

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Tímový projekt RoboCup 3D

Dokumentácia k riadeniu projektu

Študijný odbor: Softvérové inžinierstvo, Informačné systémy

Predmet: Tímový projekt

Akademický rok: 2012/2013

Téma: RoboCup 3D

Číslo tímu: 15

Názov tímu: A55 Kickers

Vedúci tímu: Ing. Marián Lekavý, PhD.

Kontakt: tptim15@googlegroups.com

Bc. Matej Červeňák

Bc. Jaroslav Grega

Bc. Martin Gregor

Bc. Gábor Nagy

Bc. Matúš Ondrejko

Bc. Filip Sucháč

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Ponuka | 2 |
| 2.1 | Predstavenie tímu | 2 |
| 2.2 | Ponuka 1: RoboCup – tretí rozmer | 3 |
| 2.2.1 | Motivácia..... | 3 |
| 2.2.2 | Koncepcia riešenia..... | 3 |
| 2.3 | Ponuka 2: Odporúčanie pre inteligentnú TV..... | 4 |
| 2.3.1 | Motivácia..... | 4 |
| 2.3.2 | Koncepcia riešenia..... | 4 |
| 2.4 | Ponuka 3: Simulácia demonštrácie v meste | 5 |
| 2.4.1 | Motivácia..... | 5 |
| 2.4.2 | Koncepcia riešenia..... | 6 |
| 2.5 | Zoradenie tém podľa priority | 7 |
| 2.6 | Rozvrh členov tímu | 8 |
| 3 | Úlohy členov tímu | 9 |
| 3.1 | Dlhodobé úlohy | 9 |
| 3.2 | Autorstvo v dokumentácii k riadeniu..... | 9 |
| 3.3 | Autorstvo v dokumentácii k inžinierskemu dielu | 10 |
| 3.4 | Autorstvo na úlohách projektu..... | 11 |
| 4 | Podporné prostriedky..... | 14 |
| 4.1 | Komunikácia | 14 |
| 4.2 | Manažment projektu..... | 14 |
| 4.3 | Manažment verzií a zmien..... | 14 |
| 4.4 | Podporné prostriedky pre správu Gitbus..... | 14 |
| 5 | Používané metodiky | 15 |
| 5.1 | Manažment kvality | 15 |
| 5.1.1 | Refaktoring..... | 15 |
| 5.2 | Manažment rizík..... | 23 |
| 5.2.1 | Spracovanie chýb..... | 23 |
| 5.3 | Manažment rozvrhu a plánovania | 30 |
| 5.3.1 | Tvorba rozvrhu | 30 |
| 5.4 | Manažment podpory vývoja | 31 |
| 5.4.1 | Verziovanie zdrojového kódu | 31 |
| 5.5 | Monitorovanie projektu | 40 |
| 5.5.1 | Manažment prehľadok – Vytvorenie zápisnice o vykonanej prehliadke..... | 40 |
| 5.6 | Manažment komunikácie | 47 |

| | | |
|-------|---|-----|
| 5.6.1 | Proces zberu požiadaviek | 47 |
| 5.6.2 | Kroky zaradenia požiadavky do backlogu | 50 |
| 5.7 | Manažment tvorby dokumentácie | 51 |
| 5.7.1 | Tvorba písomnej technickej dokumentácie | 51 |
| 6 | Záznamy zo stretnutí | 52 |
| 6.1 | Zápis z 1. stretnutia | 52 |
| 6.2 | Zápis z 2. stretnutia | 54 |
| 6.3 | Zápis z 3. stretnutia | 57 |
| 6.4 | Zápis z 4. stretnutia | 60 |
| 6.5 | Zápis z 5. stretnutia | 65 |
| 7 | Preberacie protokoly | 7-A |

História zmien

Tabuľka 7.1: História zmien

| Por.č. | Vykonaná zmena | Dátum zmeny | Vykonal | Platné od verzie |
|--------|--|--------------|-------------|------------------|
| 1. | Vytvorená kostra dokumentu | 9. 11. 2012 | M. Červeňák | 1.1 |
| 2. | Pridané časti: Úvod, Ponuka, Úlohy členov tímu, Podporné prostriedky, Manažment rozvrhu a plánovania, Preberacie protokoly | 10. 11. 2012 | M. Červeňák | 1.1 |
| 3. | Úprava údajov tabuľky 3.4 Úprava zápisu stretnutia 4 Pridanie metodiky Manažment komunikácie | 11. 11. 2012 | G. Nagy | 1.1 |
| 4. | Pridanie metodiky Manažment rizík | 11. 11. 2012 | J. Grega | 1.1 |
| 5. | Pridanie metodiky Monitorovania projektu | 11. 11. 2012 | M. Gregor | 1.1 |
| 6. | Pridanie metodiky Manažment kvality | 12. 11. 2012 | F. Sucháč | 1.1 |
| 7. | Pridanie metodiky Manažment tvorby dokumentácie | 12. 11. 2012 | M. Červeňák | 1.1 |
| 8. | Pridanie metodiky o zbere požiadaviek | 12. 11. 2012 | M. Ondrejko | 1.1 |
| 9. | Finálna úprava dokumentu pred 1. kontrolným bodom | 13. 11. 2012 | M. Červeňák | 1.1 |

1 Úvod

Tento dokument predstavuje dokumentáciu k riadeniu tímového projektu *RoboCup – tretí rozmer*, ktorý je vypracovávaný členmi tímu 15 s názvom *A55 Kickers*. Riešenie tímového projektu je obsahom kurzu Tímový projekt na FIIT STU v akademickom roku 2012/2013.

Dokument vytvoril tím 15 a spolu s ostatnými dokumentmi patrí medzi ich internú dokumentáciu. Slúži členom samotného tímu, ako i vedúcemu celého projektu. Takisto ho môžu využiť i ostatní čitatelia ako inšpiráciu pri riešení podobných projektov. Dokument obsahuje osem kapitol.

V druhej kapitole sa nachádza ponuka, ktorú tím vypracoval v počiatočných fázach predmetu Tímový projekt. Obsahuje projekty na ktoré, o ktoré sa náš tím zaujímal.

V tretej kapitole sú spísané úlohy a zodpovednosti jednotlivých členov tímu. Obsahuje zoznam dlhodobých manažérskych úloh členov ale i zodpovednosti za vypracovanie jednotlivých častí dokumentácie k inžinierskemu dielu a tohto dokumentu.

V štvrtej kapitole sú vymenované a opísané podporné prostriedky, ktoré tím využíva počas riešenia projektu.

V piatej kapitole sú spísané metodiky uplatnené a využívané v tomto projekte.

V šiestej kapitole sú vložené zápisy z pravidelných týždenných tímových stretnutí.

V siedmej kapitole sú prílohy a preberacie protokoly.

2 Ponuka

2.1 Predstavenie tímu

V tíme číslo 15 sa stretli ľudia s vedomosťami rôznych oblastí, ktoré nadobudli na rôznych inštitúciách a pracovných miestach. Tím vytvorený z nových známostí spája takto ľudí s rôznymi schopnosťami, skúsenosťami a zameraním.

Členovia nášho tímu sa môžu chváliť vysokoškolským vzdelaním prvého stupňa, získaným na univerzitách ako Slovenská technická univerzita v Bratislave, Masarykova univerzita v Brne a Žilinská univerzita, v odboroch Informatika, Aplikovaná informatika a Počítačové a komunikačné systémy a siete.

Užívateľské a vývojové skúsenosti máme s mnohými platformami a operačnými systémami ako Windows, Linux, Mac OSX, Android, iOS a Cisco IOS.

Viac členov nášho tímu sa zaoberalo, ako aj v rámci bakalárskeho projektu, tak aj v zamestnaní s webovými a databázovými technológiami. Novým výzvam v týchto oblastiach môžeme čeliť vďaka pokročilým zručnostiam: MySQL, PostgreSQL, Oracle, CMS systému Drupal, SOAP, Message Passing Service technológií.

S touto oblasťou sa spája samozrejme práca v skriptovacích jazykoch: Ruby, PHP, JavaScript; publikovacích jazykoch: HTML, XHTML, XML, JSON, CSS, SEO a návrh grafiky v Adobe Photoshop, Ilustrátor a Flash.

K návrhu a realizácii softvérových a informačných systémov využívame jazyk UML a CASE nástroje, ktoré ovládame na mierne pokročilej až pokročilej úrovni. Po úspešnom zvládnutí bakalárskych projektov, ale aj projektov z praxe máme skúsenosti v oblasti analýzy, návrhu, testovania a celkového vývoja softvéru.

Nechýbajú ani naše mierne pokročilé znalosti v sieťových a telekomunikačných technológiách, a to vďaka praktickým cvičeniam jedného nášho člena počas štúdie v odbore Počítačové a komunikačné systémy a siete.

Členovia nášho tímu majú bohaté skúsenosti v programovaní. Medzi ovládané programovacie jazyky patria Java, C, C#, Object C, z ktorých s jazykmi Java a C má väčšie skúsenosti každý člen. Začiatkové skúsenosti máme aj v správe a verziovaní zdrojových kódov s Git a SVN.

Okrem znalostí v odbore informatika máme v tíme ľudí s ukončeným odborným vzdelaním v odbore elektrotechniky s osvedčením §21.

Na základe prvotného zoznámenia a doterajších stretnutí si myslíme, že máme dobré predpoklady na kvalitné vypracovanie našich preferovaných tém v rámci predmetu Tímový projekt.

2.2 Ponuka 1: RoboCup – tretí rozmer

2.2.1 Motivácia

Hlavnou motiváciu nášho tímu je možnosť posunutia hraníc a výsledkov našej fakulty v prestížnom projekte, ktorý má dlhú tradíciu ako vo svete, tak aj na našej škole. Myslíme si, že v tejto disciplíne inteligentných robotov je vždy možnosťou a aj výzvou niekam sa ďalej posunúť.

Okrem očí informatika sa na problematiku pozeráme aj očami športovca, ktorý pre svoj úspech musí neustále pracovať a zlepšovať sa. Keďže v tejto disciplíne toho športovca tvorí niekoľko generácií poctivých študentov, veľmi radi by sme pokračovali v "hre" a zdokonaľovali nášho spoločného "hráča".

Veľmi nás zaujal aj pohľad na takúto transformáciu reálneho sveta do virtuálnej podoby. S nadšením sme čítali správy o dosiahnutých výsledkoch v snahe napodobnenia schopností človeka, ako aj o zatiaľ futuristických plánoch vývoja robotického tímu, ktorý by bol schopný poraziť tím skladaný z ľudí.

Ako ďalšiu motiváciu by sme spomenuli zlepšovanie sa v zaujímavých oblastiach nášho odboru. Myslíme tu najmä na vedecké disciplíny ako umelá inteligencia a robotika, ktoré sú zamerané od ich začiatkov na pomoc ľuďom v súkromných aj globálnych problémoch.

Ako tretiu vec chceme spomenúť genialitu vo fakulte, že tento projekt nás môže cez ciele športovca a súperenia obohacovať cennými vedomosťami. Vedomosťami, ktoré môžu byť nápomocné pri riešení iných problematík alebo výskumnej práce v tejto oblasti.

2.2.2 Koncepcia riešenia

Prvou úlohou na ceste k lepšiemu a kvalitnejšiemu agentovi je štúdium a analýza súčasného stavu projektu, ktorý je na fakulte riešený posledných trinásť rokov. Keďže najväčším nedostatkom projektu, podľa našich vedomostí, je jeho nekonzistentnosť a nekomplexnosť, našou prvotnou prioritou bude zistenie reálneho stavu a zamyslenie sa nad nápravou nedostatkov. Pokúsime sa teda dať do poriadku zdrojové kódy a prerobiť, resp. vytvoriť dôkladnú dokumentáciu, aby sme takto vytvorili lepšie podmienky pre vývoj naším tímom a aj tímami budúcich generácií.

Súčasne s refaktoringom kódu a dokumentácie začneme pracovať na vylepšovaní samotného agenta. Po prejdení dôkladnou analýzou súčasného stavu a zistení skutočností sa budeme zaoberať rôznymi vylepšeniami, ktoré agenta posunú vpred vo vývoji. Tieto nápady a vylepšenia skúsime zapracovať do podoby návrhu a implementovať. Každá implementovaná myšlienka musí prejsť testami, či naozaj spĺňa realistické správanie hráča futbalu, následne bude zapracovaná do komplexnej dokumentácie.

Cieľom je zamerať sa aj na vylepšenie a rozšírenie pohybov agenta novými schopnosťami, aby jeho pohyb pôsobil viac realistickejšie a takto dosiahol lepšie výsledky. Takýmto spôsobom skúsime priblížiť napodobnenie reálneho hráča.

Plánujeme sa inšpirovať aj prácou a radami minuloročného tímu. Pokúsime sa zrealizovať ich naznačené vylepšenia. Konkrétne napr. schopnosť otáčania hráča počas chôdze, ktorú plánujeme doplniť možnosťou zmeny polomeru otáčania.

Ďalšou úlohou bude vylepšiť spoluprácu agenta s loptou a celkový pohyb s loptou. Naša snaha by mala vyústiť do komplexného pohybu hráča s loptou, ako aj vylepšenie nahrávok. Bude potrebná aj lepšia a rýchlejšia orientácia hráča na ihrisku.

2.3 Ponuka 2: Odporúčanie pre inteligentnú TV

2.3.1 Motivácia

Každý z nás si určite pamätá na chvíle strávené sledovaním televízie. V dobách, v ktorých nebol internet tak rozšírený ako dnes, bolo pre nás toto médium bohatým zdrojom informácií a zábavy. Dnes máme internet dostupný zo svojich počítačov, notebookov, tabletov či mobilných telefónov, ktoré sa taktiež medzičasom stali súčasťou našich životov. Televízia sa však postupne tiež prepája s webom a tento projekt je pre náš tím šanca pozdvihnúť jej popularitu u ľudí uprednostňujúcich práve spomínané novšie technológie. Toto by malo hodnotu aj pre verných používateľov TV, ktorým by tento projekt výrazne vylepšil zážitok pri sledovaní programu. Pomocou personalizovaných odporúčaní by si používatelia mohli vybrať práve to, čo ich zaujíma z tej veľkej a rôznorodej ponuky TV.

Členovia tímu spĺňajú viaceré predpoklady nielen na úspešné zvládnutie projektu, ale aj na veľké prínosy k pôvodnému konceptu zadania. Tímová loď plná skúseností ako personalizácia vyhľadávania, odporúčanie dopytov pomocou sociálnych sietí v personalizovanom vyhľadávaní, tvorba webových a mobilných aplikácií, dizajn, analýza a návrh mnohých úspešne dokončených projektov vyvíjaných s podporou systému kontroly verzii projektu vidí výzvu v projekte Odporúčanie pre inteligentnú TV. Medzi úspešné projekty patrí Social Web Search, ktorý bol aj publikovaný v zborníku školskej konferencie IIT.src 2012 v sekcii 'Web Science and Technologies', VizRC iPadová aplikácia ktorej otcom je náš tímový kolega.

Dajte nám šancu vplaviť sa do mora možností a spoznať ešte nespoznané, ktoré táto téma ponúka. Chceme sa nie len priučiť novému kumštu ale priniesť a preskúmať nové možnosti personalizácie a odporúčania multimediálneho obsahu webu.

2.3.2 Koncepcia riešenia

Pre takýto typ projektu, ktorý je zameraný na odporúčania, je potrebné zanalyzovať záznamy o používateľoch, záznamy o ich sledovaní multimediálneho obsahu a opisy tohto obsahu. Výsledkom tejto analýzy bude zistenie, čo bude obsahovať model používateľa a ako ho najefektívnejšie uchovávať.

Keďže chceme odporúčať aj pre rôzne skupiny ľudí, bude potrebné si zvoliť metódu na zisťovanie podobnosti medzi používateľmi, pomocou ktorých ich budeme do skupín začleňovať. Tu by sme si tiež vybrali stratégiu, podľa ktorej budeme vyhodnocovať, čo budeme odporúčať. Teda či používateľovi odporučíme obsah, ktorý je podobný niektorému obsahu, ktorý ho zaujal doteraz (obsahovo založené odporúčanie), či mu odporučíme obsah, ktorý zaujal niekoho podobne zmýšľajúceho (kolaboratívne filtrovanie), alebo použijeme mix týchto stratégií. Pri odporúčaní budeme záznamy v profile zoraďovať podľa časovej priority a budeme brať do úvahy aj časové obdobie, z ktorého má používateľ záznamy.

Ďalej navrhujeme typy skupín. Skupiny budú buď tvorené okruhom používateľov, ktorí sú si vo všeobecnosti do istej miery podobní, alebo si určíme skupiny zamerané na konkrétne oblasti (napr. športovci, filmovo chtiví, sledovači spravodajstva atď.), z ktorých si prípadne bude môcť používateľ vybrať a tak ovplyvniť vývoj svojho profilu. Multimediálny obsah bude odporúčaný z oblastí filmov, hudby, videí, správ, vedeckých článkov, hier, aplikačných webových skladov (appstores) a bude rozdelený do rôznych kategórií.

Ak chceme odporúčať multimediálny obsah pre viacero typov zariadení (TV, PC, mobilné), musíme pre každé zvlášť navrhnuť klientske aplikácie, ktoré by naše odporúčania podávali používateľovi a zároveň by získavali ďalšie dáta do modelu používateľa. Analyzovali a zvolili by sme si čo najvhodnejšie spôsoby zobrazovania odporúčaní, aby bol používateľ čo najviac spokojný. Používateľ bude mať na výber z dvoch atraktívnych a jednoduchých rozložení prvkov rozhrania a to mriežkové rozloženie (grid layout) a kartové rozloženie (tab layout). Dáta do modelu používateľa môžu byť získavané interakciou používateľa s pôvodnými funkcionalitami zariadení, interakciou s našou klientskou aplikáciou, ktorá podáva odporúčania, alebo nami navrhnutou aplikáciou špeciálne pre tento účel (anketa, hra atď.). Používateľ v aplikáciách bude mať možnosť štatistického a chronologického náhľadu na svoj profil. V náhľade budú v časovej osi alebo v počte zachytené tagy a oblasti multimediálneho obsahu, ktoré používateľ navštívil.

Aby bolo riešenie kompletne, je potrebné navrhnuť komunikáciu medzi klientskymi aplikáciami na zariadeniach a serverovou aplikáciou s modelmi používateľov a hlavnou logikou tvorby odporúčaní. Na PC, prípadne mobilnom zariadení, je jedno z riešení spraviť klientsku aplikáciu tvorenú JavaScriptom v prehliadači, ktorý bude komunikovať so serverom pomocou JSON alebo XML. Pre TV by sme museli analyzovať možnosti komunikácie jeho softvéru.

2.4 Ponuka 3: Simulácia demonštrácie v meste

2.4.1 Motivácia

V dnešnej uponáhľanej dobe plnej demonštrácii a výtržnosti je vhodné byť na všetko pripravený. Keďže nie je vhodné robiť pokusy s ľuďmi, môžeme na to využiť simulácie. Pomocou simulácií rôznych typov demonštrácií, rôznych typov ľudí a poriadkových zložiek vieme približne odhadnúť, ako sa demonštrácia bude vyvíjať. Po získaní informácií zo simulácie vieme lepšie zabezpečiť organizáciu pri demonštráciách.

