoradených podľa priority, opis všetkých možných funkcionalít). Z tohto product backlogu vyberie product owner tie požiadavky, ktoré chce počas nasledujúceho šprintu dokončiť. S pomocou tímu a scrum mastera sa vytvorí časový odhad jednotlivých požiadaviek a funkcionalít a časový plán a funkcionality sa umiestnia do sprint backlogu (zoznam úloh zoradených podľa priority tak ako pri product backlogu, len s tým rozdielom, že sa jedná

o funkcionality implementujúce sa počas nasledujúceho šprintu). Jednotlivé funkcionality sa rozbiť na jednotlivé úlohy. Na tejto úrovni už každý člen tímu vie a rozumie, čo má robiť a sú mu pridelené úlohy.

Počas šprintu, obdobia obvyčajne 2 až 4 týždňov sa vykonávajú jednotlivé úlohy. Pridávanie nových požiadaviek a funkcionalít sa po túto dobu nevykonáva a sústreďuje sa na zadané úlohy. V prípade potreby však product owner môže prerušiť šprint a vstúpiť doň s novými požiadavkami. Členov tímu, ktorým prislúchajú pig roles, nijako neovplyvňujú šprint z časového hľadiska, i z hľadiska toho, čo sa bude implementovať. Po uplynutí tejto doby je výsledkom práce fungujúci inkrement softvéru.

6.2 Štábná kultúra

Táto kapitola opisuje metodiku písania dokumentácie a zdrojových kódov.

6.2.1 Konvencia písania dokumentácia

V tejto časti sa nachádza zoznam definovaných pravidiel pre písanie dokumentácie.

Nadpisy

Štruktúra dokumentu je rozdelená do troch úrovní nadpisov. Jednotlivé nadpisy sú pre lepšiu prehľadnosť úrovni číslované.

1 Nadpis prvej úrovne (Cambria 30b)

1.1 Nadpis druhej úrovne (Cambria 26b)

1.1.1 Nadpis tretej úrovne (Cambria 22b)

1.1.1.1 Nadpis štvrtej úrovne (Cambria 18b)

Text

Text je písaný písmom Calibri 12b, pričom je vhodne štruktúrovaný do odstavcov. Odstavce sú odsadené z ľavej strany kraja dĺžkou 1,25cm.

Obrázky a tabuľky

Obrázky a tabuľky spolu so svojimi popismi sú zarovnané na stred dokumentu. Text v tabuľke a popisy sú písané štandardným textom, teda Calibri 12b.

Názov stĺpca	Názov stĺpca	Názov stĺpca

Tabuľka 6-1 – Popis tabuľky (Calibri 12b)



Obrázok 6-1 – Popis obrázku (Calibri 12b)

Šprinty

Dokument je rozdelený do šprintov, ktorého šablóna bola vytvorená Bc. Miroslavom Vasiľom. Ukážka šablóny:

Šprint č. XX_(Cambria 26b)

Číslo šprintu	XX
Počet príbehov	XX
Začiatok šprintu	DD.MM.YYYY
Koniec šprintu	DD.MM.YYYY

Príbehy (Cambria 22b)

XXX
XXX
XXX

Každý šprint je ďalej rozdelený do kapitol Analýza, Návrh, Implementácia a Testovanie.

Zápis zo stretnutia

Zápisnica č. XX (Cambria 26b)

Téma	Prebratá téma
Miesto stretnutia	miestnosť
Dátum stretnutia	DD.MM.YYYY
Čas stretnutia	HH:MM – HH:MM
Prítomní	Zoznam prítomných
Zapisovateľ	Meno zapisovateľa
Overil	Meno overujúceho

Priebeh

Priebeh stretnutia v bodoch

Úlohy

ID	Riešiteľ	Popis	Dátum zadania	Termín
YY.XX	Meno riešiteľa	Popis úlohy	DD.MM.YYYY	DD.MM.YYYY

6.2.2 Konvencia písania zdrojových kódov

V tejto časti sa nachádza zoznam pravidiel definovaných pre písanie kódu.

Názvy

Názvy súborov tvoria podstatné mená v angličtine a začínajú veľkými písmenami. V jave musí byť názov triedy rovnaký ako názov súboru.

Meno súboru:

Main.java

Trieda:

```
Public Main  
{  
}  
}
```

Názvy funkcií, premenných sú v angličtine a začínajú malými písmenami. Názvy funkcií by mali byť tvoriť slovesá, aby sa z názvu dalo ľahko určiť čo daná funkcia robí.

```
getInstance();  
load();
```

Premenné sú pomenované podstatným menom, ktoré ich vystihuje. Nezačínajú znakmi podtržník alebo dolár (_, \$).

```
String name;  
int playerId;
```

Názvy balíkov tvoria malé písmená zložené z názvu krajiny podľa normy ISO 3166. Nasledujúce časti balíka sú tvorené podľa vlastnej organizácie vnútornej štruktúry.

```
sk.fiit.jim  
sk.fiit.jim.init  
sk.fiit.jim.gui
```

Názvy konštánt sa pomenovávajú veľkými písmenami, pričom slová sú oddelené podtržníkmi (_).

```
static final int MIN_WIDTH = 4;  
static final int MAX_WIDTH = 999;
```

Odsadenie

Kvôli prehľadnosti sa každé vnútro bloku zdrojového kódu odsadzuje. Na odsadenie zdrojového sa používajú tabulátory, pričom na jednu úroveň odsadenia je použitý jeden tabulátor. Odsadenie sa používa v rámci: definície tela triedy, metódy, konštruktora, v rámci blokov uzavretých kučeravými zátvorkami ('{' a '}'), príkazov switch a case.

```
/**  
 * Odsadenie  
 */  
class Example  
{  
    int[] myArray = { 1, 2, 3, 4, 5, 6 };  
    int theInt = 1;  
    String someString = "Hello";  
}
```

```

double aDouble = 3.0;

void foo(int a, int b, int c, int d, int e, int f)
{
    switch (a)
    {
        case 0:
            Other.doFoo();
            break;
        default:
            Other.doBaz();
    }
}

void bar(List v)
{
    for (int i = 0; i < 10; i++)
    {
        v.add(new Integer(i));
    }
}
}

