

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

FIIT KINECT Connection kin

Dokumentácia k riadeniu

Vedúci tímu:

Ing. Michal Kottman

Členovia tímu:

Bc. Ivana Bohunická

Bc. Ján Greppel

Bc. Juraj Muránsky

Bc. František Nagy

Bc. Dominik Rerko

Bc. Matúš Ujhelyi

Bc. Zuzana Ujhelyiová

Kontakt: timak_tim11@googlegroups.com

Obsah

1	Úvod	1-1
1.1	Účel dokumentu	1-1
2	Ponuka	2-1
2.1	Členovia tímu	2-1
2.2	Motivácia – FIIT Kinect	2-2
2.3	Koncept riešenia – FIIT Kinect	2-3
2.4	Motivácia – Odporúčanie pre inteligentnú TV	2-4
2.5	Koncepcia riešenia – Odporúčanie pre inteligentnú TV	2-5
3	Zápisy zo stretnutí	3-1
3.1	Šablóna zápisnice	3-1
3.2	Zápis zo stretnutia č. 1	3-3
3.3	Zápis zo stretnutia č. 2	3-5
3.4	Zápis zo stretnutia č. 3	3-7
3.5	Zápis zo stretnutia č. 4	3-10
3.6	Zápis zo stretnutia č. 5	3-12
3.7	Zápis zo stretnutia č. 6	3-15
3.8	Zápis zo stretnutia č. 7	3-18
3.9	Zápis zo stretnutia č. 8	3-22
4	Úlohy členov tímu	4-1
4.1	Dlhodobé úlohy	4-1
4.2	Krátkodobé úlohy	4-1
5	Manažment rozvrhu	5-1
5.1	Dôležité termíny	5-1
5.2	Product backlog	5-1
5.3	Výstupy jednotlivých šprintov	5-2
5.4	Harmonogram úloh	5-3
5.4.1	Šprint 0- Biela veľryba	5-3
5.4.2	Šprint 1- Čierna ovca	5-3
5.4.3	Šprint 2- Ružový panter	5-4
5.4.4	Šprint 3- Fialová krava	5-6

5.4.5	Šprint 4- Zlatá rybka	5-8
6	Manažment rizík	6-1
6.1	Identifikované riziká pre projekt	6-1
7	Monitorovanie projektu.....	7-1
7.1	Vymedzenie pojmov v manažmente úloh	7-1
7.2	Typy úloh	7-2
7.3	Priority úloh	7-2
7.4	Roly vystupujúce v rámci manažmentu úloh.....	7-2
7.5	Proces manažmentu úloh	7-3
8	Manažment komunikácie.....	8-1
8.1	Schôdze.....	8-1
8.2	Mail	8-1
8.3	Interná wiki.....	8-2
8.4	Telefón a Skype	8-2
8.5	Dokumentový server a GIT.....	8-2
8.6	Oficiálna stránka	8-2
8.7	Topológia zainteresovaných v projekte	8-2
9	Manažment požiadaviek	9-1
9.1	Pohovor s produktovým vlastníkom	9-1
9.2	Prioritizácia základných požiadaviek produktu.....	9-2
9.3	Šprint	9-2
9.4	Validácia požiadaviek.....	9-2
9.5	Definícia používateľských príbehov v Microsoft Word.....	9-2
9.6	Šablóna používateľských príbehov	9-3
10	Manažment kvality	10-1
10.1	Konvencie pre písanie zdrojového kódu	10-1
10.2	PMD.....	10-4
10.3	Cobertura	10-5
10.4	Dohodnuté konvencie názvov v GUI.....	10-6
11	Manažment dokumentácie	11-1

11.1	<i>Tvorba dokumentácie</i>	11-1
11.2	<i>Tvorba zápisnice zo stretnutia</i>	11-1
11.3	<i>Dokumentácia zdrojových kódov v JavaDoc - u</i>	11-1
11.3.1	<i>Značky definované v JavaDoc</i>	11-1
11.3.2	<i>Tvorba komentárov zdrojového kódu</i>	11-4
12	Manažment podpory vývoja	12-1
12.1	<i>Nightly build</i>	12-1
12.2	<i>Git tutoriál</i>	12-2
12.3	<i>Maven</i>	12-5
12.4	<i>Jigloo GUI Builder</i>	12-7
13	Manažment chýb (Príloha A)	13-1
13.1	<i>Úvod</i>	13-1
13.2	<i>Použité pojmy</i>	13-1
13.3	<i>Súvisiace metodiky</i>	13-1
13.4	<i>Roly a zodpovednosti</i>	13-1
13.5	<i>Životný cyklus chyby</i>	13-2
13.6	<i>Proces nahlásenia chyby</i>	13-3
13.7	<i>Proces pridelenia a opravy chyby</i>	13-3
13.8	<i>Proces verifikácie opravy</i>	13-4
13.9	<i>Proces doručenia opraveného inkrementu</i>	13-4
13.10	<i>Proces vytvorenie chyby v systéme redmine</i>	13-5
13.11	<i>Vyriešenie chyby v systéme redmine</i>	13-6
14	Manažment iterácií projektu (Príloha B)	14-1
14.1	<i>Úvod</i>	14-1
14.2	<i>Pojmy</i>	14-1
14.3	<i>Použitá literatúra</i>	14-1
14.4	<i>Roly a zodpovednosti</i>	14-2
14.5	<i>Proces plánovania metódou Scrum</i>	14-2
14.6	<i>Plánovanie metódou Scrum v Redmine</i>	14-7
14.7	<i>Vytváranie úloh v Redmine</i>	14-9
14.8	<i>Ukončenie iterácií projektu v Redmine</i>	14-10

15	Manažment úloh v jednotlivých etapách projektu (Príloha C)	15-1
15.1	Vymedzenie pojmov.....	15-1
15.2	Súvisiace metodiky.....	15-2
15.3	Roly vystupujúce v rámci manažmentu úloh.....	15-2
15.4	Vymedzenie typov úloh a priorit.....	15-3
15.5	Proces manažmentu úloh.....	15-4
15.6	Opis jednotlivých krokov v rámci manažmentu úloh.....	15-4
15.7	Činnosti spojené s manažmentom úloh.....	15-7
15.8	Vykazovanie postupu prác na úlohe v systéme Redmine.....	15-8
15.9	Kontrola postupu prác na úlohe v systéme Redmine.....	15-10
16	Manažment testovania (Príloha D)	16-1
16.1	Úvod.....	16-1
16.2	Použité pojmy.....	16-1
16.3	Roly a zodpovednosti.....	16-2
16.4	Použité procesy.....	16-2
16.5	Podrobný popis krokov.....	16-4
16.6	Použitie testovacieho frameworku JUnit.....	16-9
16.7	Záver.....	16-11
17	Šprint review (Príloha E)	17-1
18	Preberací protokol (Príloha G)	18-1

1 Úvod

1.1 Účel dokumentu

Tento dokument je vytvorený pre potreby zdokumentovania riadenia projektu Fiit Kinect. Úlohou je vymedzenie postupov a procesov pre jednotlivých členov, čím sa má zabezpečiť správny postup k dosiahnutiu funkčného produktu podľa požiadaviek vedúceho projektu Ing. Michala Kottmana. Projekt spadá pod predmety Tímový projekt I a Tímový projekt II, ktoré absolvujeme v akademickom roku 2012/2013 inžinierskeho štúdia.

2 Ponuka

2.1 Členovia tímu

Ivana Bohunická – záleží jej na estetickosti v IT produktoch, čiže sa zaujíma o používateľské rozhrania, o vizualizáciu v počítači, o interakciu človeka s PC. Téma Kinect rozhodne súvisí so spracovaním obrazu a interakciou človeka s PC, prináša novú formu grafického rozhrania a ovládania pohybom a hlasom. Aj vo svojej bakalárskej práci sa zaoberala vizualizáciou. V súčasnosti má trojmesačné pracovné skúsenosti na pozícii Java programátor, kde pracuje v tíme na rozsiahlom IS.

Juraj Muránsky – už na strednej škole sa zaujímal o optimalizované programovanie v súčinnosti s obrazom, kde v Delphi naprogramoval rôzne hry. Odvtedy číta literatúru zaoberajúcu sa optimálnym spracovaním a prekreslením obrazu. Profesionálne sa venuje web programovaniu už piaty rok a popri obyčajných stránkach pocítil potrebu tvorby aj inteligentnejších stránok, ktoré agregujú informácie, sledujú správanie používateľa a na základe toho robia aj určité odporúčania. Počas tejto doby napísal už aj zopár jednoduchých crawlerov a študoval aj zložitejšie.

Matúš Ujhelyi – trištvrte roka sa podieľal na návrhu a implementácii veľkého sieťovo-informačného logistického riešenia (projekt APVV - ILOS) zastrešujúceho viaceré spôsoby komunikácie medzi modulmi aplikácie a samotným hardvérom. Z uvedeného aj iných projektov má skúsenosti s protokolmi reálneho času, viacerými druhmi architektúr, databáz ako aj s prezentačnou vrstvou v prostredí JEE. Momentálne sa venuje vývoju ovládačov na linuxové jadro pre špecifickú platformu.

František Nagy – od strednej školy sa zaujíma o programovanie ako také. Bakalárske štúdium vyštudoval aplikovanú informatiku na Fakulte matematiky, fyziky a informatiky Univerzity Komenského, takže do tímu vnesie trochu iného myslenia. Momentálne sa v práci najviac zaoberá programovaním pre mobilné zariadenia so systémom Android, ale zaujíma sa o všetky technológie.

Ján Greppel – od strednej školy sa venuje tvorbe webových stránok a portálov. Pracoval aj na väčších projektoch (do 10 ľudí, do 2 rokov) ako programátor i vedúci. V posledných rokoch sa sústreďuje na objektovo-orientovaný návrh a využívanie aplikačných rámcov (angl. frameworks) pre webové rozhrania. V študentskej organizácii Ynet vyvíja informačný systém pre správu všetkých užívateľov a taktiež vedie tento projekt.

Dominik Rerko – od svojich informatických začiatkov sa zaujíma o oblasť vizualizácie a spracovania obrazu, aj keď nie v až tak širokom rozsahu. Úzko súvisiacou vizualizáciou

2 | Ponuka

dát sa čiastočne zaoberal aj vo svojej bakalárskej práci. Zaujíma sa tiež o oblasť mobilných aplikácií, ktorá by v konečnom dôsledku mohla byť prínosom.

Zuzana Ujhelyiová – v rámci svojej praxe pracovala na viacerých projektoch. Ako prvý bol vývoj fakturačného systému pre telekomunikačného operátora (v jazyku Visual Basic). Pracovala aj na vývoji webových aplikácií (informačný systém) v jazyku Java (s využitím HTML v rámci klientskej časti, MYSQL databáz a webových služieb). V súčasnosti sa zaoberá v profesijnej oblasti vývojom sieťových aplikácií v Jave.

2.2 Motivácia – FIIT Kinect

Kinect zaujal väčšinu členov nášho tímu už pri prvom vzhliadnutí ponúkaných tém. Niektorí boli touto tematikou oslovení už skôr, keď sme videli minuloročný tímový projekt na konferencii IIT.SRC, kde Kinect predviedli. Boli sme nadchnutí samotnou myšlienkou.

Téma je veľmi zaujímavým prepojením novej technológie a reálneho života. Predstava aplikácie ovládanej hlasom či vizuálnym podnetom je úplne nový trend v oblasti softvérového inžinierstva, aj keď myšlienka tu bola už dávnejšie. To považujeme za výzvu, avšak s dosiahnuteľnými a realistickými cieľmi. Naplnenie tejto výzvy vidíme ako produkt, ktorý možno priamo poskytnúť používateľovi, čo je pre nás veľmi motivujúce.

Pre niektorých členov tímu je takáto téma aj čosi neznáme. My sa však toho nebojíme. Teší nás, že by sme sa naučili niečo nové a mali možnosť pracovať na pútavej téme. Mnohí členovia majú vnútornú potrebu pracovať na novej a jedinečnej problematike, zároveň sa nám páči, že by sme mali možnosť pracovať s novými technológiami. Pri bežných zadaniach sa ťažko stretáme s takouto technológiou.

Sme názoru, že téma je veľmi aktuálna a predstavuje istú výzvu do budúcnosti. Dôkazom sú aj iné projekty rozbehnuté po svete, ktoré sme objavili pri prvotnom skúmaní danej problematiky¹. Tému zároveň chápeme aj ako veľmi modulárnu, skladajúcu sa z rôznych častí, viac či menej prepojených. Tu vidíme príležitosť pre každého člena tímu sa zapojiť a dokonca si nájsť niečo pre jeho osobnosť najvhodnejšie.

Môžeme povedať, že v tíme máme člena, ktorého teší, že by analyzoval takúto problematiku. Máme aj členov zaujímavých sa o počítačovú grafiku a interakciu človeka s počítačom. Tím, ktorý sa chce zaujímať o tému ako je Kinect potrebuje isto aj človeka so znalosťami z oblasti hardvéru a sietí. V našom tíme sú ľudia, ktorí majú skúsenosti

¹Kinect-powered Email, <http://www.youtube.com/watch?v=xWG7edyw7B8>

2 | Ponuka

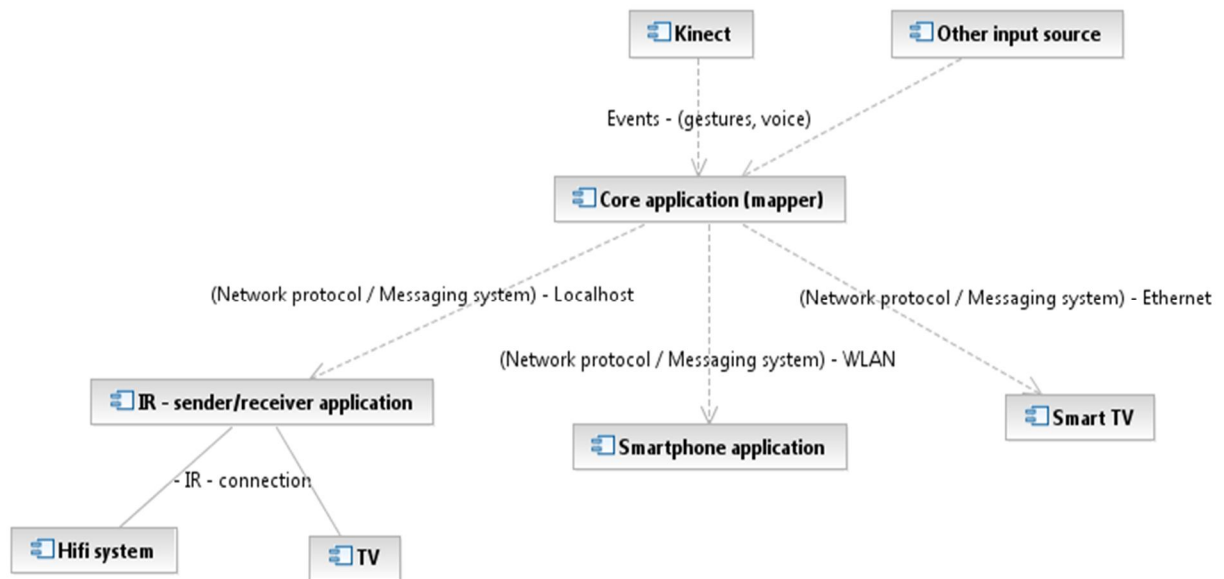
s problematikou sietí a komunikácie a preto veríme, že budú prínosom. Väčšina členov má aj praktické skúsenosti s prácou v tíme a preto sa považujeme za tím, ktorý je schopný túto tému kvalitne vypracovať.

2.3 Koncept riešenia – FIIT Kinect

Po spoločnej diskusii v tíme na túto tému sme prišli k relatívne jednoznačným záverom. Výsledok tejto témy vidíme vo vysoko modulárnom softvérovom systéme bežiacom na jednom osobnom počítači, na ktorý by sa ale mohli pripájať po štandardných sieťach ďalšie klientské aplikácie. Základným kameňom by bola aplikácia schopná na jednej strane prijímať podnety vo forme gest alebo zvukových povelov zo senzora Kinect a na druhej strane notifikovať a komunikovať s inými aplikáciami cez štandardné sieťové protokoly alebo systémy správ. Takéto rozvrhnutie modulov poskytne možnosti pre jednoduché budúce rozšírenie inými modulmi, ktoré by boli vlastne len novými aplikáciami pripájajúcimi sa na hlavnú aplikáciu.

Poskytne sa tak možnosť zdieľania používateľského prostredia, akým by bol Kinect, pre rôzne platformy a zariadenia. Jednotné gestá alebo zvukové povely použité pre viaceré zariadenia sú zjednodušeným štandardizovaným procesom ako ovládať rôzne druhy zariadení. Zabezpečenie jednotnosti prenáša znalosť „ako ovládať zariadenie“ z používateľa na tvorcov systému. Proces ovládania samostatne zabezpečujú jednotlivé klientské moduly. Jednotlivé moduly sa tak stávajú samostatnými aplikáciami, čo vylepšuje aj stabilitu a reakcie systému v reálnom čase. Každá z klientských aplikácií sa môže sama prispôbiť zariadeniu, na ktorom bude bežať alebo, s ktorým bude komunikovať. Spôsob mapovania jednotlivých gest na akcie vyvolané na zariadení očakávame ako výsledok analýzy.

Sme schopní navrhnuť viaceré riešenia zaoberajúce sa touto problematikou. Jednou z možností je vybrať si pomocou napevno nadefinovaných gest zariadenia na ovládanie ako aj akciu vykonateľnú na zariadení. Inou možnosťou by bolo navrhovanie vlastných gest používateľom a následným mapovaním týchto gest na jednotlivé akcie. Definovanie vlastných udalostí, ktoré je možné vykonávať na zariadeniach ale považujeme za špecificky závislé na klientských moduloch. Tieto ale aj ďalšie špecifiká očakávame až ako výsledok analýzy problému a očakávania zákazníka. Zamýšľanú koncepciu najlepšie vyjadríme jednoduchým diagramom.



Obrázok 2-1: Koncept riešenia

2.4 Motivácia – Odporúčanie pre inteligentnú TV

Téma inteligentnej TV sa v internej diskusii nášho tímu stretla s veľmi pozitívnym ohlasom. Všetci sme si hneď predstavili množstvo nadbytočného multimedialneho obsahu, s ktorým prichádzame každý deň do styku a ako by sme sa ho konečne zbavili. Pri malom množstve času, ktorý nám zostáva na prezeranie multimédií by bolo ideálne, keby sme ho nemuseli strácať výberom programu, ktorý budeme sledovať. Bolo by super mať aplikáciu, ktorá nám oznámi, že o 2 hodiny ide na 38. kanáli nová časť nášho obľúbeného televízneho seriálu.

V rámci tímu máme viacerých členov, ktorí sa dlhšiu dobu venovali, resp. stále venujú vývoju v oblasti webových technológií, čo by bolo určite prínosom pri riešení tejto témy.

Mať priestor, ktorý by ponúkal náhľad na multimedialny obsah v roztriedenej a prehľadnej forme, by používateľom určite uľahčilo rozhodovanie pri akom programe strávia svoj voľný čas. Notifikovanie o obľúbených programoch v dostatočnom predstihu by tiež bolo veľmi praktické.

Sme názoru, že téma poskytuje dostatočný priestor pre kreatívnu prácu a prístup k riešeniu s použitím viacerých technológií a platforiem. Veríme, že každý z členov nášho tímu by si na tejto téme našiel to svoje, vedel by prispieť vhodnými nápadmi a tým by sa nám podarilo vytvoriť ucelený výstup, ktorý by mohol mať aj reálne využitie.

2.5 Konceptcia riešenia – Odporúčanie pre inteligentnú TV

Základná koncepcia by sa mohla rozdeliť do 4 častí:

- zberač údajov,
- vyhľadávač,
- vyhodnocovač,
- prezentátor.

Zberač údajov by parsoval záznamy, ktoré by boli k dispozícii a poskytoval aplikačné rozhranie pre ostatné dve časti. Tento zberač by sa taktiež staral aj o prípadé ukladanie novo-získaných údajov do databázy na základe interakcie používateľa.

Za získavanie informácií z rôznych zdrojov by bol zodpovedný vyhľadávač, kde by získaval údaje o rozvrhoch daného multimediálneho obsahu (televízne programy z internetových stránok televízií, rozhovory v rádiu a pod.).

Vyhodnocovač by, na základe dostupných údajov a modelu používateľa, robil rozhodnutia pre výber samotných odporúčaní. V prípade, že by sa nevedel rozhodnúť, identifikoval by chýbajúcu informáciu a snažil by sa ju získať od používateľa. Takéto získavanie informácií by prebiehalo nepriamo, napr. cez navrhovanie experimentálnych možností alebo vyhodnocovanie ich úspešnosti a priradovanie pravdepodobnosti stupňa záujmu o daný obsah.

Priamu komunikáciu s používateľom by zastrešoval prezentátor. Zobrazoval by relevantné informácie a notifikoval by používateľa o prichádzajúcich programoch cez e-mail, mobilnú aplikáciu alebo webové rozhranie. Webové rozhranie by zobrazovalo všetky zozbierané údaje z vyhľadávača a taktiež aj preferovaný rozvrh. Zberač a vyhodnocovač by sa realizovali ako webová služba, ktorá by komunikovala s prezentátorom. Dôvod pre toto riešenie by bola rôznorodosť zariadení, ktoré by mohli vystupovať ako prezentátor. Externé zariadenia by mohli zároveň slúžiť na notifikáciu používateľov.