Túto tému sme si vybrali preto, lebo všetci sme už boli účastníkmi demonštrácií a naše zážitky z nich neboli najlepšie. Pri každej demonštrácii sa našlo veľa nezvládnutých častí, a to hlavne zo strany poriadkových zložiek. Poriadkové zložky neboli na svojom mieste alebo podcenili situáciu a bolo ich nedostatok. Keďže na strane demonštrantov sa nikdy nedá odhadnúť, ako sa bude demonštrácia vyvíjať a ako budú demonštranti reagovať na poriadkové zložky, je potrebné počítať s viacerými alternatívami. Simulovaním demonštrácie v dostatočnom predstihu pred reálnou demonštráciou by sme vedeli niektoré nedostatky na strane poriadkových zložiek odstrániť. Takto by sme vedeli predísť zbytočným zraneniam a veľkým škodám.

Keďže všetci členovia tímu máme rôzne skúsenosti z demonštrácií, našou prioritou by bolo vytvoriť takých agentov, ktorí by sa svojimi povahovými vlastnosťami podobali reálnym ľuďom. Tým by sme

dosiahli vernejšie kópie reálnych simulácií a na základe toho vedeli zabezpečiť lepšiu pripravenosť na demonštráciách.

2.4.2 Koncepcia riešenia

Simulovanie demonštrujúceho davu budeme simulovať pomocou agentov. Ide o multiagentový systém, ktorý by obsahoval dva hlavné typy agentov podľa zadania, protestantov a poriadkové zložky. Základom bude dobre navrhnuť správanie agentov, ktorý budú konať jednotlivo, ale aj skupinovo. Agenti budú môcť ovplyvňovať správanie iných agentov a reagovať na aktuálnu situáciu. Podkladom pre splnenie tejto požiadavky bude nutné analyzovať psychológiu davu a psychológiu človeka, jeho správanie a počínanie si vo väčších skupinách ľudí a zaužívané stratégie, prostriedky poriadkových zložiek používaných v demonštráciách. Týmto by sme zabezpečili čo najvyššiu vierohodnosť a využiteľnosť aplikácie.

Demonštrácií sa zväčša zúčastňuje veľký počet demonštrantov v rozsahu niekoľko stoviek až tisícov a preto ďalším kritériom bude minimalizovať výpočtové náklady vzhľadom na náročnosť simulácie všetkými dostupnými prostriedkami.

Ďalšími dôležitými bodmi bude:

- Široká parametrizácia programu vedie k väčšiemu množstvu situácií. Agentom priradíme dostatočnú škálu vlastností, na základe ktorých budú konať.
- Reálne a presné simulovanie situácií aby sme dostali kvalitné výsledky.
- Zbieranie a vyhodnocovanie dát počas simulácie, prípadne predpovedanie správania davu demonštrantov.

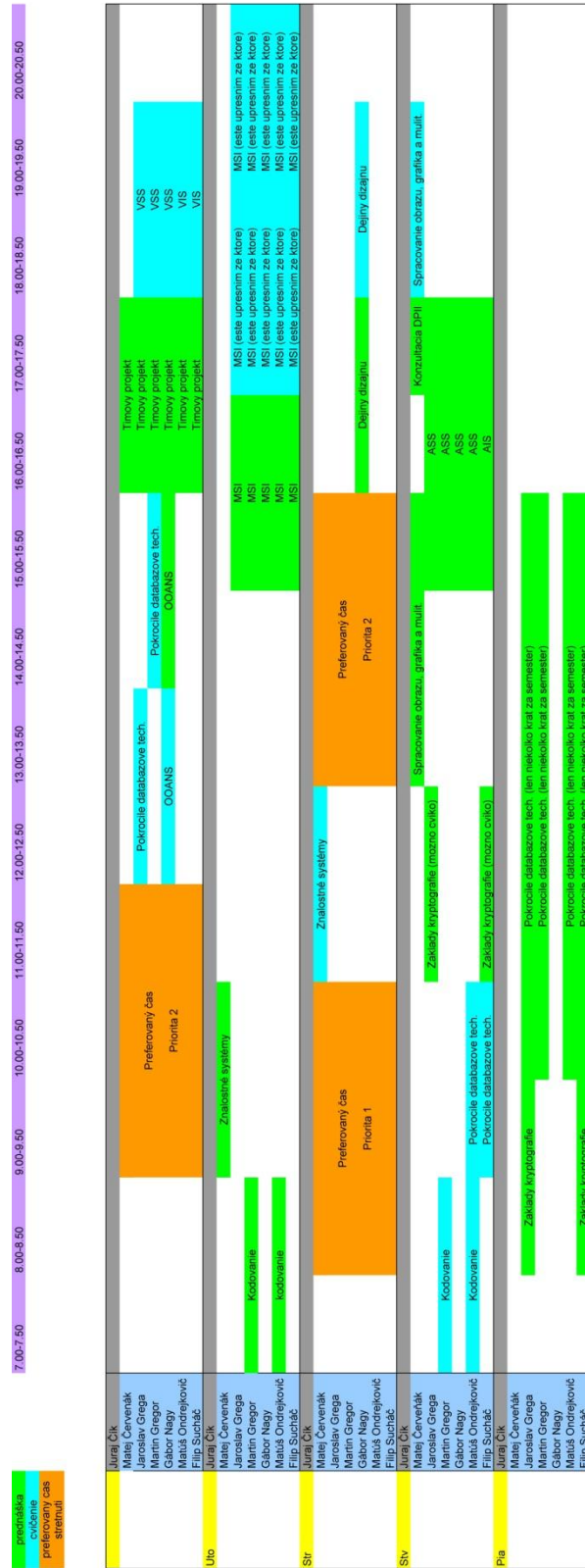
Hlavnou úlohou poriadkových zložiek je vytláčať dav z určeného priestoru pričom demonštranti sa snažia vzdorovať poriadkovým zložkám. Tento cieľ zahŕňa nasledujúce body, ktoré by v projekte bolo potrebné riešiť:

- Demonštrácie sú vedené lídrami. Záleží či ide o extrémistických vodcov alebo organizátorov s nenásilnými cieľmi demonštrácie. Lídri majú najväčší vplyv na pokojnosť a agresivitu demonštrujúceho davu a snažia sa ovplyvňovať jednotlivých ľudí.
- Demonštrácie odohrávajúce sa na jednom mieste alebo viacerých miestach kedy je potrebné dohliadať na presun demonštrantov.
- Pokojnosť demonštrácie nie je daná len vodcami protestu, ľudia môžu byť agresívny aj bez extrémistických vodcov.
- Vonkajšie vplyvy na protestujúcich. Príliš dlhý protest, zlé počasie môžu spôsobiť rezignáciu niektorých pretestujúcich alebo nečakanú zmenu správania.
- Dohliadanie na protestujúcich demonštrantov poriadkovými zložkami alebo ich zásah v prípade násilia a výtržností. Je potrebná adekvátna a neprehnaná reakcia poriadkových zložiek na situáciu.

2.5 Zoradenie tém podľa priority

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

2.6 Rozvrh členov tímu



3 Úlohy členov tímu

Táto kapitola obsahuje a popisuje dlhodobé a krátkodobé úlohy jednotlivých členov tímu. Takisto obsahuje aj autorstvo vytvorených dokumentácií a ich častí.

3.1 Dlhodobé úlohy

V tabuľke č. 3.1 je prehľad dlhodobých úloh členov tímu.

Tabuľka 3.1: Dlhodobé úlohy členov tímu

| Člen tímu | Úloha | Popis |
|----------------|--|--|
| Matej Červeňák | - Manažér rozvrhu a plánovania - Manažér dokumentácie | - Plánovanie úloh, dohľad nad termínmi. - Starostlivosť o dokumentáciu. |
| Jaroslav Grega | - Manažér rizík | - Identifikácia a riadenie rizík. |
| Martin Gregor | - Manažér monitorovania - Webmaster | - Monitorovanie aktuálneho progresu a vykonávanie prehliadok. - Starostlivosť o webovú stránku. |
| Gábor Nagy | - Manažér komunikácie - Vedúci tímu | - Riadenie komunikácie medzi členmi tímu. - Delegovanie úloh. |
| Matúš Ondrejko | - Manažér podpory vývoja | - Podpora vývoja, podporné prostriedky pre vývoj. |
| Filip Sucháč | - Manažér kvality | - Dohľad nad kvalitou produktu. |

3.2 Autorstvo v dokumentácii k riadeniu

V tabuľke č. 3.2 je rozdelenie práce na kapitolách dokumentácie k riadeniu. Ku každému členovi tímu sú priradené kapitoly, na ktorých sa podieľal vrátane percentuálneho podielu.

Tabuľka 3.2: Autorstvo v dokumentácii k riadeniu

| Autor | Kapitola | Podiel práce |
|----------------|------------------------------------|--------------|
| Matej Červeňák | 1 Úvod | 100 % |
| | 2 Ponuka | 17 % |
| | 3 Úlohy členov tímu | 100 % |
| | 4 Podporné prostriedky | 100 % |
| | 5.3 Manažment rozvrhu a plánovania | 100 % |
| | 5.7 Manažment tvorby dokumentácie | 100 % |
| | 6.1 Zápis z 1 stretnutia | 100 % |
| | 7 Preberacie protokoly | 100 % |
| Jaroslav Grega | 2 Ponuka | 17 % |
| | 5.2 Manažment rizík | 100 % |
| Martin Gregor | 2 Ponuka | 17 % |
| | 6.2 Zápis z 2 stretnutia | 100 % |

| | | |
|----------------|------------------------------|-------|
| Gábor Nagy | 2 Ponuka | 17 % |
| | 5.6 Manažment komunikácie | 100 % |
| | 6.3 Zápis z 3 stretnutia | 100 % |
| | 6.4 Zápis z 4 stretnutia | 100 % |
| Matúš Ondrejko | 2 Ponuka | 17 % |
| | 5.4 Manažment podpory vývoja | 100 % |
| Filip Sucháč | 2 Ponuka | 17 % |
| | 6.1 Manažment kvality | 100 % |
| | 6.5 Zápis z 5 stretnutia | 100 % |

3.3 Autorstvo v dokumentácii k inžinierskemu dielu

V tabuľke č. 3.3 je rozdelenie práce na kapitolách dokumentácie k riadeniu. Ku každému členovi tímu sú priradené kapitoly, na ktorých sa podieľal vrátane percentuálneho podielu.

Tabuľka 3.3: Autorstvo v dokumentácii k inžinierskemu dielu

| Autor | Kapitola | Podiel práce |
|----------------|--|--------------|
| Matej Červeňák | 1 Úvod | 100 % |
| | 2 Plán projektu na zimný semester | 100 % |
| | 3 Šprinty – základný opis a tabuľky | 100 % |
| | 3.1.3 Projekt od konkurenčného tímu | 100 % |
| | 3.1.5 Vytvorenie nového pohybu | 100 % |
| | 3.1.6.1 Analýza tímu Androids | 100 % |
| | 3.1.7 Analyzovanie RoboCup wikipedie | 100 % |
| | 3.1.8.2 Analýza tímu rUNSWift | 100 % |
| | 3.2.1 Návrh celkovej práce počas projektu | 100 % |
| | 3.2.4 Tvorba dokumentácie | 100 % |
| Jaroslav Grega | 3.1.8.5 Analýza tímu magmaOffenburg | 100 % |
| | 3.1.9.2 Analýza pohybov | 100 % |
| Martin Gregor | 3.1.8.6 Analýza tímu Austin Villa | 100 % |
| | 3.1.1 Webová stránka | 100 % |
| | 3.1.2 Server na RoboCup | 100 % |
| | 3.1.4 Spustenie hráča na server | 100 % |
| | 3.1.10 Git | 100 % |
| | 3.2.3 Analýza zdrojových kódov z minulého roka | 100 % |
| Gábor Nagy | 3.1.6.2 Analýza tímu High 5 | 100 % |
| | 3.1.8.3 Analýza tímu beeStambul | 100 % |

| | | |
|----------------|-------------------------------------|-------|
| | 3.1.11 Manažérsky softvér | 100 % |
| Matúš Ondrejko | 3.1.8.1 Analýza tímu Karachi Koalas | 100 % |
| | 3.1.9.2 Analýza pohybov | |
| | 3.2.2 Import a kríženie hráčov | 100 % |
| | 3.2.6 Návod na používanie Git-u | 100 % |
| Filip Sucháč | 3.1.6.3 Analýza tímu TÍM 17 ŽIJE... | 100 % |
| | 3.1.8.4 Analýza tímu Nexus3D | 100 % |
| | 3.1.9.1 Analýza robota ASIMO | 100 % |
| | 3.2.5 Návod na písanie kódu | 100 % |

3.4 Autorstvo na úlohách projektu

V tabuľke č. 3.4 je rozdelenie práce na projektových úlohách. Ku každému členovi tímu sú priradené jeho úlohy, na ktorých sa podieľal vrátane percentuálneho podielu a pridelených bodov.

Tabuľka 3.4: Autorstvo na úlohách projektu

| Člen tímu | Šprint | Úloha | Podiel práce | Body |
|----------------|--------|--|--------------|------|
| Matej Červeňák | 1 | Rozbehať server na RoboCup. | 11,5 % | 1,5 |
| | 1 | Získať projekt od konkurenčného tímu z posledného roka. | 100 % | 0,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať dokumentáciu a projekty z minulých rokov. | 100 % | 1 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Vytvoriť dokumentáciu aj so šablónou | 100 % | 3 |
| | | SPOLU | | |

| | | | | |
|----------------|---|---|--------|-----|
| Jaroslav Grega | 1 | Rozbehať server na RoboCup. | 42,3 % | 5,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |
| | 1 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | 100 % | 1 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |

| | | | | |
|--|--------------|--|--------|-------------|
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | 20 % | 4 |
| | SPOLU | | | 13,7 |

| | | | | |
|---------------|--------------|--|--------|-------------|
| Martin Gregor | 1 | Rozbehať webovú stránku. | 100 % | 8 |
| | 1 | Rozbehať server na RoboCup. | 11,5 | 1,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |
| | 1 | Rozbehať Git | 100 % | 0,5 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Git – importovanie a kríženie hráčov | 66 % | 2 |
| | 2 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | 20 % | 4 |
| | SPOLU | | | 19,2 |

| | | | | |
|------------|--------------|--|--------|-------------|
| Gábor Nagy | 1 | Rozbehať server na RoboCup. | 11,5 % | 1,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať dokumentáciu a projekty z minulých rokov. | 100 % | 1 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |
| | 1 | Rozbehať manažérsky softvér | 100 % | 5 |
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Git – importovanie a kríženie hráčov | 33 % | 1 |
| | 2 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | 20 % | 4 |
| | 2 | Analýza chôdze nášho (kríženého) hráča | 0 % | 0 |
| | SPOLU | | | 15,7 |

| | | | | |
|----------------|---|--|--------|-----|
| Matúš Ondrejko | 1 | Rozbehať server na RoboCup. | 11,5 | 1,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |

| | | | | |
|--|--------------|---|--------|-------------|
| | 1 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | 100 % | 1 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | 20 % | 4 |
| | 2 | Metodika na používanie Git-u | 100 % | 3 |
| | SPOLU | | | 12,7 |

| | | | | |
|--------------|---|---|--------|-----|
| Filip Sucháč | 1 | Rozbehať server na RoboCup. | 11,5 | 1,5 |
| | 1 | Spustiť hráča na server | 16,7 % | 0,5 |
| | 1 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | 100 % | 0,5 |
| | 1 | Študovať dokumentáciu a projekty z minulých rokov. | 100 % | 1 |
| | 1 | Študovať wiki k RoboCup-u. | 100 % | 0,5 |
| | 1 | Študovať zahraničné tímy a zápasy. | 100 % | 0,5 |
| | 1 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | 100 % | 1 |
| | 1 | Nainštalovať Git u seba. | 100 % | 0,5 |
| | 2 | Návrh celkovej práce počas projektu | 16,7 % | 0,7 |
| | 2 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | 20 % | 4 |
| | 2 | Koding guide (šablóna-metodika) | 100 % | 3 |
| | | SPOLU | | |

4 Podporné prostriedky

4.1 Komunikácia

Na komunikáciu používame konto na Google Groups, kde komunikujeme pomocou vytvorených tém, na ktoré odpovedáme, alebo prostredníctvom mailov. Ďalšou formou komunikácie je Gmail chat, ktorý využívame ako rýchly spôsob riešenia vzniknutých nejasností. V prípade potreby sa stretávame osobne.

4.2 Manažment projektu

Ako prostriedok na manažment projektu využívame nástroj Redmine. Tento nám umožňuje sledovať vývin všetkých úloh, umožňuje monitorovanie odhadovaného času potrebného na dokončenie jednotlivých úloh, prípadne na dokončenie aktuálneho šprintu. Vytvárajú sa tu *User Stories*, teda príbehy. K *User Stories* sa v každom šprinte vytvárané úlohy a následne sa pridávajú jednotlivým členom tímu.

4.3 Manažment verzií a zmien

Na manažment verzií a zmien využívame nástroj Gitbus. Vďaka nemu si vždy každý člen tímu môže stiahnuť z repozitára aktuálnu verziu projektu a na nej pracovať. Gitbus dokáže tiež riešiť súčasnú redundanciu zmien. Repozitár obsahuje všetky zdrojové kódy projektu.

4.4 Podporné prostriedky pre správu Gitbus

Ako podporné prostriedky pre správu Gitbus používame nástroj Git. Niektorí z nás využívajú pluginy do Eclipse alebo NetBeans, ktoré umožňujú prehľadne a jednoducho pracovať s naším repozitárom.

5 Používané metodiky

5.1 Manažment kvality

5.1.1 Refaktoring

5.1.1.1 Pojmy

- **NetBeans IDE** – integrované vývojové prostredie pre vývoj aplikácií v jazyku Java a mnohých iných.
- **Pach kódu** – štruktúra časti zdrojového kódu, ktorú je možné vylepšiť pomocou konkrétneho postupu refaktoringu.
- **Repozitár** – úložisko zdrojových kódov.
- **Synchronizované (zosynchronizované)** – aktualizované, aktuálne.
- **Záznam** – záznam vykonaného refaktoringu s poradovým číslom problému zo zápisu revízie. Jednotlivé záznamy sa nachádzajú v dokumente o refaktoringu.
- **Signatúra metódy** – názov metódy, počet a typy jej parametrov.
- **Wrapper** – trieda, ktorá obalí funkcionality inej triedy.
- **Getters a setters** – metódy zabezpečujúce zapuzdrenie atribútov.
- **RSA** – Rational Software Architect.
- **Redmine** – systém pre manažment projektov.

5.1.1.2 Prostredie

Vývojové prostredie používané na projektoch je NetBeans IDE.

5.1.1.3 Zodpovednosť

Za refaktoring je zodpovedný autor zdrojových kódov, ktoré majú byť refaktorované. V špeciálnych prípadoch je zodpovednosť daná inému vývojárovi/programátorovi.

5.1.1.4 Vstupy

Vstup tvorí zdrojový kód s potrebou refaktoringu, unit testy a zápis z revízie.

5.1.1.5 Zápis z revízie

Štruktúra zápisu revízie:

| Poradové číslo problému | Trieda | Metóda | Rozsah riadkov kódu | Identifikovaný problém (pach kódu) | Pokyny pre refaktoring (postup refaktoringu) |
|-------------------------|--------|--------|---------------------|------------------------------------|--|
| | | | | | |

5.1.1.6 Zdrojový kód

Je potrebné mať vo vývojovom prostredí nainportované príslušné zdrojové kódy, ktoré ich autor nedávno vytvoril. V prípade, že novovytvorený alebo zmenený bol už v minulosti odovzdaný do systému pre správu verzií zdrojových kódov, je potrebné mať zdrojové kódy zosynchronizované s repozitárom.

5.1.1.7 Unit testy

Je potrebné mať vo vývojovom prostredí nainportované, otvorené a zosynchronizované unit testy k príslušným zdrojovým kódom.