```

Zložené zátvorky

Tieto zátvorky sú tvorené znakmi '{' a '}', pričom ohraničujú blok kódu. Pri deklarácii triedy, metódy, konštruktora, blokov, príkazov switch a case je každá zátvorka je umiestnená na nový riadok, pričom na tomto riadku sa nenachádza žiadny iný znak okrem tabulátora. Pri definícii poľa sa tieto zátvorky nachádzajú na tom istom riadku.

```

/**
 * Kučeravé zátvorky

```



```

*/

interface Empty
{
}

enum MyEnum
{
    UNDEFINED(0)
    {
        void foo()
        {
        }
    }
}

@interface SomeAnnotationType
{
}

class Example
{
    SomeClass fField = new SomeClass()
    {
    };

    int[] myArray = { 1, 2, 3, 4, 5, 6 };
    int[] emptyArray = new int[] {};

    Example()
    {
    }
}

```

```

void bar(int p)
{
    for (int i = 0; i < 10; i++)
    {
    }

    switch (p)
    {
        case 0:
            fField.set(0);
            break;

        case 1:
        {
            break;
        }

        default:
            fField.reset();
    }
}
}

```

Prázdné riadky a medzery

Prázdné riadky nasledujú po deklarácii balíka, pred importom balíka, po definovaní triedy, pred deklaráciou rovnakého typu (trieda - trieda, metóda - metóda).

Medzery sa používajú pri viacnásobnej deklarácii premenných, medzi parametre metódy, medzi kľúčovými riadiacich štruktúr a ich zátvorkami, medzi výrazy vo for cykle, pred a po binárnom operátore.

Komentovanie zdrojového kódu

Komentáre sú rozdelené do troch typov komentárov:

- Riadkové – začínajú od znakov „//“ a končia na konci riadku
- Blokové – začínajú znakmi „/*“ a končia znakmi „*/“
- Dokumentačné – začínajú znakmi „/**“ a končia znakmi „*/“

Jednoriadkové komentáre sa môžu nachádzať na tom istom riadku ako kód, viacriadkový komentár sa píše nad daný kód, ktorý komentujeme.

Dokumentačné komentáre musia písať hneď pred deklaráciu triedy, rozhrania, konštruktora, metódy alebo premennej. Tento komentár sa skladá z hlavného opisu, za ktorým nasleduje tag sekcia. Všetky komentáre sú písané v anglickom jazyku.

```
/**
 * An example for comment formatting. This example is meant to illustrate
 the various possibilities offered by <i>Eclipse</i> in order to format
 comments.
 */

package mypackage;

/**
 * This is the comment for the example interface.
 */
interface Example
{
    // long comment with whitespace that should be split in multiple
    // line comments in case the line comment formatting is enabled
    int foo3();

    // void commented() {
    // System.out.println("indented");
    // }

    // void indentedCommented() {
    // System.out.println("indented");
}
```

```
// }

/* block comment on first column */
int bar();

/*
 *
 * These possibilities include: <ul><li>Formatting of header
 * comments.</li><li>Formatting of Javadoc tags</li></ul>
 */
int bar2(); // long comment that should be split in multiple line
           // comments in case the line comment formatting is enabled

/**
 * The following is some sample code which illustrates source
 * formatting within javadoc comments:
 *
 * <pre>
 * public class Example
 * {
 *     final int a = 1;
 *     final boolean b = true;
 * }
 * </pre>
 *
 * Descriptions of parameters and return values are best appended at
 * end of the javadoc comment.
 *
 * @param a
 *
 *         The first parameter. For an optimum result, this
 *
 *         should be an odd number between 0 and 100.

```

```
* @param b
*
*         The second parameter.
*
* @return The result of the foo operation, usually within 0 and
*         1000.
* /
int foo(int a, int b);
}
```

6.3 Podporné prostriedky

Na manažment projektu sa využíva nástroj JTrack, ktorý umožňuje zadávanie a sledovanie úloh na projekte, čiže v prípade vyvíjania metodikou vývoja SCRUM plánovanie šprintov a vytváranie backlogu.

K vývoju vzhľadom na výber hráča implementovanom v Java sa využíva integrované vývojové prostredie Eclipse, rozšírené nástrojom na správu nástroja manažmentu verzií Mercurial Eclipse. Na manažment verzií sa teda využíva nástroj Mercurial, ku ktorému je prístup priamo z integrovaného vývojového prostredia pomocou spomenutého rozšírenia. K vývoju sú využívané aj automatizované testy pomocou technológie JUnit.

Komunikácia v rámci tímu je riešená viacerými spôsobmi. K individuálne komunikácií medzi členmi a nie celým tímom sa využívajú aplikácie typu Instant Messaging. Napríklad ICQ, Skype alebo Google Talk. Globálna komunikácia celého tímu prebieha pomocou emailu a diskusnej skupiny vytvorenej pomocou Google Groups. Emaily na globálnu komunikáciu sú písané normovaným spôsobom.

TO: [riešiteľ problému] // využíva sa z dôvodu určenia riešiteľa problému

CC: fiit-tim-04@googlegroups.com //využíva sa z dôvodu informovania celého tímu

BCC: [] //nevyužíva sa

Priority: [HIGH, NORMAL,LOW] // využíva sa z dôvodu určenia závažnosti problému

Subject: TP – [text] //predmet problému

Opis problému

Očakávania

Attachment: //prílohy potrebné k riešeniu problému

7 Manažment projektu

7.1 Manažment verzií

Pre manažment verzií v projekte bol zvolený nástroj Mercurial. Mercurial je distribuovaný systém pre podporu revízií a verziovania zdrojových kódov a iných dokumentov. V tíme využívame server a lokálne úložiská s kópiami zdrojových kódov, ktoré predstavujú klientov.