3 Zápisy zo stretnutí

V tejto kapitole sú všetky zápisy zo stretnutí vrátane šablóny pre vytvorenie zápisnice. Obsahujú informácie ako: téma, dátum, čas a miesto stretnutia, zoznam účastníkov a zapisovateľa. V pár bodoch je popísaný vždy priebeh stretnutia, zadanie úloh a zhodnotenie úloh z predchádzajúceho stretnutia. Na záver stretnutia je vždy krátke zhodnotenie alebo zhrnutie stretnutia.

3.1 Šablóna zápisnice

Zápis zo stretnutia č. Y

Téma stretnutia: Stretnutie v strede X. šprintu | Plánovanie X. šprintu

Dátum stretnutia:

Čas stretnutia:

Dĺžka stretnutia:

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
 Ivana Bohunická Ján Greppel Juraj Muránsky
 František Nagy
 Ing. Michal Kottman

Zapisovateľ:

Vedúci stretnutia:

Priebeh stretnutia:

- úloha

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Šprint X: Šprint X-1: príklady		OK Pokračuje Zrušená Predĺžovaná Nedokončená Zmena riešiteľa		
Napísať dokumentáciu	Všetci	OK		
Šprint 1: Knižnica XY	Janko Hrasko	Pokračuje		

3 | Zápisy zo stretnutí

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Šprint X: (bez)				

Zhodnotenie stretnutia:

3.2 Zápis zo stretnutia č. 1

Téma stretnutia: Plán projektu, rozdelenie úloh, analýza problému

Dátum stretnutia: 10.10.2012

Čas stretnutia: 9:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Ján Greppel

Vedúci stretnutia: Ing. Michal Kottman

Priebeh stretnutia:

- vedúci poskytol všeobecné informácie o minuloročnom projekte, ktorého časť bude použitá ako súčasť nášho riešenia
- definovanie manažérskych úloh a zodpovedností
- výber nástroja na manažment verzii – GIT, výber nástroja na manažment úloh – Redmine
- spísanie základných požiadaviek na systém v podobe používateľských príbehov. Najprioritnejšie: komunikácia po sieti, ukladanie konfigurácie, detegovanie gest, odosielanie príkazov, spúšťanie programov.

3 | Zápisy zo stretnutí

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Poslať dokumenty a kódy z minulého roku	Zuzana Ujhelyiová	10.10	ASAP	ihneď
Tímová web - stránka	Ján Greppel	10.10	21.10	8h
Spojzdniť server	Ján Greppel	10.10	17.10	
Vytvoriť projekt na gitbus.fiit.stuba.sk	Matúš Ujhelyi	10.10	17.10	2h
Pripraviť git a napísať návod na Git	Matúš Ujhelyi	10.10	17.10	
Zapísať úlohy do Redmine	Ivana Bohunická	10.10	17.10	1h
Získať prístupy do Gitbusu pre Zuzku a Jura	Michal Kottman	10.10	17.10	ihneď
Analýza existujúceho riešenia	všetci	10.10	17.10	7x3h
Nainštalovať Git, vygenerovať kľúče, pridať kľúče na gitbus	všetci	10.10	17.10	7x30min

Zhodnotenie stretnutia:

Výstupom stretnutia bolo oboznámenie sa s témou, nárokmi a požiadavkami súvisiacimi s výsledkom projektu. Boli rozdelené úlohy do ďalšieho stretnutia a spísaná prvá verzia productbacklog - u.

3 | Zápisy zo stretnutí

3.3 Zápis zo stretnutia č. 2

Téma stretnutia:	Prioritizácia product backlogu, dátový model, vytvorenie kostry projektu Plánovanie 1. šprintu		
Dátum stretnutia:	17.10.2012		
Čas stretnutia:	9:00		
Dĺžka stretnutia:	180 min.		
Miesto stretnutia:	softvérové štúdio		
Účastníci:	Dominik Rerko Ivana Bohunická František Nagy Ing. Michal Kottman	Zuzana Ujhelyiová Ján Greppel	Matúš Ujhelyi Juraj Muránsky
Zapisovateľ:	Ján Greppel		
Vedúci stretnutia:	Ing. Michal Kottman		

Priebeh stretnutia:

- prezentácia minuloročného projektu, ktorého časť bude slúžiť ako základ pre rozpoznávanie gest
- definovanie pravidiel šprintu a naplánovanie úloh
- spolu s pedagogickým vedúcim tímu sme prioritizovali už definované požiadavky
- vytvorenie prvotného dátového modelu

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Poslať dokumenty a kódy z minulého roku	Zuzana Ujhelyiová	OK	ihneď	ihneď
Tímová web- stránka	Ján Greppel	OK	8h	11h
Spojzdníť server	Ján Greppel	OK		
Vytvoriť projekt na gitbus.fiit.stuba.sk	Matúš Ujhelyi	OK	2h	3h
Napísať návod na GIT	Matúš Ujhelyi	OK		
Zapísať úlohy do Redmine	Ivana Bohunická	OK	1h	1,5h
Získať prístupy do Gitbusu pre Zuzku a Jura	Michal Kottman	OK	ihneď	ihneď
Analýza existujúceho riešenia	všetci	OK	7x3h	Individuálne
Nainštalovať GIT, vygenerovať kľúče, pridať kľúče na gitbus	všetci	OK	7x30min	Individuálne

3 | Zápisy zo stretnutí

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Extrakcia dôležitých častí z minuloročného projektu	Matúš Ujhelyi	17.10	24.10	13h
Návrh protokolu	všetci	17.10	22.10	7x6h
Pripraviť projektový plán	Ivana Bohunická	17.10	24.10	2h
Uverejniť schôdze- na stránke	Ján Greppel	17.10	24.10	ihneď
Založiť všetky dokumentácie	Dominik Rerko	17.10	24.10	2h
Zdokumentovanie návrhu protokolu	Zuzana Ujhelyiová	22.10	24.10	2,5h
Pripraviť v GITE štruktúru projektu	František Nagy	17.10	24.10	30min
Pripraviť podklady pre svoju manažérsku pozíciu	všetci	17.10	24.10	individuálne

Zhodnotenie stretnutia:

Na stretnutí bol vytvorený plán prvého šprintu a prioritizovali sme si product backlog.

3.4 Zápis zo stretnutia č. 3

Téma stretnutia: Komunikačný protokol, mapovanie gest
Stretnutie v strede 1.šprintu

Dátum stretnutia: 24.10.2012

Čas stretnutia: 9:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Zuzana Ujhelyiová

Vedúci stretnutia: Matúš Ujhelyi

Priebeh stretnutia:

- vyhodnotenie odvedenej práce
- Juro – odprezentovanie návrhu protokolu (diagram s popisom, vysvetlenie)
 - broadcast,
 - komunikácia nad TCP protokolom,
 - používa správy (sekvenčne posielané správy, že je zariadenie pripojené => server si počíta, kedy mal naposledy správu (connection) od daného klienta,
 - návrh definuje typy správ still alive, event, action, going off, response on close.
- protokol je rozdelený na 2 časti:
 - sender – obsah správy: guid zariadenia, guid akcie, ktorá nastala, posledné priradenie (riešenie problému mapovania gest), na vyžiadanie nejaký typ správy (posielanie multimédií) a daný súbor.
 - server si vypýta akcie, ktoré klient pozná, protokol a port na začatie komunikácie.
- definovaný problém exportu/importu konfigurácie na zariadení na strane servera
- Matúš – odprezentovanie protokolu

3 | Zápisy zo stretnutí

- návrh mapovania akcií sme na udalosti v hlavnej aplikácii:
 - pre potreby user - friendly rozhrania,
 - nastavenie na základe hlasového povelu komu je aktuálne gesto určené (TV, všetci, mobil, skupina),
 - kategórie eventov, ovládania (všetkým, len niektorým a pod.),
 - definovanie histórie mapovaní,
 - možnosť pre používateľa prepnúť sa do nejakého módu, ale aj zadávať globálne gesta bez zadania módu, v rámci ktorého je dané gesto zadávané,
 - gesta namapované na viaceré zariadenia sú aktívne len po aktivovaní daného módu,
 - používateľ si vie nastaviť gesto prepnutia do ďalšieho módu,
 - na začiatku má používateľ možnosť buď namapovania zariadenie - udalosť, alebo vytvorenia skupiny a namapovania gesta,
 - gesta, ktoré prepínajú módy musia byť globálne,
 - v prototypy len 1 kus z každého typu vstupu (v rámci aplikácie môže byť unikátne a nemeniteľné GUID).
- výber Matúšovho návrhu protokolu
 - problém serializácie Json objektov v c++ (qjson)
- manažérske úlohy - požiadavky:
 - Jano: komunikácia e - mailom, icq, skype, atď.
 - ako majú vyzerat' e – maily (bude uvedené v metodike),
 - zoznam bodov na stretnutie bude nachystaný vopred (článok na internej wiki, kam každý doplní, čo chce v rámci stretnutia riešiť).

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Extrakcia dôležitých častí z minuloročného projektu	Matúš Ujhelyi	OK	13h	17h
Návrh protokolu	všetci	OK	7x6h	individuálne

3 | Zápisy zo stretnutí

Pripraviť projektový plán	Ivana Bohunická	OK	2h	2h
Uverejniť zápisy zo schôdzí na stránke	Ján Greppel	OK	ihneď	ihneď
Založenie dokumentácií	Dominik Rerko	OK	2h	2h
Zdokumentovanie návrhu protokolu	Zuzana Ujhelyiová	Pokračuje	2,5h	1,5h
Pripraviť v GITE štruktúru projektu	František Nagy	OK	30min	30min
Pripraviť podklady pre svoju manažérsku pozíciu	všetci	OK	individuálne	individuálne

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Napísať dokumentáciu riadenia	všetci	24.10.	30.10.	individuálne
Úprava webdesign	Juraj Muránsky	24.10	28.10	2h
Návrh designu stránky	Zuzana Ujhelyiová, Ivana Bohunická	24.10	26.10	2x2h
Analýza a product backlog - zdokumentovať	Ján Greppel	24.10	31.10	4h
Návrh knižnice v C++	Ivana Bohunická	25.10	26.10	3h
Návrh knižnice v C++	Matúš Ujhelyi	25.10	26.10	2h
Návrh knižnice Java	Dominik Rerko	25.10	26.10	2h
Návrh knižnice Java	František Nagy	25.10	26.10	2h
Implementácia protokolu v C++	Matúš Ujhelyi	24.10	30.10	5h
Implementácia protokolu v Jave	František Nagy	24.10	30.10	5h
Doplniť dokumentáciu riadenia	Dominik Rerko	24.10	31.10	1h
Návrh GUI – prototyp	Ivana Bohunická	24.10	29.10	4h
Implementácia GUI – prototyp	Juraj Muránsky	24.10	31.10	3h
Zdokumentovanie mapovania	Zuzana Ujhelyiová	24.10	31.10	2h
Pretestovať komunikáciu	Matúš Ujhelyi, František Nagy	24.10	30.10	2x5h
Zdokumentovanie alternatívneho protokolu	Juraj Muránsky	24.10	31.10	2h
Zadovážiť Kinect	Michal Kottman	24.10	31.10	ihneď

Zhodnotenie stretnutia:

Na stretnutí boli vyriešené 2 blokujúce problémy. Jedným bol komunikačný protokol a druhým problém mapovania udalostí na akcie.

3.5 Zápis zo stretnutia č.4

Téma stretnutia: Plánovanie 2. šprintu

Dátum stretnutia: 31.10.2012

Čas stretnutia: 09:00

Dĺžka stretnutia: 195 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Ujhelyi Matúš

Vedúci stretnutia: Ivana Bohunická

Priebeh stretnutia:

- zhodnotenie postupu práce na úlohách
- výber jigloo na tvorbu GUI v Jave
- ukážka stavu komunikačných knižníc: C++ - Matúš, Java - Fero
- doplnenie ping timeout a response timeout do init messagu protokolu
- ukážka návrhu GUI – Ivka: následná konzultácia a doladenie v spolupráci s ostatnými členmi tímu a product ownerom
 - spokojnosť s návrhom
 - nenaplnilo sa riziko: *Nesprávne navrhnuté používateľské rozhranie*, aj vďaka konzultáciám a priebežným reportom práce
- dohoda na názve prvého šprintu - čierna ovca
- výber názvu druhého šprintu - ružový panter
- na čistý nákres rozhrania bol vybraný nástroj Moqups
- pri návrhu entít je nutné zohľadniť históriu dát
- bolo dohodnuté, že každý task musí mať merateľný výstup

3 | Zápisy zo stretnutí

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Napísať dokumentáciu riadenia	všetci	OK	individuálne	individuálne
Úprava webdesignu	Juraj Muránsky	OK	2h	3h
Návrh designu stránky	Zuzana Ujhelyiová, Ivana Bohunická	OK	2x2h	Individuálne
Analýza a product backlog - zdokumentovať	Ján Greppel	OK	4h	5h
Návrh knižnice C++	Ujhelyi Matúš	OK	2h	4h
Návrh knižnice C++	Ivana Bohunická	OK	3h	3h
Návrh knižnice Java	František Nagy	Pokračuje	2h	1h
Návrh knižnice Java	Dominik Rerko	Pokračuje	2h	2h
Implementácia protokolu v Jave	František Nagy	Pokračuje	5h	15h
Implementácia protokolu v C++	Ujhelyi Matúš	Pokračuje	5h	18h
Návrh GUI – prototyp	Ivana Bohunická	OK	4h	5h
Implementácia GUI – prototyp	Juraj Muránsky	OK	3h	4h
Zdokumentovanie mapovania	Zuzana Ujhelyiová	OK	2h	2h
Pretestovať komunikáciu	Matúš Ujhelyi, František Nagy	OK	2x5h	individuálne
Zdokumentovanie alternatívneho protokolu	Juraj Muránsky	Pokračuje	2h	2h
Zdokumentovanie návrhu protokolu	Zuzana Ujhelyiová	Predlžovaná, OK	2,5h	2h

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Príprava návrhu mapovania (Business entity)	Dominik Rerko	31.10	5.10	4h
Príprava návrhu mapovania (Business entity)	Zuzana Ujhelyiová	31.10	5.10	4h
Príprava návrhu mapovania (Business entity)	Ján Greppel	31.10	5.10	2h
Návrh GUI	Ivana Bohunická	31.10	5.10	6h
Testovanie protokolu	Matúš Ujhelyi	31.10	7.10	8h
Testovanie protokolu	František Nagy	31.10	7.10	10h
Prototyp GUI	Juraj Muránsky	31.10	5.10	15h
Dokumentácia GUI	Ivana Bohunická	31.10	5.10	3h
Zdokumentovať architektúru a analýzu existujúceho riešenia	Ján Greppel	31.10	5.10	4h

Zhodnotenie stretnutia:

Skutočne každý člen tímu bol ochotný spolupracovať a zavládlo profesionálne prostredie, v ktorom sa dobre pracovalo. Neprišlo ku žiadnym konfliktom, vyriešilo sa veľa dôležitých vecí a naplánovali sa konkrétne úlohy pre ďalší týždeň.

3.6 Zápis zo stretnutia č.5

Téma stretnutia: Mapovanie udalostí a akcií
Stretnutie v strede 2. šprintu

Dátum stretnutia: 7.11.2012

Čas stretnutia: 09:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Ivana Bohunická

Vedúci stretnutia: Matúš Ujhelyi

Priebeh stretnutia:

- zhodnotenie postupu práce na jednotlivých úlohách
- Matúš, Fero: predvedenie sieťového spojenia
- Matúš: ukážka modifikácie FiitKinect knižnice pre naše použitie
- Dominik: neposkytnutie dokumentácii včas, nemohli byť dopredu skontrolované
- Juro: neodovzdaná dokumentácia
- dohoda, že zápisnica musí byť zverejnená na stránke ešte v deň stretnutia
- odhad času jednotlivých úloh bude zapisovaný do zápisnice pri zápise úloh na najbližší týždeň
- Zuzka: logovanie taskov a reportovanie ich stavu, komentáre, všetko dávať na wiki, zverejniť aby bola úloha skontrolovateľná
- Zuzka, Jano, Dominik: návrh bussines entít a dátového modelu
 - návrh mapovania na implementačnej úrovni - metódy pre rozhrania na prepojenie s ďalšími vrstvami - s GUI a so sieťou
- diskusia o ukladaní mapovania a výber implementácie
- hlasovaním bola vybraná databáza SqlLite z týchto možností:

3 | Zápisy zo stretnutí

- xml, xml schema
- JSON
- databáza
 - SQLite
- diskusia o GUI implementácii, napokon prehodnotenie a zmena rozdelenia úloh.

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Príprava návrhu mapovania (Bussiness entity)	Dominik Rerko	OK	4h	4h
Príprava návrhu mapovania (Bussiness entity)	Zuzana Ujhelyiová	OK	2h	2h
Príprava návrhu mapovania (Bussiness entity)	Ján Greppel	OK	2h	3h
Návrh GUI	Ivana Bohunická	OK	6h	7h
Testovanie protokolu	Matúš Ujhelyi	OK	2h	2h
Testovanie protokolu	František Nagy	OK	2h	2h
Prototyp GUI	Juraj Muránsky	zmena riešiteľa	15h	-
Dokumentácia GUI	Ivana Bohunická	OK	3h	3h
Zdokumentovať architektúru a analýzu existujúceho riešenia	Ján Greppel	OK	4h	5,5h
Návrh knižnice Java	František Nagy	Predĺžovaná, OK	2h	2h
Návrh knižnice Java	Dominik Rerko	Predĺžovaná, OK	2h	4h
Implementácia protokolu v Jave	František Nagy	Predĺžovaná, OK	5h	22h
Implementácia protokolu v C++	Ujhelyi Matúš	Predĺžovaná, OK	5h	25h
Zdokumentovanie alternatívneho protokolu	Juraj Muránsky	Predĺžovaná, OK	2h	4h 45min

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Zdokumentovanie core aplikácie_- návrh, opis Class diagramu	František Nagy	7.12.2012	12.11.2012	2h
Class diagram Core aplikácie	Matúš Ujhelyi	7.12.2012	9.11.2012	2h
Implementácia gui – úvod	Dominik Rerko	7.12.2012	14.12.2012	20
Implementácia gui – lokálny mód, detail situácie	František Nagy	7.12.2012	14.12.2012	7
Implementácia gui – tvorba situácie	Matúš Ujhelyi	7.12.2012	14.12.2012	15
Implementácia gui – prehľad módov	Ivana Bohunická	7.12.2012	14.12.2012	20
Vylepšenie FiitKinect knižnice	Matúš Ujhelyi	7.12.2012	priebežne	10h

3 | Zápisy zo stretnutí

Databáza - komplet pripraviť, schéma, prázdna databáza, fyzický model	Juraj Muránsky	7.12.2012	9.11.2012	2h
Zdokumentovať databázu a jej fyzický model	Juraj Muránsky	7.12.2012	12.11.2012	3h
Implementácia connect, load, save databázy	Ján Greppel	7.12.2012	12.11.2012	8h
Vyrobenie objektov pre mapovacie API	Zuzana Ujhelyiová	7.12.2012	9.11.2012	3h
Objekty – implementácia metód rozhrania - zatiaľ prázdne metódy	Ivana Bohunická	7.12.2012	12.11.2012	4h
Revízia a kontrola dokumentácií pred odovzdaním	Zuzana Ujhelyiová	7.12.2012	12.11.2012	5h
Revízia a kontrola dokumentácií pred odovzdaním	Ivana Bohunická	7.12.2012	12.11.2012	5h
Vytvorenie prihlášky do TP Cup	Zuzana Ujhelyiová	7.12.2012	12.11.2012	2h

Zhodnotenie stretnutia:

Matúš a Fero predviedli fungujúcu komunikáciu po sieti, čo nám poskytuje priestor pre ďalšiu prácu. Vyriešilo a navrhlo sa mapovanie udalostí a akcií, čo považujeme za kľúčové - podstata centrálnej aplikácie. Riešili sa tímové problémy ohľadom fungovania a organizácie tímu, snaha o zlepšenie.