5.1.1.8 Spustenie a výstup unit testu

V prípade, že sa po spustení testov preukážu chyby, je potrebné vytvoriť záznam o chybách a chyby hlásiť. Tento krok je možné ukončiť až po úspešných výstupoch unit testov.

5.1.1.9 Vykonanie zmeny v kóde

Podľa zápisu zo stretnutia revízie je potrebné v zdrojovom kóde vykonať zmeny pre všetky identifikované problémy. Jednotlivé identifikované problémy sú v zápise vyjadrené pachom kódu, ktorý bol objavený pri revízii. Podľa pokynov pre refaktoring, vyjadrených konkrétnym postupom refaktoringu, je potrebné refaktorovať časť zdrojového kódu, pre ktorú bol problém identifikovaný. Správnosť zrefaktorovanej časti kódu je nutné overiť spustením príslušných unit testov k danej časti kódu.

5.1.1.10 Zdokumentovanie zmeny v kóde

Po vykonaní zmien pre konkrétny problém a po úspešnom výstupe unit testov je nutné zmeny zdokumentovať. Zdokumentovať znamená vytvoriť dokument o refaktoringu typu Microsoft Office Word 97-2003 s koncovkou „doc“.

Dokument musí byť pomenovaný podľa názvu dokumentu zápisu revízie, ktorým bol refaktoring riadený, pridaním textu „-refaktoringX“ pred koncovku súboru do názvu nového dokumentu, kde namiesto „X“ bude poradové číslo verzie dokumentu o refaktoringu. Príklad:

- pomenovanie dokumentu zápisu revízie:
robocup-jim-pohyb14-revizia2.doc
- pomenovanie dokumentu o refaktoringu:
robocup-jim-pohyb14-revizia2-refaktoring1.doc

Prvá strana dokumentu musí obsahovať nasledovné polia:

- Meno refaktorujúceho: <<meno a priezvisko autora dokumentu a zároveň autora refaktoringu zdrojového kódu>>
- Email: <<email refaktorujúceho>>
- Dátum: <<dátum ukončenia práce na refaktorovaní zdrojového kódu v formáte dd.MM.rrrr>>

Prvá strana dokumentu musí byť ukončená zlomom strany (Microsoft Word – Vložiť – Zlom strany).

Od druhej strany dokumentu nasledujú záznamy o jednotlivých refaktoringoch podľa poradia problémov zo zápisu revízie. Každý záznam začína nadpisom štýlu „Nadpis 1“, ktorého text je „Problém X“, kde na mieste „X“ bude uvedené poradové číslo problému zo zápisu revízie.

Text záznamu pod nadpisom záznamu musí byť v štýle „Normálny“. Prvé polia textu záznamu musia byť rovnaké ako v zápise revízie, teda:

- Trieda: <<obsah bunky „Trieda“ pre daný problém zo zápisu revízie>>
- Metóda: <<obsah bunky „Metóda“ pre daný problém zo zápisu revízie>>
- Rozsah riadkov kódu: <<...>>
- Identifikovaný problém (pach kódu): <<...>>
- Pokyny pre refaktoring (postup refaktoringu): <<...>>

Ďalšie polia záznamov o refaktoringu sa odlišujú na základe toho, aký konkrétny postup refaktoringu bol pri riešení daného problému použitý. V ďalších podkapitolách nasleduje zoznam jednotlivých postupov refaktoringu s poľami, ktoré musí obsahovať záznam o refaktoringu, ktorý dokumentuje použitie daného postupu refaktoringu.

5.1.1.10.1 Extrahovanie metódy (extract method)

- Signatúra novovytvorenej metódy: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.2 Absorbovanie metódy (inline method)

- Signatúry metód, ktoré absorbovali telo metódy: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.3 Nahradenie pomocnej premennej metódou (replace temp with query)

- Typ a názov pomocnej premennej: <<napr. „int pom“>>
- Signatúra novovytvorenej metódy: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.4 Zavedenie vysvetľujúcej premennej (introduce explaining variable)

- Typ a názov vytvorenej premennej: <<napr. „int pom“>>

5.1.1.10.5 Rozdelenie dočasnej premennej (split temporary variable)

- Typ a názov dočasnej premennej: <<napr. „int pom“>>
- Typy a názvy vytvorených premenných: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.6 Nahradenie metódy objektom metódy (replace method with method object)

- Názov vytvorenej triedy: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.7 Presunutie metódy (move method)

- Názov triedy, kam sa presunula metóda: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.8 Presunutie atribútu (move field)

- Názov triedy, kam sa presunul atribút: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.9 Extrahovanie triedy (extract class)

- Názov vytvorenej triedy: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>
- Signatúry metód presunutých do novovytvorenej triedy: <<čiarkami oddelený zoznam metód v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>
- Typy a názvy atribútov presunutých do novovytvorenej triedy: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.10 Absorbovanie triedy (inline class)

- Názov absorbujúcej triedy: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>
- Signatúry metód presunutých do absorbujúcej triedy: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>
- Typy a názvy atribútov presunutých do absorbujúcej triedy: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.11 Ukrytie delegáta (hide delegate)

- Signatúry vytvorených delegujúcich metód v triede poskytovateľa (server-a): <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>
- Signatúry odstránených prístupujúcich metód (accessors) z triedy poskytovateľa (server-a): <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.12 Odstránenie prostredníka (remove middle man)

- Signatúry vytvorených prístupujúcich metód (accessors) v triede poskytovateľa (server-a): <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>
- Signatúry odstránených delegujúcich metód v triede poskytovateľa (server-a): <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.13 Zavedenie cudzej metódy (introduce foreign method)

- Signatúra vytvorenej metódy: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.14 Zavedenie lokálneho rozšírenia (introduce local extension)

- Názov vytvorenej rozširujúcej triedy: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>
- Názov triedy, od ktorej rozšírenie dedí alebo je jej wrapper: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.15 Zapuzdrenie atribútu (encapsulate field)

- Typy a názvy atribútov, pre ktoré boli vytvorené getters a setters: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.16 Nahradenie dátovej hodnoty objektom (replace data value with object)

- Názov vytvorenej triedy nahradzujúcej dátovú hodnotu: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.17 Nahradenie poľa objektom (replace array with object)

- Názov vytvorenej triedy, ktorej objekt nahradil pole: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.18 Nahradenie čísla konštantou (replace magic number with symbolic constant)

- Typ a názov vytvorenej konštanty: <<napr. „int CONST“>>

5.1.1.10.19 Zapuzdrenie kolekcie (encapsulate collection)

- Signatúry vytvorených metód v triede využívajúcej kolekciu: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.20 Nahradenie typového kódu triedou/podtriedami (replace type code with class/subclasses)

- Názov vytvorenej triedy alebo vytvorených tried nahradzujúcich typové kódy: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.21 Nahradenie typového kódu stavom/stratégiou (replace type code with state/strategy)

- Názov vytvorenej triedy predstavujúcej stavový objekt: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>
- Názvy podtried stavového objektu: <<čiarkami oddelené názvy tried spolu s cestou, napr. „sk.fiit.robocup.Jim, sk.fiit.robocup.Test“>>

5.1.1.10.22 Nahradenie podtried atribútmi (replace subclass with fields)

- Názvy odstránených podtried: <<čiarkami oddelené názvy tried spolu s cestou, napr. „sk.fiit.robocup.Jim, sk.fiit.robocup.Test“>>

- Typy a názvy vytvorených atribútov: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.23 Dekompozícia podmienky (decompose conditional)

- Signatúry vytvorených metód: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.24 Konsolidácia výrazu podmienky (consolidate conditional expression)

- Signatúry vytvorených metód: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.25 Nahradenie vnorených podmienok (replace nested conditional with guard clauses)

- Signatúry vytvorených metód: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.26 Nahradenie podmienok polymorfizmom (replace conditional with polymorfism)

- Názvy vytvorených dediacich tried: <<čiarkami oddelené názvy tried spolu s cestou, napr. „sk.fiiit.robocup.Jim, sk.fiiit.robocup.Test“>>

5.1.1.10.27 Zavedenie „Null“ objektu (introduce null object)

- Názov „Null“ triedy: <<názov triedy spolu s cestou, napr. „sk.fiiit.robocup.Jim“>>

5.1.1.10.28 Premenovanie metódy (rename method)

- Signatúra metódy s novým názvom: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.29 Pridanie/odstránenie parametra (add/remove parameter)

- Signatúra metódy po pridaní/odstránení parametra: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.30 Parametrizácia metódy (parameterize method)

- Signatúra novej metódy: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>
- Signatúry odstránených metód: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>

5.1.1.10.31 Zavedenie objektu parametrov (introduce parameter object)

- Názov novej triedy pre objekt parametrov: <<názov triedy spolu s cestou, napr. „sk.fiiit.robocup.Jim“>>

5.1.1.10.32 Nahradenie konštruktora „factory“ metódou (replace constructor with factory method)

- Signatúra „factory“ metódy: <<v tvare „nazovMetody(Typ1 parameter1, Typ2 parameter2, ...)“>>

5.1.1.10.33 Nahradenie chybového kódu výnimkou (replace error code with exception)

- Názov vytvorenej triedy výnimky: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>

5.1.1.10.34 Extrahovanie podtriedy/nadtriedy/rozhrania (extract subclass/superclass/interface)

- Názov vytvorenej podtriedy/nadtriedy/rozhrania: <<názov triedy spolu s cestou, napr. „sk.fiit.robocup.Jim“>>
- Signatúry metód presunutých do podtriedy/nadtriedy/rozhrania: <<čiarkami oddelený zoznam metód v tvare napr. „nazovMetody(Typ1 parameter1, Typ2 parameter2), metoda2(Typ1 parameter1)“>>
- Typy a názvy atribútov presunutých do podtriedy/nadtriedy: <<čiarkami oddelený zoznam napr. „int pom, String text“>>

5.1.1.10.35 Rozuzlenie dedenia (tease apart inheritance)

- Diagram novej hierarchie dedenia a delegácie: <<obrázok typu PNG, vytvorený nástrojom RSA, maximálna veľkosť na 1 A4>>

5.1.1.10.36 Extrahovanie hierarchie (extract hierarchy)

- Názvy vytvorených tried: <<čiarkami oddelené názvy tried spolu s cestou, napr. „sk.fiit.robocup.Jim, sk.fiit.robocup.Test“>>
- Diagram extrahovanej hierarchie: <<obrázok typu PNG, vytvorený nástrojom RSA, maximálna veľkosť na 1 A4>>

V prípade, že pri použití konkrétneho postupu refaktoringu musel byť vykonaný aj iný ďalší postup refaktoringu, je potrebné to uviesť. Za poľami prvého postupu bude vtedy nasledovať nadpis štýlu „Nadpis 2“ s textom „Ďalší použitý postup X“, kde sa namiesto „X“ uvedie poradové číslo ďalšieho postupu refaktoringu použitého v danom zázname. Pod týmto nadpisom sa uvedie pole „Postup refaktoringu“ v ktorom sa identifikuje konkrétny postup. Ďalej sa uvedú polia daný postup určené zoznamom jednotlivých postupov refaktoringu (kapitoly 6.1.1.10.x).

Príklad záznamu:

Problém 14 (v štýle „Nadpis 1“)

Trieda: sk.fiit.robocup.Jim

Metóda: calculatePosition()

Rozsah riadkov kódu: 15 - 40

Identifikovaný problém (pach kódu): dlhá metóda (long method)

Pokyny pre refaktoring (postup refaktoringu): extrahovanie metódy (extract method)

Signatúra novovytvorenej metódy: calculateCornerPosition()

Ďalší použitý postup 1 (v štýle „Nadpis 2“)

Postup refaktoringu: presunutie metódy (move method)

Názov triedy, kam sa presunula metóda: sk.fiit.robocup.helpers.PositionCalculator

5.1.1.11 Výstup

Akceptovateľný výstup tohto procesu sú zrefaktorované zdrojové kódy, ktoré na koniec úspešne prešli unit testami, ktoré nehlásia žiadne chyby. Ďalší výstup tvorí dokument o refaktoringu, ktorý jeho autor po vypracovaní odovzdá do súborového systému v Redmine.

5.2 Manažment rizík

5.2.1 Spracovanie chýb

5.2.1.1 Úvod

Táto metodika slúži na určenie postupov, ktorými sa bude riadiť tím A55 Kickers počas práce na tímovom projekte RoboCup 3D.

5.2.1.2 Súvisiace metodiky

- Testovanie softvéru
- Zhotovenie dokumentácie

5.2.1.3 Zodpovednosti pri hlásení a zapisovaní chýb

V tejto časti metodiky sú popísané jednotlivé roly, ktoré zasahujú do manažmentu chýb a zodpovednosti účastníkov v tomto procese.

5.2.1.3.1 Vedúci projektu

Osoba, ktorá ma na starosti vedenie celého tímu. Ma prehľad o celom projekte, o všetkých členoch a vie povedať, kto je za čo zodpovedný.

- Nahlásenie chyby
- Pridelenie chyby
- Pridanie hlásenia v nástroji Redmine

5.2.1.3.2 Manažér plánovania

Osoba zodpovedná za dodržanie plánu pri odstraňovaní chýb.

- Návrh na pridanie ľudí k odstráneniu chyby

5.2.1.3.3 Tester

Člen tímu, ktorý má na starosti testovanie funkčnosti systému.

- Nahlásenie chyby
- Hľadanie chyby
- Testovanie opravenej chyby
- Hlásí, či je chyba vyriešená
- Pridanie hlásenia v nástroji Redmine
- Spravovanie chyby v nástroji Redmine

5.2.1.3.4 Vývojár

Člen tímu, ktorý systém vyvíja. Ma prehľad o celom systéme.

- Nahlásenie chyby
- Oprava chyby
- Posúdenie chyby, či sa dá chyba opraviť alebo nie
- Pridanie hlásenia v nástroji Redmine

- Spravovanie chyby v nástroji Redmine

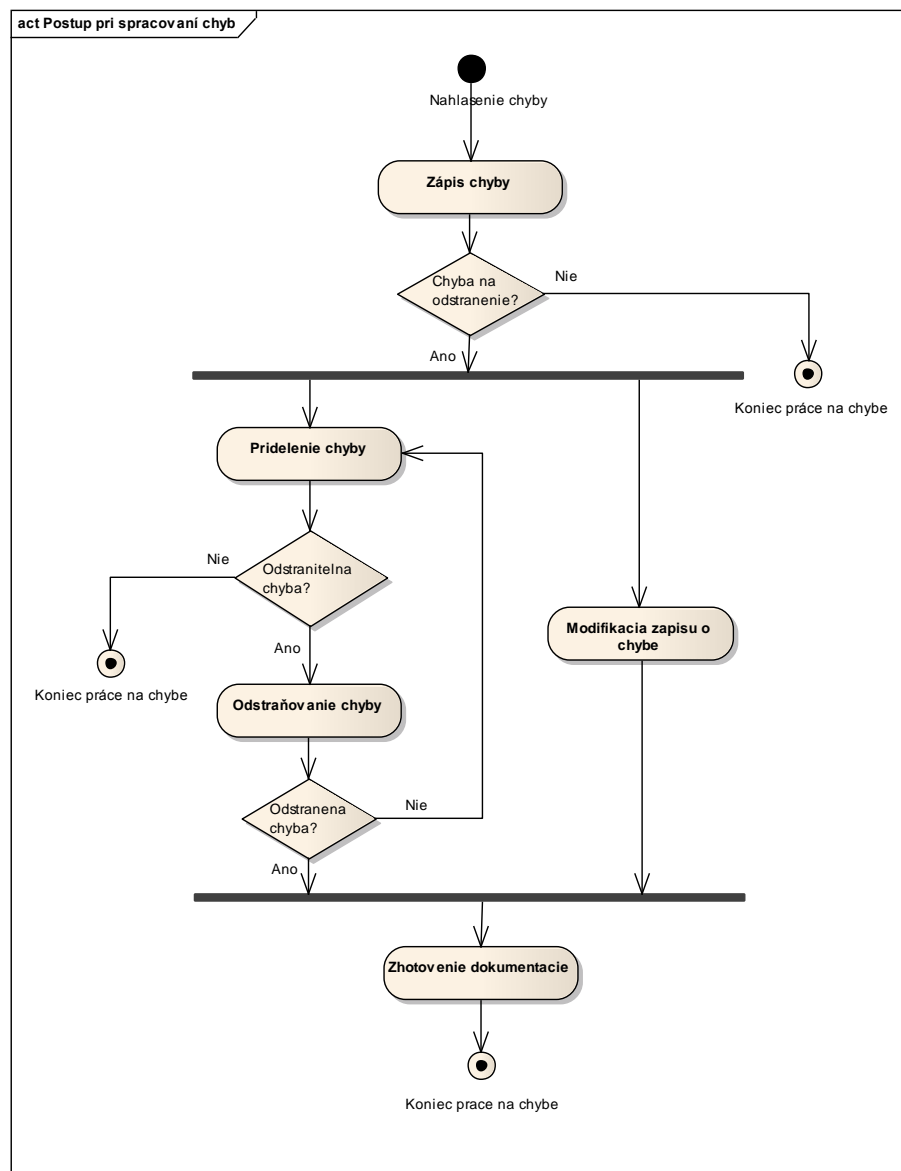
5.2.1.3.5 Manažér dokumentácie

Člen tímu, ktorý ma na starosti tvorbu dokumentácie po vyriešení chyby.

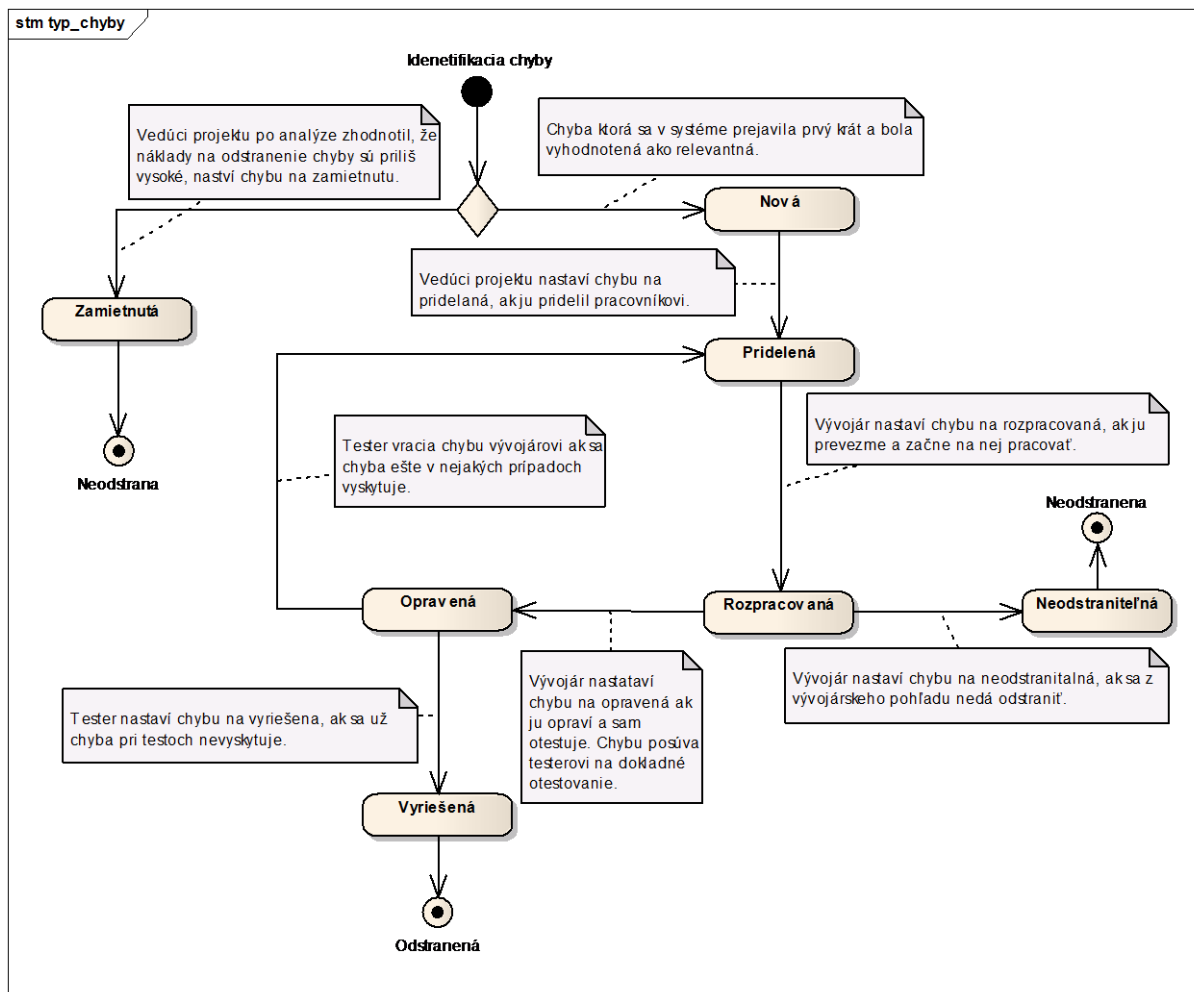
- Zaistenie vytvorenia dokumentácie

5.2.1.4 Postup pri spracovaní chyb

V tejto časti sú popísané procesy súvisiace s manažmentom chýb. Postup procesov spracovania chyby je zobrazený na obrázku 5.1. Každý proces spracovania je opísaný v nasledujúcich kapitolách. Súčasťou spracovania chyby je aj jej životný cyklus. Životný cyklus chyby je zobrazený a popísaný na obrázku 5.2.