Server pre manažovanie verzií máme zriadení na webovej službe bitbucket.org. Služba ponúka vytvorenie centrálneho úložiska pre projekty, ktoré sú prístupné na stiahnutie pre všetkých používateľov služby. Vytvárať zmeny na serveri však môžu len schválení používatelia. Úložisko sa nachádza na adrese <https://bitbucket.org/xpagaca/team-04>. V úložisku sa nachádzajú zdrojové súbory a ďalšie dokumenty, ktoré sú predmetom verziovania.

Manažment verzií sa vo všeobecnosti skladá z nasledujúceho postupu:

1. Vytvorenie lokálnej kópie úložiska (pozri postup Clone)
2. Zvolenie vetvy pre práce (pozri výber Vetvy)
3. Úprava súboru v lokálnom úložisku
4. Potvrdenie zmien (pozri postup Commit)
5. Vloženie zmien na server do centrálneho úložiska (pozri postup Push)
6. Stiahnutie aktuálnej verzie z centrálneho úložiska (pozri postup Pull)
7. Aktualizácia súborov v lokálnom úložisku (pozri postup Update)
8. Riešenie konfliktov

Pre prehľad v projekte boli vytvorené tri typy vetiev:

- **Master** – obsahuje zdrojové súbory, ktoré sú otestované. Zdrojové súbory v tejto vetve sú 100% funkčné, program je stabilný a kedykoľvek prezentovateľný.
- **Default** – slúži ako úložisko pre prvé vloženie zmien od členov tímu. V tomto úložisku sa riešia prípadne konflikty. Otestované súbory sa vyberajú do vetvy Master.
- **Vetvy jednotlivých členov** – vytvárajú si členovia tímu podľa potreby (napríklad v prípade rozvíjania špeciálnej funkcie). Vetvy sú pomenované podľa funkcie, ktorá sa v nich rozvíja.

Prístup do úložiska

Úložisko je prístupné tromi spôsobmi :

- Priamou návštevou úložiska na serveri bitbucket.org na adrese <https://bitbucket.org/xpagaca/team-04>. Súbory je možné manažovať prostredníctvom webového rozhrania.
- Úprava súborov v repozitári, ktorý sa nachádza na lokálnom disku.
- Prístupom do vzdialeného úložiska prostredníctvom protokolov Mercurial.

Zoznam použitých klientov

Nástroje, ktoré využívame na verziovanie prostredníctvom systému Mercurial:

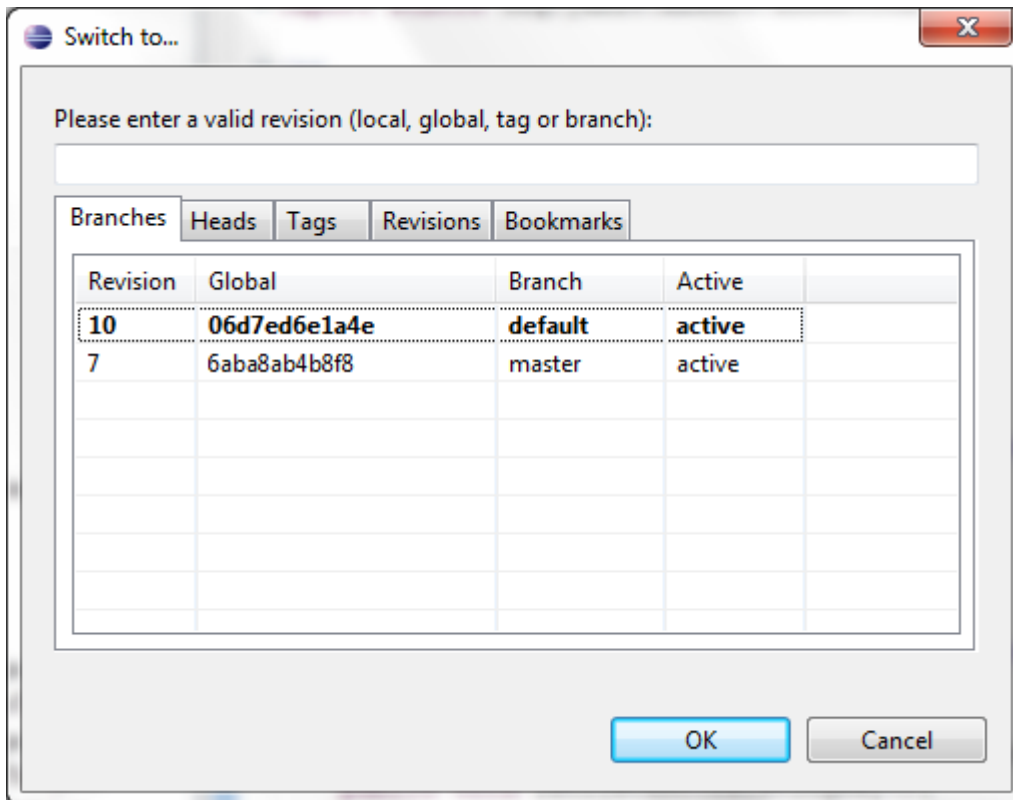
- TortoiseHg, dostupný na adrese:
<http://tortoisehg.bitbucket.org/download/index.html>
- zásuvný modul do Eclipse – MercurialEclipse, ktorý je prístupný v Eclipse prostredníctvom Help -> install new software

Pre zmeny v kóde, ich revíziu a verziovanie sa využíva zásuvný modul v Eclipse. Nástroj TortoiseHg sa používa pre verziovanie dodatočných súborov v úložisku.

Výber vetvy

Na začiatku úpravy zdrojových kódov je potrebné vybrať si aktuálnu vetvu, s ktorou sa bude pracovať. Výber vetvy pre prácu prebieha podľa nasledujúcich krokov:

1. V Eclipse zobrazíme formulár pre prepínanie vetiev (Obrázok. 7-1). Klikneme na projekt a vyberieme možnosť Team -> Switch to.
2. Dvojklikom vyberieme vetvu s ktorou chceme pracovať.
3. Aktuálna vetva je vyznačené s tučným písmom.