3.7 Zápis zo stretnutia č. 6

Téma stretnutia: Plánovanie 3. šprintu

Dátum stretnutia: 14. 11. 2012

Čas stretnutia: 9:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: František Nagy

Vedúci stretnutia: Dominik Rerko

Priebeh stretnutia:

- zhodnotenie výsledkov od posledného stretnutia
- Zuzka: šprint review, vyhodnotenie uzavretých úloh
 - nesplnené: vylepšenie kinect - dlhodobá úloha, dorobenie gui - dlhodobá úloha, load / save databázy, problémy v komunikácii a v plánovaní z Janovej strany – prebehla diskusia a návrhy, ako postupovať, aby sa podobná situácia neopakovala
- Jano: konzultácia otázok k úlohe load/save databázy so zvyškom tímu
- Matúš upozornil, že nastalo riziko: *Nesplnenie naplánovaných úloh v korektnom časovom horizonte*, a to zo strany Jana a jeho úlohy pre implementáciu load a save metód nad databázou
 - nebolo možné vyhnúť sa tomuto riziku, pretože Jano nepostupoval podľa Matúšových odporúčaní na predchádzanie rizika a nekomunikoval svoj problém včas.
- Ivka, Matúš, Dominik, Fero: prezentácia výsledkov v implementácii používateľského rozhrania
- Jano: návrh presunúť dokumentáciu do google - docs z dôvodu možnosti úprav všetkými a v rámci jednej verzie - návrh bol zamietnutý
- Jano: návrh na zjednotenie, kam nahrávať súbory (dokumentácia) - súbory sa budú nahrávať na ftp server a bude sa na ne odkazovať

3 | Zápisy zo stretnutí

- Matúš: pripomenutie manažérskych úloh niektorých členov tímu (Juro, Jano, Fero). Matúš tým vystríhal pred naplnením rizika: *Nezodpovedné plnenie manažérskych rolí*.
 - spolu s pedagogickým vedúcim sa uskutočnila dôrazná výzva na zlepšenie plnenia manažérskych rolí
 - po diskusii bolo určené, kto sa bude čím bližšie zaoberať
- Ivka: plán na 3. šprint
- odovzdanie dokumentácie pedagogickému vedúcemu a podpísanie preberacieho protokolu

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Zdokumentovanie core aplikácie- návrh, opis Class diagramu	František Nagy	OK	4	6
Class diagram Core aplikácie	Matúš Ujhelyi	OK	4	3
Implementácia gui – úvod	Dominik Rerko	OK	20	8
Implementácia gui – lokálny mód, detail situácie	František Nagy	OK	7	8
Implementácia gui – tvorba situácie	Matúš Ujhelyi	OK	15	17
Implementácia gui – prehľad módov	Ivana Bohunická	OK	20	10
Vylepšenie FiitKinect knižnice	Matúš Ujhelyi	OK	10	3
Databáza- komplet pripraviť, schéma, prázdna databáza, fyzický model	Juraj Muránsky	OK	2	4,5
Zdokumentovať databázu a jej fyzický model	Juraj Muránsky	OK	3	4
Implementácia connect, load, save databázy	Ján Greppel	Nedokončená, (v stredu o polnoci 100%)	8	13,5(v stredu o polnoci)
Vyrobenie objektov pre mapovacie API	Zuzana Ujhelyiová	OK	3	5
Objekty - implementácia metód rozhrania – zatiaľ prázdne metódy	Ivana Bohunická	OK	4	2,5
Revízia a kontrola dokumentácií pred odovzdaním	Zuzana Ujhelyiová	OK	5	9
Revízia a kontrola dokumentácií pred odovzdaním	Ivana Bohunická	OK	5	7,5
Vytvorenie prihlášky do TP Cup	Zuzana Ujhelyiová	Pokračuje	2h	1h

3 | Zápisy zo stretnutí

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
IR do stredy vysielač, sieť (možno prijímač) + dokumentácia	Juraj Muránsky	14. 11.	21.11	20
Android vstup - základ (pár tlačítok, komunikácia po sieti, indikátor pripojenia)	Ján Greppel	14. 11.	19.11	15
Android vstup – dokončenie gui + dokumentácia	Ján Greppel	14. 11.	21.11	8
Android výstup – komunikácia po sieti, prijímanie eventov, spúšťanie akcii (toggle mute, volume up/down, zahranie zvuku, toggle led...) + dokumentácia	Dominik Rerko	14. 11.	21.11	20
Core – mapper + testy	František Nagy	14. 11.	21.11	20
Core – mapping classes + testy	Zuzana Ujhelyiová	14. 11.	19.11	4
Core – gui design	Ivana Bohunická	14. 11.	21.11	8
Core – gui handling	Ivana Bohunická, Matúš Ujhelyi, Zuzana Ujhelyiová	14. 11.	21.11	10, 20, 6
Rozbehať projekt všetkým	Juraj Muránsky	14. 11.	28.11	x
Nightly build systém	Juraj Muránsky	14. 11.	28.11	x
Dorobiť obrázky do kinect	Matúš Ujhelyi	14. 11.	19.11	3
Dorobiť obrázky do serializácie	Zuzana Ujhelyiová	14. 11.	19.11	2

Zhodnotenie stretnutia:

Stretnutie bolo prínosné. Boli odprezentované výsledky práce od posledného stretnutia, ale aj vydiskutované nesplnené úlohy, ich dôvody a ako takýmto situáciám predísť.

3.8 Zápis zo stretnutia č. 7

Téma stretnutia: Stretnutie v strede3. šprintu

Dátum stretnutia: 21. 11. 2012

Čas stretnutia: 9:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Juraj Muránsky

Vedúci stretnutia: František Nagy

Priebeh stretnutia:

- Diskusia o prezentácii aplikácie poznatkov riadenia v tíme: termín 6.12.
 - obsah prezentácie: zadanie, cieľ projektu, čo bolo spravené
 - vedúci: nepoužívať v rámci prezentácie video, radšej screenshoty
 - rozdelenie častí prezentácie na základe manažérskych úloh a ujasnenie si zodpovednosti členov tímu
 - dohoda, že 28.11. každý predvedie svoju pripravenú časť prezentácie
- zmena vedúceho tímu z Jana na Matúša
- Prebehla zmena vedúceho, a teda nastalo riziko: *Nezodpovedné plnenie manažérskych rolí*. Vzhľadom na predošlé upozornenia bolo nevyhnutným riešením zmena role v tíme.
- Zuzka: nezabudnúť na Javadoc komentáre v súlade s Dominikovou metodikou (nemusia byť v testoch)
- Fero: použitie tabulátorov na miesto medzier, maximálna dĺžka riadku smie byť 120 znakov a aj to len v špeciálnom prípade, v prázdnych riadkoch nemajú byť tabulátory (čo treba nastaviť vo formattery), v kóde nemá byť zbytočne vysoká úroveň vnorenia pokiaľ sa tomu dá vyhnúť.
- pondelkové stretnutie: 18:30

3 | Zápisy zo stretnutí

- Jano: poslať e - mail, čo spravil a aké má pripomienky (nebude sa môcť zúčastniť stretnutia)
- potreba upraviť entitu *situácia*: prídanie atribútu *value* (úprava v GUI, databáze a príslušných dokumentáciách), aktuálna hodnota *value* v entite *akcia* sa použije ako prednastavená hodnota, v databáze zmena typu obrázku z blobu na string
- Dominik, Jano: prezentácia Android aplikácií
- Ukážka pripojenia 3 zariadení (kinect, android prijímač a android vysielateľ) na core aplikáciu
- vedúci: definícia GUI pri android aplikáciách
- vedúci: ikony majú byť konzistentné medzi zariadeniami (toto sa týka hlavne obrázkov zariadení a akcií) – zodpovedná osoba => Ivka
- definícia módu na zákalde mena: v databáze potreba pridať unikátny index na meno módu, Fero úprava core aplikácie
- potreba pridať do protokolu pole *description* pre zariadenia
- Ivka: doplnenie metód do mappera (zoznam voľných eventov - tie ktoré nie sú v globálnom móde pre lokálne módy a naopak, zoznam prepínacích akcií - tie, ktoré sú v globálnom móde definované ako switching mode akcie, pre zoznam voľných akcií - komplement k prepínacím akciám, na premenovanie módu)
- dohoda pridať core aplikáciu ako typ zariadenia do databázy pre potreby referencovania na prepínacie akcie
- vedúci: pripomienky k zápisniciam a k dokumentácií
- vedúci: príklad zdrojového kódu na opakovanie zvukovej sekvencie pre Android.

3 | Zápisy zo stretnutí

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
IR do stredy vysielač, sieť (možno prijímač) + dokumentácia	Juraj Muránsky	Pokračuje	20	5
Android vstup - základ (pár tlačítok, komunikácia po sieti, indikátor pripojenia)	Ján Greppel	Pokračuje	15	8
Android vstup – dokončenie gui + dokumentácia	Ján Greppel	Pokračuje	8	2
Android výstup – komunikácia po sieti, prijímanie eventov, spúšťanie akcii + dokumentácia	Dominik Rerko	Pokračuje	20	14
Core – mapper + testy	František Nagy	Pokračuje	20	14
Core – mapping classes + testy	Zuzana Ujhelyiová	OK	4	2
Core – gui design	Ivana Bohunická	Pokračuje	8	9
Core – gui handling	Ivana Bohunická, Matúš Ujhelyi, Zuzana Ujhelyiová	Pokračuje	10, 20, 6	8, 16, 5
Rozbehať projekt všetkým	Juraj Muránsky	Pokračuje	x	0
Nightly build systém	Juraj Muránsky	Pokračuje	x	0
Dorobiť obrázky do kinect	Matúš Ujhelyi	OK	3	2
Dorobiť obrázky do serializácie	Zuzana Ujhelyiová	OK	2	1,5
Vytvorenie prihlášky do TP Cup	Zuzana Ujhelyiová	Predĺžovaná, OK	2h	1h 45min

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Dorobenie GUI	Zuzana Ujhelyiová, Matúš Ujhelyi, Ivana Bohunická	21. 11.	28.11	10+10+1 5
Úprava DB modelu	Juraj Muránsky	21. 11.	21.11	0.5
Úpravy load a save tak, aby reagoval na zmeny v DB	Ján Greppel	21. 11.	22.11	2
Dorábanie android aplikácie	Ján Greppel, Dominik Rerko	21. 11.	26.11	10+10
Dokončenie mappera	František Nagy	21. 11.	24.11	3
Vytvorenie nightly build systému	Juraj Muránsky	21. 11.	26.11	12
Zdokumentovanie IR aplikácie	Juraj Muránsky	21. 11.	26.11	3
Implementovanie IR prijímača/vysielača	Juraj Muránsky	21. 11.	26.11	0,5/8
Pripraviť sa na rozbehávanie C++ prostredia ostatným členom tímu	Juraj Muránsky	21. 11.	26.11	1

3 | Zápisy zo strenutí

Zhodnotenie strenutia:

Stretnutie bolo prínosné. Bolo vyriešených množstvo nezrovnalostí v aplikácii a boli upresnené postupy pri ďalšej práci. Odprezentovaná ukážka prepojenia core aplikácie s Kinectom a vstupnou aj výstupnou Android aplikáciou mala motivačný efekt.

3 | Zápisy zo stretnutí

3.9 Zápis zo stretnutia č. 8

Téma stretnutia: Plánovanie 4. šprintu

Dátum stretnutia: 28. 11. 2012

Čas stretnutia: 9:00

Dĺžka stretnutia: 180 min.

Miesto stretnutia: softvérové štúdio

Účastníci: Dominik Rerko Zuzana Ujhelyiová Matúš Ujhelyi
Ivana Bohunická Ján Greppel Juraj Muránsky
František Nagy
Ing. Michal Kottman

Zapisovateľ: Dominik Rerko

Vedúci stretnutia: Juraj Muránsky

Priebeh stretnutia:

- zhodnotenie postupu práce na jednotlivých úlohách
- Zuzka, Ivana: review šprintu, vyhodnotenie uzavretých úloh
 - Jano: nesplnené obidve úlohy v rámci šprintu (prvá ešte z 2. šprintu) – prebehla diskusia o neplnení úloh a vyvodení sankcií
- Dominik, Juro: predvedenie implementovaných aplikácií komunikujúcich s centrálnou aplikáciou (android receiver, IR modul)
- nácvik prezentácie aplikácie poznatkov riadenia
- Ivka: plán 4. Šprintu

Zhodnotenie úloh z predchádzajúceho stretnutia:

Popis úlohy	Zodpovedný	Stav	Odhadovaný čas	Strávený čas
Dorobenie GUI	Zuzana Ujhelyiová, Matúš Ujhelyi, Ivana Bohunická	OK	10+10+15	7+9+8
Úprava DB modelu	Juraj Muránsky	OK	0.5	0.5
Úpravy load a save tak, aby reagoval na zmeny v DB.	Ján Greppel	Nesplnená Zmena riešiteľa	2	14.5
Dorábanie android aplikácie	Ján Greppel,	Nesplnená Pokračuje	10	2.5
Dorábanie android aplikácie	Dominik Rerko	OK	10	9

3 | Zápisy zo stretnutí

Testy (sieťová komunikácia, mapper)	František Nagy	Pokračuje	10	10
Dokončenie mappera	František Nagy	OK	3	9
Vytvorenie nightly build systému	Juraj Muránsky	Pokračuje	12	1
Zdokumentovanie IR aplikácie	Juraj Muránsky	OK Pokračuje	3	2
Implementovanie IR prijímača/vysielača	Juraj Muránsky	OK Pokračuje	0,5/8	11/5
Pripraviť sa na rozbehávanie C++ prostredia ostatným členom teamu	Juraj Muránsky	OK Pokračuje	1	1

Úlohy do ďalšieho stretnutia :

Popis úlohy	Zodpovedný	Od	Do	Odhadovaný čas
Dokončenie testov	František Nagy	28. 11.	3.12	11
Zdokumentovanie testov	František Nagy	28. 11.	5.12	1
Oprava load a save	Matúš Ujhelyi	28.11	5.12	2
RefaktORIZÁCIA KÓDU	Matúš Ujhelyi	28. 11.	5.12	13
Dokumentácia k implementácii GUI	Ivana Bohunická	28. 11.	5.12	3
Revízia dok. riadenia	Ivana Bohunická	28. 11.	3.12	10
Dokumentácia Cobertúra	Zuzana Ujhelyiová	28. 11.	5.12	1
Revízia dok. riadenia	Zuzana Ujhelyiová	28. 11.	3.12	15
JavaDoc komentáre	Dominik Rerko	28. 11.	3.12	6
Prezentácia MSI	Dominik Rerko	28. 11.	3.12	1
Doladenie IR	Juraj Muránsky	28. 11.	3.12	3
Dokumentácia Nightly Build	Juraj Muránsky	28. 11.	2.12(1900)	3
Dokončenie android sender	Ján Greppel	28.11	2.12	10
Oprava dokumentácie k android sender	Ján Greppel	28.11	2.12.	1
Zdokumentovanie User stories	Ján Greppel	28.11	5.12.	2

Zhodnotenie stretnutia:

Odprezentovaná ukážka prepojenia core aplikácie s IR a výstupnou Android aplikáciou mala motivačný efekt. Pri nácviku prezentácie bolo vznesených niekoľko pripomienok.

4 Úlohy členov tímu

4.1 Dlhodobé úlohy

Pre zlepšenie prehľadnosti uvádzame rozdelenie pozícií medzi jednotlivými členmi tímu prostredníctvom tabuľky (Tabuľka 4-1).

Oblasť:	Zodpovedný:
Manažment rozvrhu a plánovania	Ivana Bohunická
Manažment komunikácie	Ján Greppel
Manažment podpory vývoja	Juraj Muránsky
Manažment kvality	František Nagy
Manažment tvorby dokumentácie	Dominik Rerko
Manažment rizík, vedúci tímu	Matúš Ujhelyi
Monitorovanie projektu	Zuzana Ujhelyiová

Tabuľka 4-1: Rozdelenie pozícií medzi členov tímu

4.2 Krátkodobé úlohy

Na zaznamenávanie prebiehajúcich, ako aj ukončených úloh používame systém na sledovanie úloh (z angl. issue tracking system) Redmine. Úlohy sú rozdelené na analytické, implementačné, podporné a dokumentačné. Zoznam všetkých krátkodobých úloh sa nachádza v zápisniciach (kapitola 2), v harmonograme úloh (kapitola 5.4) a v spomínanom podpornom nástroji Redmine.

5 Manažment rozvrhu

Zodpovedná osoba: Ivana Bohunická

5.1 Dôležité termíny

Dôležité termíny v zimnom semestri sú pevne stanovené (Tabuľka 5-1) a vyplývajú z organizácie predmetu. Definujú kedy je potrebné odovzdať požadovaný výstup.

Termín odovzdania	Úloha
12. - 16. 11. 2012	<ul style="list-style-type: none"> ▪ Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrhu riešenia ▪ Odovzdanie dokumentácie prvých dvoch šprintov plus dokumentácia k riadeniu.
26.11.2012	<ul style="list-style-type: none"> ▪ Prihláška do TP Cup
14. 12. 2012, 14:00	<ul style="list-style-type: none"> ▪ Odovzdanie prototypu vybraných častí systému spolu s dokumentáciou ▪ odovzdanie dokumentácie šprintov za zimný semester spolu s opisom vytvoreného prototypu plus dokumentácia k riadeniu
14. - 20. 12. 2012 (podľa dohody tímov)	<ul style="list-style-type: none"> ▪ Používateľská prezentácia prototypu

Tabuľka 5-1: Dôležité termíny

5.2 Product backlog

Aby sme mohli plniť požiadavky na stanovené výstupy projektu, bolo nutné pripraviť plán. Postupovali sme agilnou metódou Scrum. Najskôr vznikol Product Backlog (Tabuľka 5-2), ktorého položky boli ohodnotené a zoradené zákazníkom podľa priority (1 = najvyššia priorita).

Všetky úlohy sa plánovali a rozdeľovali do šprintov. Každý šprint trval 2 týždne. Počas jedného zimného semestra sú takéto šprinty štyri. Jednotlivé požiadavky sa rozdelili do šprintov semestra, kedy sa očakáva ich realizácia. Pričom v 4.šprinte sa plánuje prepojenie a integrácia jednotlivých modulov (FiitKinect knižnica, komunikácia cez sieť, centrálna aplikácia s mapovaním a s grafickým rozraním), ktoré boli implementované v predošlých šprintoch. V poslednom šprinte je taktiež naplánované testovanie a záťažové testy.

Priorita	Položka Product backlog	Šprint
1.	Príprava a extrakcia potrebných častí knižnice Fiit Kinect	1.Šprint
2.	Komunikácia modulov cez sieť, vlastný protokol	1-2.Šprint

3.	Vstup z počítača	1-2.Šprint
4.	Mapovanie akcií a gest, znovu použitie gest (jedno gesto - viacero akcií)	2.Šprint
5.	Používateľské rozhranie	2-3.Šprint
6.	Rozpoznať rôzne vstupy (gestá, mobil, IR)	3.Šprint
7.	Rozpoznať rôzne vstupy (zvuk)	Letný semester
8.	Vylepšiť rozpoznanie gest	Letný semester
9.	Odosielat' príkazy	Letný semester
10.	Ukladanie konfigurácie	Letný semester

Tabuľka 5-2: Product Backlog

5.3 Výstupy jednotlivých šprintov

Naplánované výstupy jednotlivých šprintov uvádzame v tabuľke (Tabuľka 5-3). Tieto výstupy predstavujú časť zoznamu Product backlog - u, ktoré boli vybrané pre zimný semester a rozdelené do menších používateľských príbehov.

Obdobie	Výstup
Šprint 1 Čierna ovca 5. – 6. Týždeň (17.10. – 31.10. 2012)	<ul style="list-style-type: none"> ▪ analýza problému, user story ▪ pripravenie knižnice Fiit Kinect ▪ návrh komunikačného protokolu ▪ navrhnutá a implementovaná komunikácia cez sieť formou knižnice ▪ testovací program pre základný vstup z počítača ▪ úprava našej stránky - návrh, implementácia ▪ prvý prototyp grafického rozhrania - návrh, implementácia ▪ dohoda o spôsobe mapovania udalostí (gest) – znovu použitie udalostí (gest)
Šprint 2 Ružový panter 7. – 8. Týždeň (1.11. – 14.11.2012)	<ul style="list-style-type: none"> ▪ návrh implementácie mapovania v hlavnej aplikácii ▪ implementácia mapovania v hlavnej aplikácii ▪ zintegrovanie knižnice FiitKinect s našou komunikačnou knižnicou ▪ testovanie komunikácie ▪ finálny návrh grafického rozhrania ▪ implementácia prezentačnej vrstvy ▪ implementácia paralelnej komunikácie, vlákien hlavnej aplikácie ▪ finalizácia a kompletizácia dokumentácií pred odovzdaním
Šprint 3 Fialová krava 9. – 10. Týždeň (15.11. – 28.11.2012)	<ul style="list-style-type: none"> ▪ implementácia prezentačnej vrstvy ▪ vstup z telefónu, výstup na telefón ▪ vstup z Kinectu ▪ vstup z IR, výstup na IR
Šprint 4 Zlatá rybka	<ul style="list-style-type: none"> ▪ integrácia implementovaných modulov ▪ testovanie, záťažové testy

11. – 12. Týždeň (29.11. – 12.12.2012)	<ul style="list-style-type: none"> ▪ refaktor zdrojového kódu ▪ finalizácia a kompletizácia riešenia pred odovzdaním ▪ finalizácia a kompletizácia dokumentácií pred odovzdaním
---	--

Tabuľka 5-3: Výstupy šprintov

5.4 Harmonogram úloh

Šprint obsahuje viacero úloh a požiadaviek vybraných z product backlog - u. Tieto zvolené úlohy sa rozdelia na najmenšie vykonateľné podúlohy a tvoria sprint backlog. Každý úlohe bol pridelený člen tímu, ktorý mal zodpovednosť za jej realizáciu.