Obrázok 5.1: Postup spracovania chyby



Obrázok 5.2: životný cyklus chyby

5.2.1.4.1 Zápis chyby

V tomto procese sa popisuje chyba a identifikuje jej stav.

Vstup – nahlásená chyba

Výstup – presný popis chyby a stav chyby

Zodpovednosť – vedúci projektu, vývojár, tester

Proces

1. Dôkladné opísanie chyby - pri akých akciách sa chyba vyskytuje
2. Overenie či je chyba v nástroji
 - Vloženie chyby a nastavenie stavu chyby na nová/zamietnutá
 - Doplnenie informácií o chybe

5.2.1.4.2 Pridelenie chyby (kto ju bude riešiť)

Tento proces popisuje postup pri pridelení chyby.

Vstup – nová/rozpracovaná chyba

Výstup – chyba pridelená konkrétnej osobe alebo chyba zamietnutá

Zodpovednosť – vedúci projektu, manažér plánovania

Proces

1. vedúci projektu spolu s manažérom plánovania chybu analyzujú
2. ak je chyba odstrániteľná (z pohľadu ceny) a je nová alebo rozpracovaná
 - pridelenie chyby pracovníkovi
3. Ak je chyba neodstrániteľná
 - nastavenie chyby na zamietnutá

5.2.1.4.3 Odstraňovanie chyby

Proces popisuje postup pri odstraňovaní chyby

Vstup – chyba pridelená

Výstup – chyba opravená/neodstrániteľná

Zodpovednosť – vývojár

Proces

1. naštudovanie analýzy a popisu chyby
2. zhodnotenie chyby, či je odstrániteľná (pohľadu vývojárskeho)
3. ak je odstrániteľná
 - práca na odstránení
 - Po odstránení nastaví chybu na opravená
4. Ak je neodstrániteľná nastaví chybu na neodstrániteľná

5.2.1.4.4 Overenie odstránenia chyby

Proces popisuje postup pri overovaní, či sa chyba odstránila.

Vstup – chyba opravená

Výstup – chyba vyriešená/pridelená

Zodpovednosť – tester

Proces

1. Tester otestuje opravenú chybu podľa metodiky Testovanie softvéru
2. Po overení chyby
 - nastaví chybu na vyriešená, ak sa už chyba nevyskytuje
 - nastaví chybu na pridelená, ak sa ešte v nejakých prípadoch chyba vyskytuje a vráti ju späť vývojárovi

5.2.1.4.5 Zhotovenie dokumentácie

Vstup – chyba vyriešená

Výstup – vytvorená dokumentácia k chybe

Zodpovednosť – manažér dokumentácie

Proces

- Postup procesu popisuje metodika Zhotovenie dokumentácie.

5.2.1.5 Vykonanie zápisu o chybe (pridanie/úprava chyby v nástroji)– Redmine

Táto časť popisuje ako vykonať zápis chybového hlásenia v nástroji Redmine. Dodržanie tohto postupu je veľmi dôležité, kvôli jednoduchej orientácii sa v chybách. Pri nedodržaní pomenovania, alebo vynechaní niektorých údajov, môže byť ťažké chybu identifikovať a následne odstrániť. Ďalším následkom môže byť hromadenie rovnakých chýb, čo je tiež neželaná situácia. Pridať hlásenie o novej chybe môže každý člen tímu. Pri vkladaní chyby a modifikovaní zápisu o chybe treba dodržiavať nasledujúci postup.

5.2.1.5.1 Opis chyby

Táto časť procesu hovorí o krokoch, ktoré sú spojené s opisom chyby.

Postup krokov pri opise chyby:

1. Objavenie chyby členom tímu, alebo nahlásenie chyby zákazníkom
2. Príprava všetkých podkladov k chybe
 - Opísať chybu, viď tabuľka 1 - Opis chyby
 - Pripraviť obrázok/video z monitora pri prejavení chyby
3. Zistiť či je sa chyba nachádza v nástroji Redmine

5.2.1.5.2 Overenie, či je chyba v nástroji

Táto časť procesu popisuje, ako overiť či sa už chyba nachádza v nástroji.

Postup krokov pri overení existencie chyby v nástroji:

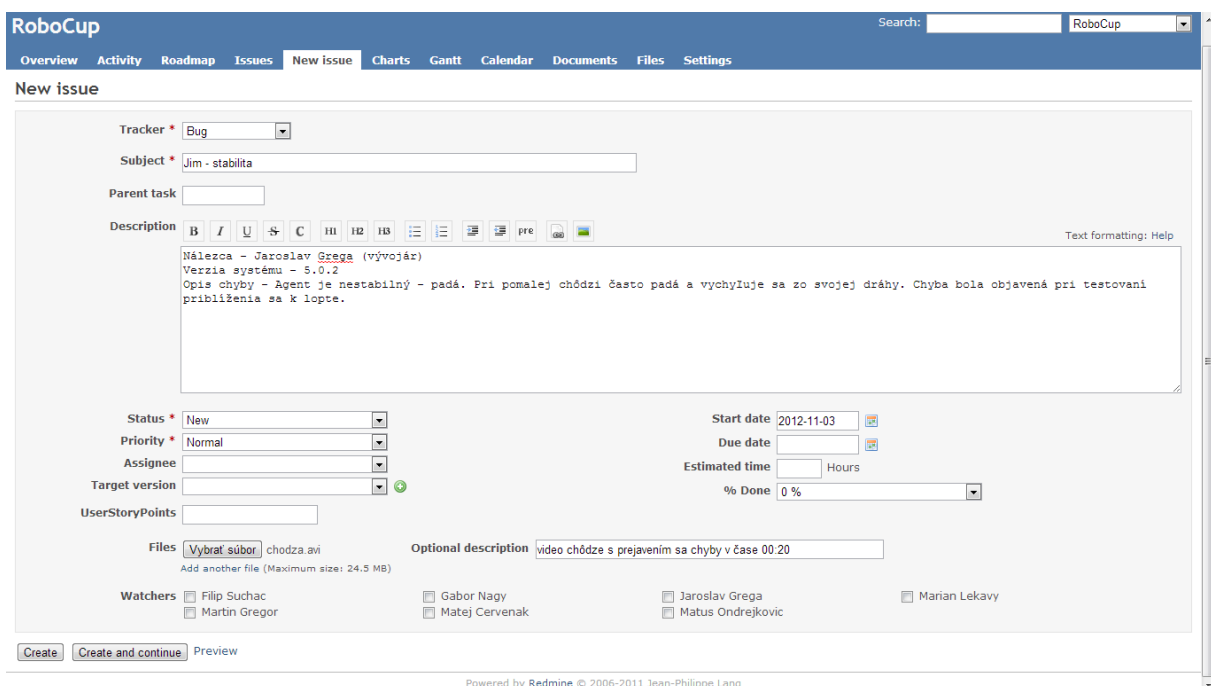
1. Prihlásenie sa do nástroja Redmine
2. Po prihlásení na pravej strane obrazovky zvolíť projekt
3. Filtrovanie chyby
 - Kliknúť na možnosť **Issues**
 - V pravej časti obrazovky pridať filter- kliknúť na combobox **add filter** a zvolíť možnosť **Tracker**
 - Následne po pridaní filtra v prvom comboboxe zvolíť možnosť **is** a v druhom **Bug**
 - Potvrdíť filtrovanie kliknutím na **Apply**
4. Po použití filtra sa zobrazia všetky chyby, ktoré sú v nástroji Redmine. Prihlásený používateľ prejde chyby a vyhladá podobnú chybu tej, ktorú chce pridať. Ak sa mu to nepodarí pokračuje vložení novej chyby. Ako vložiť novu chybu je popísané v časti **5.3 Vloženie novej chyby**. Ak podobnú chybu nájde, môže túto chybu modifikovať. Pre modifikáciu chyby sa riadi časťou **5.4 Modifikácia zápisu chyby**.

5.2.1.5.3 Vloženie novej chyby

Táto časť popisuje, ako bude vkladať novu chybu do nástroja tím A55 Kickers.

Postup krokov pri vkladaní novej chyby:

1. Prihlásenie sa do nástroja Redmine
2. Po prihlásení na pravej strane obrazovky zvolíť projekt
3. Pridanie novej chyby
 - Zvolíť možnosť **New issue**
 - Možnosť **Tracker** nastaviť na **Bug**
 - Možnosť **Subject** – vložiť vhodný názov chyby, vid' tabuľka 1
 - Možnosť **Description** – stručne popísať chybu (ako k nej došlo, ako sa prejavila, ...) vid' tabuľka 1
 - Možnosť **Status** nastaviť na **New**
 - Možnosť **Priority** nastaviť na **Normal**
 - Ak je obrázok/video monitora s chybou zvolíť možnosť **Files** a pridať obrázok
 - Ak je pridaný obrázok/video v možnosti **Optional description** výstižne popísať obrázok, vid' tabuľka 1
 - Ostatné možnosti nechať nevyplnené
 - Pre vloženie chyby zvolíť možnosť **Create**



Obrázok 5.3: Vloženie chyby v nástroji Redmine

5.2.1.5.4 Modifikácia zápisu chyby

V tejto časti je popísaný postup krokov pri modifikácii zápisu chyby v tíme A55 Kickers.

Postup krokov pri modifikácii chyby:

1. Prihlásenie sa do nástroja Redmine
2. Po prihlásení na pravej strane obrazovky zvolíť projekt
3. Modifikácia chyby
 - Zvolíť možnosť **Issues**

- Nájst chybu, ktorú je treba modifikovať podľa postupu **5.2 Overenie, či je chyba v nástroji**
- Pri nájdení chyby kliknúť na chybu v stĺpci **Subject**
- Vo vrchnej časti obrazovky nájsť možnosť **Update** a kliknúť na ňu.
- Doplniť potrebné informácie
 - o pre pridelenie pracovníka zvoliť možnosť **Assignee** a vybrať pracovníka (pracovník, ktorý ma chybu na starosti)
 - o pre zmenu statusu chyby voliť nasledovne
 - o New → Nová chyba
 - o Assigned → Pridelená
 - o Resolved → Opravená
 - o Feedback → Rozpracovaná
 - o Closed → Vyriešená
 - o Rejected → Zamietnutá/Neodstrániteľná
 - o pre pridelenie ďalších pracovníkov k chybe upraviť možnosť **Description** pridaním položky **Ďalší pracovníci** a dopísať k nej požadovaných pracovníkov
- Potvrdiť požadované zmeny kliknutím na možnosť **Submit**

Tabuľka 5.1: Príklady popisov

| Popis | Príklad popisu |
|----------------------|---|
| Opis chyby | Agent je nestabilný - padá. Pri pomalej chôdzi agent často padá a vychýľuje sa zo svojej dráhy. Chyba bola objavená pri testovaní priblíženia sa k lopte. |
| Subject | Jim – stabilita Jim – kop do lopty |
| Description | Nálezca – Jaroslav Grega (vývojár) Verzia systému – 5.0.2 Opis chyby – Agent je nestabilný - padá. Pri pomalej chôdzi agent často padá a vychýľuje sa zo svojej dráhy. Chyba bola objavená pri testovaní priblíženia sa k lopte. Ďalší pracovníci – Filip Sučač, Martin Gregor |
| Optionla description | Video chôdze s prejavením sa chyby v čase 00:20 |

5.3 Manažment rozvrhu a plánovania

5.3.1 Tvorba rozvrhu

Nakoľko sa pri tvorbe produktu riadime metodikou SCRUM, nemáme dopredu určený rozvrh projektu. Ten sa tvorí vždy pre konkrétny šprint. Táto tvorba prebieha vždy na prvom tímovom stretnutí v rámci šprintu.

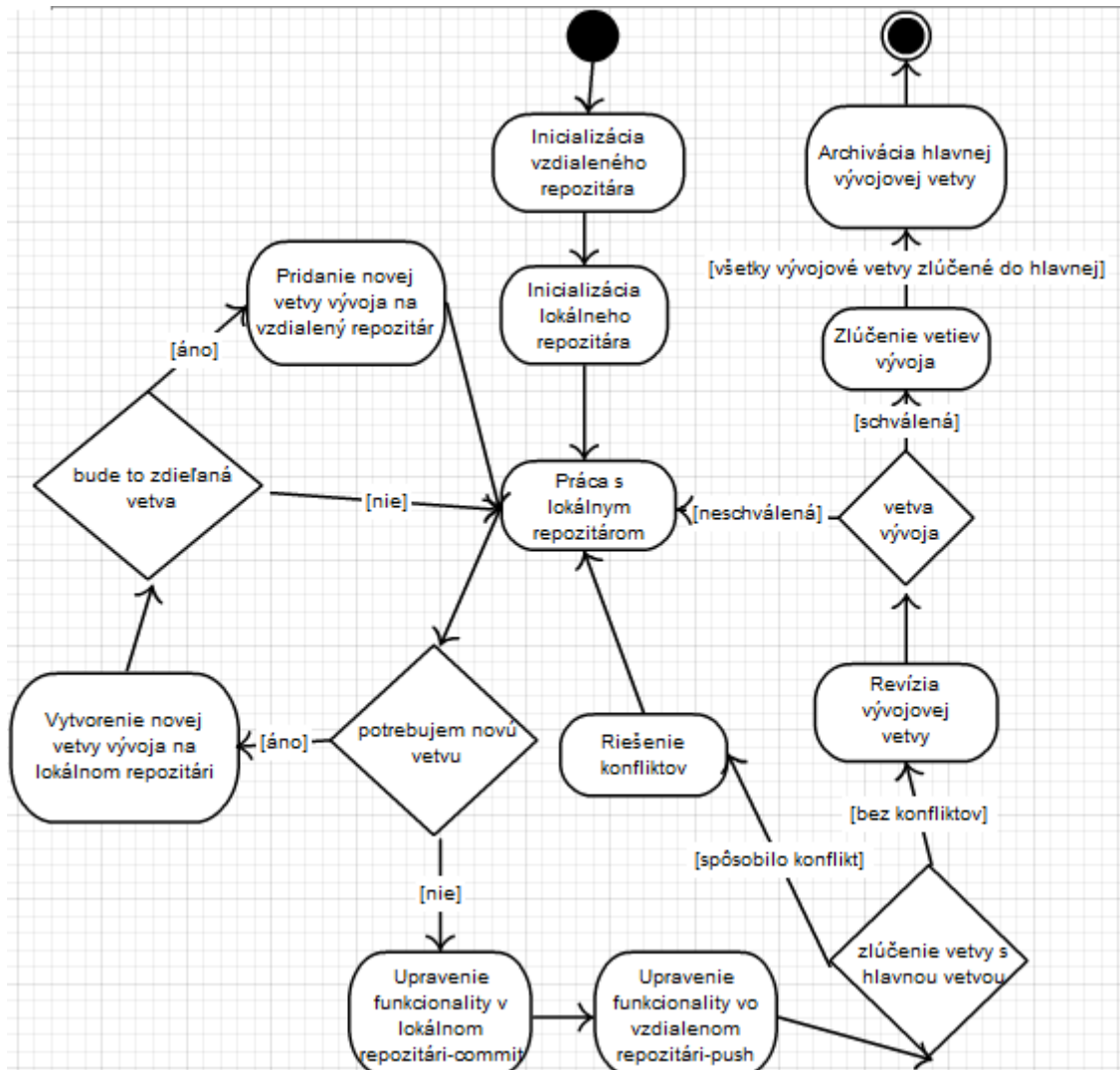
5.3.1.1 Postup tvorby rozvrhu

1. Zhodnotí sa predchádzajúci (končiaci) šprint. Všetky úlohy, ktoré sa členom tímu nepodarilo z tohto šprintu dokončiť, sa im automaticky prenášajú do nasledujúceho šprintu.
2. Na základe dohody medzi tímom a vedúcim tímu sa určí, ktoré používateľské príbehy (*User Stories*) budú vykonané.
3. Pre každý používateľský príbeh sa určí jedna zodpovedná osoba. V prípade, že sa v tíme nenájde dobrovoľník, pridelí túto rolu vedúci tímu.
4. Tím následne identifikuje úlohy jednotlivých používateľských príbehov. Úlohy musia byť identifikované tak, aby bolo možné ich rozsah jednoznačne definovať a aby mohla byť úloha pridelená na vypracovanie jednému členom tímu, ktorý ju vykoná v jednom šprinte.
5. Tím následne ohodnotí časovú náročnosť všetkých úloh pomocou *Poker cards* metódy. Táto zložitost' je vyjadrená v bodoch, ktoré budú po skončení šprintu za dané úlohy pridelené.
6. Náročnosť všetkých používateľských príbehov sa potom ohodnotí na základne hodnotenia ich úloh.
7. Na základe tohto hodnotenia náročnosti a vlastných priorít dôležitosti sa vlastník produktu rozhodne pre jednotlivé používateľské príbehy, ktoré chce nechať v nasledujúcom šprinte vypracovať.
8. Zodpovedné osoby za jednotlivé používateľské príbehy následne pridelia úlohy jednotlivým členom tímu. Pri tomto rozdelení sa dbá na to aby všetci členovia tímu mali celkovo približne rovnaký počet bodov.
9. V prípade nespokojnosti niektorých členov tímu so zadelením úloh sa tím snaží nájsť kompromisné riešenie tak, aby boli spokojní všetci členovia tímu. Posledné slovo má však vždy vedúci tímu.
10. Keď sú všetci členovia tímu s rozvrhom spokojný, tvorba rozvrhu pre daný šprint sa skončila

5.4 Manažment podpory vývoja

5.4.1 Verziovanie zdrojového kódu

5.4.1.1 Procesy



Obrázok 5.4: Procesy

5.4.1.2 Roly

- Vývojár
 - inicializácia lokálneho repozitára
 - implementácia
 - testovanie
 - vytváranie vetiev
 - zlučovanie vetiev
- Manažér kvality
 - kontrolovanie vetiev

- schvaľovanie a odmietanie
- oboznamovanie vývojárov výsledkami revízie
- Manažér podpory vývoja
 - inicializácia vzdialeného repozitára
 - archivácia súčasného stavu hlavnej vývojovej vetvy na konci šprintu
 - zlučovanie vetiev

5.4.1.3 Pojmy

- git – voľne dostupný, distribuovaný systém riadenia verzií
- vzdialený repozitár – uložený na serveri a uchováva zdieľané zdrojové kódy projektu, prípadne iné súbory v jednotlivých vývojových vetvách s dátami popisujúcimi jednotlivé verzie projektu
- lokálny repozitár – uložený na lokálnom počítači a uchováva zdrojové kódy projektu, prípadne iné súbory v jednotlivých vývojových vetvách s dátami popisujúcimi jednotlivé verzie projektu
- vetva – počas vývoja sa vytvárajú rôzne vetvy za účelom oddeľovania vývojových procesov, fixovania chýb
- zlúčenie vetiev – obsah súborov dvoch vetiev sa spojí do jednej vetvy
- konflikt – vzniká ak sa niekto snaží zlúčiť dve vetvy vývoja, ktoré obsahujú súbory s rôznym obsahom v jednotlivých riadkoch, git vtedy nevie aké riadky majú byť výsledkom zlúčenia vetiev
- junit – testovací unit framework
- redmine – open source projekt obsahujúci množstvo funkcií na manažovanie projektov, realizovaný ako webová aplikácia
- šprint – základná jednotka v agilnej metodike vývoja softvéru scrum
- commit – upravenie zmien funkcionality na danej vetve vývoja v repozitári
- push – vloženie vetvy vývoja s upravenou funkcionalitou na inú vetvu v inom repozitári

5.4.1.4 Inicializácia vzdialeného repozitára

Vstup: požiadavka na vytvorenie vzdialeného repozitára na serveri

Výstup: na serveri je vytvorený vzdialený repozitár

Zodpovedný: manažér podpory vývoja

Repozitár servera slúži ako zdieľaný centrálny repozitár, na(z) ktorý sa zasielajú(stahujú) vývojové vetvy so zdrojovými textami systému.