Obrázok 7-1 – Výber vetvy

Clone

Pre prácu s projektom je potrebné najprv vytvoriť lokálnu kópiu zdrojových kódov. Zdrojové kódy sú prístupné komukoľvek. Postup vytvorenia lokálnej kópie v nástroji Eclipse sa skladá z nasledujúcich krokov:

1. V Eclipse zobrazíme formulár pre klonovanie projektu zo serveru : file -> import -> mercurial -> clone existing project , potvrdíme stlačením tlačidla Next.
2. Do otvoreného formuláru (Obrázok. 1) napíšeme do položky URL adresu centrálného úložiska: <https://bitbucket.org/xpagaca/team-04>.
3. Vyplníme prihlasovacie meno (položka Username) a heslo (položka Password) (v prípade, že sú pridelené manažérom verziovania).
4. Proces vytvorenia lokálneho úložiska spustíme stlačením tlačidla Finish.
5. Projekt, je skopírovaný do aktuálneho Workspace.

Obrázok 7-2 – Formulár pre klonovanie projektu

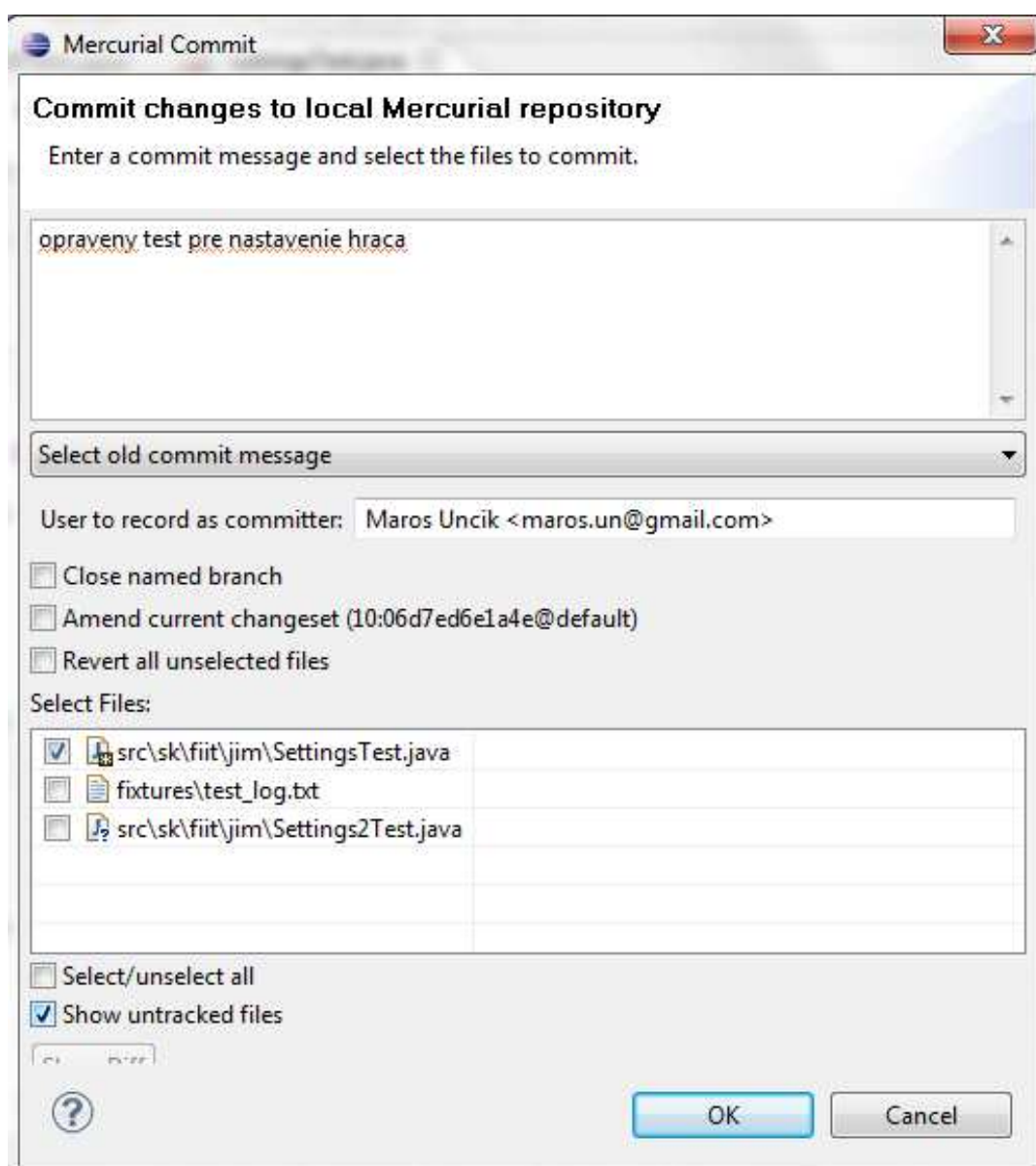
Commit

Po úpravách zdrojových kódov alebo dokumentov je potrebné pridať zmeny do lokálneho úložiska. Pre pridanie zmien platia nasledujúce pravidlá:

- pridávať sa môžu len zmeny, ktoré nie sú chybné
- pri každom pridaní zmeny sa musí zapísať stručná správa o zmenách, ktorá je súčasťou, pridávania zmeny
- zmeny sa pridávajú do lokálneho úložiska priebežne, po vykonaní dostatočne veľkej zmeny

Postup pre vloženie zmeny do lokálneho úložiska je nasledovný:

1. Pravým tlačidlom myši sa klikne na projekt.
2. Vyberie sa možnosť Team > Commit, čím sa zobrazí formulár pre vloženie zmien.
3. Napíše sa správa o zmene. Správa sa píše v slovenskom jazyku v súvislých vetách.
4. V časti Select Files sa vyberú súbory, ktoré chceme vložiť do lokálneho úložiska.
5. Vloženie sa potvrdí tlačidlom Ok a zmeny sa vložia do lokálneho úložiska.