Niektoré úlohy kvôli svojej rozsiahlosti a zložitosti zasahovali do viacerých šprintov. Pre označenie úlohy pokračujúcej v nasledujúcom šprinte uvádzame pri jej názve symbol: ➡

5.4.1 Šprint 0- Biela veľryba

Úloha	Riešiteľ	Začiatok	Koniec
Vytvoriť štruktúru Git - u	František Nagy	18.10.2012	22.10.2012
Analyzovať existujúce riešenia	Všetci	10.10.2012	17.10.2012
Pripraviť Git a napísať návod	Matúš Ujhelyi	10.10.2012	16.10.2012
Vytvoriť web stránku tímu, založiť tímovú wiki	Ján Greppel	10.10.2012	22.10.2012

Tabuľka 5-4: Šprint 0

5.4.2 Šprint 1- Čierna ovca

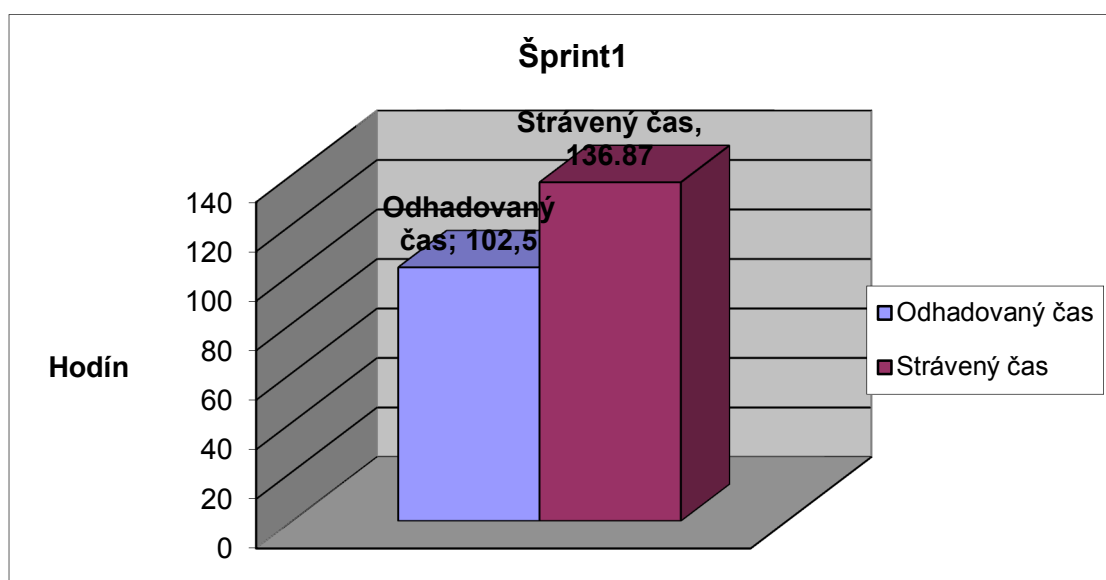
Úloha	Riešiteľ	Začiatok	Koniec
Implementovať používateľské rozhranie - prvotný prototyp	Juraj Muránsky	27.10.2012	29.10.2012
Navrhnuť používateľské rozhranie - prvotný prototyp	Ivana Bohunická	27.10.2012	29.10.2012
Konfigurácia Java aplikácie + serializácia JSON objektov	Zuzana Ujhelyiová	26.10.2012	27.10.2012
Skompletizovať a zdokumentovať Product backlog, user stories	Ján Greppel	25.10.2012	30.10.2012
Zadovážiť Kinect	Michal Kottman	25.10.2012	31.10.2012
Zdokumentovať mapovanie udalostí a akcií	Zuzana Ujhelyiová	25.10.2012	30.10.2012
Prerobiť stránku podľa navrhnutého designu	Juraj Muránsky	25.10.2012	30.10.2012
Návrh designu stránky	Zuzana Ujhelyiová	25.10.2012	26.10.2012
Návrh designu stránky, úprava obsahu stránky	Ivana Bohunická	25.10.2012	26.10.2012
Návrh C++ knižnice	Matúš Ujhelyi	24.10.2012	26.10.2012
Návrh C++ knižnice, dokumentácia ku knižnici	Ivana Bohunická	25.10.2012	26.10.2012
Návrh Java knižnice	František Nagy	25.10.2012	26.10.2012
Návrh Java knižnice	Dominik Rerko	25.10.2012	26.10.2012

5 | Manažment rozvrhu

Vytvoriť testovací program pre C++ knižnicu ➡	Matúš Ujhelyi	24.10.2012	6.11.2012
Vytvoriť testovací program pre Java knižnicu ➡	František Nagy	24.10.2012	6.11.2012
Implementovať protokol v Java ➡	František Nagy	24.10.2012	6.11.2012
Implementovať protokol v C++ ➡	Matúš Ujhelyi	24.10.2012	6.11.2012
Navrhnuť a zdokumentovať alternatívny komunikačný protokol	Juraj Muránsky	24.10.2012	31.10.2012
Zdokumentovať komunikačný protokol	Zuzana Ujhelyiová	22.10.2012	24.10.2012
Návrh protokolu	Matúš, František, Zuzana, Ivana, Juraj	18.10.2012	22.10.2012
Komunikácia cez sieť - návrh	Všetci	18.10.2012	30.10.2012
Pripraviť existujúcu Fiit Kinect knižnicu	Matúš Ujhelyi	18.10.2012	23.10.2012

Tabuľka 5-5: Šprint 1

V grafe (Obrázok 5-1) je uvedené porovnanie odhadovaných a reálne vykázaných hodín za šprint Čierna ovca. Keďže išlo o prvý šprint, na ktorý boli odhadované časy, rozdiel medzi odhadovaným a stráveným časom je cca. 30 hodín.



Obrázok 5-1: Porovnanie štatistiky časov za šprint Čierna ovca

5.4.3 Šprint 2- Ružový panter

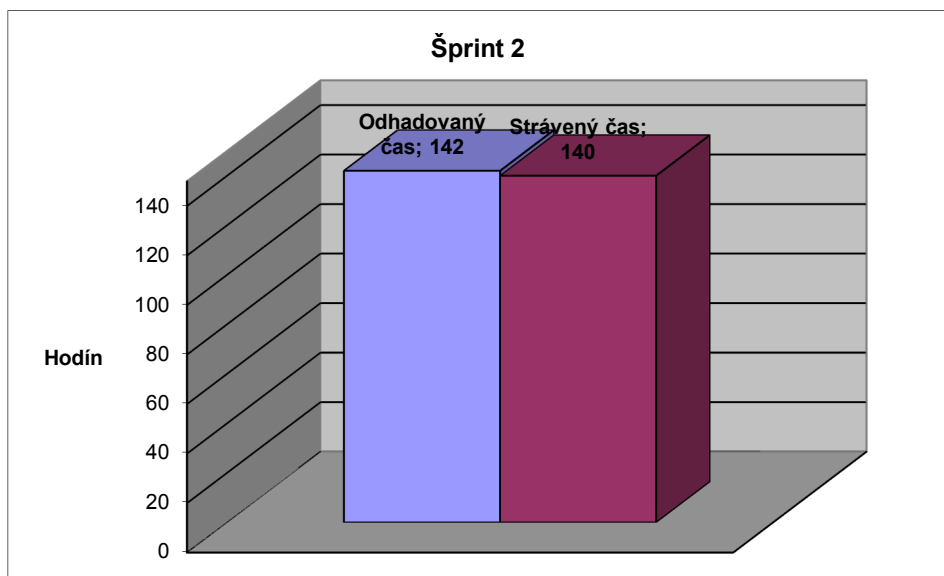
Úloha	Riešiteľ	Začiatok	Koniec
Revízia a kontrola dokumentácií pred odovzdaním	Ivana Bohunická	7.11.2012	11.11.2012
Revízia a kontrola dokumentácií pred odovzdaním	Zuzana Ujhelyiová	7.11.2012	11.11.2012
Implementácia metód rozhrania na mapovanie objektov	Ivana Bohunická	7.11.2012	12.11.2012
Implementácia - vytvorenie mapovacích objektov	Zuzana Ujhelyiová	7.11.2012	9.11.2012
Implementácia connect, load, save databázy ➡	Ján Greppel	7.11.2012	12.11.2012

5 | Manažment rozvrhu

Zdokumentovať databázu a jej fyzický model	Juraj Muránsky	7.11.2012	12.11.2012
Pripraviť databázu, databázovú schému	Juraj Muránsky	7.11.2012	9.11.2012
Vylepšenie a implementačné prispôsobenie FiitKinect knižnice	Matúš Ujhelyi	7.11.2012	14.11.2012
Implementácia a vytvorenie obrazoviek používateľského rozhrania	Dominik Rerko	7.11.2012	14.11.2012
Implementácia a vytvorenie obrazoviek používateľského rozhrania	František Nagy	7.11.2012	14.11.2012
Implementácia a vytvorenie obrazoviek používateľského rozhrania	Ivana Bohunická	7.11.2012	14.11.2012
Implementácia a vytvorenie obrazoviek používateľského rozhrania	Matúš Ujhelyi	7.11.2012	14.11.2012
Vytvoriť diagram tried centrálnej aplikácie	Matúš Ujhelyi	7.11.2012	9.11.2012
Zdokumentovať návrh centrálnej aplikácie	František Nagy	7.11.2012	12.11.2012
Otestovať výsek minuloročného riešenia s kinectom	Matúš Ujhelyi	3.11.2012	7.11.2012
Zdokumentovať komplexný návrh a architektúru nášho riešenia	Ján Greppel	1.11.2012	6.11.2012
Zdokumentovať analýzu existujúceho riešenia - knižnica Fiit Kinect	Ján Greppel	1.11.2012	6.11.2012
Zdokumentovať návrh používateľského rozhrania	Ivana Bohunická	1.11.2012	3.11.2012
Finálny návrh používateľského rozhrania	Ivana Bohunická	1.11.2012	3.11.2012
Návrh mapovania (Bussiness entity)	Dominik Rerko	1.11.2012	5.11.2012
Návrh mapovania (Bussiness entity)	Ján Greppel	1.11.2012	5.11.2012
Návrh mapovania (Bussiness entity)	Zuzana Ujhelyiová	1.11.2012	5.11.2012

Tabuľka 5-6: Šprint 2

V nasledujúcom grafe (Obrázok 5-2) sa nachádza porovnanie odhadovaných a strávených časov za 2. šprint. Rozdiel bol 2h, čo značí posun k lepšiemu, čo sa odhadov týka.



Obrázok 5-2: Porovnanie štatistiky časov za 2. šprint

5.4.4 Šprint 3- Fialová krava

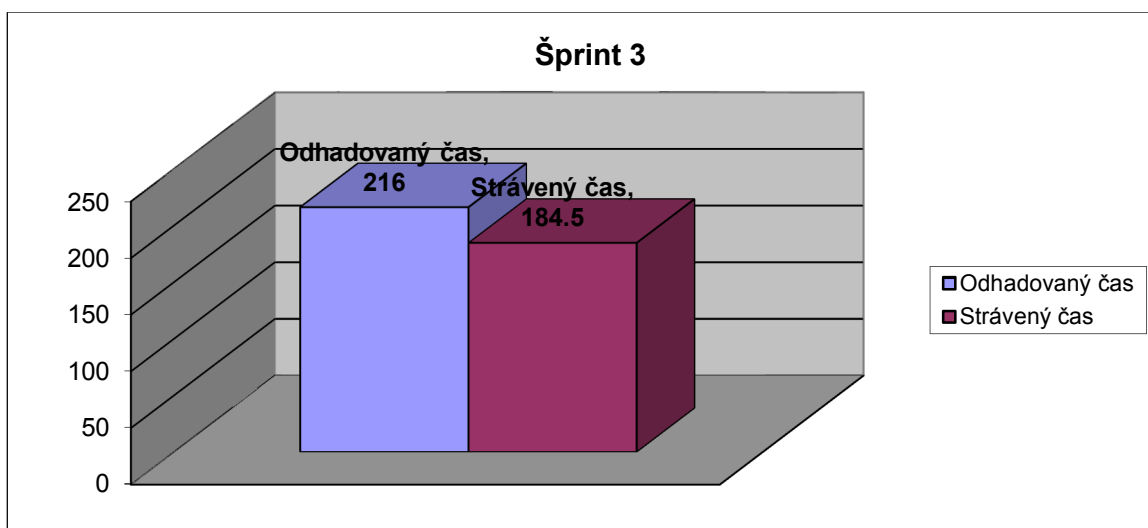
Úloha	Riešiteľ	Začiatok	Koniec
Zdokumentovať Android Sender	Jan Greppel	21.11.2012	02.12.2012
Zdokumentovať Android Receiver	Dominik Rerko	21.11.2012	26.11.2012
Rozbehanie C++ prostredia všetkým členom tímu	Juraj Muransky	21.11.2012	26.11.2012
Implementácia IR vysielача ➡	Juraj Muransky	21.11.2012	03.12.2012
Zdokumentovať IR	Juraj Muransky	21.11.2012	25.11.2012
Zdokumentovať doplnenie a vylepšenie FiitKinect knižnice	Matus Ujhelyi	21.11.2012	26.11.2012
Unit testy sieťovej komunikácie	Frantisek Nagy	21.11.2012	03.12.2012
Implementácia funkcionality vybraných metód v Mapper class - e	Ivana Bohunicka	17.11.2012	25.11.2012
Dokončenie implementácie load, save metód	Jan Greppel	14.11.2012	28.11.2012
Implementácia GUI handling	Zuzana Ujhelyiova	14.11.2012	27.11.2012
Implementácia GUI handling	Matus Ujhelyi	14.11.2012	27.11.2012
Implementácia GUI handling	Ivana Bohunicka	14.11.2012	27.11.2012
Implementácia GUI dizajnu	Ivana Bohunicka	14.11.2012	25.11.2012
Doimplementovať obrázky do FiitKinect aplikácie	Matus Ujhelyi	14.11.2012	19.11.2012
Doimplementovať obrázky do serializácie	Zuzana Ujhelyiova	14.11.2012	19.11.2012
Unit testy - mapovacie triedy	Zuzana Ujhelyiova	14.11.2012	21.11.2012
Implementácia metód mapovacích tried	Zuzana Ujhelyiova	14.11.2012	19.11.2012
Unit testy - Mapper class	Frantisek Nagy	14.11.2012	28.11.2012
Implementácia metód v Mapper class - e	Frantisek Nagy	14.11.2012	24.11.2012
Konzultácia a pomoc pri vývoji Android aplikácií	Frantisek Nagy	14.11.2012	21.11.2012

5 | Manažment rozvrhu

Implementácia Android Receiver - spúšťanie akcií	Dominik Rerko	14.11.2012	26.11.2012
Implementácia Android Receiver - príjem udalostí z Core app	Dominik Rerko	14.11.2012	26.11.2012
Android Sender - grafické rozhranie ➡	Jan Greppel	28.11.2012	04.12.2012
Implementácia Android Sender - komunikácia po sieti s Core	Jan Greppel	14.11.2012	20.11.2012
Implementácia IR prijímača	Juraj Muransky	14.11.2012	26.11.2012

Tabuľka 5-7: Šprint 3

V grafe č. 3 (Obrázok 5-3) sa nachádza porovnanie odhadovaného času pre jednotlivé úlohy s časom stráveným a vykázaným v systéme Redmine.



Obrázok 5-3: Porovnanie štatistiky časov za šprint Fialová krava

5.4.5 Šprint 4- Zlatá rybka

Úloha	Riešiteľ	Začiatok	Koniec
Core: Prepínanie medzi globálnym a lokálnym módom (Bug)	Matus Ujhelyi	28.11.2012	5.12.2012
Ir: Po odpojení Core padne vysielac a uz sa nepripoji po opatovnom nastartovani Core (Bug)	Juraj Muransky	29.11.2012	5.12.2012
Zdokumentov' user stories 3. a 4. šprintu	Jan Greppel	28.11.2012	5.12.2012
Core: nefunuguje loadovanie lokálnych módoz databázy (Bug)	Matus Ujhelyi	28.11.2012	3.12.2012
Finálna refaktorizácia a revízia zdrojového kódu	Matus Ujhelyi	28.11.2012	5.12.2012
Zdokumentovanie testov	Frantisek Nagy	28.11.2012	5.12.2012
Záťažové testy	Frantisek Nagy	28.11.2012	5.12.2012
Revízia Javadoc	Dominik Rerko	28.11.2012	3.12.2012
Revízia a kontrola dokumentácií pred odovzdaním	Ivana Bohunicka	28.11.2012	8.12.2012
Implementácia Android Sender - administracia udalosti + ovladac	Jan Greppel	21.11.2012	2.12.2012
Revízia a oprava dokumenatacii	Zuzana Ujhelyiova	25.11.2012	8.12.2012
Zdokumentovať implementáciu Gui	Ivana Bohunicka	27.11.2012	5.12.2012

Tabuľka 5-8: Šprint 4

6 Manažment rizík

Zodpovedná osoba: Matúš Ujhelyi

6.1 Identifikované riziká pre projekt

Riziko #1	
Názov: Implementovanie inej ako požadovanej funkcionality	
Zodpovedný:	Ján Greppel
Časové ohraničenie:	Dlhodobé
Pravdepodobnosť výskytu:	Nízka
Dopad na úspech projektu:	Vysoký
Popis:	Vytvorenie funkcionality, ktorá nezodpovedá predstavám zákazníka/zadávatel'a projektu.
Aktivita predchádzania spusteniu:	Časté stretnutia so zákazníkom. Častá komunikácia ohľadom prototypu.
Spustenie:	-
Náprava:	-
Dátum uzatvorenia:	Koniec projektu

Tabuľka 6-1: Riziko 1

Riziko #2	
Názov: Nesplnenie naplánovaných úloh v korektnom časovom horizonte	
Zodpovedný:	Každý
Časové ohraničenie:	Krátkodobé
Pravdepodobnosť výskytu:	Vysoká
Dopad na úspech projektu:	Stredný
Popis:	Jednotlivé úlohy nie sú ukončované podľa plánu.
Aktivita predchádzania spusteniu:	Komunikácia s manažérom plánovania ohľadom úloh pred pridelením predpokladaného časového horizontu. Zohľadnenie časového kontextu iných ako tímových aktivít pri komunikácii s manažérom plánovania.
Spustenie:	Nesplnenie naplánovanej úlohy, meškanie projektu.
Náprava:	Predĺženie úlohy s vyvodenými dôsledkami. Realistickejší odhad budúcej činnosti.
Dátum uzatvorenia:	Koniec projektu

Tabuľka 6-2: Riziko 2

Riziko #3	
Názov: Neschopnosť implementovať požadovanú funkcionálnosť z dôvodu neznalosti jazyka C++	
Zodpovedný:	Bohunická, Ujhelyiová, Greppel, Rerko
Časové ohraničenie:	Krátkodobé
Pravdepodobnosť výskytu:	Nízka
Dopad na úspech projektu:	Stredný
Popis:	Funkcionálnosť nie je programátorom implementovaná z dôvodu, že nepozná C++.
Aktivita predchádzania spusteniu:	Pridelenie tejto implementácie inému programátorovi. Dôraz na modulárnu architektúru nezávislú na použitých jazykoch až na špecifické časti systému.
Spustenie:	Častá opakovaná chybovosť modulu dodávaného v C++.
Náprava:	Preplánovanie úlohy inému programátorovi.
Dátum uzatvorenia:	Korektné prebehnutie testov daného modulu.

Tabuľka 6-3: : Riziko 3

Riziko #4	
Názov: Nesprávne navrhnuté používateľské rozhranie	
Zodpovedný:	Bohunická
Časové ohraničenie:	Krátkodobé
Pravdepodobnosť výskytu:	Stredná
Dopad na úspech projektu:	Nízky
Popis:	Používateľské rozhranie nie je dostatočne intuitívne, resp. nie je vyhovujúce pre zákazníka.
Aktivita predchádzania spusteniu:	Analýza existujúcich riešení, časté a efektívne komunikovanie so zákazníkom oproti prototypu.
Spustenie:	Časté prerábanie existujúcich obrazoviek.
Náprava:	Vytvorenie návrhu rozhrania spolu so zákazníkom

Tabuľka 6-4: : Riziko 4

Riziko #5	
Názov: Nesplnenie termínov v organizácii predmetu	
Zodpovedný:	Ujhelyi, Greppel, Bohunická
Časové ohraničenie:	Strednodobé
Pravdepodobnosť výskytu:	Nízky
Dopad na úspech projektu:	Fatálny
Popis:	Niektorá z požadovaných častí projektu nebude odovzdaná alebo nebude odovzdaná načas.
Aktivita predchádzania spusteniu:	Každý člen tímu je oboznámený s plánom projektu aj organizáciou predmetu.
Spustenie:	Nesplnenie niektorého z daných termínov.
Náprava:	-

Tabuľka 6-5: : Riziko 5

6 | Manažment rizík

Riziko #6	
Názov: Neschopnosť naprogramovať multithreading spojenia	
Zodpovedný:	František Nagy
Časové ohraničenie:	Krátkodobé
Pravdepodobnosť výskytu:	Stredná
Dopad na úspech projektu:	Vysoký
Popis:	Neschopnosť naprogramovať paralelné vykonávanie spojení so synchronizáciou.
Aktivita predchádzania spusteniu:	Komunikácia ohľadom problému, správny návrh, naštudovanie problematiky paralelného programovania.
Spustenie:	Meškanie termínu dodania jadra hlavnej aplikácie.
Náprava:	Preplánovanie úlohy inému programátorovi.

Tabuľka 6-6: Riziko 6

Riziko #7	
Názov: Slabá a neefektívna komunikácia v tíme	
Zodpovedný:	Ján Greppel, čiastočne všetci
Časové ohraničenie:	Dlhodobé
Pravdepodobnosť výskytu:	Vysoká
Dopad na úspech projektu:	Stredný
Popis:	Slabé alebo žiadne komunikovanie niektorých členov v tíme
Aktivita predchádzania spusteniu:	Dohodnutie komunikačných kanálov, mailu, IRC a podobne.
Spustenie:	Neschopnosť zastihnúť členov, keď je to nutné, resp. žiadna odozva na otázky a výzvy poskytnuté prostredníctvom dohodnutých kanálov.
Náprava:	Dôsledný dohovor problémovým členom tímu.