5.4.1.5 Inicializácia lokálneho repozitára

Vstup: požiadavka na vytvorenie lokálneho repozitára u každého vývojára

Výstup: každý vývojár má vytvorený lokálny repozitár

Zodpovedný: vývojár

Lokálny repozitár si inicializuje každý vývojár na svojom počítači.

- Nainštalovať git
- Vybrať adresár repozitára a vytvoriť git repozitár
- Vygenerovať SSH kľúč a pridať ho na server so vzdialeným repozitárom
- Klonovať vzdialený repozitár

5.4.1.6 Práca s lokálnym repozitárom

Vstup: požiadavka na vytvorenie funkcionality, dostupné dokumenty s návrhom implementácie funkcionality

Výstup: implementovaná a otestovaná funkcionality v lokálnom repozitári vývojára

Zodpovedný: vývojár

Vývojár implementuje funkcionality podľa návrhu. Po jej dokončení vytvorí unit testy pomocou junit, podľa metodiky testovania MTPJU. Vývojár overí funkčnosť vytvorenej funkcionality a opraví prípadne odhalené chyby. Vytvorenú funkcionality vývojár okomentuje. Každá pridaná alebo zmenená funkcionality musí byť popísaná komentárom. Jej presná štruktúra je popísaná v metodike spravovanie dokumentov MSD v časti komentovanie zdrojových textov.

5.4.1.7 Vytvorenie novej vetvy vývoja na lokálnom repozitári

Vstup: založený projekt na vzdialenom serveri

Výstup: nová vetva vývoja na lokálnom repozitári

Zodpovedný: vývojár

Nová vetva sa vytvára za účelom pridania novej ucelenej funkcionality. Lokálnych vetiev si vývojár môže vytvárať ľubovoľný počet a pracovať s nimi ako uzná za vhodné.

5.4.1.8 Pridanie novej vetvy vývoja na vzdialený repozitár

Vstup: vytvorená vetva v lokálnom repozitári

Výstup: pridaná vetva na vzdialenom repozitári

Zodpovední: vývojár

Zdieľané vetvy sú vetvy uložené na serveri. Všetci zainteresovaní majú k nim prístup a môžu ich meniť. Tieto vetvy vytvárajú vývojári. Ich počet nie je presne stanovený a odvíja sa od cieľov projektu. Ak tím pracuje na dvoch ucelených funkcionality stačia dve vývojové vetvy. Počet týchto vetiev by nemal byť vysoký, ideálne by mali stačiť max 2-3 vetvy. Vytvoreniu týchto vetiev predchádza konzultácia medzi vývojármi a dohoda kto bude na danej funkcionality pracovať. Vzor pre mená zdieľaných vetiev feature-version. Slová z feature sú oddelené podčiarkovníkmi a version sú dve čísla oddelené bodkou.

- main-1.0 je hlavná vývojová vetva
- archive-1.0 je archivačná vetva
- walk-1.0 je vývojová vetva zameraná na implementáciu chodenia robota

Version sa mení po desatinnej bodke.

5.4.1.9 Upravenie funkcionality v lokálnom repozitári - commit

Vstup: okomentované a otestované zdrojové texty programu

Výstup: zdrojové texty v lokálnom repozitári

Zodpovedný: vývojár

Vloženie zdrojových textov s upravenou funkcionality do určenej vetvy vývoja v lokálnom repozitári. Vloženie musí prebiehať so správou a jej štruktúra je uvedená v dolnej časti metodiky.

5.4.1.10 Upravenie funkcionality vo vzdialenom repozitári - push

Vstup: upravené texty programu v lokálnom repozitári
 Výstup: upravené texty programu na vzdialenom repozitári
 Zodpovedný: vývojár

Vývojár prispieva svojou funkcionalitou raz za deň, v ktorom upravil funkcionalitu programu, samozrejme za predpokladov, že upravená funkcionalita je otestovaná a okomentovaná ako bolo popísané v predchádzajúcich fázach procesu. Priebeh vkladania zmien na vzdialený repozitár:

- Vývojár stiahne poslednú verziu projektu zo vzdialeného repozitára do vývojovej vetvy svojho repozitára
- Zlúči vetvu vývoja so svojou vetvou vývoja v repozitári
- Ak prebehne zlúčenie vetiev bezchybne, pošle svoju verziu na vzdialený repozitár, v prípade konfliktov sa postupuje podľa časti riešenie konfliktov
- Nepovinne v nástroji redmine upraví stav úlohy, na ktorom pracoval. Redmine odosiela e-mail o zmene stavu úlohy a oboznámi ostatných o zmene stavu funkcionality. Tento krok je povinný ak ide o ukončenie úlohy a úloha nadobúda stav 100%. Prácou s redminom sa zaoberá metodika s názvom ADPSR.
- Vývojár ďalej pokračuje implementovaním funkcionality popísanej v časti Práca s lokálnym repozitárom.

5.4.1.11 Riešenie konfliktov

Vstup: konflikt v repozitári
 Výstup: vyriešený konflikt
 Zodpovedný: vývojár

Konflikty sa môžu objaviť na lokálnych aj zdieľaných vetvách vzdialeného repozitára. V lokálnych repozitároch pri získavaní zmien zo vzdialeného repozitára a vo vzdialenom repozitári pri odosielaní zmien z lokálneho repozitára. Dôvod vzniku konfliktu spočíva v dvoch rôznych a navzájom odlišných zmenách z pôvodného stavu vetvy. Vzdialený repozitár nesmie obsahovať vetvu s nevyriešeným konfliktom. Za riešenie konfliktu na vzdialenom repozitári je zodpovedný vývojár, ktorý konflikt spôsobil. Pri riešení konfliktu musí vývojár presne vedieť čo robí a musí rozumieť zdrojovým textom, nad ktorými rieši konflikt. Po vyriešení konfliktu sa vývojár uistí či pôvodná funkcionalita zostala nezmenená. Ak sa podarí konflikt vyriešiť, postupuje sa podľa upravenie funkcionality vo vzdialenom repozitári. V prípade ak nie je možné konflikt vyriešiť, projekt musí vývojár vrátiť do pôvodného stavu.

5.4.1.12 Revízia vývojovej vetvy

Vstup: Zmena implementovanej funkcionality na vývojovej vetve
 Výstup: Schválená alebo neschválená vývojová vetva
 Zodpovední: manažér kvality

Za revíziu je zodpovedný manažér kvality. Revízia môže byť vykonávaná aj priebežne počas vývoja kedy ešte vývoj vetvy nie je ukončený. Revízia sa má spraviť určite pred koncom šprintu alebo keď skončil vývoj vetvy. O ukončení vývoja vetvy informujú vývojári, ktorí na vetve pracovali. Manažér kvality môže vetvu schváliť alebo odmietnuť. Pri jej schválení sa pokračuje podľa časti zlúčenie

vetiev vývoja. V prípade odmietnutia musí manažér kvality stanoviť dôvod odmietnutia a môže byť podaný námet k vylepšeniu, odstraňujúci nedostatky, kvôli ktorým bola vetva odmietnutá. Odmietnutie spolu s prípadným námetom na vylepšenie je podaný zodpovedajúcim osobám e-mailom alebo osobne.

5.4.1.13 Zlúčenie vetiev vývoja

Vstup: vývoj vetvy úspešne prešiel revíziou
 Výstup: Vývojová vetva bola zlúčená s hlavnou vývojovou vetvou
 Zodpovedný: Manažér podpory vývoja

Na vzdialenom repozitári bude viacero vetiev

- Hlavná vývojová vetva obsahuje aktuálny stav projektu. Na túto vetvu sa priamo funkcionality nepridáva. Táto vetva sa rozrastá zlučovaním ostatných vývojových vetiev. Zlučovanie bude prebiehať najneskôr pred koncom šprintu alebo skôr ak vývoj vetvy skončil a môže byť začlenená do hlavnej vývojovej vetvy. Zodpovedný manažér podpory vývoja.
- Ostatné vývojové vetvy, ich počet nie je presne stanovený a viac sa o nich píše v časti pridanie novej vetvy vývoja na vzdialený repozitár. K týmto vetvám vývojári prispievajú.
- Archivačná vetva obsahuje archívy hlavnej vývojovej vetvy počas jednotlivých šprintov. Takto je pozorovateľný reálny výsledok každého šprintu napr. pre účely sledovania ako tím napredoval.

Proces zlúčenia vetvy s hlavnou vývojovou vetvou sa vykonáva po úspešnej revízii vetvy. Vetva vývoja je zlúčená do hlavnej vývojovej vetvy.

5.4.1.14 Archivácia hlavnej vývojovej vetvy

Vstup: všetky vývojové vetvy prešli zlúčením s hlavnou vývojovou vetvou
 Výstup: archivovaná hlavná vetva vývoja
 Zodpovední: vývojár, manažér podpory vývoja

Proces zlúčenia vetiev musí prebehnúť pre každú vetvu vývoja.

- Archivuje sa hlavná vývojová vetva
- Po úspešnej archivácii hlavnej vývojovej vetvy a po ukončení vývoja ostatných vetiev je zodpovedný manažér podpory vývoja za vymazanie ostatných vývojových vetiev na vzdialenom repozitári. Ak vývoj vetiev neskončil ale prešli revíziou a zlúčením s hlavnou vývojovou vetvou tak vývojové vetvy na vzdialenom repozitári zostávajú nedotknuté. (Revízia a zlúčenie vetiev prebehlo kvôli blížiacemu sa koncu šprintu, nie kvôli ukončeniu vývoja vetiev). Postup mazania vzdialených vetiev platí aj pre lokálne vetvy ale za mazanie lokálnych vetiev sú zodpovední vývojári.

5.4.1.15 Práca s gitom

Vzdialený a lokálny repozitár je inicializovaný. V adresári projektu sa vytvorí git repozitár. V netbeanse pravý klik na projekt => Versioning => Initialize Git Repository. Pri práci sa budú používať vývojové vetvy.

5.4.1.15.1 Vytvorenie vetvy vývoja

Vytvorí sa nová vetva pre vývoj. V tejto vetve bude upravovaná funkcionálnosť. V uvedenom príklade(Obrázok 5.5) bola vytvorená vetva s menom develop.

```
$ git branch develop
```

Obrázok 5.5

Pred samotnou prácou sa treba „prepnúť“ sa na danú vetvu vývoja(Obrázok 5.6).

```
$ git checkout develop
Switched to branch 'develop'
```

Obrázok 5.6

5.4.1.15.2 Pridanie funkcionality

Aktualizuje sa vývojová vetva z vetvy vzdialeného repozitára(Obrázok 5.7).

```
$ git pull git@gitbus.fiit.stuba.sk:robocup-a55-kickers/robocup-a55-kickers.git
Enter passphrase for key '/c/Users/Jigsaw/.ssh/id_rsa':
warning: no common commits
remote: Counting objects: 18, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 18 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (18/18), done.
From gitbus.fiit.stuba.sk:robocup-a55-kickers/robocup-a55-kickers
 * branch          HEAD          -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 team15.txt | 5 +++++
 1 file changed, 5 insertions(+)
 create mode 100644 team15.txt
```

Obrázok 5.7

Pred začatím samotnej práce vývojár overí či vetva nie je rozbitá a je konzistentná. Skúsi zostaviť program a spustiť testy. Teraz vývojár vytvorí zdrojový kód. Po skončení implementácie sa navrhnujú junit testy. Testovaním sa bližšie zaoberá metodika testovania. Až po akceptácii testov sa pokračuje ďalej. Treba explicitne povedať, ktoré súbory podstúpili zmeny, prípadne pribudli ako nové v projekte a treba ich brať v úvahu pre nasledujúci commit. Nad git repozitárom, ktorý je aj adresárom projektu označíme všetky súbory adresára.

```
$ git add .
```

Zmeny nahrá do repozitára a spravíme ich viditeľnými v rámci lokálneho repozitára(Obrázok 5.8).

```
$ git commit -m "test-TestApp::f,g|zatiaľ neuplne"
[master c471925] test-TestApp::f,g|zatiaľ neuplne
 2 files changed, 1 insertion(+), 1 deletion(-)
```

Obrázok 5.8

Commit prebieha vždy so správou(Obrázok 5). Správa musí obsahovať:

- Meno úlohy, s ktorým doplnené zdrojové texty súvisia(test)
- Popis čo bolo z úlohy spravené, názvy funkcií sú oddelené čiarkami(f,g)
- Meno úlohy oddelené od popisu znakom -
- Nepovinný vlastný komentár oddelený znakom |

5.4.1.15.3 Riešenie konfliktov

Vývojár zmenil funkcionality a pridal ju do lokálneho repozitára. Teraz stiahne vetvu zo servera na svoju vetvu v lokálnom adresári(Obrázok 5.9).

```
$ git pull git@gitbus.fiit.stuba.sk:robocup-a55-kickers/robocup-a55-kickers.git
test
Enter passphrase for key '/c/Users/Jigsaw/.ssh/id_rsa':
From gitbus.fiit.stuba.sk:robocup-a55-kickers/robocup-a55-kickers
 * branch          test          -> FETCH_HEAD
Auto-merging nbproject/project.xml
CONFLICT (add/add): Merge conflict in nbproject/project.xml
Auto-merging nbproject/project.properties
CONFLICT (add/add): Merge conflict in nbproject/project.properties
Auto-merging nbproject/genfiles.properties
CONFLICT (add/add): Merge conflict in nbproject/genfiles.properties
Auto-merging nbproject/build-impl.xml
CONFLICT (add/add): Merge conflict in nbproject/build-impl.xml
Auto-merging manifest.mf
CONFLICT (add/add): Merge conflict in manifest.mf
Auto-merging build.xml
CONFLICT (add/add): Merge conflict in build.xml
Automatic merge failed; fix conflicts and then commit the result.
```

Obrázok 5.9

Vetva test v lokálnom repozitári je nekonzistentná. Súbory teraz obsahujú časti textov, v ktorých sa lokálna a verzia servera a líšili. V takom prípade vývojár prezerá texty a ručne rieši konflikty. Pre vizualizáciu zdrojových textov pomôže gitk. Časti textov jednej či druhej verzie vetvy sú zvýraznené a pomáhajú pri riešení konfliktov. Ak sa konflikt nepodarí vyriešiť treba vrátiť zmeny

```
git reset --hard
```

Príkaz napravi stav vetvy na posledný commit. Hard spôsobí stratu posledných zmien. V prípade vrátenia sa na predposledný commit:

```
git reset --hard HEAD~1
```

Alternatívou k uvedenému riešeniu konfliktov je najprv zachytenie zmien pomocou fetch a následne vývojár použije diff na zobrazenie konfliktov(Obrázok 5.10) a ručne vyrieši konflikt.

```
diff --git a/src/testapp/TestApp.java b/src/testapp/TestApp.java
index 8e26a43..4491981 100644
--- a/src/testapp/TestApp.java
+++ b/src/testapp/TestApp.java
@@ -7,13 +7,13 @@ package testapp;
 public class TestApp {

    public static void main(String[] args) {
-       fO;
+       gO;AM
    }
    public static void fO{
        System.out.println("f");
    }

    public static void gO{
-       System.out.println("g");
+       System.out.println("gg");AM
    }
}
```

Obrázok 5.10

Po vyriešení konfliktu vývojár pridá zmeny, vykoná commit(Obrázok 5.11) a pošle stav projektu na server(Obrázok 5.12).

```
git add .
```

```
git commit -m "sprava"
[develop 9b10b8d] sprava
3 files changed, 6 insertions(+)
create mode 100644 build/classes/.netbeans_automatic_build
create mode 100644 build/classes/.netbeans_update_resources
create mode 100644 build/classes/testapp/TestApp.class
create mode 100644 nbproject/private/private.properties
```

Obrázok 5.11

```
$ git push git@gitbus.fiit.stuba.sk:robocup-a55-kickers/robocup-a55-kickers.git
Enter passphrase for key '/c/Users/Jigsaw/.ssh/id_rsa':
```

Obrázok 5.12

5.4.1.15.4 Archivácia hlavnej vývojovej vetvy

Archivovaná vetva bude vo formáte tar. Archivovať sa bude iba hlavná vývojová vetva.

```
git archive branchname > sprintX.tar
```

Formát sprintX

- Sprint je názov šprintu
- X je číslo šprintu

Takto je pozorovateľný výsledok každého šprintu napr. pre účely sledovania ako tím napredoval. Archív sa pridáva do archivačnej vetvy. Rozbalenie archivovanej vetvy:

```
tar -xf branchname.tar -C directory_to_extract
```

Za archiváciu je zodpovedný manažér podpory vývoja. Rozbaľovať archív môže každý. Počas archivácie, vetvy, ktoré boli ukončené, treba zmazať. Za vymazanie vetvy vzdialeného repozitára je zodpovedný manažér podpory vývoja.

```
git push origin :branchname
```

Za vymazanie vetvy na lokálnom repozitári je zodpovedný vývojár.

```
git branch -d branchname
```

5.5 Monitorovanie projektu

5.5.1 Manažment prehliadok – Vytvorenie zápisnice o vykonanej prehliadke

5.5.1.1 Úvod

Metodika MP28102012/1ZP popisuje systematický priebeh prehliadok vývoja softvérového produktu podľa normy ISO88988/§56MP s cieľom zabezpečiť vysokú kvalitu produktu v oblasti zdrojového kódu, dokumentácie a splnených požiadaviek vyplývajúcich zo špecifikácie. Riadi sa ňou manažment monitorovania projektu pri výkone svojej činnosti.