Obrázok 7-3 - Vloženie zmeny do lokálneho úložiska

Push

Pre zdieľanie zmien s ostatnými členmi tímu, je potrebné vykonané zmeny vložiť do centrálného úložiska. Zmeny do centrálného úložiska je možné vložiť až po ich vložení do lokálneho úložiska. Zmeny sa vložia nasledujúcim spôsobom:

1. Pravým tlačidlom myši sa klikne na projekt.
2. Vyberie sa možnosť Team > Push, čím sa zobrazí formulár pre vloženie zmien do centrálného úložiska.
3. Vo formulári je vidieť zmeny pridané do lokálneho úložiska. Zmeny sa vložia do centrálného úložiska kliknutím na tlačidlo Ok.

Pull

Stiahnutie aktuálnej verzie z centrálného úložiska je potrebné vykonať pred začatím práce na projekte a pred každým vložením zmien do centrálného úložiska. Stiahnutie aktuálnej verzie sa vykoná prostredníctvom nasledujúceho postupu:

1. Pravým tlačidlom myši sa klikne na projekt
2. Vyberie sa možnosť Team > Pull, čím sa zobrazí formulár pre stiahnutie zmien z centrálného úložiska
3. Zmeny sa stiahnu z centrálného úložiska kliknutím na tlačidlo Ok

Update

Aktualizácia zdrojových súborov je potrebná po každom stiahnutí novej verzie z centrálného úložiska, nakoľko sa nevykonáva automaticky. Aktualizácia súborov sa vykoná prostredníctvom nasledujúceho postupu:

1. Pravým tlačidlom myši sa klikne na projekt
2. Vyberie sa možnosť Team > Update
3. Následne prebehne aktualizácia projektu

7.2 Manažment dokumentácie

7.2.1 Dokumentácia k inžinierskemu dielu

Súčasťou softvéru je aj dokumentácia, ktorá opisuje samotný softvér a spôsob jeho vývoja. Pomocou dokumentácie sa dá zistiť stav softvéru a projektu v jednotlivých fázach vývoja, dôvody jednotlivých rozhodnutí a fungovanie jednotlivých častí softvéru.

Tímový projekt Robocup 3D je vyvíjaný pomocou metodiky vývoja SCRUM. Dokumentácia k inžinierskemu dielu vyvíjaným touto metodikou vývoja sa vyznačuje inou štruktúrou dokumentácie ako je známa pri metodike vývoja inkrementálnym alebo iteratívnym spôsobom. Vývoj je dokumentovaný po jednotlivých šprintoch a v každom šprinte je zaznamenaná analýza, návrh, implementácia a testovanie každého jedeného príbehu šprintu.

Dokumentácia je výsledkom práce celého tímu, ale v konečnej fáze má byť konzistentná obsahom aj formou v celej svojej dĺžke. Z tohto dôvodu je vhodné, aby za dokumentáciu zodpovedal práve jeden člen tímu. V tíme The A Team je za dokumentáciu zodpovedný Bc. Miroslav Vasiľ. Jeho úlohou však nie je písanie celej dokumentácie, ale vytvorenie kostry dokumentácie, integrácia jednotlivých častí dokumentácie, revízia dokumentácie a formálna úprava. Za účelom jednoduchšej integrácie jednotlivých častí dokumentácie je vytvorený formát názvu súborov vytváraných ostatnými členmi tímu.

[číslo šprintu]-[príbeh]-[etapy príbehu]-[meno priezvisko]-[dátum]

Príklad:

01-Vytvoriť základnú kostru dokumentácie-ANIT-Miroslav Vasiľ-10.10.2010

Tento príklad zobrazuje názov súboru, ktorý obsahuje časti dokumentácie patriace do šprintu číslo jedna s názvom príbehu Vytvoriť základnu kostru dokumentu. Súbor obsahuje časti dokumentácie pre analýzu, návrh, implementáciu a testovanie. Autorom je Miroslav Vasiľ a dátum vytvorenia je 10.10.2010.

Jednotlivé časti dokumentácie sú vytvárané počas celého šprintu. Ukladané sú v repozitári a koncom šprintu sú integrované do dokumentácie, zrevidované a formálne upravené do finálnej podoby.

7.2.2 Dokumentácia k riadeniu

Dokumentácia k riadeniu obsahuje ponuku, plánovanie projektu, ktoré je pravidelne aktualizované, úlohy členov tímu, zápisnice zo stretnutí, metodiky, manažment projektu a preberacie protokoly. Dokumentácia k riadeniu rovnako ako dokumentácia k inžinierskemu dielu nie je vytváraná jedným členom tímu, ale za jednotlivé časti sú zodpovední rôzni členovia tímu. Aktualizácie plánovania a zápisnice sú aktualizované pri každom šprinte resp. stretnutí.

7.2.3 Zápisnice

Zápisnice sú vytvárané po každom stretnutí. Na začiatku každého stretnutia je určený zapisovateľ, ktorý po stretnutí vytvorí zápisnicu, ktorá je ešte overená iným členom tímu. Formálnu úpravu zápisnice vykonáva Bc. Miroslav Vasiľ. Zápisnica obsahuje základne informácie o téme, dátume a mieste stretnutia a prítomných členov na stretnutí. Ďalej obsahuje informácie o priebehu stretnutia, zhodnotenie úloh a informácie o ďalšom plánovaní.

7.3 Manažment úloh

Na manažment úloh bol vybraný webový systém JTrack, ktorý je dostupný na webovej adrese www.tonyb.sk. Systém je postavený všeobecne, nie je úzko zameraný na metodiku SCRUM. Napriek tomu poskytuje základné nástroje pre manažment úloh v projekte.