Tabuľka 6-7: Riziko 7

Riziko #8	
Názov: Nezodpovedné plnenie manažérskych rolí	
Zodpovedný:	Všetci
Časové ohraničenie:	Dlhodobé
Pravdepodobnosť výskytu:	Stredná
Dopad na úspech projektu:	Stredný
Popis:	Jednotliví členovia si nedôsledne plnia svoje manažérske role.
Aktivita predchádzania spusteniu:	Diskutovanie ohľadne rolí na tímových stretnutiach.
Spustenie:	Neschopnosť zdokumentovať svoje výsledky manažmentu v tíme.
Náprava:	Dôsledný dohovor problémovým členom tímu. Naplánovanie úloh súvisiacich s manažérskymi rolami.

Tabuľka 6-8: Riziko 8

6 | Manažment rizík

Riziko #9	
Názov: Neschopnosť komunikovať s IR modulom	
Zodpovedný:	Juraj Muránsky
Časové ohraničenie:	Strednodobé
Pravdepodobnosť výskytu:	Nízka
Dopad na úspech projektu:	Stredný
Popis:	Nemožnosť vyslať z IR USB modulom.
Aktivita predchádzania spusteniu:	Dôsledné prečítanie dokumentácie k zariadeniu. Predčasné zaujímanie sa o túto problematiku. Komunikácia s väčším odborníkom v tejto oblasti.
Spustenie:	Nemožnosť vytvoriť aplikáciu vysielajúcu IR kódy na ovládanie TV alebo HIFI.
Náprava:	Priradenia IR modulu inému členovi tímu.

Tabuľka 6-5: Riziko 9

Riziko #10	
Názov: Odovzdanie produktu alebo dokumentácie v zlej kvalite	
Zodpovedný:	Všetci
Časové ohraničenie:	Dlhodobé
Pravdepodobnosť výskytu:	Nízka
Dopad na úspech projektu:	Vysoký
Popis:	Odovzdanie niektorých výstupov projektu v zlej, alebo neakceptovateľnej kvalite.
Aktivita predchádzania spusteniu:	Zabezpečenie vysokej motivácie členov, dohodnutie pravidiel a postupov v tímovom projekte. Prieběžná časť predčasná revízia.
Spustenie:	Opakované nedodržiavanie pravidiel.
Náprava:	-

Tabuľka 6-6: Riziko 10

Riziko #11	
Názov: Strata možnosti skompilovať kinect aplikáciu v dôsledku straty kľúčového člena	
Zodpovedný:	Juraj Muránsky, Matúš Ujhelyi
Časové ohraničenie:	Dlhodobé
Pravdepodobnosť výskytu:	Nízka
Dopad na úspech projektu:	Vysoký
Popis:	Neschopnosť skompilovať kinect C++ aplikáciu.
Aktivita predchádzania spusteniu:	Zabezpečenie kompilovania aplikácie aspoň 4 členom tímu.
Spustenie:	Strata Matúša Ujhelyiho ako jediného vývojára C++ kinect aplikácie.
Náprava:	-

Tabuľka 6-7: Riziko 11

7 Monitorovanie projektu

Zodpovedná osoba: Zuzana Ujhélyiová

7.1 Vymedzenie pojmov v manažmente úloh

Úloha – aktivita, ktorá má byť vykonaná vo vopred definovanom čase.

Manažment úloh – proces riadenia úloh počas životného cyklu softvéru. Zahŕňa plánovanie, testovanie, sledovanie a vykazovanie.

Nástroj na podporu plánovania – softvérový nástroj, ktorý umožňuje zlepšiť produktivitu a kvalitu výstupu projektu. Dovoľuje vytvárať, zaznamenávať, plánovať úlohy a taktiež i vykazovať činnosti s nimi spojené.

Začiatkový dátum – je dátum, ktorý definuje začiatok práce na pridelenej úlohe.

Dátum ukončenia – predpokladaný dátum dokedy má byť úloha ukončená.

Odhadovaný čas – je odhad času, ktorý je potrebný na splnenie danej úlohy.

Vykazovanie práce – je aktivita, počas ktorej riešiteľ zaznamenáva čas, ktorý strávil pri plnení úlohy.

Priorita úlohy – určuje dôležitosť danej úlohy. Je vyjadrená hodnotami *okamžite* (immediate), *urýchlene* (urgent), *vysoká* (high), *stredná* (normal) a *nízka* (low). Platí, že úlohy označené hodnotou *okamžite* sú najdôležitejšie.

Typ úlohy – väčšina softvérových nástrojov definuje viacero typov úloh, ktoré môžu byť vytvorené. Medzi základné typy patria *úloha* (task), *funkcionalita* (feature), *chyba* (bug) a *požiadavka na zmenu* (change request).

Stav – každá vytvorená úloha má v danom čase vždy jeden konkrétny stav. Môže ísť o stavy *nová* (new), *otvorená* (open), *priradená* (assignee), *riešená* (in progress), *vyriešená* (resolved), *testovaná* (under test), *uzavretá* (closed), *znovuotvorená* (reopened), *zamietnutá* (rejected). Stav úlohy je možné meniť podľa potreby.

Riešiteľ – osoba, ktorej je úloha pridelená a je zodpovedná za jej vyriešenie.

Cielová verzia – verzia produktu, v ktorej má byť daná úloha zahrnutá.

7.2 Typy úloh

Typ úlohy	Použitie
Úloha (task)	<ul style="list-style-type: none"> všeobecný typ úlohy najčastejšie používaný typ úlohy
Funkcionalita (feature)	<ul style="list-style-type: none"> používa sa na zadefinovanie úloh, ktorých výsledkom je nová funkcionalita
Chyba (bug)	<ul style="list-style-type: none"> vytvára sa v prípade nájdenia chýb v procese testovania
Zmenová požiadavka (change request)	<ul style="list-style-type: none"> používa sa na zadefinovanie zmenových požiadaviek zákazníka, ktoré neboli definované v dohodnutej funkcionalite a spravidla predstavujú zvýšenie ceny projektu

Tabuľka 7-1: Typy úloh

7.3 Priority úloh

Priorita úlohy	Použitie
Nízka (low)	<ul style="list-style-type: none"> úlohy nízkej priority nemajú dopad na celkový výsledok projektu
Stredná (normal)	<ul style="list-style-type: none"> štandarde používaný typ úlohy
Vysoká (high)	<ul style="list-style-type: none"> úlohy s vysokou prioritou
Urýchlene (urgent)	<ul style="list-style-type: none"> úlohy, ktoré treba riešiť čo možno najskôr
Okamžite (immediate)	<ul style="list-style-type: none"> úlohy, ktoré je nutné riešiť okamžite, predstavujú prekážku v pokračovaní projektu

Tabuľka 7-2: Priority úloh

7.4 Roly vystupujúce v rámci manažmentu úloh

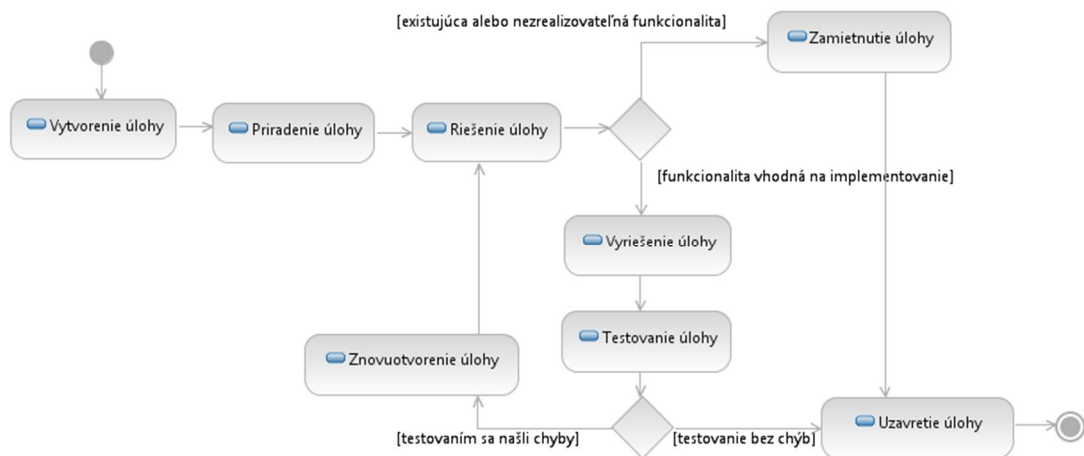
Rola	Zodpovednosť
Zákazník	<ul style="list-style-type: none"> zadefinovanie požiadaviek priorita požiadaviek
Vedúci tímu	<ul style="list-style-type: none"> komunikácia so zákazníkom ohľadom požiadaviek rozdelenie funkcionality do modulov a vytvorenie štruktúry
Manažér plánovania	<ul style="list-style-type: none"> vytvorenie úloh v rámci projektu a priradenie do príslušnej cieľovej verzie stanovenie priorít, začiatkový dátum, dátum ukončenia, odhadovaného času a priradenie úlohy riešiteľovi

7 | Monitorovanie projektu

	<ul style="list-style-type: none"> ▪ komunikácia s manažérom monitorovania
Programátor	<ul style="list-style-type: none"> ▪ rieši priradenú úlohu v odhadovanom čase ▪ odovzdá úlohu najneskôr v deň kedy má byť splnená ▪ opravuje nájdené chyby
Manažér kvality	<ul style="list-style-type: none"> ▪ testuje funkcionality jednotlivých úloh ▪ zaznamená nájdené chyby a oznámi ich programátorovi ▪ zabezpečí odovzdanie projektu v bezchybnom stave
Manažér monitorovania	<ul style="list-style-type: none"> ▪ kontrola postupu práce na projekte ▪ komunikácia o postupe prác s manažérom plánovania

Tabuľka 7-3: Roly

7.5 Proces manažmentu úloh



Obrázok 7-1: Proces manažmentu úloh

8 Manažment komunikácie

Zodpovedná osoba: Ján Greppel

8.1 Schôdze

Osobné stretnutia všetkých členov tímu sú 2-krát týždenne. Prvá schôdza je oficiálna (v časoch podľa rozvrhu), druhá je neoficiálna. Zápisnice z oficiálnych stretnutí sú zverejnené na oficiálnej stránke tímu.

Každá schôdza má vždy moderátora a zapisovateľa, ktorí sa týždenne menia. Úlohou moderátora je:

- začínať a ukončovať stretnutie,
- usmerňovať diskusiu správnym smerom (upozorňovať na zbytočné odbočenie od témy, nerelevantné poznámky, skákanie po bodoch),
- rekapitulovať a sumarizovať závery diskusie,
- moderovať s citom.

Úlohou členov tímu je aktívne sa zapájať do diskusie a rešpektovať moderátora.

8.2 Mail

Tím má vlastný mailing list, v ktorom sú uvedení všetci členovia tímu a taktiež pedagogický vedúci. Každý člen má povinnosť sa priebežne (minimálne každý druhý deň) informovať o obsahu mailovej komunikácie a včas reagovať. Pre zachovanie zmysluplnej mailovej komunikácie je dôležité dodržiavať základné pravidlá mailovej etikety:

- písať mail iba tým, ktorým je správa určená,
- dávať kópiu tým, ktorí majú byť informovaní o maile, ale nemusia priamo reagovať,
- písať výstižné predmety správ, ktoré sumarizujú obsah mailu,
- nepísať o veciach, ktoré sú predmetom väčšej diskusie (písať iba v prípade, ak to musí byť vybavené skôr ako do najbližšej schôdze),
- ak sa odpovedá na časť predchádzajúceho mailu, treba danú časť citovať,
- ak adresát potrebuje od ostatných potvrdenie, explicitne tak napíše (napr.: “Prosím, každý sa vyjadrite do <dátum a čas>”).

8.3 Interná wiki

Každý člen tímu, vrátane pedagogického vedúceho, má zriadený účet do internej wiki tímu. Sú tu rôzne návody, prístupy ku službám, kontakty na členov, poznámky zo schôdzí, interné dohody a rozpracované dokumenty.

8.4 Telefón a Skype

Telefónne čísla aj Skype účty všetkých členov sú zverejnené na internej wiki. V prípade potreby prebieha komunikácia aj týmtito spôsobmi.

8.5 Dokumentový server a GIT

Na dokumentovom serveri sa nachádzajú niektoré dokumenty, obrázky a súbory, ktoré majú rádovo niekoľko desiatok MB a nezmestia sa do mailových príloh.

Zdrojové kódy a iné textové súbory, ktoré sú často editované, sa nachádzajú v distribuovanom systéme pre riadenie revízií – GIT-e.

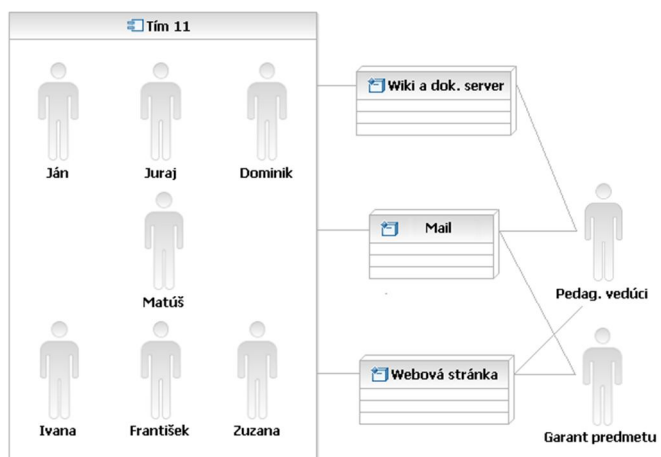
8.6 Oficiálna stránka

Oficiálna stránka tímu informuje verejnosť o priebehu projektu. Obsahuje stručné informácie o riešenej problematike, členoch tímu, zápisnice z oficiálnych stretnutí, priebežný plán projektu, dokumentácie k inžinierskemu dielu a riadeniu.

8.7 Topológia zainteresovaných v projekte

Na obrázku 8-1 je znázornená topológia zainteresovaných v tomto projekte, vzhľadom na používané komunikačné kanály. V rámci tímu prebieha osobná komunikácia, telefónne a Skype hovory viac krát do týždňa, vo všetkých smeroch. Pedagogický vedúci má prístup k internej wiki, dokumentovému serveru, vnútornému aliasu a príležitosť osobnej komunikácie s každým členom v rámci oficiálnych stretnutí. Garant predmetu komunikuje s tímom prostredníctvom mailového aliasu a o stave projektu sa môže informovať cez oficiálnu webovú stránku.

8 | Manažment komunikácie

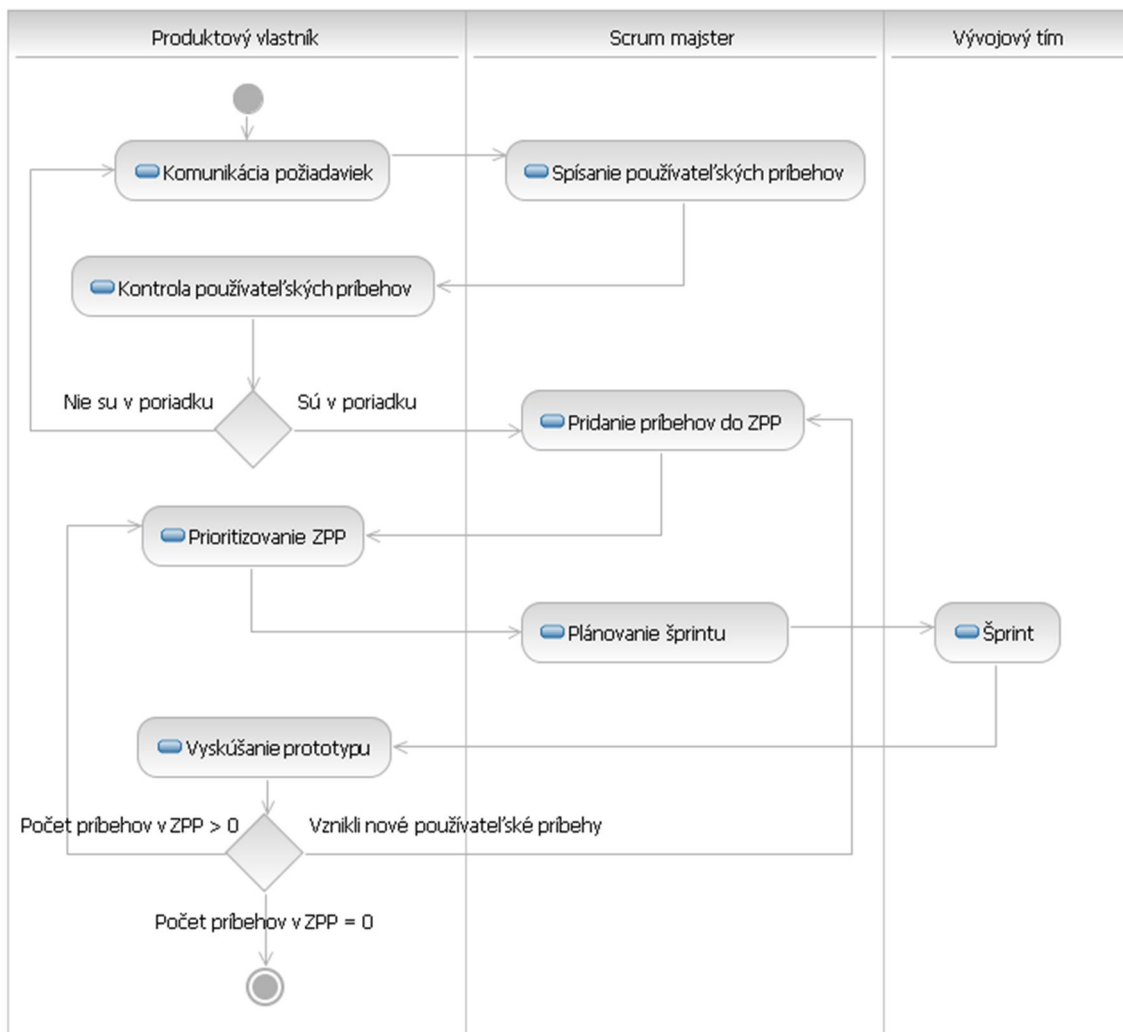


Obrázok 8-1: Topológia zainteresovaných

9 Manažment požiadaviek

Zodpovedná osoba: Ján Greppel

Obrázok 9-1 popisuje proces manažmentu požiadaviek, ktorý je následne konkrétne popísaný.



Obrázok 9-1: Manažment požiadaviek

9.1 Pohovor s produktovým vlastníkom

Vedúci projektu (angl. scrum master) na základe osobného pohovoru s produktovým vlastníkom spíše neformálne požiadavky do základných požiadaviek produktu (skr. ZPP). Každý používateľský príbeh musí obsahovať:

- rolu, ktorá vystupuje v príbehu,

9 | Manažment požiadaviek

- funkcionality, ktorá má byť implementovaná,
- pridanú hodnotu resp. benefit, ktorý vyplýva z implementovania funkcionality.

Okrem toho musí byť používateľský príbeh špecifikovaný tak, aby bol:

- nezávislý od ostatných príbehov,
- v jazyku produktového vlastníka,
- časovo a rozsahovo odhadnuteľný,
- primerane malý,
- testovateľný.

9.2 Prioritizácia základných požiadaviek produktu

Pri každej iterácii vývoja sa implementujú iba tie najdôležitejšie používateľské príbehy. Preto produktový vlastník zoradí ZPP podľa jeho úsudku od najdôležitejších po najmenej dôležité.

9.3 Šprint

Vedúci projektu vyberie niekoľko najprioritnejších príbehov, ktorých splnenie bude obsahom najbližšieho šprintu. Počet týchto príbehov závisí od počtu všetkých príbehov v ZPP a počtu všetkých šprintov. Výstupom tohto procesu bude nasadenie funkčného prototypu, obsahujúceho implementované a otestované používateľské príbehy definované produktovým vlastníkom.

9.4 Validácia požiadaviek

Nasadený funkčný prototyp bude predložený produktovému vlastníkovi, ktorý sa vyjadrí, či daná funkcionality (definovaná na začiatku procesu)

- je implementovaná podľa predstáv,
- predstavuje nejakú pridanú hodnotu.

Produktový vlastník má v tejto fáze lepšiu predstavu o produkte a takto môžu vzniknúť nové požiadavky, ktoré sa (1) špecifikujú do používateľských príbehov, (2) zaradia do ZPP, (3) prioritizujú a (4) naplňujú do ďalšieho šprintu.

9.5 Definícia používateľských príbehov v Microsoft Word

Obsah príbehov je určený podľa pravidiel:

- používať slová, ktorým používateľ rozumie,

9 | Manažment požiadaviek

- uvádzať vhodné príklady funkcionalít,
- v prípade ak je používateľský príbeh nejasný, pre ozrejmienie pridať opis prípadu použitia.

Forma príbehov je určená podľa pravidiel:

- v slovenskom jazyku,
- písmo iba Times New Roman, veľkosť 12,
- názov príbehu tučným písmom,
- rola, funkcionalita, benefit a prípad použitia
 - do tabuľky
 - odsadenie bunky zo všetkých strán 0.2“,
 - bez viditeľných okrajov.
 - kurzívou
- názov prípadu použitia podčiarknutý.

9.6 Šablóna používateľských príbehov

<názov>

Rola: <rola>

Funkcionalita: <funkcionalita>

Benefit: <benefit>

Prípad použitia: <názov prípadu použitia 1>

<1. krok>

<2. krok>

...

<názov prípadu použitia 2>

<1. krok>

...