5.5.1.2 Súvisiace dokumenty

ISO88988/§56MP - norma dostupná na www.vymyslenyarchiv.sk/docs/isonorms/15545.php

5.5.1.3 Slovník pojmov

V tabuľke 5.2. je slovník pojmov metodiky MP28102012/1ZP.

Tabuľka 5.2: Slovník pojmov

| Výraz | Opis |
|-------|--|
| PDF | Portable document format - formát dokumentov |
| .ppt | Koncovka súboru formátu power point |

5.5.1.4 Roly účastníkov prehliadky

5.5.1.4.1 Manažér prehliadky

Organizácia a vedenie prehliadky podľa stanoveného plánu prehliadok, určenie zodpovedností jednotlivým účastníkom prehliadky.

Zodpovedosti:

- Vytvorenie plánu prehliadok formou zoznamu dátumov prehliadok a ich predmetov prehliadania
- Vytvorenie rozvrhov jednotlivých prehliadok podľa metodiky MP28102012/3ZP
- Upozornenie jednotlivých účastníkov prehliadky o najbližšie konanej prehliadke formou e-mailového, telefónneho alebo ústneho kontaktu.
- Pridelenie zodpovedností a úloh účastníkom vyplývajúcich z jednotlivých zodpovedností prehliadky ústnou formou, ktorú poskytne zapisovateľovi prehliadky
- Organizovanie priebehu prehliadky podľa vytvoreného rozvrhu, vedenie diskusie počas prehliadky, usmernenie účastníkov počas prehliadky

5.5.1.4.2 Inšpektor prehliadky

Kontrola kvality a stavu predmetu prehliadky podľa metodiky MP28102012/§57KKS.

Zodpovednosti:

- Vytvorenie zoznamu metrík prehliadky nad daným predmetom prehliadky formou tabuľky nameraných hodnôt metrík prehliadky, ktorá obsahuje stĺpce *Metrika*, *Nameraná hodnota*,

Požadovaná hodnota, Poznámka. Zoznam metrík zapíše do stĺpca *Metrika*, kde jedna metrika bude najviac trojslovné spojenie presne vyjadrujúce metriku, ktorá sa bude merať.

- Vytvorenie zoznamu požadovaných hodnôt metrík pre daný predmet prehliadky. Požadované hodnoty zapíše do stĺpca *Požadovaná hodnota* v tabuľke nameraných hodnôt metrík prehliadky. Požadovaná hodnotu zapíše formou čísla a mernej jednotky ak výsledkom merania bude číselná hodnota. Inak maximálne 10 slovami presne opíše čo sa požaduje na výstupe merania.
- Kontrola predmetu prehliadky podľa vytvoreného zoznamu metrík a zápis nameraných hodnôt do tabuľky nameraných hodnôt metrík prehliadky do stĺpca *Nameraná hodnota* formou čísla a mernej jednotky ak výsledkom merania je číselná hodnota. Inak maximálne 10 slovami presne opíše čo je výstupom merania.
- Postúpenie tabuľky nameraných hodnôt metrík prehliadky zapisovateľovi prehliadky formou e-mailu
- Prezentácia nameraných hodnôt prehliadky pomocou výtlačnej tabuľky nameraných hodnôt metrík prehliadky na stretnutí k prehliadke ústnou formou.

5.5.1.4.3 Zapisovateľ prehliadky

Vyhotovenie zápisnice prehliadky softvérového produktu podľa metodiky MP28102012/1ZP a vytvorenie zoznamu zúčastnených na prehliadke s ich úlohami a zodpovednosťami formou tabuľky v zápisnici prehliadky.

Zodpovednosti:

- Vytvorenie a archivácia zápisnice z vykonanej prehliadky produktu podľa metodiky MP28102012/1ZP
- Vytvorenie zoznamu účastníkov prehliadky s ich zodpovednosťami a úlohami formou tabuľky z údajov poskytnutých od manažéra prehliadky

5.5.1.4.4 Zodpovedná osoba za prehliadanú časť produktu

Prezentácia predmetu prehliadky a obhajoba výsledkov vykonanej práce nad predmetom prehliadky.

Zodpovednosti:

- Príprava na prezentáciu predmetu prehliadky, vytvorenie vizuálnej a ústnej prezentácie predmetu prehliadky. Vizuálna prezentácia vo forme .ppt podľa metodiky MP28102012/1PP
- Príprava ústnej obhajoby na namerané hodnoty prehliadky inšpektorom

5.5.1.5 Vstupy prehliadky

V tabuľke 5.3 sú vstupy prehliadky podľa metodiky MP28102012/1ZP.

Tabuľka 5.3: Vstupy prehliadok

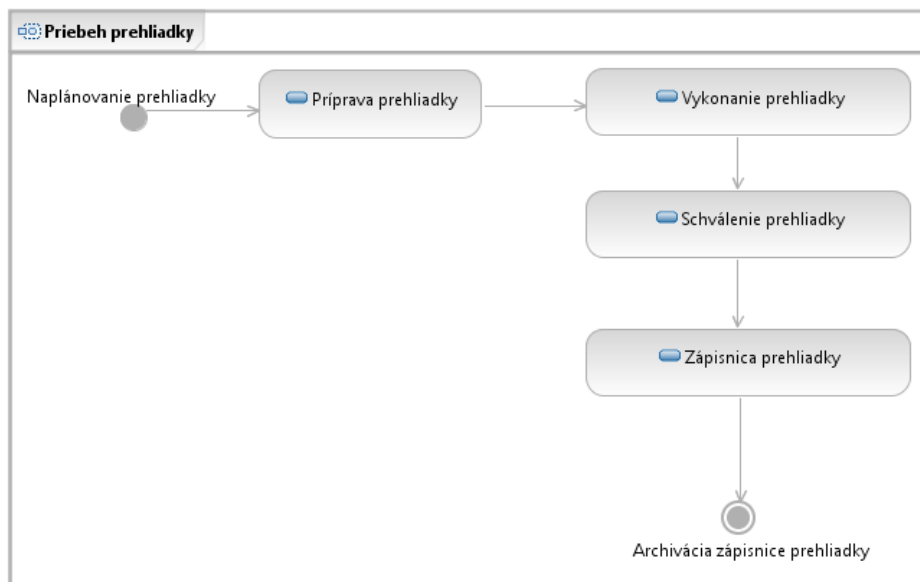
| Vstup | Popis |
|--------------------|---|
| Predmet prehliadky | Zdrojové súbory, Dokumentácia, Zoznam splnených požiadaviek |

| | |
|---|---|
| Prezentácia predmetu prehliadky | .ppt prezentácia dôležitých častí predmetu prehliadky podľa metodiky MP28102012/1PP |
| Tabuľka nameraných hodnôt metrík prehliadky | metriky, podľa ktorých sa predmet prehliadky kontroloval, ich požadované a namerané hodnoty |
| Rozvrh prehliadky | rozvrh priebehu prehliadky podľa metodiky MP28102012/3ZP |
| Zoznam účastníkov s rozdelením zodpovedností a úloh | zoznam účastníkom s ich zodpovednosťami a úlohami formou tabuľky |

5.5.1.6 Priebeh prehliadky

Následnosť prehliadok sa riadi plánom prehliadok. V pláne prehliadok sú prehliadky naplánované na konci každého šprintu a aj počas šprintu po ukončení významných častí vývoja softvérového produktu. Priebeh prehliadky je znázornený na obrázku 5.13. Pozostáva z týchto častí:

- Naplánovanie prehliadky
- Príprava prehliadky
- Vykonanie prehliadky rozdelené do nasledujúcich častí:
 - Prezentácia predmetu prehliadky
 - Prezentácia nameraných hodnôt inšpektorom
 - Diskusia k nameraným hodnotám a obhajoba osoby zodpovednej za prehliadanú časť produktu
 - Uznesenie prehliadky
 - Schválenie prehliadky prípadne naplánovanie ďalšej prehliadky
- Zápisnica prehliadky - dokončenie zápisnice
- Archivácia zápisnice



Obrázok 5.13: Priebeh riadenia prehliadok

5.5.1.6.1 Naplánovanie prehliadky

Vstup: Plán prehliadok

Výstup: Zoznam účastníkov s rozdelením zodpovedností a úloh, Rozvrh prehliadky

Zodpovednosť: Manažér prehliadky, Zapisovateľ prehliadky

Manažér prehliadky sa drží plánu prehliadok. Týždeň pred naplánovanou prehliadkou alebo nenaplánovanou požadovanou prehliadkou vytvorí zoznam účastníkov prehliadky, ktorý poskytne zapisovateľovi prehliadky a upozorní všetkých účastníkov prehliadky o konaní prehliadky ústnou formou, e-mailom alebo telefonicky. Následne vytvorí rozvrh nadchádzajúcej prehliadky podľa metodiky MP28102012/1RP. Zapisovateľ prehliadky vytvorí zoznam účastníkov prehliadky, ich úloh a zodpovedností formou tabuľky *Účastníci*, ktorá obsahuje slúpce *Rola*, *Meno a Priezvisko*, *Úlohy*.

5.5.1.6.2 Príprava prehliadky

Vstup: Zoznam účastníkov s rozdelením zodpovedností a úloh, Predmet prehliadky

Výstup: Prezentácia predmetu prehliadky, Tabuľka nameraných hodnôt metrík prehliadky

Zodpovednosť: Inšpektor prehliadky, Osoba zodpovedná za prehliadanú časť produktu

Z upozornených účastníkov prehliadky osoba zodpovedná za prehliadanú časť produktu poskytne kompletný predmet prehliadky inšpektorovi a pripraví .ppt prezentáciu predmetu prehliadky podľa metodiky MP28102012/1PP. Inšpektor prehliadky vypracuje zoznam metrík prehliadky formou tabuľky nameraných hodnôt metrík prehliadky. Následne zvolí požadované hodnoty metrík v závislosti od predmetu prehliadky podľa metodiky MP28102012/§57KKS a skontroluje predmet prehliadky podľa zoznamu metrík. Zapiše namerané hodnoty do tabuľky nameraných hodnôt metrík prehliadky.

5.5.1.6.3 Vykonanie prehliadky

Vstup: Prezentácia predmetu prehliadky, Tabuľka nameraných hodnôt metrík prehliadky, Predmet prehliadky, Zoznam účastníkov s rozdelením zodpovedností a úloh, Rozvrh prehliadky

Výstup: Zápisnica

Zodpovednosť: Manažér prehliadky, Inšpektor prehliadky, Osoba zodpovedná za prehliadanú časť produktu, Zapisovateľ prehliadky

Prehliadka prebehne len za účasti všetkých požadovaných účinkujúcich prehliadky so všetkými požadovanými vstupmi prehliadky a bude organizovaná pokynmi manažéra prehliadky. Zapisovateľ má predprípravenú zápisnicu prehliadky podľa metodiky MP28102012/1ZP. Otvorenie prehliadky vyhlási manažér prehliadky. Postupne predvolá osobu zodpovednú za prehliadanú časť produktu aby prezentovala svoju prácu. Následne pozve na predvedenie metrík a výsledkov merania podľa metrík inšpektora prehliadky. Nasledujúcim krokom je otvorenie diskusie, kde osoba zodpovedná za prehliadanú časť produktu zaujme postoj obhajcu k svojej práci. Ostatní účastníci diskutujú k jednotlivým metrikám, nameraným a požadovaným hodnotám metrík. Zapisovateľ zápisnice

prehliadky zapisuje k jednotlivým metrikám prípadné poznámky od účastníkov prehliadky do poľa *Poznámka* v tabuľke *Namerané hodnoty metrick prehliadky*.

5.5.1.6.4 Schválenie prehliadky

Vstup: Zápisnica, Prezentácia nameraných hodnôt prehliadky

Výstup: Schválenie prehliadky alebo naplánovanie novej opravnej prehliadky

Zodpovednosť: Manažér prehliadky

Na konci prehliadky, po diskusii, manažér prehliadky vyhlási prehliadku za uzavretú a predvedie uznesenie z prehliadky, ktoré zapisovateľ prehliadky zapíše do zápisnice podľa metodiky MP28102012/1ZP. Ak pri prehliadke predmetu vznikla potreba opraviť predmet prehliadky tak manažér prehliadky naplánuje novú opravnú prehliadku.

5.5.1.7 Zápisnica prehliadky

Výstupom prehliadky podľa metodiky MP28102012/1ZP je zápisnica s názvom „*Produktová prehliadka č.* “ vo formáte pdf. Vzor zápisnice je v prílohe A metodiky MP28102012/1ZP. Zápisnica je výstupom práce zapisovateľa prehliadky vykonávanej počas prehliadky a po prehliadke.

5.5.1.7.1 Štruktúra zápisnice

- Nadpis zápisnice – „*Produktová prehliadka č.* “, kde za č. dopísať číslo prehliadky
- Hlavička zápisnice – obsahuje polia:
 - Predmet prehliadky – názov súboru/-ov, dokumentácii, špecifikovaných požiadaviek. Názov súborov a dokumentácii musí obsahovať informáciu o verzii súboru a dokumentu vo formáte *názov_súboru-V.číslo_verzie*.
 - Dňa – dátum dňa vypracovania zápisnice vo formáte *dd.mm.yyyy*, kde *dd* je deň, *mm* je mesiac a *yyyy* je rok vypracovania zápisnice
 - Vypracoval – meno a priezvisko zapisovateľa prehliadky
- Účastníci – zoznam účastníkov z ich rolami a úlohami. Zoznam je pripravený a zapisovateľom prehliadky vložený v predpripravenej zápisnici. Zoznam je v tabuľke, ktorá obsahuje polia:
 - Rola - sú 4 základné roly (zodpovednosti) a to *Manžér prehliadky, Inšpektor prehliadky, Zodpovedná osoba za prehliadanú časť produktu, Zapisovateľ prehliadky*
 - Meno a Priezvisko - meno a priezvisko účastníka prehliadky
 - Úlohy - maximálne 5 slovesami alebo podstanými menami opísané úlohy
- Správa – správa manažéra prehliadky. 50 - 150 slov, kde prvá veta je štrukturovaná následovne "*Prehliadka č. číslo konaná dňa dátum k predmetu prehliadky názov_predmetu_prehliadky bola prevedená za účasti všetkých účastníkov podľa zoznamu, ktorí boli náležite pripravený.*". Ďalej obsahuje naplánovaný priebeh prehliadky vo forme nečíslovaného zoznamu slovných spojení o dĺžke maximálne 5 slov, ktoré presne určujú bod priebehu prehliadky.
- Namerané hodnoty metrick prehliadky – zoznam metrick a požadované hodnoty, ktoré určil inšpektor prehliadky, namerané hodnoty metrick, ktoré nameral inšpektor prehliadky a poznámky účastníkov prehliadky k metrikám. Tabuľka je vložená zapisovateľom do predpripravenej zápisnice prehliadky. Zoznam je formou tabuľky, ktorá obsahuje polia:

- Metrika - slovné spojenie maximálne 3 slov, ktoré určuje čo sa na predmete prehliadky meria
- Nameraná hodnota - číselná hodnota s mernou jednotkou alebo maximálne 10 slovami opísaná hodnota, ktorú pre danú metriku namerá inšpektor prehliadky
- Požadovaná hodnota - číselná hodnota s mernou jednotkou alebo maximálne 10 slovami opísaná hodnota, ktorú pre danú metriku odhadol inšpektor prehliadky podľa metodiky MP28102012/§57KKS
- Poznámka - poznámky účastníkov prehliadky dopíše zapisovateľ prehliadky počas diskusie k predmetu prehliadky
- Uznesenie prehliadky – výsledok prehliadky, vyjadrenie sa k priebehu prehliadky účastníkov tímu. Predvedie ho manažér prehliadky. Uznesenie prehliadky je minimálne 50 slov a maximálne 150 slov, v ktorých prvá veta má formu "*Predmet názov_predmetu prehliadky konanej dňa dátum je ne-/úspešne ukončený/-á/-é. Ne-/Úspešné ukončenie predmetu prehliadky potvrdzujú dolupodpísaní účastníci prehliadky.*" Pri neúspešnom ukončení predmetu prehliadky pribudne veta vo forme "*Z vyplývajúceho neúspešného ukončenia predmetu prehliadky je naplánovaná ďalšia prehliadka na dátum*". Ďalšie slová uznesenia určí manažér prehliadky.
- Súvisiace dokumenty – názvy zdrojových súborov, dokumentácií, súboru so špecifikáciami požiadaviek, ktorých obsah bol predmetom prehliadky alebo ktoré boli použité ako prezentačný materiál počas prehliadky. Názov zdrojových súborov, dokumentácií, špecifikácií požiadaviek musí obsahovať informáciu o verzii súboru vo formáte *názov_súboru-V.číslo_verzie*, kde za *názov_súboru* napíšeme názov súboru a za *číslo_verzie* číslo verzie.
- Poznámky – prípadné poznámky účastníkov počas prehliadky, ktoré zapíše zapisovateľ prehliadky počas priebehu prehliadky, mimo diskusie k metrikám a nameraným hodnotám metrik k predmetu prehliadky. Formát zápisu je na nový riadok napísať "*meno_účastníka - poznámka_účastníka*", kde za *meno_účastníka* doplniť meno a priezvisko účastníka a pomlčkou odeliť poznámku účastníka.

5.5.1.7.2 Postup vypracovania zápisnice

1. Prípravným krokom je vypísanie tabuľky účastníkov prehliadky. Zapisovateľ ku každej roli v stĺpci *Rola* vypíše hodnotu v stĺpci *Meno a Priezvisko* celým menom účastníka a hodnotu v stĺpci *Úlohy* úlohami účastníkov, ktoré obdržali od manažéra prehliadky pri plánovaní prehliadky podľa metodiky MP28102012/1ZP.
2. Zapisovateľ pokračuje vyplnením tabuľky *Namerané hodnoty metrik prehliadky*. Hodnoty v stĺpcoch *Metrika*, *Nameraná hodnota*, *Požadovaná hodnota* vyplní podľa zoznamu metrik, nameraných a požadovaných hodnôt metrik, ktorý obdrží od inšpektora prehliadky. Hodnoty v stĺpci *Poznámka* vyplní z prípadných poznámok účastníkov prehliadky počas diskusie ku konkrétnej metrike.
3. Na začiatku sedenia prehliadky zapisovateľ vypíše pole *Predmet prehliadky* názvami súborov, dokumentácií a špecifikácií požiadaviek, ktoré sú predmetom prehliadky v požadovanom formáte s informáciou o verzii súboru podľa metodiky MP28102012/1ZP.
4. Pole *Správa* vyplní zapisovateľ prehliadky na základe pokynu sa manažéra prehliadky podľa metodiky MP28102012/1ZP.
5. *Poznámky* vyplní na základe poznámok účastníkov počas priebehu prehliadky okrem poznámok k metrikám a nameraným hodnotám metrik podľa metodiky MP28102012/1ZP.

6. Na konci prehliadky zapisovateľ zapíše do poľa *Uznesenie prehliadky* vyjadrenie manžéra k celkovému priebehu prehliadky, k výsledkom prehliadky, a ak bola naplánovaná ďalšia prehliadka v dôsledku neuspokojivých výsledkov súčasnej prehliadky, tak aj vyjadrenie k ďalšej prehliadke podľa metodiky MP28102012/1ZP.
7. Predposledným bodom je vyplnenie poľa *Súvisiace dokumenty*, kde vyplní názvy všetkých dokumentov, súborov a prezentácií, ktoré boli použité na prehliadke formou zoznamu. Názvy obsahujú informáciu o verzii dokumentu, súboru alebo prezentácie podľa metodiky MP28102012/1ZP.
8. Zápisnica produktovej prehliadky sa uzavrie vypísaním poľa *Dňa* dátumom vypracovania prehliadky a poľa *Vypracoval* menom a priezviskom zapisovateľa prehliadky podľa metodiky MP28102012/1ZP. Zápisnicu nakoniec podpíšu všetci účastníci prehliadky v spodnej časti zápisnice.