Hoci systém nie je úzko špecializovaný na metóda SCRUM, práca so systém sme odvodili od postupov, ktoré sa v tejto metóde využívajú. Základné postupy môžeme zhrnúť do piatich bodov:

1. Na stretnutí k projektu sa po naplánovaní úloh na ďalší šprint, vloží nová úloha do systému pre manažment úloh podľa postupu.
2. Úlohe sa prideli identifikátor v závislosti od šprintu, určí sa priorita, zaznačí odhadovaný čas a určí sa člen, ktorý je zodpovedný za jej vykonanie (pozri postup (2)).
3. Ľubovoľný člen tímu môže pridať úlohu do systému.
4. Členovia zaznamenávajú prácu na úlohe podľa postupu.
5. Na najbližšom stretnutí k projektu sa úlohy vyhodnotia a vzniknuté ohodnotenie sa zaznačí v systéme JTrack.


Prihlásenie do systému JTrack

Prvým krokom je prihlásenie sa do systému:

1. Člen tímu navštíví adresu www.tonyb.sk a zadá svoj login a heslo (password) do webového formulára.
2. Stlačí tlačidlo Submit.

Vytvorenie úlohy

Vytvorenie úlohy môže vykonať každý z členov tímu, táto možnosť je však prístupná len používateľom, ktorí sú prihlásení do systému JTrack. Prvým krokom je prihlásenie sa do systému podľa postupu.

Člen tímu je prihlásený do systému. Následne je potrebné zvoliť položku New označenú ikonou  v stĺpci Action. Tým sa vyvolá formulár pre pridanie novej úlohy. Formulár obsahuje niekoľko položiek, ktoré sa vyplňajú pre každú úlohu. Jednotlivé položky sa vyplňajú podľa nasledujúcich pravidiel:

Položka	Pravidlo
Summary	Určí sa názov úlohy stanovený členom tímu, ktorý úlohu vytvoril.
Detail	Určí sa bližší a krátky opis úlohy. Stanovuje člen tímu, ktorý úlohu vytvoril.
Priority	Určuje prioritu úlohy. Vyberá sa jeden z piatich stupňov. Štandardne sa vyberá možnosť medium.
Estimate time	Odhadovaný čas, ktorý zaberie vypracovanie úlohy.
Spended time	Strávený čas na úlohe. Používa sa v prípade, že sa úloha presunula z predchádzajúceho šprintu. Vypĺňa sa celočíselne, alebo v tvare X.X, kde je potrebné použiť desatinnú bodku.
Performed by	Kto sa na úlohe podieľa, v prípade, že na úlohe pracujú viacerí.

Area of issue	Oblasť činnosti, do ktorej spadá úloha. Je možné vybrať jednu z 8 typov – Stretnutie, Plánovanie, Analýza, Návrh, Implementácia, Testovanie, Dokumentácia a Údržba.
Type of issue	Typ úlohy: Bug – opravenie chyby Task – bežná úloha Improvement – vylepšenie existujúcej úlohy New feature – nová funkcia
Assign to	Vyberie sa člen tímu zodpovedný za úlohu.
Notify by Email	Vyberú sa členovia tímu, ktorý budú o úlohe notifikovaný prostredníctvom emailu.
Attachment	Pridanie prílohy, ktorá má súvis s danou úlohou.

Tabuľka 7-1 - Pravidlá

Možnosť send email notification sa necháva zaškrtnutá, členovia tímu zaškrtnutí v časti Notify by Email budú notifikovaní prostredníctvom mailu. Formulár sa odošle kliknutím na tlačidlo Submit, čím sa zaznamená úloha v systéme.

Aktualizácia úlohy

Počas práce na úlohách každý člen tímu aktualizuje priebežne svoj stav úlohy v systéme JTrack. Aktualizuje sa stav práce a strávený čas. Prvým krokom je prihlásenie do systému podľa postupu.

1. Člen tímu zobrazí úlohy, ktoré sú mu priradené – v stĺci Assigned To Me klikne na počet úloh.
2. Vyberie z formulára úlohu, ktorú chce aktualizovať, tým že klikne na ID úlohy v stĺpci ID.
3. Zobrazí sa formulár pre aktualizáciu úlohy, ktorý sa vyplňa podľa nasledujúcich pravidiel:

Položka	Pravidlo
Priority	Určuje prioritu úlohy. Vyberá sa jeden z piatich stupňov. Štandardne sa vyberá možnosť medium. Pri aktualizácii sa priorita úlohy mení (zvyšuje, alebo znižuje) len na základe pokynov ScrumMastera.
Estimate time	Odhadovaný čas, ktorý bol zadáný pri vytvorení úlohy. Táto položka sa vyplňa len na základe pokynov ScrumMastera.
Spended time	Čas, ktorý člen tímu doposiaľ strávil na inkremente v úlohe.
Performed by	Kto sa na splnení úlohy podieľal.
Area of issue	Oblasť činnosti, do ktorej spadá úloha. Je možné vybrať jednu z 8 typov – Stretnutie, Plánovanie, Analýza, Návrh, Implementácia, Testovanie, Dokumentácia a Údržba. Hodnota musí byť zhodná s hodnotou, ktorá bola zadaná pri vytvorení úlohy.
Type of issue	Typ úlohy: Bug – opravenie chyby Task – bežná úloha Impovement – vylepšenie existujúcej úlohy New feature – nová funkcia Hodnota musí byť zhodná s hodnotou, ktorá bola zadaná pri vytvorení úlohy.
New status	Člen tímu vyberie jednu z troch možností: In-progress – ak je úloha v štádiu riešenia Resolved – ak je úloha považovaná za vyriešenú Chucked – ak bola úloha prehodnotená

	a ďalej sa na nej nepracuje, túto hodnotu môže nastaviť len ScrumMaster
Assign to	Vyberie sa člen tímu zodpovedný za úlohu. Mení sa iba v prípade, že sa zmenila zodpovedná osoba za úlohu.
Comment	Člen zapíše aktuálne poznámky k stavu úlohy tak, aby sa dalo identifikovať, v akom stave je daná úloha. Píše sa stručne v odrážkach.
Notify By Email	Vyberú sa členovia tímu, ktorý budú o úlohe notifikovaný prostredníctvom emailu.
Attachment	Pridanie prílohy, ktorá má súvis s danou úlohou.