10 Manažment kvality

Zodpovedná osoba: František Nagy

10.1 Konvencie pre písanie zdrojového kódu

Hlavným dôvodom pre písanie zdrojového kódu predpísanými konvenciami je jednoduchšia údržba, ale hlavne pochopiteľnosť pre ostatných a v konečnom dôsledku aj pre autora kódu. Jednotnosť taktiež zvyšuje relevantnosť nástrojov pre analýzu kódu.

Štruktúra triedy

- dokumentačný komentár
- deklarácia triedy
- statické premenné (v poradí public, protected, private)
- premenné (v poradí public, protected, private)
- konštruktory
- metódy

Štruktúra programu

- jedna inštrukcia (deklarácia, volanie funkcie) na riadok
tj. (názov funkcie doSomething je použitý kvôli príkladu)

```
pocet++;  
doSomething(pocet);  
nie:  
pocet++; doSomething(pocet);
```

- odsadzovanie 4 medzerami
- deklarácia premenných až pri ich potrebe
- inicializácia premenných pri deklarácii
- nepridávať zbytočne nový riadok pred "{"

tj.

```
if (...){  
    ...  
}  
nie
```

```
if (...)  
{
```

10 | Manažment kvality

- ```
...
}
```
- koniec bloku ("}") je zarovnaný so začiatkom (viď napr. príklad if)
- po čiarke nasleduje medzera  
tj. (názvy argumentov arg1, arg2. nie sú vhodné, opäť viď nižšie)  

```
doSomething(arg1, arg2, arg3)
nie
```
- pred a po operátore sú medzery  
tj. (a + b)  
nie (a+b)
- if - else blok pokračuje na rovnakom riadku ako } (ak tam je, inak nový riadok)  
tj.  

```
if (...){
} else {
...
}
alebo
if (...)
...
else
...

```
- nový if po else pokračuje v riadku  
tj.  

```
if (...){
...
} else if (...) {
...
}

```
- preferovaný je for cyklus pred while cyklom
- preferované je while (...) do {...} oproti do {...} while (...)
- zapuzdrenie premenných - použitie setterov a getterov na prístup
- funkcia maximálne na jednu obrazovku (preferované menej)
- dĺžku riadku udržiavať do 80 znakov, ak je to potrebné, zalomiť po čiarke, pred operátorom

## 10 | Manažment kvality

- nepoužívať číselné konštanty v porovnávaniach, cykloch (výnimka napr. 0, aj to treba zvážiť)
- 1 metóda robí 1 vec na 1 úrovni abstrakcie
- čím menej argumentov v metóde, ak treba, zhrnúť súvisiace argumenty do objektu

tj.

```
saveUser(User newUser)
nie
```

```
saveUser(String firstName, String lastName, long phoneNumber, String address, ...)
```

- nepoužívať boolean argumenty

tj. nie

```
render(true);
pre render(boolean isSuite)
```

ale

```
renderForSuite() a renderForSingleTest()
```

### Pomenovania

- popisujúce, radšej viacslonné (výnimky sú premenné cyklu)
- štruktúrovať mená, aby sa ľahšie hľadali (tj. nie studentName, studentId, studentSubject, ale nameStudent, idStudent, subjectStudent)
- triedy: podstatné mená, začínajú veľkým písmenom, potom CamelCase
- premenné: podstatné mená, začínajú malým písmenom, potom CamelCase
- konštanty: veľkými písmenami, oddeľované "\_"
- metódy: slovesá, začínajú malým písmenom
- vyhýbať sa menám ako data, processor, info
- všetky súvisiace pomenovania by mali byť podobné

tj. napr. nepoužívať fetch, retrieve, get ako ekvivalentné v rôznych triedach, ak sa raz niečo nejak pomenuje, príbuzné triedy (premenné, metódy) by mali byť pomenované podobne

- nepoužívať rovnaké pomenovanie vo viacerých významoch

### 10.2 PMD

Existuje viacero nástrojov na meranie kvality zdrojového kódu. Jedným z nástrojov statickej kontroly kvality je aj PMD. Pomocou neho je možné analyzovať zdrojový kód a identifikovať:

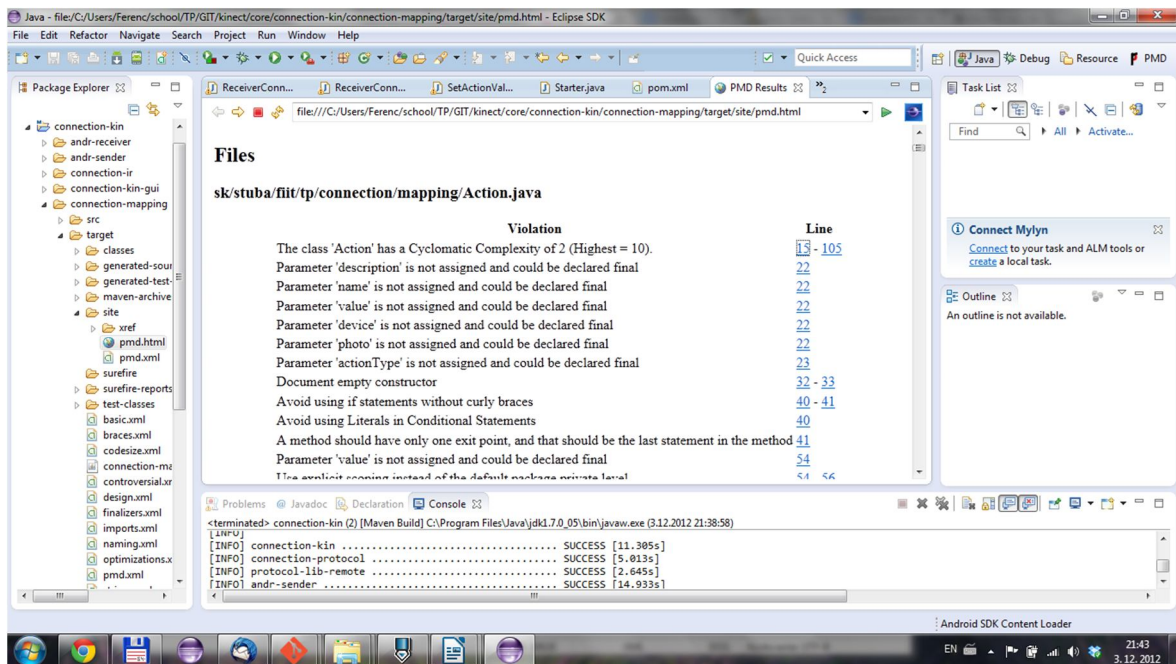
- možné chyby,
- mŕtvy kód,
- neoptimálny kód,
- duplicitný kód a iné.

Tento nástroj bol vybraný hlavne pre jeho jednoduchú integráciu s buildovacím nástrojom maven, ktorý už bol na projekte nasadený. PMD je do mavenu jednoducho dokonfigurovateľné vo forme plug-inu.

Nástroj je možné konfigurovať viacerými prednastavenými súbormi pravidiel, no dajú sa vytvoriť aj vlastné pravidlá. Pre potreby tohto projektu sme sa rozhodli vybrať rozumnú sadu pravidiel, ktorých dodržiavanie sme vyhodnotili ako prínosné. Aplikujeme pravidlá, ktoré skúmajú:

- Prázdne bloky kódu
- Nepoužité premenné, parametre
- Importy (viacnásobné, nepoužité)
- Premenné, ktoré by mohli byť finálne
- Veľkosť kódu (tried, funkcií)
- Cyklomatickú komplexitu
- Nesprávnu manipuláciu s textovými reťazcami

Po analýze, ktorá sa vykonáva jednoduchým vybuildovaním projektu s príslušným prepínačom (jxr:jxr pmd:pmd) sa vygeneruje pre každý modul správa vo formáte html so zoznamom chýb a varovaní. Zároveň s PMD sa používa aj plug-in JXR, ktorý do správy vygeneruje krížové odkazy na príslušné miesta v zdrojovom kóde.



Obrázok 10-1:Vygenerovaný výstup so zapnutými viacerými pravidlami

### 10.3 Cobertura

Zodpovedná osoba: Zuzana Ujhelyiová

V súvislosti so zabezpečením kvality výsledného produktu je vhodné sledovať mieru pokrytia zdrojového kódu testami. Vďaka tomu je možné určiť, či sú pokryté všetky vetvy, ako aj jednotlivé podmienky a ich časti. Existuje viacero nástrojov, ktoré spĺňajú túto úlohu. Keďže v rámci projektu bol vybraný nástroj *maven*, bolo vhodné a najjednoduchšie vybrať niektorý z pluginov, ktoré poskytujú túto funkcionality. Výbratý bol plugin Cobertura. Ten poskytuje možnosť konfigurácie hranice pre úspešné testy, teda na koľko % musí byť trieda pokrytá testami aby zbehol build aplikácie korektne. Pre náš projekt bola optimisticky zvolená hranica 85%. Okrem toho nástroj poskytuje export testových výstupov vo formáte .html, kde sú presne označené podmienky, riadky, metódy v kóde, ktoré sú a ktoré nie sú <sup>2</sup> pokryté testami. V prípade podmienky

```
if(premenna!=null){...}
```

je podmienka pokrytá testami na 100% v prípade, že je otestovaný pozitívny aj negatívny scenár, teda

```
premenna = null
premenna != null
```

[2] [http://www.e-reading.org.ua/bookreader.php/134601/Clean Code - A Handbook of Agile Software Craftsmanship.pdf](http://www.e-reading.org.ua/bookreader.php/134601/Clean_Code_-_A_Handbook_of_Agile_Software_Craftsmanship.pdf)

## 10 | Manažment kvality

V prípade, že kód vyhadzuje kontrolované výnimky (napr. `Preconditions.checkNotNull(...)`) je nutné odtestovať aj prípady, kedy výnimka má nastať. Na obrázku 1. je uvedený príklad reportu vytvoreného pomocou pluginu Cobertura.

The screenshot displays the Cobertura report interface. On the left, there are two panes: 'All Packages' and 'Classes'. The 'Classes' pane lists several classes with their coverage percentages, such as 'Action (65%)', 'ActionType (100%)', 'ConnKinObjectInformation (75%)', 'ConnectionKinGuiInterface (N/A)', 'ConnectionKinNetworkInterface (N/A)', 'DefaultNetworkInterface (0%)', 'Device (80%)', 'DeviceType (100%)', 'Event (58%)', 'GlobalMode (100%)', 'LocalMode (100%)', 'Mapper (49%)', 'Mode (70%)', 'ReceiverDevice (53%)', 'SenderDevice (53%)', and 'Situation (67%)'. The main pane shows the source code of a class with line numbers and coverage counts. The code is highlighted in green (2% coverage) or red (0% coverage). The code includes comments and a public method `loadConfiguration()` that handles database connections and schema creation.

Obrázok 10-2: Príklad reportu z Cobertury

### 10.4 Dohodnuté konvencie názvov v GUI.

Zodpovedná osoba: Ivana Bohunická

Implementácia a názvoslovie grafického rozhrania má svoje špecifiká. Je dôležité používať jednotné názvoslovie premenných - inšancií Swing tried, pretože grafické rozhranie bude zložité a bude obsahovať veľa komponentov a kontajnerov. Ak chceme s týmito komponentmi pracovať a volať ich pri implementácii, je nutné, aby boli ich názvy ľahko pochopiteľné z ponuky dopĺňania, ktorú poskytuje eclipse (Ctrl+ medzerník).

#### Názvoslovie pre swing triedy- Containers:

názov vo všeobecnosti :

<skratka názvu swing triedy (swing kontajner)><podstatné meno vyjadrujúce objekt, ktorého sa daný kontajner týka>

Skratky názvov swing tried - Containers:

- JScrollPane- scrolpane
- JTabbedPane- tabpane
- JPanel - panel
- JFrame - frame
- JDialog - dial

## 10 | Manažment kvality

Konkrétne príklady pomenovania inštancií swing tried - Containers:

- scroll pane pre celé okno zobrazujúce globálny aplikačný mód – `scrollPaneGlobalMode`
- panel obsahujúci všetky informácie o udalosti, nachádzajúci sa v obrazovke zobrazujúcej jednu konkrétnu namapovanú situáciu módu - `panelSituationDetailEvent`

### Názvoslovie pre swing triedy- Components:

názov vo všeobecnosti :

<skratka názvu swing triedy(swing komponent)><podstatné meno vyjadrujúce objekt, ktorého sa daný komponent týka><názov vyjadrujúci aktivitu vyvolanú nad objektom>

Skratky názvov swing tried- Components:

JButton- btn  
JRadioButton- radBtn  
JCheckBox- check  
JComboBox- comb  
JLabel- label  
JTextField- txtfield  
JEditorPane- editpane  
JTextPane- txtpane  
JTextArea- txtarea  
JTable- table  
JSeparator- separ

Konkrétne príklady pomenovania inštancií swing tried - Components:

- tlačidlo Späť pre návrat z obrazovky na definovanie novej situácie – `btnNewSituationBack`
- textové pole pre filtrovanie udalostí – `txtFieldEventFilter`
- názov (label) udalosti v ponuke pre definovanie novej situácie – `labelNewSituationEventOffered`
- tlačidlo pre pridanie lokálneho módu - `btnLocalModeAdd`

## 11 Manažment dokumentácie

---

Zodpovedná osoba: Dominik Rerko

### 11.1 Tvorba dokumentácie

Pri vytváraní dokumentácie k riadeniu ako aj k inžinierskemu dielu je potrebné použiť príslušnú šablónu. Tá sa nachádza v Git - e v priečinku „doc“. V týchto šablónach sú preddefinované štýly, ktoré musia byť korektne používané. Na označenie obrázkov, tabuliek a podobne sa používa štýl „Popis“. Pri odkazovaní sa na zdroje využívame automatické citácie štýlu „ISO 690 – Číselný odkaz“. Pri dokumentácii k inžinierskemu dielu je štruktúra (podľa možnosti) nasledovná: analýza, návrh, implementácia (testovanie).

Každý člen tímu je povinný čo najskôr dokončiť danú časť dokumentácie a následne ju odoslať manažérovi dokumentácie na integráciu. Zodpovednosť za nedodanú dokumentáciu na seba berie člen tímu, ktorý mal dokumentáciu dodať.

### 11.2 Tvorba zápisnice zo stretnutia

Pre vytvorenie zápisnice zo spoločného stretnutia sa používa pripravená šablóna. Tá je odstupná v Git - e v priečinku „doc“. Ukážka tejto šablóny sa nachádza v kapitole 3.1 (Šablóna zápisnice).

### 11.3 Dokumentácia zdrojových kódov v JavaDoc - u

JavaDoc parsuje deklarácie a komentáre zo zdrojových kódov a vytvára korešpondujúcu množinu HTML stránok popisujúcich triedy, rozhrania, konštruktory, metódy a premenné.

Pri dodržaní pravidiel definovaných v JavaDoc - u je zaručené, že zo zdrojových kódov sa bude dať automaticky generovať dokumentácia pomocou IDE nástroja Eclipse.

#### 11.3.1 Značky definované v JavaDoc

V rámci JavaDoc - u je definovaný veľký počet značiek využívaných pri tvorbe komentárov, pričom každá z nich má špecifický význam. Každá značka začína znakom @. Podrobný popis všetkých značiek je uvedený v JavaDoc dokumentácii.

V nasledujúcej tabuľke (Tabuľka 11-1) sú zobrazené značky dôležité pre každý projekt vytvorený v jazyku Java. Tabuľka obsahuje tvar značky a parametra, popis parametra, ďalej elementy, kde sa značka môže použiť a poradie. Poradie je dôležité v prípade použitia viacerých značiek. Nižšie číslo znamená prednosť.



## 11 | Manažment dokumentácie

V prípade viacnásobného použitia rovnakej značky je potrebné riadiť sa abecedným poradím parametra.

| Značka parameter | a                      | Popis parametra                                                                                                                                                                                                                                                                                                 | Element                                               | Poradie |
|------------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|---------|
| @author          | meno                   | <ul style="list-style-type: none"> <li>meno autora sa uvádza v tvare: „KrstnéMeno Priezvisko“</li> <li>meno je napísané veľkými písmenami v jednom riadku</li> <li>je možné uviesť viacerých autorov</li> <li>je povinné uviesť autora (autorov) daného elementu</li> <li>pr.: @author DOMINIK RERKO</li> </ul> | Balík<br>Trieda<br>Rozhranie                          | 1.      |
| @param           | MenoParametra<br>popis | <ul style="list-style-type: none"> <li>značka povinná pre každý parameter</li> <li>meno parametra predstavuje skutočné meno parametra, nie dátový typ</li> <li>popis oddelený od mena tabulátorom</li> <li>popis parametra uvedený v anglickom jazyku</li> <li>pr.: @param x the x-coordinate</li> </ul>        | Metóda<br>Konštruktor                                 | 3.      |
| @return          | popis                  | <ul style="list-style-type: none"> <li>značka povinná len pre metódy, ktorých návratová hodnota nie je void</li> <li>popis návratovej hodnoty uvedený v anglickom jazyku oddelený tabulátorom</li> <li>pr. @return the image at the specified URL</li> </ul>                                                    | Metóda                                                | 4.      |
| @see             | referencia             | <ul style="list-style-type: none"> <li>referencia uvedená v tvare textu, URL odkazu, alebo odkazu na konkrétny element</li> <li>pr.: @see JavaDoc documentation</li> </ul>                                                                                                                                      | Balík<br>Trieda<br>Rozhranie<br>Metóda<br>Konštruktor | 7.      |

## 11 | Manažment dokumentácie

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Premenná                                                          |    |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|----|
| <i>@since CisloVerzie</i>           | <ul style="list-style-type: none"> <li>• číslo verzie popisuje číslo verzie, od ktorej je daný element platný</li> <li>• po pridaní nového balíka nutné uviesť v rámci popisu balíka, aj každej triedy, ktoré obsahuje</li> <li>• po vytvorení novej triedy, alebo rozhrania uviesť značku len v popise triedy</li> <li>• po pridaní nového elementu uviesť značku len v prípade, ak bol element vytvorený v neskoršej verzii ako trieda</li> <li>• pri zmene prístupu elementu z protected na public nie je nutné uvádzať danú značku</li> <li>• pr.: @since 1.3.2</li> </ul> | Balík<br>Trieda<br>Rozhranie<br>Metóda<br>Konštruktor<br>Premenná | 6. |
| <i>@throws DruhChyby PopisChyby</i> | <ul style="list-style-type: none"> <li>• značka povinná pre každý element obsahujúcu klauzulu throws</li> <li>• druh chyby označuje typ chyby, ktorá môže nastať</li> <li>• popis chyby oddelený tabulátorom, napísaný v anglickom jazyku, krátko popisujúci danú chybu</li> <li>• pr.: @throws IOException If an input or output exception occurred</li> </ul>                                                                                                                                                                                                                | Metóda<br>Konštruktor                                             | 5. |
| <i>@version CisloVerzie</i>         | <ul style="list-style-type: none"> <li>• použiteľné maximálne raz v rámci elementu</li> <li>• povinná značka</li> <li>• uvedené v jednom riadku</li> <li>• číslo verzie v tvare „XXX.YYY.ZZZ“</li> <li>• prvá časť znamená hlavnú verziu a zmena nastáva v okamihu, keď pribudnú významné nové funkcie</li> </ul>                                                                                                                                                                                                                                                              | Balík<br>Trieda<br>Rozhranie                                      | 2. |

## 11 | Manažment dokumentácie

|  |                                                                                                                                                                                                                                                        |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|
|  | <ul style="list-style-type: none"><li>• druhá časť predstavuje podverziu a mení sa, keď príde k vylepšeniu existujúcich funkcií</li><li>• tretia časť predstavuje zmenu po oprave chýb, či iných malých zmien</li><li>• pr. : @version 1.3.2</li></ul> |  |  |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|

Tabuľka 11-1

### 11.3.2 Tvorba komentárov zdrojového kódu

Pre vytvorenie jednotnej dokumentácie v rámci projektu je potrebné dodržiavať pravidlá popísané v častiach 11.3.2.1 až 11.3.2.4.

#### 11.3.2.1 Štruktúra JavaDoc komentáru

Základná štruktúra JavaDoc komentáru má striktné daný formát. Komentár musí začínať postupnosťou znakov `/**`, nasledovaných znakom konca riadku. V každom nasledujúcom riadku musí byť znak `*` nasledovaný medzerou a následne značkou alebo textom. Koniec komentáru je umiestnený na novom riadku a má tvar `*/`.

Vzorový príklad:

```
/**
 * Komentar 1
 * Komentar 2
 */
```

#### 11.3.2.2 Komentovanie triedy

Vo vzorovom príklade je uvedený minimálny rozsah komentovanie triedy. Krátky popis obsahuje stručné, výstižné a dôležité informácie o triede. V rámci podrobnejšieho popisu je potrebné uviesť čoho sa trieda týka, aká je jej funkcia a čo robí. Taktiež je potrebné uviesť aj zobrazené značky.

Vzorový príklad:

```
/**
 * Kratky popis triedy
 *
 * Podrobnejši popis triedy
 *
 * @author
 * @version
 * @since
 */
```

## 11 | Manažment dokumentácie

### 11.3.2.3 Komentovanie metódy

Obsah komentára každej metódy začína krátkym opisom, ktorý podáva všeobecné informácie o metóde. Podrobnejší opis musí informovať o logických súvislostiach metódy, aby bolo každému programátorovi jasné, čo metóda robí. Metóda musí obsahovať aj uvedené značky.