5.5.1.7.3 Archivácia zápisnice

Zápisnica sa vyhotoví tlačenu formou v 4 exemplároch, kde 3 exempláre zápisnice si uchovávajú manažér prehliadky, osoba zodpovedná za prehliadanú časť produktu a inšpektor prehliadky. Posledný podpísaný exemplár všetkými účastníkmi prehliadky sa uloží do archívu. Elektronická forma zápisnice sa uloží do manažérskeho systému *Redmine*, ktorý tím používa. Manažérsky systém *Redmine* je dostupný na adrese <https://redmine.fiit.stuba.sk>. Po prihlásení do systému konkrétneho projektu k archivácii zápisnice prehliadky treba kliknúť na *Documents* v hornom menu. Potom kliknúť na *New document* vpravo hore obrazovky na pridanie nového dokumentu. Zobrazí sa formulár ako na obrázku 5.14. Vo výberovom poli *Category* vybrať *User documentation*. Do poľa *Title* vložiť nadpis *Produktová prehliadka č.*, kde za č. dopísať číslo prehliadky. Do textového poľa *Description* vložiť text vo forme *Archivácia zápisnice prehliadky č. konanej dňa*, kde za č. dopísať číslo prehliadky a na koniec vety dátum konania prehliadky a bodku. Potom kliknúť na tlačidlo *Prehľadávať*. Zobrazí sa dialógové okno na výber súboru. Vybrať súbor zápisnice prehliadky vo formáte pdf určený na archiváciu a potvrdiť výber. Potvrdenie výberu závisí od operačného systému a jazyka operačného systému počítača, na ktorom archivácia prebieha. Nakoniec kliknúť na *Create* pre potvrdenie formulára archivácie zápisnice prehliadky.

Obrázok 5.14: Formulár archivácie zápisnice v Redmine

5.6 Manažment komunikácie

5.6.1 Proces zberu požiadaviek

5.6.1.1 Hlavné kroky zberu požiadaviek

| Kapitola | Krok |
|-----------|---|
| 5.6.1.2.1 | Konzultácia zadávateľov projektu |
| 5.6.1.2.2 | Návrh požiadavky |
| 5.6.1.2.3 | Konzultácia požiadavky |
| 5.6.1.2.4 | Analýza požiadavky a návrh postupu na splnenie požiadavky |
| 5.6.1.2.5 | Úprava požiadavky |
| 5.6.1.2.6 | Zaradenie požiadavky do backlogu |
| 5.6.1.2.7 | Vytváranie úloh z konkrétnej požiadavky |

5.6.1.2 Diagram procesu zberu požiadaviek

5.6.1.2.1 Konzultácia zadávateľov projektu

Vykonáva: Konzultanti product ownera, product owner

Vstup: Predstava požiadavky, ako súčasť výsledného riešenia

Výstup: Požiadavka vo forme návrhu pre vývojový tím

Požiadavka musí prejsť aspoň základným spresnením a musí byť sformulovaná jednoducho a zrozumiteľne. Požiadavka musí byť schválená všetkými zúčastnenými tohto procesu.

5.6.1.2.2 Návrh požiadavky

Vykonáva: Product owner

Vstup: Požiadavka v prvotnej formulácii

Výstup: Vývojový tím oboznámený s požiadavkou

Product owner slovne sformuluje požiadavku na vývojový tím. Táto požiadavka už musí byť prekonzultovaná a schválená jeho konzultantmi. Táto činnosť sa vykonáva na začiatku projektu. Neskôr počas vývoja sa môže vykonávať výhradne na začiatku šprintov.

5.6.1.2.3 Konzultácia požiadavky

Vykonáva: Product owner, vývojový tím

Vstup: Požiadavka v slovnom podaní

Výstup: Požiadavka vhodná na analýzu a návrh

Vykonáva sa hneď po podaní požiadavky product ownerom, najneskôr však počas najbližšieho stretnutia. V tomto procese sa vývojový tím vyjadří k požiadavke a keď je potrebné, uvedie požiadavku do tvaru vhodného na analýzu.

5.6.1.2.4 Analýza požiadavky a návrh postupu na splnenie požiadavky

Vykonáva: Vývojový tím

Vstup: Prekonzultovaná požiadavka

Výstup: Požiadavka v podobe na zaradenie do backlogu

Hlavnou úlohou tohto kroku je odhaliť nedostatky požiadavky. Vývojový tím preskúma existujúce riešenia a metódy. Analýza má byť vykonaná do takej hĺbky, aby sa vývojový tím vedel rozhodnúť, či je požiadavka splniteľná. Z analýzy majú byť jasné aj možné návrhy na riešenie.

5.6.1.2.5 Úprava požiadavky

Vykonáva: Vývojový tím

Vstup: Analyzovaná požiadavka

Výstup: Návrh na úpravy požiadavky

Vývojový tím preformuluje požiadavku do podoby, ktorá neobsahuje odhalené nedostatky pôvodnej požiadavky. Vychádza sa z vykonanej analýzy k požiadavke.

5.6.1.2.6 Zaradenie požiadavky do backlogu

Vykonáva: Product owner, vývojový tím

Vstup: Analyzovaná požiadavka

Výstup: Požiadavka zaradená do backlogu

Požiadavka sa fyzicky zaradí do backlogu, ktorý je dostupný všetkým členom tímu, aj product ownerovi. Na pridávanie požiadaviek má právo každý člen tímu, aj product owner. Avšak, keď sa neurčí inak, zodpovedný za aktualizáciu backlogu je člen tímu, ktorý spísal zápisnicu z poslednej schôdze. Zápisnica schôdze a aktualizácia backlogu musí byť vykonaná do 48 hodín po skončení schôdze.

5.6.1.2.7 Vytváranie úloh z konkrétnej požiadavky

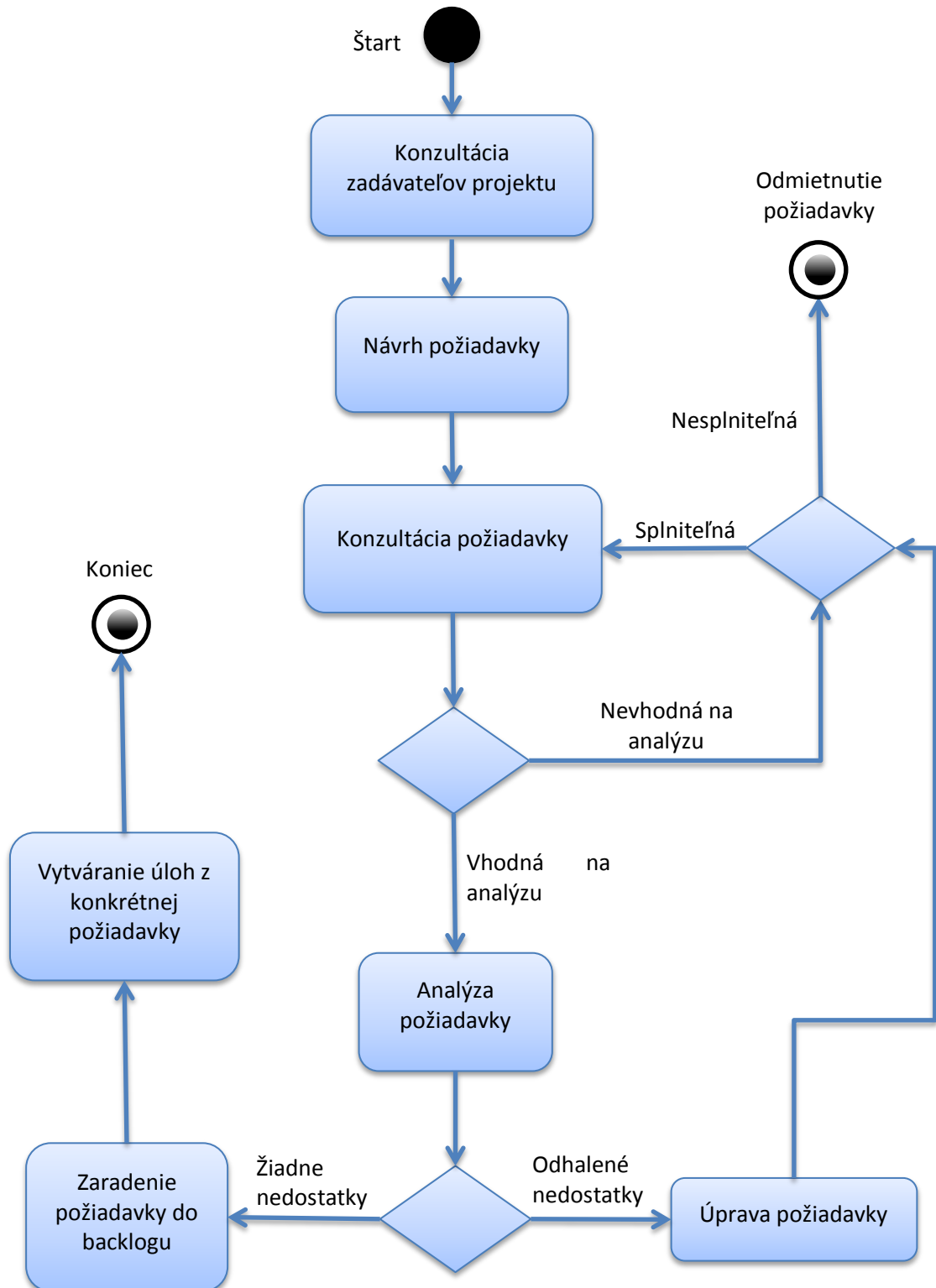
Vykonáva: Manažér plánovania, vývojový tím

Vstup: Požiadavka z backlogu

Výstup: Úlohy na splnenie požiadavky

Vytváranie úloh sa vykonáva na začiatku šprintov počas vývoja. Vytvorené úlohy majú zabezpečiť splnenie jednotlivých požiadaviek.

5.6.1.3 Diagram procesu zberu požiadaviek



5.6.2 Kroky zaradenia požiadavky do backlogu

| Kapitola | Krok |
|----------|------------------------------------|
| 5.6.2.1 | Otvorenie backlogu na Google drive |
| 5.6.2.2 | Pridanie požiadavky |
| 5.6.2.3 | Kontrola uloženia zmien |

5.6.2.1 Otvorenie backlogu na Google drive

Prístup k zdieľanej tabuľke je zabezpečený viacerými cestami. Prístupová URL adresa je dostupná na web stránke tímu a na spoločnej e-mailovej adrese.

5.6.2.2 Pridanie požiadavky

Požiadavka sa zapisuje do zdieľanej tabuľky s pripravenými názvami stĺpcov.

Názvy stĺpcov hrubým písmom sú povinné. Formátovanie polí sa riadi podľa vopred určených pravidiel a možností.

| Stĺpec | Popis | Formát |
|------------------------------------|--|---|
| Požiadavka | Stručné pomenovanie požiadavky | Text |
| Dátum záznamu | Dátum schôdze, na ktorej bola požiadavka schválená | Dátum (autoformát: DD/MM/RR) |
| Dátum požadovaného splnenia | Dátum alebo inak označený termín požadovaného splnenia | Text: Dátum (DD/MM/RR) Obdobie (zimný/ letný semester) |
| Zaznamenal | Meno a priezvisko autora príspevku | Text |
| Stav | Stav požiadavky | Text: Nová Rozpracovaná Dokončená |
| Podmienky | Závislosť splnenia požiadavky od inej požiadavky | Text |
| Popis | Popis požiadavky | Text |

5.6.2.3 Kontrola uloženia zmien

Zmeny sa uchovávajú automaticky po každom kroku. Automatické uchovávanie vykonáva funkcia Google driveu. O kontrole stavu je potrebné sa uistiť identifikovaním výpisu: „*All changes saved in Drive*“. Výpis sa má objaviť napravo od hlavného menu nad tabuľkou

5.7 Manažment tvorby dokumentácie

5.7.1 Tvorba písomnej technickej dokumentácie

5.7.1.1 Nástroj na tvorbu dokumentácie

Na tvorbu dokumentácie sme vybrali program Microsoft Word 2010.

5.7.1.2 Konvencie pri písaní dokumentácie

Pre uľahčenie spájania jednotlivých častí dokumentácie do jedného konzistentne vyzerajúceho a celistvého dokumentu bola vypracovaná šablóna, ktorá obsahuje základné konvencie pri písaní dokumentu.

Pri odovzdávaní dokumentu na spájanie je vhodné jeho pomenovanie v tvare autor_názovDokumentácie (napr. ján-komunikácia).

Najdôležitejšou požiadavkou je dodržiavanie štýlov, ktoré sú obsiahnuté v dostupnej šablóne. Ich použitie je popísané v tabuľke 5.4.

Tabuľka 5.4: Zoznam a použitie štýlov dokumentácie

| Názov štýlu | Použitie |
|----------------|---------------------------|
| R_Normálny | Bežný text |
| V_Nadpis1 | Nadpis 1. úrovne |
| V_Nadpis2 | Nadpis 2. úrovne |
| V_Nadpis3 | Nadpis 3. úrovne |
| V_Nadpis4 | Nadpis 4. úrovne |
| V_Nadpis5 | Nadpis 5. úrovne |
| R_Tabulka | Štýl pre tabuľku |
| R_Normálny_tab | Text v tabuľkách |
| R_Popis | Popis obrázkov a tabuliek |

5.7.1.2.1 Vkladanie obrázkov a tabuliek

Pre konzistenciu číslovanie obrázkov a tabuliek je potrebné dodržiavanie istých zásad pri ich vkladaní.

Vloženie obrázku (tabuľky):

1. vložiť obrázok (tabuľku)
2. vložiť popis
 - a. označiť obrázok (tabuľku)
 - b. stlačiť pravé tlačidlo na myši
 - c. zvoliť *Vložiť popis*
 - d. napísať názov obrázku (tabuľky)
 - e. potvrdiť *OK*

6 Záznamy zo stretnutí

6.1 Zápis z 1. stretnutia

| | |
|----------------|---|
| Dátum: | 10.10.2012 |
| Trvanie: | 8:00 – 10:00 |
| Miestnosť: | blok D – softvérové štúdio |
| Prítomní | |
| Pedagóg: | Ing. Marián Lekavý, PhD. |
| Členovia tímu: | Matej Červeňák, Jaroslav Grega, Martin Gregor, Gábor Nagy, Matúš Ondrejkovič, Filip Sucháč |
| Nepřítomní: | - |
| Zapísovateľ: | Matej Červeňák |

Téma stretnutia:

Úvod k projektu RoboCup.
Spustenie prvého šprintu.

Vyhodnotenie úloh šprintu / Priebežný stav úloh šprintu:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav |
|----|-------|------------|----------------|----------|------|
| - | - | - | - | - | - |

Opis stretnutia:

- Zoznámenie sa s vedúcim.
- Ustanovenie SCRUM, ako metodiky vývoja.
- Úvodné informácie k projektu RoboCup na fakulte (server, vývoj v Jave, simulačné prostredie, wiki...).
- Rozhodnutie o rozbehaní projektov z posledného ročníka.
- Rozhodnutie o našťudovaní projektov z posledného ročníka.
- Rozhodnutie o našťudovaní projektov od zahraničných tímov.
- Diskusia o webovej stránke.
- Určenie verziovacieho nástroja = Git
- Určenie nástroja na manažment = Redmine (?)

Stanovené úlohy:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav |
|-----|--|---------------|----------------|----------|------|
| 1.1 | Rozbehať webovú stránku. Skupinový mail ako obrázok. | Martin Gregor | 19/10/12 | normálna | nová |
| 1.2 | Rozbehať server na | Martin | 17/10/12 | vysoká | nová |

| | | | | | |
|------|---|--|----------|----------|------|
| | RoboCup. | Gregor | | | |
| 1.3 | Získať projekt od konkurenčného tímu z posledného roka. | Matej Červeňák | 17/10/12 | vysoká | nová |
| 1.4 | Spustiť hráča na server | Martin Gregor | 17/10/12 | vysoká | nová |
| 1.5 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | všetci | 24/10/12 | normálna | nová |
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | všetci | 17/10/12 | normálna | nová |
| 1.7 | Študovať wiki k RoboCup-u. | všetci | 17/10/12 | normálna | nová |
| 1.8 | Študovať zahraničné tímy a zápasy. | všetci | 17/10/12 | normálna | nová |
| 1.9 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | Matúš Ondrejko, Jaroslav Grega, Filip Sucháč | 24/10/12 | normálna | nová |
| 1.10 | Rozbehať Git | Martin Gregor | 17/10/12 | normálna | nová |
| 1.11 | Nainštalovať Git u seba. | všetci | 24/10/12 | normálna | nová |
| 1.12 | Rozbehať manažérsky softvér (Redmine?) | Gábor Nagy | 17/10/12 | vysoká | nová |
| 1.13 | Databáza | Martin Gregor, Matúš Ondrejko | 24/10/12 | normálna | nová |

Poznámky:

-

6.2 Zápis z 2. stretnutia

Dátum: 17.10.2012
 Trvanie: 8:00 – 10:00
 Miestnosť: blok D – softvérové štúdio
 Prítomní
 Pedagóg: Ing. Marián Lekavý, PhD.
 Členovia tímu: Matej Červeňák, Jaroslav Grega, Martin Gregor, Gábor Nagy, Matúš Ondrejko, Filip Sucháč
 Neprítomní: -
 Zapisovateľ: Martin Gregor

Téma stretnutia:

Kontrola výsledkov z prvého šprintu.
 Určenie výstupov jednotlivých úloh a ich hodnotenia náročnosti

Vyhodnotenie úloh šprintu / Priebežný stav úloh šprintu:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|-----|---|----------------|----------------|----------|--------------|------|--|
| 1.1 | Rozbehať webovú stránku. Skupinový mail ako obrázok. | Martin Gregor | 19/10/12 | normálna | rozpracovaná | 8 | Bežiaci stránka na správnej url, naplnená podľa pokynov. |
| 1.2 | Rozbehať server na RoboCup. | Martin Gregor | 17/10/12 | vysoká | rozpracovaná | 13 | Nainštalovaný server u všetkých, na ktorom vedia pustiť hráča. |
| 1.3 | Získať projekt od konkurenčného tímu z posledného roka. | Matej Červeňák | 17/10/12 | vysoká | splnená | ½ | Získané zdrojové súbory. |
| 1.4 | Spustiť hráča na server | Martin Gregor | 17/10/12 | vysoká | rozpracovaná | 3 | U každého člena spustený hráč na svojom serveri. |
| 1.5 | Vytvoriť pohyb | vsetci | 24/10/12 | normálna | nová | 3 | U každého vytvorený |

| | | | | | | | |
|------|--|---|----------|----------|--------------|---|--|
| | alebo upraviť existujúci pohyb hráča v editore. | | | | | | pohyb svojho hráča. |
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | Filip, Matej, Gabor | 17/10/12 | normálna | rozpracovaná | 1 | Písomná analýza minimálne 3 tímov. |
| 1.7 | Študovať wiki k RoboCup-u. | všetci | 17/10/12 | normálna | rozpracovaná | ½ | Každý člen povie niečo zaujímavé, čo sa dočítal na wiki. |
| 1.8 | Študovať zahraničné tímy a zápasy. | všetci | 17/10/12 | normálna | rozpracovaná | 3 | Analýza minimálne 6 tímov zo zahraničia so zaujímavými prístupmi |
| 1.9 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie. .. | Matúš Ondrejko č, Jaroslav Grega, Filip Sucháč | 24/10/12 | normálna | rozpracovaná | 3 | Prehľad existujúcich prístupov + analýza minimálne 2 prístupov reálnych modelov. |
| 1.10 | Rozbehať Git | Martin Gregor | 17/10/12 | normálna | ukončená | ½ | Založený repozitár na git |
| 1.11 | Nainštalovať Git u seba. | všetci | 24/10/12 | normálna | nová | ½ | - |
| 1.12 | Rozbehať manažérsky softvér (Redmine?) | Gábor Nagy | 17/10/12 | vysoká | rozpracovaná | 5 | Každý člen tímu má prístup do manažérskeho softvéru. |

Opis stretnutia:

- Prezentácia doteraz vykonanej práce.