Tabuľka 7-2 - Pravidlá

1. Povinnými položkami pri aktualizácii úlohy sú položky: Priority, Spended Time, New Status a Comment.
2. Možnosť send email notification sa necháva zaškrtnutá, členovia tímu zaškrtnutý v časti Notify by Email budú notifikovaný prostredníctvom mailu. Formulár sa odošle kliknutím na tlačidlo Submit, čím sa zaznamená úloha v systéme.

7.4 Manažment testovania

Pri testovaní softvérového výrobku využívame testovací pracovný rámec (framework) JUnit 4.0. Pre testovanie boli zvolené pravidlá popísané v tejto kapitole.

7.4.1 Tvorba testovacích tried a metód

Nasledujúca kapitola popisuje princípy tvorenia názvov tried a metód pri testovaní, rovnako popisuje aj obsah testovacích tried a metód. Konvencie sa odvíjajú od zaužívaných praktík v testovaní a rovnako aj štandardných konvenciách pri tvorbe programov v jazyku Java.

Testovacia trieda

Testovacia trieda sa používa na testovanie konkrétnej triedy. Pre jej pomenovanie a obsah patrí nasledovné:

1. Testovacia trieda musí dediť z triedy `junit.framework.TestCase`.
2. Testovacia trieda, ktorá slúži na pomenovanie triedy s názvom `[Class]`, musí byť pomenovaná `[Class]Test` a musí sa nachádzať v súbore `[Class]Test.java`.
3. Názov tried sa vytvára výhradne v anglickom jazyku.
4. Testovacia trieda obsahuje iba požiadavky na testovanie, vzťahujúce sa na triedu, ktorá je testovaná (v jednej testovacej triede sa testuje iba jedna trieda).
5. Každá testovacia trieda musí mať na začiatku napísaný komentár s informáciou čo sa testuje.

Metóda v testovacej triede

Testovacia trieda sa skladá z metód. Pre ich pomenovanie a obsah platí nasledovné:

1. Metóda v testovacej triede musí byť pomenovaná ako `[name_of_test]`, kde `[name_of_test]` je názov testu.
2. Názov testu sa vytvára výhradne v anglickom jazyku.
3. Návratová hodnota metódy je `void`.
4. Metóda je deklarovaná ako `public`.
5. Pred každou metódou je potrebné uviesť `@Test` a prípadne ďalšie argumenty, podľa pravidiel pracovného rámca JUnit.

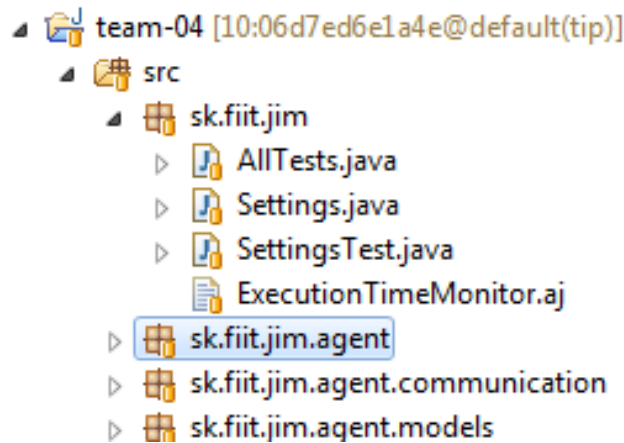
Množina testov

Množina testovacích tried je zoskupenie všetkých testovacích tried.

1. Množina testov musí byť pomenovaná `AllTest.java`.
2. Množina testov musí dediť z triedy `junit.framework.TestSuite`.
3. V každom balíku sa nachádza jedna trieda `AllTest`, ktorá obsahuje zoznam všetkých testov balíka.
4. Po vytvorení testu je potrebné pridať názov novej testovacej triedy do triedy `AllTest`. To zabezpečí, aby sa nový test spustil pri testovaní všetkých súčastí produktu.

Všetky testy v podobe testovacích tried nachádzajúce sa v projekte musia zodpovedať nasledujúcim pravidlám:

1. Testovacie triedy patria do rovnakého balíka ako testovaná trieda.
2. Testovacia trieda sa fyzicky nachádza pri testovanej triede. Umiestnenie testov je zobrazené na Obrázku 7-4



Obrázok 7-4 – Umiestnenie testovacích tried

7.4.2 Vytvorenie testu

Test sa vytvára ihneď po vytvorení novej triedy. Test sa vytvára vo vývojom prostredí Eclipse na základe nasledujúcich pravidiel:

1. Vytvorí sa nová testovacia trieda: File -> New -> JUnit Test Case. Zobrazí sa formulár na pridanie testovacej triedy.
2. Pri pomenovaní triedy (položka Name) sa postupuje ako je uvedené v časti Testovacia trieda.
3. Zvyšné možnosti vo formulári sú nepovinné.
4. Testovacia trieda sa vytvorí kliknutím na tlačidlo Finish.