Vzorový príklad:

```
/**
 * Kratky opis
 *
 * Podrobnejši opis
 *
 * @param
 * @return
 * @see
 */
```

### 11.3.2.4 Komentovanie premennej

Každá premenná musí byť zdokumentovaná spôsobom uvedeným vo vzorovom príklade. Popis premennej obsahuje krátky opis načo daná premenná slúži.

Vzorový príklad:

```
/**
 * Popis premennej.
 */
```

## 12 Manažment podpory vývoja

---

Zodpovedná osoba: Juraj Muránsky

### 12.1 Nightly build

V rámci riešenia projektu sme identifikovali potrebu pravidelne kontrolovať funkčnosť zdrojového kódu. Existuje viacero integračných systémov používaných v procese vývoja softvéru, v rámci ktorých je možné nastaviť viacero činností, ktoré sa majú spúšťať periodicky alebo po zmene domovského repozitára projektu. Všetky tieto riešenia sa vyznačujú rôznorodou funkcionalitou a robustnosťou, ktorú by sme v našom projekte nevyužili a preto sme sa rozhodli pre vlastné riešenie na mieru. Vytvorili sme skript, ktorý bude automaticky prekladať projekt a spúšťať všetky testy.

Základné požiadavky na buildovací systém boli pravidelné sťahovanie zmien z repozitára projektu a následný automatický build projektu aj so spustením testov. Ďalej bolo potrebné vyriešiť spôsob dostupnosti výstupov na našom projektovom server a odosielanie informácií o výstupe formou emailu.

V riešení sme použili už známe nástroje a technológie. Skript je nastavený periodicky, každú pol hodinu sa pokúsi spustiť build. V prípade, že ide o prvý beh skriptu v daný deň, pokúsi sa stiahnuť všetky zmeny z repozitára projektu a spustí build projektu pomocou nástroja maven. V rámci buildu sa spustia aj všetky testy, ktoré jednotlivé moduly obsahujú. Následne po ukončení buildu sa výsledky uložia na tímový FTP server a zároveň sa členom tímu odošle email o výsledku.

Cieľom bolo mať k dispozícii nástroj, ktorý je schopný informovať o stave poslednej – aktuálnej verzii hlavnej vetvy projektu v domovskom repozitári. Tento cieľ sme dosiahli vlastným a jednoduchým riešením integračného systému.

## 12 | Manažment podpory vývoja

```
Summarisation:
Result: SUCCESS
Tests runned: 39
Tests failed (errors or failures): 0
Elapsed maven time: 4:57.646s
Time runned: 03.12.2012 18:09:45
Memory usage: 33M/498M

Projects descriptions:

Project: sk.stuba.fiit.tp.connection.kin.connection.protocol.JSONToMessageObjectSerializerTest
Tests: 13
Failures: 0
Errors: 0
Skipped tests: 0
Time elapsed for tests: 0

Project: sk.stuba.fiit.tp.connection.kin.connection.protocol.MessageObjectToJSONDeserializerTest
Tests: 7
Failures: 0
Errors: 0
Skipped tests: 0
Time elapsed for tests: 0

Project: sk.stuba.fiit.tp.connection.mapping.ActionTest
Tests: 3
Failures: 0
Errors: 0
Skipped tests: 0
Time elapsed for tests: 0
```

Obrázok 12-1: Príklad výstupu z Nightly buildu

### 12.2 Git tutoriál

Zodpovedná osoba: Matúš Ujhelyi

#### URL adresy

Hlavný repozitár:

- `git@gitbus.fiit.stuba.sk:conneciton-kin-app/connection-kin-app.git`

Testovací repozitár (na zoznámenie sa s gitom):

- `git@gitbus.fiit.stuba.sk:test1/test1.git`

Externý tutoriál: <http://www.fuzzy.cz/cs/projekty/git-uzivatelska-prirucka/>

#### Návod:

Stiahnuť a nainštalovať: <http://code.google.com/p/msysgit/downloads/detail?name=Git-1.7.11-preview20120710.exe&can=2&q=full+installer+official+git>

##### 12.2.1.1.1 Základné nastavenie (stačí raz)

1. Spustiť Git Bash
2. `git config --global user.name "Your full name"`
3. `git config --global user.email "x_vase_@is.stuba.sk"`

## 12 | Manažment podpory vývoja

### 12.2.1.1.2 Vygenerovanie a nastavenie kľúčov (potrebné na komitovanie do repozitára, stačí raz)

1. Spustiť Git GUI
2. Help → Show SSH Key
3. Generate Key → Copy to Clipboard
4. Prihlásiť sa na gitbus.fiit.stuba.sk
5. Vo svojom profile “Manage SSH Keys” → Add SSH Key
6. Paste from clipboard → Save

### 12.2.1.1.3 Základný prehľad git - u

- **clone** - nakopírovanie vzdialeného repozitára k sebe na počítač
  - Git gui: Repository → Clone
  - Git bash: git clone 'URL'
- **commit** - uloženie zmien na svoju lokálnu kópiu repozitára
  - Git gui: Odklikat' súbory na commit, potom napísať hlášku, a commit
- **push** - natlačenie svojich zmien alebo vetiev na vzdialený repozitár
  - Git gui: Remote → Push → vybrať vzdialený repozitár → Vybrať vetvu na push → Push
  - Git bash: git push repository branch
- **pull** - prijatie zmien zo vzdialených repozitárov a pokus o spojenie a vyriešenie konfliktov (ekvivalent **git fetch** nasledovaný **git merge**)
  - Git bash: git pull
- **fetch** - prijatie zmien zo vzdialeného repozitára (spojenie musí byť manuálne cez **merge** alebo **rebase**), - odporúčam viac ako pull
  - Git Gui: Remote → Fetch from → vybrať repozitár
  - Git bash: git fetch repozitar
- **amend** - zmena posledného lokálneho commitu
  - Git Gui: Amend last commit → postup ako pri commite
- **branch** - vetva v strome, vytvorenie vetvy
  - Git Gui: Branch → Create
  - Git bash: git branch nazov
- **checkout** - presun medzi vetvami v strome
  - Git Gui: Branch → Checkout
  - Git bash: git checkout nazov
- **merge** - spojenie zmien z inej vetvy do aktuálnej vetvy
  - Git bash: git merge zdrojova\_vetva
- **rebase** - (pokročilé) presun commitov vetvy za commity z inej vetvy - mení históriu repozitára, umožňuje napr. presunúť lokálne zmeny za nové commity zo vzdialeného repozitára získané cez **fetch**
  - Git bash: git rebase vzdialena\_vetva

Binárne súbory sa pushujú len ak je to nevyhnutné. Inak sa tomu treba vyhýbať.

## 12 | Manažment podpory vývoja

### 12.2.2 Git naming

Zodpovedná osoba: Matúš Ujhelyi

#### Git commit tvar

##### Krátky tvar:

Prvý riadok: max 80 znakov.

Tvar: “priečinok: podpriečinok: anglický činný rod čo commit robí”

Druhý riadok: prázdny

Tretí riadok: Signed off ...

##### Príklad:

```
modules: kinect: connector-protocol: change udp communication to be unblocking
```

```
Signed-off-by: Matus Ujhelyi <xujhelyi@is.stuba.sk>
```

Ak je potrebný väčší komentár tak sa použije **dlhý tvar**:

##### Dlhý tvar:

Prvý riadok: max 80 znakov.

Tvar: “priečinok: podpriečinok: anglický činný rod čo commit robí”

Druhý riadok: prázdny

Tretí riadok: komentár po anglicky, najlepšie s určením priečinkov ak je to možné

.....

N-tý riadok komentáru(posledný):

N+1 riadok prázdny

N+2 riadok: Signed off ...

##### Príklad:



```
modules: kinect: connector-protocol: create InitMessage abstract class, add timeouts to init messages
```

```
modules: kinect: connector-protocol: create InitMessage class as parent to ReceiverInitMessage and SenderInitMessage
```

```
modules: kinect: connector-protocol: insert response timeout and ping timeout to init messages
```

Signed-off-by: Matus Ujhelyi <xujhelyi@is.stuba.sk>

### Git branch tvar

Nová vetva sa vytvára vždy z aktuálneho vrchu vetvy “master”. Tvar pomenovania pozostáva z prvých 3 písmen ako iniciálok mena tvoriťela vetvy a funkcionality oddelenej pomlčkou. Tvar je:

INC-popis-funkcionality-danej-vetvy

Príklad:

MUJ-kinect-gesture-recognition-improvement

## 12.3 Maven

Zodpovedná osoba: Zuzana Ujhelyiová

### Inštalácia:

1. zo stránky <http://maven.apache.org/download.html> stiahnuť verziu 3.0.4
2. rozbaľiť tam, kde sa má nachádzať home dir pre knižnice
3. pridať M2\_HOME do premenných prostredia tak aby cesta končila na úrovni bin adresára
4. pridať premennú M2 do premenných používateľa z hodnotou %M2\_HOME%\bin
5. do path-u pridať %M2% pre možnosť build-u z príkazového riadku.
6. skontrolovať, či je nastavené JAVA\_HOME a cesta je k JDK a v path - e sa nachádza %JAVA\_HOME%\bin
7. otvoriť príkazový riadok a skúsiť mvn – version (pod windowsom) výstup by mal vyzeráť:

## 12 | Manažment podpory vývoja

Apache Maven 3.0.4 (r1232337; 2012-01-17 09:44:56+0100)

Maven home: /usr/share/maven

Java version: 1.6.0\_37, vendor: Apple Inc.

Java home:

/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home

Default locale: en\_US, platform encoding: MacRoman

OS name: "mac os x", version: "10.6.8", arch: "x86\_64", family: "mac"

Pre integráciu s eclipsom je potrebné stiahnuť a nainštalovať plugin, konkrétne <http://eclipse.org/m2e/download/>.

V eclipse je potrebné mať nastavenú správnu cestu k JDK.

Build (cmd): na build z cmd je potrebné spustiť

```
mvn clean install
```

pre build bez spustených testov

```
mvn clean install -Dmaven.test.skip=true
```

pre pustenie iba testov bez buildu stačí zadať

```
mvn test
```

Build z eclipsu: pre build treba spustiť

```
run as --> maven build
```

pre spustenie testov stačí použiť

```
run as --> maven test
```

Pre vytvorenie nového projektu je potrebné nad parentom (connection-kin) vytvoriť nový maven module s archetype - om (vybrať default a následne upraviť pom.xml).

V novom module treba vymazať duplikátne groupId.

```
<groupId>sk.stuba.fiit.tp.connection.kin.test</groupId>
```

```
<version>1.0</version>
```

## 12 | Manažment podpory vývoja

Pre pridanie novej knižnice stačí nájsť potrebnú maven dependenciu v maven repository napr.:

```
<dependency>
 <groupId>com.google.code.gson</groupId>
 <artifactId>gson</artifactId>
 <version>2.2.2</version>
</dependency>
```

Ak ide o testovaciu dependenciu stačí pridať scope test:

```
<dependency>
 <groupId>junit</groupId>
 <artifactId>junit</artifactId>
 <version>4.10</version>
 <scope>test</scope>
</dependency>
```

### 12.4 Jigloo GUI Builder

Zodpovedná osoba: Ivana Bohunická

Domovská stránka pre editor Jigloo GUI Builder: <http://www.cloudgarden.com/jigloo/>

Jigloo GUI Builder sa inštaluje ako bežný plugin do Eclipse. Postup:

Eclipse → Help → Install new Software → Work with: <http://cloudgarden1.com/update-site>

Vytvorenie Swing triedy pomocou Jigloo: pravým na projekt → New → Other → Gui Forms → Swing → vyberiem napríklad: JFrame

Otvorenie Swing triedy pomocou Jigloo: pravým na triedu v projektovom strome → Open with → Form editor

Oficiálny tutoriál pre prácu s Jigloo:

[http://www.cloudgarden1.com/swing\\_tutorial/index.html](http://www.cloudgarden1.com/swing_tutorial/index.html)

## 13 Manažment chýb (Príloha A)

---

Zodpovedná osoba: Matúš Ujhelyi

### 13.1 Úvod

Dôvodom vytvorenia tejto metodiky je definovanie postupu identifikácie, nahlásenia a opravy chýb technického charakteru v jednom softvérovom inkremente dodanom zákazníkovi a popísanie postupu hlásenia a riešenia chýb v systéme Redmine.

### 13.2 Použité pojmy

**chyba** – odhalená funkcionálnosť systému spôsobujúca zmätenie alebo úplné narušenie činnosti systému

**repozitár** – úložisko zdrojového kódu a dokumentácie

**modul** – samostatne oddelená softvérová časť vyvíjaná nezávisle od systému

### 13.3 Súvisiace metodiky

Metodika práce so systémom git.

Metodika vytvárania a overovania akceptačných testov.

Metodika hodnotenia výsledkov testovania.

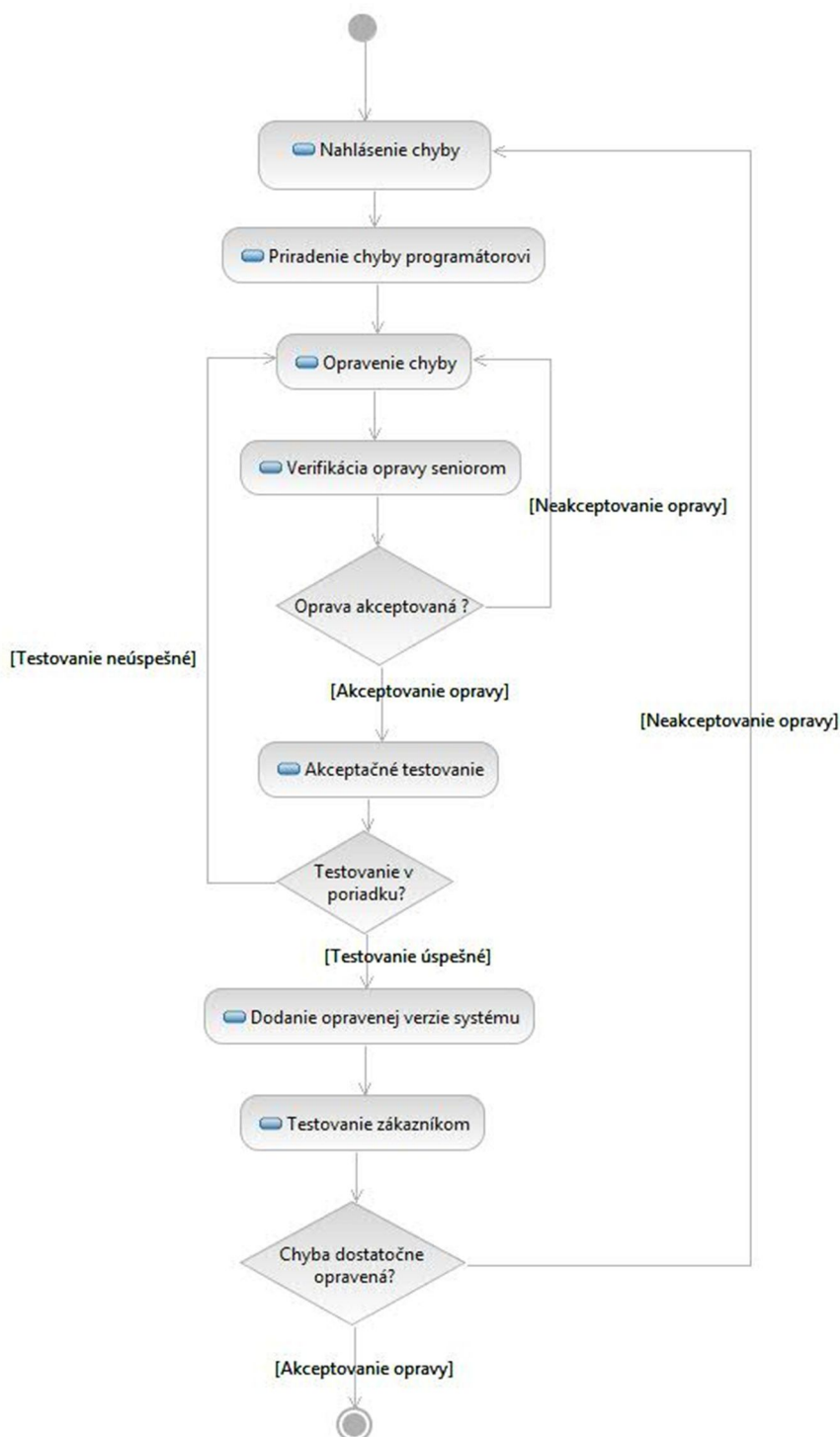
### 13.4 Roly a zodpovednosti

Rola	Zodpovednosť
Vedúci vývojového tímu	- získanie informácií o chybe - zabezpečenie dodania opravenej verzie prototypu
Zákazník (vlastník produktu)	- informovanie dodávateľa prostredníctvom systému na reportovanie chýb - akceptovanie opraveného prototypu
Tester	- vykonanie testovania čiernou skrinkou - vypracovanie správy o testovaní
Programátor	- prezentácia a vysvetlenie zdrojového kódu - oprava nefunkčného bloku kódu

## 13 | Manažment chýb (Príloha A)

	<ul style="list-style-type: none"> <li>- overenie funkčnosti opravy kódu</li> <li>- zaslanie požiadavky testerovi na testovanie</li> </ul>
Senior programátor	<ul style="list-style-type: none"> <li>- analýza a verifikácia opravného kódu programátora</li> <li>- príprava opravenej verzie prototypu</li> <li>- vypracovanie správy o oprave</li> </ul>
Manažér kvality	- akceptovanie/neakceptovania opravy

### 13.5 Životný cyklus chyby



Obrázok 13-1: Životný cyklus chyby

### 13.6 Proces nahlásenia chyby

*Vstup:* Identifikovaná a reprodukovateľná chyba zistená zákazníkom

*Výstup:* Požiadavka na opravenie chyby u dodávateľa doručená seniorovi programátorovi

*Identifikované roly:* zákazník, vedúci vývojového tímu, senior programátor

Základné kroky procesu:

- 1) Zákazník vyplní správu o chybe v dodávateľovom systéme
  - popíše aktuálnu konfiguráciu systému
  - popíše aktivity vykonávané počas identifikovania chyby
  - podá informáciu o verzii prototypu a závažnosti chyby
- 2) Vedúci vývojového tímu prezrie správu o chybe
  - a. správa neobsahuje dostatok informácií
    - vedúci kontaktuje zákazníka a požiada o ďalšie informácie
  - b. správa obsahuje dostatok informácií
    - vedúci vyberie tím a seniora, ktorý za danú chybu zodpovedá
- 3) Vedúci odošle požiadavku na opravu vybratému seniorovi

### 13.7 Proces pridelenia a opravy chyby

*Vstup:* Požiadavka na opravenie chyby

*Výstup:* Požiadavka na zaradenie chyby do verifikačného procesu

*Identifikované roly:* vedúci vývojového tímu, programátor, tester, senior programátor

Základné kroky procesu:

- 1) Senior programátor preskúma systém revízií a zistí zodpovedného programátora
- 2) Senior programátor poskytne prístup k požiadavke zodpovednému programátorovi
- 3) Programátor opraví chybnú implementáciu
  - zodpovedá za opravu chyby
  - otestuje funkčnosť opraveného kódu
- 4) Programátor oznámi a ozrejní opravu seniorovi
- 5) Senior zanalyzuje opravu a vyhodnotí či je:
  - a. chyba nedostatočne opravená

## 13 | Manažment chýb (Príloha A)

- vrátenie programátorovi požiadavku na opravu
  - b. chyba dostatočne opravená
- 6) Ak je chyba podľa seniora dostatočne opravená, programátor odošle testerovi požiadavku na otestovanie opravy

### 13.8 Proces verifikácie opravy

*Vstup:* Požiadavka na verifikáciu opravy chyby

*Výstup:* Správa o oprave chyby

*Identifikované roly:* vedúci vývojového tímu, programátor, tester, senior programátor

Základné kroky procesu:

- 1) Tester verifikuje korektnosť opravy a na základe testov rozhodne:
  - a. chyba nedostatočne opravená
    - tester pripraví správu o testovaní
    - v správe podrobne popíše proces testovania
    - podrobne popíše problémové spôsoby použitia systému
  - b. chyba dostatočne opravená
    - tester pripraví správu o testovaní
- 2) Tester odošle seniorovi správu o akceptačnom testovaní
- 3) Senior vyhodnotí testovanie ako:
  - a. neúspešné
    - senior vráti požiadavku na opravu programátorovi spolu s výsledkami a správou z testovania
  - b. úspešné
    - senior pripraví správu o oprave a poskytne ju manažérovi kvality

### 13.9 Proces doručenia opraveného inkrementu

*Vstup:* Správa o oprave chyby

*Výstup:* Opravený inkrement u zákazníka

*Identifikované roly:* zákazník, vedúci vývojového tímu, senior programátor, manažér kvality

## 13 | Manažment chýb (Príloha A)

### Základné kroky procesu:

- 1) Manažér kvality na základe správy verifikuje opravu a oznámi to vedúcemu vývojového tímu
- 2) Vedúci vývojového tímu pošle požiadavku na pripravenie nového prototypu seniorovi
- 3) Senior pripraví nový prototyp, v ktorom môže byť aj viac opravených chýb a na diaľku ho dodá zákazníkovi
- 4) Vedúci informuje zákazníka o aktualizácii systému
- 5) Zákazník verifikuje opravu chyby:
  - a. oprava neúspešná
    - upraví požiadavku na opravu a doplní informácie o novom prototypu
  - b. oprava úspešná
    - zatvorí požiadavku na opravu chyby v dodávateľovom systéme

### **13.10 Proces vytvorenie chyby v systéme redmine**

*Vstup:* Identifikovaná chyba

*Výstup:* Nahlásená chyba v systéme Redmine

*Identifikované roly:* zákazník (vlastník produktu), tester, programátor

Všetky textové polia sa vyplňajú po slovensky. Chyby môže programátorom hlásiť vlastník produktu, tester alebo iný programátor.