- Určenie výstupov jednotlivých úloh.
- Určenie váh úloh pomocou scrum pokeru.

Stanovené úlohy:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|----|-------|------------|----------------|----------|------|------|--------|
| - | | - | - | - | - | - | |

Poznámky:

- Poslať prístup na manažérsky sofvért a git Marianovi.

6.3 Zápis z 3. stretnutia

Dátum: 24.10.2012
 Trvanie: 8:00 – 10:00
 Miestnosť: blok D – softvérové štúdio
 Prítomní
 Pedagóg: Ing. Marián Lekavý, PhD.
 Členovia tímu: Matej Červeňák, Jaroslav Grega, Martin Gregor, Gábor Nagy, Matúš Ondrejko, Filip Sucháč
 Neprítomní: -
 Zapisovateľ: Gábor Nagy

Téma stretnutia:

Kontrola priebežných výsledkov z prvého šprintu.

Vyhodnotenie úloh šprintu / Priebežný stav úloh šprintu:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|-----|---|----------------|----------------|----------|--------------|------|--|
| 1.1 | Rozbehať webovú stránku. Skupinový mail ako obrázok. | Martin Gregor | 19/10/12 | normálna | ukončená | 8 | Bežiaci stránka na správnej url, naplnená podľa pokynov. |
| 1.2 | Rozbehať server na RoboCup. | Martin Gregor | 17/10/12 | vysoká | rozpracovaná | 13 | Nainštalovaný server u všetkých, na ktorom vedú hrať. |
| 1.3 | Získať projekt od konkurenčného tímu z posledného roka. | Matej Červeňák | 17/10/12 | vysoká | ukončená | ½ | Získané zdrojové súbory. |
| 1.4 | Spustiť hráča na server | Martin Gregor | 17/10/12 | vysoká | rozpracovaná | 3 | U každého člena spustený hráč na svojom serveri. |
| 1.5 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča | vsetci | 24/10/12 | normálna | nová | 3 | U každého vytvorený pohyb svojho hráča. |

| | | | | | | | |
|------|---|---|----------|----------|--------------|---|--|
| | v editore. | | | | | | |
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | Filip, Matej, Gabor | 17/10/12 | normálna | rozpracovaná | 1 | Písomná analýza minimálne 3 tímov. |
| 1.7 | Študovať wiki k RoboCup-u. | všetci | 17/10/12 | normálna | ukončená | ½ | Každý člen povie niečo zaujímavé, čo sa dočítal na wiki. |
| 1.8 | Študovať zahraničné tímy a zápasy. | všetci | 17/10/12 | normálna | ukončená | 3 | Analýza minimálne 6 tímov zo zahraničia so zaujímavými prístupmi |
| 1.9 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | Matúš Ondrejkovič, Jaroslav Grega, Filip Sucháč | 24/10/12 | normálna | rozpracovaná | 3 | Prehľad existujúcich prístupov + analýza minimálne 2 prístupov reálnych modelov. |
| 1.10 | Rozbehať Git | Martin Gregor | 17/10/12 | normálna | ukončená | ½ | Založený repozitár na git |
| 1.11 | Nainštalovať Git u seba. | všetci | 24/10/12 | normálna | ukončená | ½ | - |
| 1.12 | Rozbehať manažérsky softvér (Redmine?) | Gábor Nagy | 17/10/12 | vysoká | ukončená | 5 | Každý člen tímu má prístup do manažérskeho softvéru. |

Opis stretnutia:

- Prezentácia doteraz vykonanej práce.
- Upresnenie niektorých úloh.
- Pridanie menších úloh (opísané nižšie)
- Konzultácia o postupe riešení nasledujúcich šprintov a navrhovaní úloh

Stanovené úlohy:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|------|------------------------------------|------------|----------------|----------|------|------|-------------------------------------|
| 1.13 | Zabezpečiť aktuálny editor pohybov | - | 31/10/12 | Vysoká | nová | - | Editor dostupný pre všetkých členov |

Poznámky:

- Zabezpečiť prístup vedúcemu projektu do tímového Redmine projektu (úloha vyriešená hneď po stretnutí)
- Analýzy zahraničných a domácich tímov postupne zdieľať na wiki

6.4 Zázpis z 4. stretnutia

Dátum: 31.10.2012
 Trvanie: 8:00 – 10:40
 Miestnosť: blok D – softvérové štúdio
 Prítomní
 Pedagóg: Ing. Marián Lekavý, PhD.
 Členovia tímu: Matej Červeňák, Jaroslav Grega, Martin Gregor, Gábor Nagy, Matúš Ondrejko, Filip Sucháč
 Neprítomní: -
 Zapisovateľ: Gábor Nagy

Téma stretnutia:

Vyhodnotenie šprintu 1
 Štart šprintu 2

Vyhodnotenie úloh šprintu 1:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|-----|--|----------------|----------------|----------|----------|------|---|
| 1.1 | Rozbehať webovú stránku. Skupinový mail ako obrázok. | Martin Gregor | 19/10/12 | normálna | ukončená | 8 | Bežiaci stránka na správnej url, naplnená podľa pokynov. |
| 1.2 | Rozbehať server na RoboCup. | Martin Gregor | 17/10/12 | vysoká | ukončená | 13 | Nainštalovaný server u všetkých, na ktorom vedú pustiť hráča. |
| 1.3 | Získať projekt od konkurenčného tímu z posledného roka. | Matej Červeňák | 17/10/12 | vysoká | ukončená | ½ | Získané zdrojové súbory. |
| 1.4 | Spustiť hráča na server | Martin Gregor | 17/10/12 | vysoká | ukončená | 3 | U každého člena spustený hráč na svojom serveri. |
| 1.5 | Vytvoriť pohyb alebo upraviť existujúci pohyb hráča v editore. | vsetci | 24/10/12 | normálna | ukončená | 3 | U každého vytvorený pohyb svojho hráča. |

| | | | | | | | |
|------|---|--|----------|----------|----------|---|--|
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | Filip, Matej, Gabor | 17/10/12 | normálna | ukončená | 1 | Písomná analýza minimálne 3 tímov. |
| 1.7 | Študovať wiki k RoboCup-u. | všetci | 17/10/12 | normálna | ukončená | ½ | Každý člen povie niečo zaujímavé, čo sa dočítal na wiki. |
| 1.8 | Študovať zahraničné tímy a zápasy. | všetci | 17/10/12 | normálna | ukončená | 3 | Analýza minimálne 6 tímov zo zahraničia so zaujímavými prístupmi |
| 1.9 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | Matúš Ondrejko, Jaroslav Grega, Filip Sucháč | 24/10/12 | normálna | ukončená | 3 | Prehľad existujúcich prístupov + analýza minimálne 2 prístupov reálnych modelov. |
| 1.10 | Rozbehať Git | Martin Gregor | 17/10/12 | normálna | ukončená | ½ | Založený repozitár na git |
| 1.11 | Nainštalovať Git u seba. | všetci | 24/10/12 | normálna | ukončená | ½ | - |
| 1.12 | Rozbehať manažérsky softvér (Redmine?) | Gábor Nagy | 17/10/12 | vysoká | ukončená | 5 | Každý člen tímu má prístup do manažérského softvéru. |
| 1.13 | Zabezpečiť aktuálny editor pohybov | - | 31/11/12 | vysoká | ukončená | - | Editor dostupný pre všetkých členov |

Rozdelenie získaných bodov za dokončené úlohy šprintu 1:

| ID | Popis | Zodpovední | Celkový počet bodov | Rozpis bodov |
|-----|--------------------------|---------------|---------------------|------------------|
| 1.1 | Rozbehať webovú stránku. | Martin Gregor | 8 | Martin Gregor: 8 |

| | | | | |
|------|---|---|-----|--|
| 1.2 | Rozbehať server na RoboCup. | Martin Gregor | 13 | Jaroslav Grega: 5,5 Ostatní: 1,5 |
| 1.3 | Získať projekt od konkurenčného tímu z posledného roka. | Matej Červeňák | 3 | Všetci: 0,5 |
| 1.4 | Spustiť hráča na server | Martin Gregor | 3 | Všetci: 0,5 |
| 1.5 | Vytvoríť pohyb alebo upraviť existujúci pohyb hráča v editore. | vsetci | 3 | Filip Sucháč: 1 Matej Červeňák: 1 Gábor Nagy: 1 |
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | Filip, Matej, Gabor | 3 | Všetci: 0,5 |
| 1.7 | Študovať wiki k RoboCup-u. | vsetci | 3 | Všetci: 0,5 |
| 1.8 | Študovať zahraničné tímy a zápasy. | vsetci | 3 | Všetci: 0,5 |
| 1.9 | Analyzovať robotov a prečítať si o metódach učenia, stabilizácie... | Matúš Ondrejkovič, Jaroslav Grega, Filip Sucháč | 3 | Matúš Ondrejkovič: 1 Jaroslav Grega: 1 Filip Sucháč: 1 |
| 1.10 | Rozbehať Git | Martin Gregor | 0,5 | Martin Gregor: 0,5 |
| 1.11 | Nainštalovať Git u seba. | vsetci | 3 | Všetci: 0,5 |
| 1.12 | Rozbehať manažérsky softvér | Gábor Nagy | 5 | Gábor Nagy: 5 |

Opis stretnutia:

- Prezentovanie výstupov úloh šprintu 1
- Určenie nového spôsobu priradovania zodpovednosti za jednotlivé úlohy
- Návrh úloh na šprint 2

Stanovené úlohy na šprint 2:

| ID | Popis | Zodpovední | Dátum splnenia | Priorita | Stav | Body | Výstup |
|-----|--|--------------------------|----------------|----------|------|------|--|
| 2.1 | Návrh celkovej práce počas projektu | Matej Č. + všetci | 7/11/12 | vysoká | nová | 8 | Spísaný hrubý návrh, ktorý je prediskutovaný |
| 2.2 | Git – importovanie a kríženie hráčov | Matúš, Martin | 14/11/12 | normálna | nová | 3 | Funkčný importovaný hráč |
| 2.3 | Analýza z. kódu hráča tímov High5 a Tím 17 žije... | Martin + všetci | 14/11/12 | vysoká | nová | 20 | Zoznam funkčných resp. nefunkčných vecí |
| 2.4 | Vytvoriť dokumentáciu aj so šablónou | Matej | 14/11/12 | vysoká | nová | 3 | Dokumentácia obsahujúca požadovanú štruktúru a doteraz vytvorené dokumenty |
| 2.5 | Analýza chôdze nášho (kríženého) hráča | Gábor N. | 14/11/12 | normálna | nová | 8 | Spísaná analýza. Návrh možností vylepšenia |
| 2.6 | Koding guide (šablóna-metodika) | Filip | 14/11/12 | normálna | nová | 3 | Krátka metodika (tutoriál) |
| 2.7 | Metodika na používanie Git-u | Matúš, Martin | 14/11/12 | normálna | nová | 3 | Krátka metodika (tutoriál) |

Poznámky:

- Pole „Zodpovední“ označuje hrubým písmenom hlavného zodpovedného za úlohu a normálnym písmom ďalších pomocníkov.
- Na základe nezhôd bodov za jednotlivé úlohy sme spravili revíziu troch úloh zo šprintu 1.

| | | | | | | | |
|------|--|---------------------|----------|----------|----------|---|--|
| 1.6 | Študovať dokumentáciu a projekty z minulých rokov. | Filip, Matej, Gabor | 17/10/12 | normálna | ukončená | 3 | Písomná analýza minimálne 3 tímov. |
| 1.7 | Študovať wiki k RoboCup-u. | všetci | 17/10/12 | normálna | ukončená | 3 | Každý člen povie niečo zaujímavé, čo sa dočítal na wiki. |
| 1.11 | Nainštalovať Git u seba. | všetci | 24/10/12 | normálna | ukončená | 3 | - |

6.5 Zázpis z 5. stretnutia

| | |
|----------------|--|
| Dátum: | 7.11.2012 |
| Trvanie: | 8:00 – 10:00 |
| Miestnosť: | blok D – softvérové štúdio |
| Prítomní | |
| Pedagóg: | Ing. Marián Lekavý, PhD. |
| Členovia tímu: | Matej Červeňák, Jaroslav Grega, Martin Gregor, Gábor Nagy, Matúš Ondrejko, Filip Sucháč |
| Neprítomní: | - |
| Zapisovateľ: | Filip Sucháč |

Téma stretnutia:

Kontrola priebežného stavu úloh z druhého šprintu a debata o návrhu/špecifikácii celkovej práce projektu.

Priebežný stav úloh šprintu:

1. Úloha 2.2 Git – importovanie a kríženie hráčov:

Gábor povedal, že v hráčovi tímu „High5“ bolo viac zdrojového kódu, tak ho porovnal s hráčom tímu „Tím 17 žije...“. Hráča tímu „High5“ dal do repozitára s pridaním funkcionality druhého hráča, ktorá v „High5“ nebola. Bol to zatiaľ len syntaktický merge.

2. Merge Test-framework-u zatiaľ robený nebol.

3. Matej hovoril o odovzdávaní dokumentácie o týždeň. Bude za to zodpovedný. Marián povedal, že pri tlačenej dokumentácii je najlepšie použiť rozširovateľné viazanie.

4. Úloha 2.3 Analýza z. kódu hráča tímov „High5“ a „Tím 17 žije...“:

Tím má prácu na úlohe rozdelenú. Martin povedal, že zistil, že sa „RoboCup Library“ už nepoužíva.

5. Vymenovali sme úlohy, na ktorých sa pracuje a ktoré ešte nie sú začaté.

6. Úloha 2.5 Analýza chôdze nášho (kríženeho) hráča:

Marián povedal, že chôdzu je nutné analyzovať spustením hráča cez server a aj cez Test-framework. Na tejto analýze by sa mal podieľať človek, ktorý v prvom šprinte analyzoval chôdzu robotov všeobecne (bol to Matúš).

Debata o návrhu/špecifikácii celkovej práce projektu:

Táto debata vychádzala z bodov, ktoré boli výstupom úlohy 2.1 Návrh celkovej práce počas projektu. K týmto bodom sa Marián vyjadril nasledovne.

1. Optimalizácia vytvorených nízkoúrovňových pohybov
 - a. Je to možné ručne alebo evolučkom.

- b. Prístupy: statický (akým kĺbom koľko pohnúť), dynamický (sledovať chyby pohybov a podľa toho upravovať), úplne dynamický (vypočítavať, ako zmeniť kĺby z jedného stavu hráča na druhý), malé statické pohyby skladať na vyššie.
 - c. Dynamické prístupy sú náročné, ale kvalitou lepšie. Tie by Marián odporúčal.
 - d. Dynamické pohyby je možné vytvárať zo statických, na ktorý sa aplikuje nejaký diff.
 - e. Strojové učenie – je náročné.
2. Vytvorenie nových pohybov
 - a. Beh: s ním opatrne, pretože to nemusí byť na tomto hráčovi fyzikálne možné. Je to náročné na stabilizáciu. Niektorí sa na to môže pozrieť, ale nemusí sa to podariť.
 - b. Otáčanie hráča počas chôdze: najprv revíziou pohybov zistiť, či funguje. Inak tento návrh môže byť.
 - c. Bránenie: pozrieť sa na diplomovku o brankárovi.
 3. Vytvorenie univerzálneho trénera (navrhol Jaro)
 - a. Je zakázané dávať hráčom všetky informácie o svete.
 4. Nástroj pre tréning strategii (navrhol Matúš)
 - a. Formáciu formovať podľa pozície lopty.
 - b. Zdrojový kód ohľadom formácií by bolo možné získať z 2D hráčov.
 5. Nástroj RoboViz (navrhol sa ním zaoberať Martin)
 - a. Matej sa spýta minuloročných na problém s kompiláciou Simsparku pre RoboViz.
 - b. Marián poradil radšej rozvíjať Test-framework.
 - i. Zlepšovanie usability, presnosti zobrazenia.
 - ii. Vizualizovať odchýlku, ruchy, absolútne pozície hráčov.
 6. Vylepšenie editora pohybov
 - a. Kontrola stability vytvoreného pohybu (navrhol Matúš)
 - i. Kontrola by sa musela odsimulovať na serveri alebo inak.
 - ii. Kontrola by mohla byť vykonaná aj výpočtom, napr. či je ťažisko v rámci oporných bodov.
 - b. Zvýšiť použiteľnosť (navrhol Gábor) - vylepšenie robenia pohybov pre začiatočníkov.
 - i. K tomuto by bolo potrebné presne navrhnúť features.
 - c. Spraviť editor pohybov odolný voči zmene štruktúry XML pohybov (navrhol Filip)
 - i. Je to ťažký cieľ, možno až nereálny.
 7. Zaviesť do hráča kontrolu stability (navrhol Gábor) – vedieť hranice, kedy hráč stráca stabilitu a zabraňovať tomu.
 - a. Takáto kontrola by musela závisieť od toho, čo hráč práve robí.
 - b. Mohol by sa to hráč učiť – na to by sa musel vytvoriť proces, akým by to fungovalo.
 8. Marián odporučil stanoviť si priority k cieľom a prioritnejšie by mali byť nižšie pohyby.

Ďalšie úlohy:

1. Je potrebné zlepšiť náš scrum

- a. Vytvárať backlog a user stories (kto požaduje čo a prečo, napr. „Hráč chce chôdzu, aby sa vedel pohybovať.“)
 - b. Navrhovať user stories také, aby boli zvládnuteľné jedným šprintom.
2. Zmeniť scrum mastera (navrhol Gábor)
 - a. Dohodnúť si poradie. Po každom šprinte vystriedať scrum mastera.
3. Zaviesť si tímové rutiny, ako napr. pri scrum pokri.
4. Úlohu 2.1 „Návrh celkovej práce počas projektu“ predefinovať z návrhu na podrobnú špecifikáciu (chceme toto a preto, toto a preto, ...).
 - a. Výstupy by mali byť špecifikácia a user stories.

Poznámky:

1. Rozhodnutie zmeniť body pre úlohy vždy najprv konzultovať s Mariánom.

7 Preberacie protokoly

Preberací protokol

Tímový projekt I

Projekt: RoboCup 3D

Odovzdávajúci tím: A55 Kickers (tím č. 15)

Preberajúci: Ing. Marián Lekavý, PhD.

Dátum odovzdania: 14.11.2011

Odovzdané dokumenty:

- Tlačená dokumentácia k inžinierskemu dielu v rozsahu 59 strán
- Tlačená dokumentácia k riadeniu projektu v rozsahu 72 strán

Vedúci tímu

Ing. Marián Lekavý, PhD.