Týmto postupom sa vytvorí testovacia trieda, ktorá neobsahuje žiadne metódy. Metódy je potrebné dopísať do novovytvoreného súboru. Pri vytvorení metód sa postupuje podľa postupu uvedeného v časti Metóda v testovacej triede. Na Obr. 2. je príklad vytvoreného testu pre testovanie triedy Settings. V príklade je použitá metóda `assertTrue` triedy `Assert`, ktorá overuje či argument predstavuje hodnotu `true`, dátového typu `Boolean`.

```

package sk.fiit.jim;

import static org.junit.Assert.assertThat;

/**
 * SettingsTest.java
 *
 * @Title      Jim
 * @author     $Author: marosurbanec $
 */
public class SettingsTest{

    @Test
    public void setAndPull(){
        Settings.setValue("someKey", true);
        assertTrue(Settings.getBoolean("someKey"));
    }
}

```

Obrázok 7-5 – Testovacia trieda SettingTest

Pri vytváraní metód pre testovanie sa používajú metódy triedy Assert a Assume. Najdôležitejšie metódy triedy Assert a ich použitie je nasledovné (Tabuľka 7-3):

Trieda	Použitie
assertArrayEquals	Porovná dve polia či sú ekvivalentné, ak je test nepravdivý, vyhodí výnimku.
assertEquals	Porovná dve hodnoty, či sú ekvivalentné, ak je test nepravdivý, vyhodí výnimku.
assertFalse	Overí, že vstupný argument má hodnotu false typu Boolean, ak je test nepravdivý, vyhodí výnimku.
assertNotNull	Overí, že vstupný argument nie je null, ak je test nepravdivý, vyhodí výnimku.
assertNotSame	Overí, že dve hodnoty nie sú rovnaké, ak je test nepravdivý, vyhodí výnimku.
assertNull	Overí, že vstupný argument je null, ak je test

	nepravdivý, vyhodí výnimku.
assertSame	Overí, že dve hodnoty sú rovnaké, ak je test nepravdivý, vyhodí výnimku.
assertTrue	Overí, že vstupný argument má hodnotu true typu Boolean, ak je test nepravdivý, vyhodí výnimku.
fail	Test skončí neúspechom za každých okolností s danou správou.

Tabuľka 7-3 - Metódy triedy Assert a ich použitie

Najdôležitejšie metódy triedy Assume a ich použitie je nasledovné (Tabuľka 7-4):

Trieda	Použitie
assumeNoException	Používa sa k predpokladu, že operácia nevyhodí výnimku.
assumeNotNull	Používa sa k predpokladu, že argument nemá hodnotu null.
assumeThat	Používa sa k predpokladu, že jeden argument zodpovedá podmienkam druhého argumentu.
assumeTrue	Ak sa zavolá s argumentom, ktorý má hodnotu false, test zastane a bude ignorovaný.

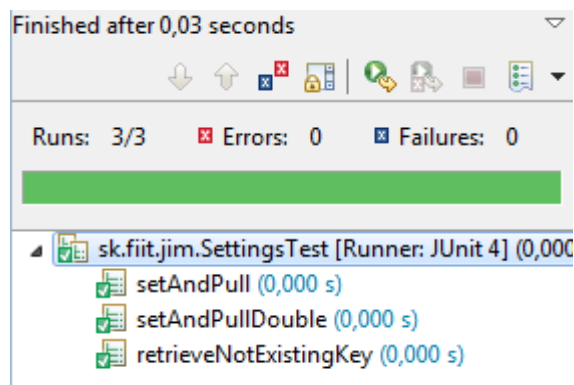
Tabuľka 7-4 - Metódy triedy Assume a ich použitie

7.4.3 Spustenie testu

Testy, ktoré sa vytvárajú sú dynamické a preto pre zistenie výsledku testu je ich potrebné spustiť. Testy sa spúšťajú pomocou nasledujúceho postupu:

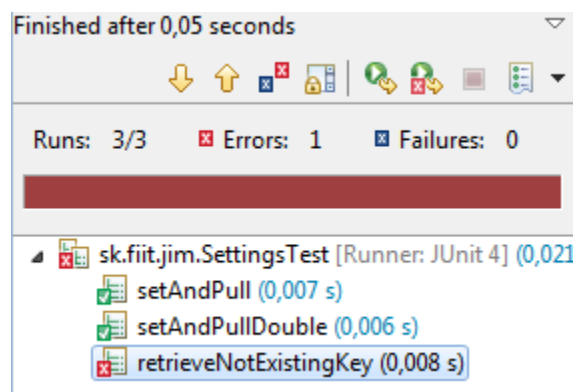
1. Pravým tlačidlom sa klikne na test, ktorý chceme spustiť.
2. Z vyskakovacieho menu vyberieme Run as -> JUnit test.
3. Test sa spustí a zobrazí sa formulár s výsledkom.
4. Týmto postupom je možné spustiť jeden test, prípadne množinu testov.

Test prebehol úspešne, ak všetky metódy testovacej triedy sú vo formulári znázornené zelenou farbou resp. je pri nich zobrazený zelená ikona (Obrázok 7-5).



Obrázok 7-5 – Úspešný výsledok testu

V prípade, že test neprebehol správne, je pri danej metóde zobrazená červená farba (Obrázok. 7-6).



Obrázok 7-6 – Neúspešný výsledok testu

7.5 Manažment zmien

Proces manažmentu zmien vyplýva z použitej metodiky vývoja Scrum. Táto metodika produkuje prototypy v každom šprinte. V každom šprinte dochádza k zvýšeniu funkcionality alebo k jej zmene. Požiadavky na pridanie funkcionality alebo jej zmenu sú zaznamenané do product backlogu. Požiadavku na pridanie funkcionality alebo jej zmenu môže zaznamenať každý účastník vývoja. Product owner na základe technickej, časovej alebo inej možnosti zrealizovať požadované návrhy zamietne alebo akceptuje. Akceptované návrhy sú zaznamenané v release backlogu a stávajú sa cieľmi daného šprintu. Zamietnuté návrhy sa nestávajú automaticky nezrealizovateľmi požiadavkami. Môžu byť naplánované do ďalších šprintov z dôvodu zlepšenia technických, časových alebo iných možnosti.

8 Preberacie protokoly

Táto kapitola obsahuje preberacie protokoly k jednotlivým kontrolným bodom.

Preberací protokol

Projekt:

- Robocup 3D

Predmet odovzdania:

- Dokumentácia k inžinierskemu dielu v rozsahu 53 strán
- Dokumentácia k riadeniu v rozsahu 73 strán

Odovzdávajúci:

- The A Team

Preberajúci:

- Ing. Marián Lekavý, PhD.

Dátum odovzdania:

- 04.11.2010

Ing. Marián Lekavý, PhD.

The A Team