### Základné kroky procesu:

- 1) Na nahlásenie chyby sa použije nová „issue“ v systéme Redmine.
- 2) V poli „Tracker“ sa vyberie možnosť „Bug“.
- 3) Do poľa „Subject“ sa vloží názov identifikujúci chybu. Súčasťou názvu je aj názov modulu, v ktorom chyba vznikla. Názov nemá viac ako 60 znakov. Tvar je: „Module: Názov“
- 4) Pole „Description“ je určené na výstižný popis situácie ako nastala. Popis obsahuje postup reprodukovateľnosti chyby. Píše sa vo vetách, slovenským jazykom s diakritikou.
- 5) Pole „Status“ sa vyplní na „New“.
- 6) Pole „Priority“ slúži na udelenie priority chyby pričom možné hodnoty sú:
  - **nízka** (low) - chyba nespôsobuje haváriu systému a je možné jej vyriešenie odložiť na neskorší termín



## 13 | Manažment chýb (Príloha A)

- **stredná** (normal) – bežná chyba objavujúca sa v systéme, vyriešenie sa očakáva v rámci najbližšieho ukončenia šprintu
  - **vysoká** (high) – chyba, ktorá zásadne mení správanie systému oproti požiadavkám, jej oprava má byť do najbližšieho tímového stretnutia
  - **akútna** (urgent) – kritická chyba ktorej oprava musí byť realizovaná do 48 hodín
- 7) Do pola „Target version“ sa uvádza najbližšie očakávaný názov šprintu po ktorom bude chyba opravená.
  - 8) Pole „Due date“ predstavuje dátum stretnutia tímového projektu, do ktorého má byť chyba opravená.
  - 9) Pole „Estimated time“ sa nastaví na očakávaný počet hodín, ktoré budú na opravu chyby využité.
  - 10) Pole „Assignee“ sa vyplní človekom, ktorý na danom module pracoval. Ak to nie je možné, vyplní sa človekom, ktorý daný modul po ňom prebral.
  - 11) Záznam sa uloží stlačením tlačidla „Submit“.

The screenshot shows a bug tracking form with the following details:

- Tracker:** Bug
- Subject:** Kinect: Spadnutie aplikácie počas nahrávania gesta
- Parent task:** (empty)
- Description:** Pri nahrávaní nového gesta je možné aplikáciu zhodiť stlačením tlačidla "STOP".
- Status:** New
- Priority:** Urgent
- Assignee:** Matus Ujhelyi
- Category:** Milestone
- Target version:** Sprint 1
- UserStoryPoints:** (empty)
- Start date:** 2012-10-28
- Due date:** 2010-10-30
- Estimated time:** 2 Hours
- % Done:** 0 %

Obrázok 13-2: Vytvorenie chyby

### 13.11 Vyriešenie chyby v systéme redmine

*Vstup:* Nahlásená chyba v systéme Redmine

*Výstup:* Opravená chyba v systéme Redmine čakajúca na overenie

*Identifikované roly:* tester, programátor, manažér kvality

## 13 | Manažment chýb (Príloha A)

Všetky textové polia sa vyplňajú po slovensky.

### Základné kroky procesu:

- 1) Nájdenie a otvorenie issue reprezentujúcu chybu.
- 2) V poli „Status“ sa vyberie možnosť „Resolved“.
- 3) Do poľa „Assignee“ sa vyberie člen tímu, ktorý to má skontrolovať. Ak na danom module pracoval človek sám, tak vyberie manažéra kvality, ak na danom module pracovali viacerí, vyberie jedného zo spolupracovníkov.
- 4) Pole „Done“ sa vyplní na 100%.
- 5) Pole „Spent time“ sa vyplní počtom hodín, ktoré boli spotrebované na opravu.
- 6) Pole „Activity“ sa vyplní na „Development“.
- 7) Pole „Notes“ sa vyplňa nasledovne:
  - prvý riadok - popis ako bola chyba opravená, nie viac ako 100 znakov
  - druhý riadok - prázdny
  - tretí riadok - URL repozitára kde je chyba opravená
  - štvrtý riadok - názov vetvy kde je chyba opravená
  - piaty riadok - úvodný riadok commitu z repozitára
  - šiesty riadok - commit ID
  - siedmy riadok - prázdny
  - ôsmy a viac – dodatočné informácie
- 8) Záznam sa uloží stlačením tlačidla „Submit“.

Ak je nutné pridať súbor, príkladá sa v pdf formáte.

### Update

Change properties (More)

Status *	Resolved	Start date	2012-10-24
Priority *	Normal	Due date	2012-10-30
Assignee	Matus Ujhelyi	Estimated time	5 Hours
Category		% Done	100 %
Target version	Sprint 1		
UserStoryPoints			

Log time

Spent time	6 Hours	Activity	Development
Comment			

Notes

**B** *I* U ~~S~~ **C** H1 H2 H3 Text formatting: Help

Presunutie blokujúceho volania do samostatného vlákna.

```
git@gitbus.fiit.stuba.sk:conneciton-kin-app/connection-kin-app.git
MUJ-fiit-kinect-init-import
modules: kinect: move waiting on socket to separate thread
eaac5ebe7ea84505f2064bad43a190ff69458ca8
```

Spôsobené nedôsledným odchyťávaním signálov socketu.

Obrázok 13-3: Vyríšenie chyby

## 14 Manažment iterácií projektu (Príloha B)

---

Zodpovedná osoba: Ivana Bohunická

### 14.1 Úvod

Táto metodika je určená všetkým členom vývojového tímu. Účelom tejto metodiky je definovať postup plánovania iterácií projektu pri vývoji agilnou metódou Scrum. Na manažment úloh bude použitý nástroj Redmine.

### 14.2 Pojmy

- **Sprint = šprint** - predstavuje vývojový cyklus, počas ktorého tím produkuje novú komplexnú časť projektu, iterácia vývoja projektu
- **Product owner = vlastník produktu** - vlastník a zadávateľ produktu, zákazník
- **Scrum master** - člen tímu zodpovedný za tím
- **Product backlog** - zoznam úloh a požiadaviek, ktoré sa očakávajú od produktu
- **Sprint backlog** - zoznam úloh na produkte, ktoré majú byť hotové do konca šprintu
- **Sprint budget** - čas, ktorý má tím k dispozícii na šprint
- **Sprint review** - stretnutie všetkých zúčastnených na konci šprintu, kde sa predvedie výsledok šprintu
- **Sprint retrospective** - stretnutie tímu s diskusiou o vylepšení tímovej práce
- **Planning poker cards** - bodovací systém

### 14.3 Použitá literatúra

[1] Fernandes, J.M., Almeida, M.: Classification and Comparison of Agile Methods. Quality of Information and Communications Technology, 2010, vol. , no. , s. 391- 396.

[2] Metodika manažmentu a vytvárania úloh

[3] Barla, M.: Agilný vývoj metodikou Scrum.

[http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/materialy/scrum\\_2012.pdf](http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/materialy/scrum_2012.pdf)

## 14.4 Roly a zodpovednosti

<i>Rola</i>		<i>Zodpovednosť</i>
<i>Vlastník produktu</i>		<ul style="list-style-type: none"> <li>▪ Definovanie požiadaviek do Product backlog</li> <li>▪ Určenie priorít definovaných požiadaviek</li> <li>▪ Poskytnutie spätnej väzby vývojovému tímu ohľadom predvedených výsledkov projektu</li> </ul>
<i>Tím</i>	<i>Bežný člen tímu</i>	<ul style="list-style-type: none"> <li>▪ Definovanie požiadaviek do Product backlog</li> <li>▪ Ohodnotenie definovaných požiadaviek</li> <li>▪ Odhad trvania jednotlivých úloh</li> <li>▪ Predvedenie výsledkov svojej práce</li> <li>▪ Zhodnotenie práce na šprinte</li> </ul>
	<i>Scrum master</i>	<ul style="list-style-type: none"> <li>▪ Zodpovednosť za celý tím</li> <li>▪ Zodpovednosť za komunikáciu v tíme</li> <li>▪ Zodpovednosť za komunikáciu tímu so zákazníkom</li> </ul>
	<i>Manažér plánovania</i>	<ul style="list-style-type: none"> <li>▪ Spočítanie času na šprint</li> <li>▪ Stanovenie dĺžky šprintu</li> <li>▪ Výber úloh do šprintu</li> <li>▪ Pridelenie úloh členom tímu</li> </ul>

Tabuľka 14-1: Roly a zodpovednosti

## 14.5 Proces plánovania metódou Scrum

Proces plánovania metódou Scrum pozostáva z viacerých krokov, ktorých prehľad sa nachádza v tabuľke (Tabuľka 14-2).

<i>Kapitola</i>	<i>Krok</i>
14.5.1	Vytvorenie Product backlog
14.5.1.1	Ohodnotenie Product backlog
14.5.1.2	Zoradenie Product backlog
14.5.2	Plánovanie šprintu
14.5.2.1	Spočítanie Sprint budget
14.5.2.2	Vytvorenie Sprint backlog
12.5.2.3	Pridelenie úloh Sprint backlog

## 14 | Manažment iterácií projektu (Príloha B)

14.5.3	Koniec šprintu
14.5.3.1	Sprint review
14.5.3.2	Sprint retrospective

Tabuľka 14-2: Proces plánovania

### 14.5.1 Vytvorenie Product backlog

Vstup: Požiadavka na vznik produktu

Výstup: Vytvorený zoznam požiadaviek a súčastí produktu

Zodpovedné roly: vlastník produktu, Tím

Na úplnom začiatku vývoja produktu sa stretne vlastník produktu s celým tímom a stanoví požiadavky vyvíjaného systému - vzniká Product backlog. Hocikto môže pridať položky do zoznamu Product backlog.

#### 14.5.1.1 Ohodnotenie Product backlog

Vstup: Vytvorený zoznam požiadaviek a súčastí produktu

Výstup: Ohodnotené jednotlivé požiadavky a súčasti produktu

Zodpovedné roly: Tím, Scrum master

Ohodnotenie Product backlog znamená stanoviť odhad jednotlivých položiek aké sú veľké a náročné. Nemyslí sa tým časový odhad, ale odhad úsilia. Ohodnotenie prebieha pomocou Planning poker cards, na ktorých je uvedená Fibonacciho postupnosť [3].

Scrum master postupne vyberá položky zo zoznamu a vyzýva členov tímu aby ich ohodnotili. Každý člen má kartičky s Fibonacciho postupnosťou a na výzvu každý otočí karičku s hodnotou akou odhaduje náročnosť danej položky.

#### 14.5.1.2 Zoradenie Product backlog

Vstup: Vytvorený zoznam požiadaviek a súčastí produktu

Výstup: Zoradený zoznam požiadaviek a súčastí produktu podľa ich priorit

Zodpovedné roly: vlastník produktu

## 14 | Manažment iterácií projektu (Príloha B)

Keď je zoznam požiadaviek kompletný, vlastník produktu ho zoradí. Zoradiť a teda určiť priority položiek môže iba vlastník produktu. Na základe ohodnotenia, ktoré vykonáva tím, môže vlastník produktu meniť svoje priority jednotlivých položiek z Product backlog.

### 14.5.2 Plánovanie šprintu

Jednotlivé vývojové inkreментy sú rozplánované do šprintov. Šprint predstavuje vývojový cyklus produktu, počas ktorého tím vytvorí novú komplexnú časť projektu. Na konci šprintu musí byť výsledkom práce odovzdateľná časť funkcionality [1].

#### 14.5.2.1 Spočítanie Sprint Budget

Vstup: Zoradený Product backlog

Výstup: Vypočítaný sprint budget

Zodpovedné roly: manažér plánovania

Je potrebné spočítať čas v hodinách, koľko má tím k dispozícii na prácu počas šprintu. Zároveň sa ohraničí začiatok a koniec šprintu, to znamená že sa určí dĺžka šprintu. Každý šprint musí mať rovnakú dĺžku [3].

#### 14.5.2.2 Vytvorenie Sprint backlog

Vstup: Zoradený Product backlog a vypočítaný Sprint budget

Výstup: Vytvorený Sprint backlog

Zodpovedné roly: Tím, Scrum master, manažér plánovania

Na začiatku šprintu manažér plánovania vyberie požiadavky a úlohy, ktoré budú vykonané a implementované počas aktuálneho šprintu. Úlohy sa vyberajú z vrchu Product backlog. Tieto úlohy sú zaradené do aktuálneho Sprint backlog. Je nutné rozdeliť ich na menšie vykonateľné úlohy - podúlohy.

Každý podúlohe sa stanoví odhadovaný čas, koľko sa predpokladá že bude trvať. Toto určujú členovia tímu použitím Planning poker cards. Odhadovaný čas všetkých úloh sa sčíta a porovná so Sprint budget, ktorý musí byť aspoň rovný alebo väčší ako odhadovaný čas úloh. V opačnom prípade sa musí zmenšiť rozsah Sprint backlog, aby bola dodržaná uvedená podmienka:

## 14 | Manažment iterácií projektu (Príloha B)

*Sprint budget [h] >= súčet odhadovaného času všetkých podúloh [h]*

### 14.5.2.3 Pridelenie úloh Sprint backlog

Vstup: Vytvorený Sprint backlog

Výstup: Rozdelené úlohy medzi členov tímu

Zodpovedné roly: manažér plánovania

Každéj úlohe je pridelená zodpovedná osoba, ktorej povinnosťou je zabezpečiť splnenie úlohy. Manažér plánovania vyberá zodpovednú osobu vzhľadom na schopnosti a vytáženie jednotlivých členov tímu.

### 14.5.3 Koniec šprintu

#### 14.5.3.1 Sprint review

Vstup: Ukončený šprint

Výstup: Stretnutie a vykonaná kontrola výstupu na konci šprintu

Zodpovedné role: vlastník produktu, Tím, manažér plánovania, Scrum master

Na konci šprintu sa stretnú všetky zainteresované osoby, kedy každý člen tímu predvedie svoju prácu. Vlastník produktu vidí, čo sa urobilo a má príležitosť poskytnúť spätnú väzbu. Vlastníkovu spätnú väzbu je veľmi dôležitý element, pretože v tomto štádiu vývoja produktu, po šprinte, je ešte možné produkt pozmeniť, ak je to potrebné.

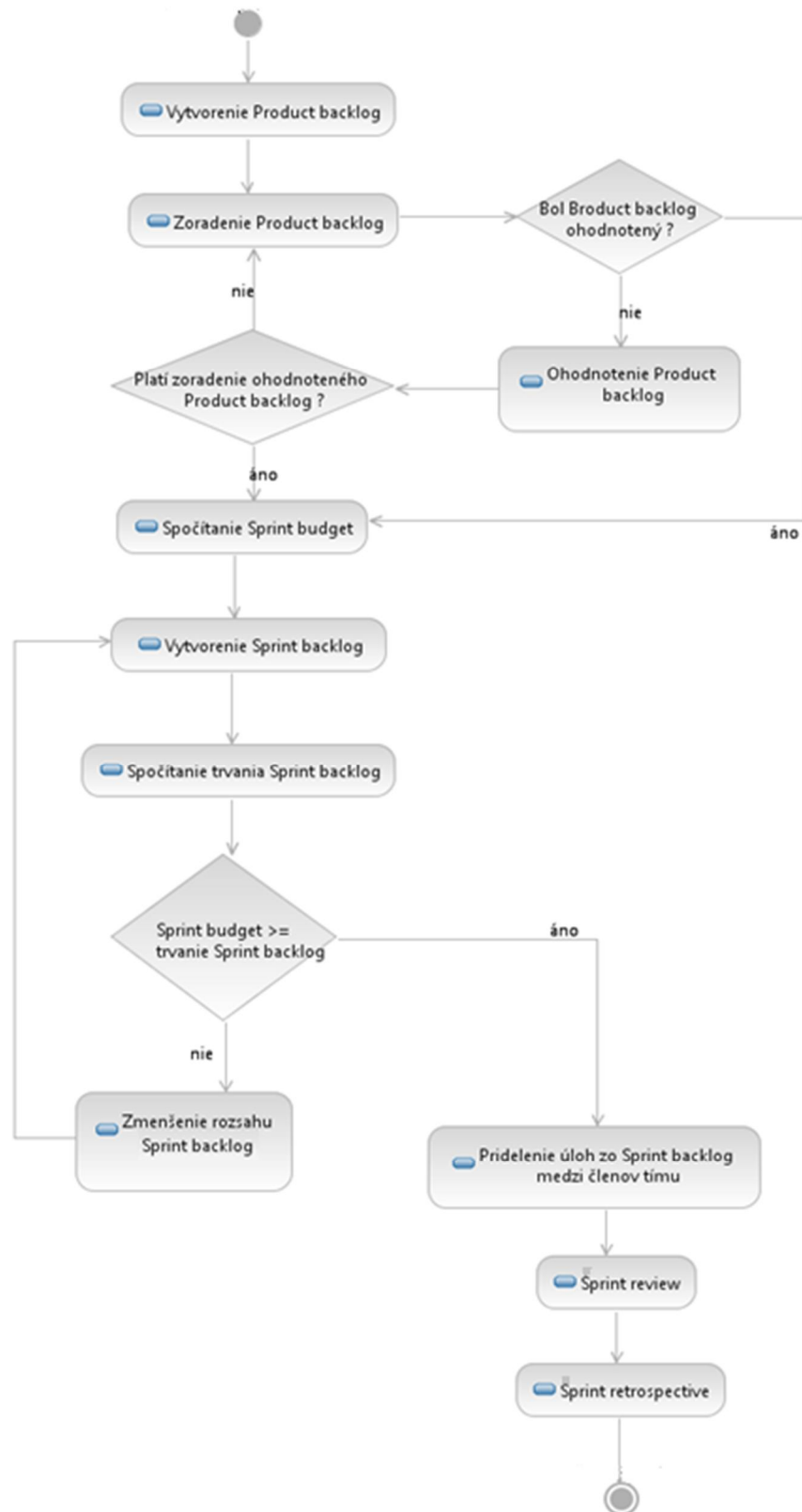
#### 14.5.3.2 Sprint retrospective

Vstup: Koniec šprintu

Výstup: Vydiskutované problémy a nedostatky ukončeného šprintu

Zodpovedné role: Tím, Scrum master

Stretnutie členov tímu, prípadne aj vlastníka produktu, ktoré slúži na diskusiu o priebehu šprintu. Zaoberá sa otázkami čo bolo dobré, čo bolo zlé a čo treba urobiť v nasledujúcom šprinte inak. Cieľom je zlepšenie tímovej práce v ďalších šprintoch.



Obrázok 14-1: Priebeh šprintu



### 14.6 Plánovanie metódou Scrum v Redmine

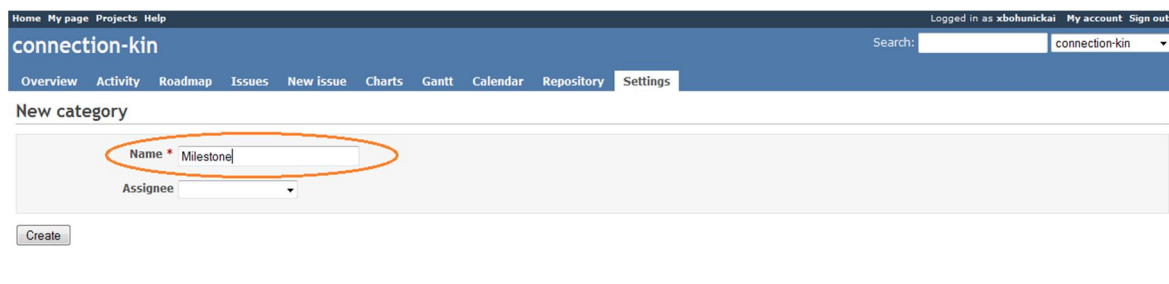
Táto kapitola poskytuje opis zaznamenávania plánov do softvérového nástroja Redmine určeného pre manažment úloh. Redmine však nie je v princípe určený na manažment agilného vývoja. Je tu ale možnosť prispôbiť si funkcie, ktoré ponúka tak, aby adekvátne reprezentovali agilné postupy plánovania. Cieľom je ukázať ako možno prispôbiť a používať Redmine pre plánovanie a tvorbu úloh Scrum metódou.

#### 14.6.1 Vytvorenie Product backlog

V rámci nástroja Redmine neexistuje konkrétny typ úlohy, ktorý by sa nazýval alebo by charakterizoval Product backlog. Produkt Backlog sa bude reprezentovať pomocou kategórie milestone.

Postup vytvorenia novej kategórie Milestone (Obrázok 14-2)

1. Otvoriť záložku „Settings“
2. Otvoriť záložku „Issue Categories“
3. Zvoliť „New Category“
4. Vyplniť údaje o novej kategórii- zadať názov: Name → „Milestone“
5. Potvrdiť tlačidlom „Create“



Obrázok 14-2: Vytvorenie novej kategórie

Keď už existuje kategória úloh typu Milestone, založí sa nové „Issue“ s názvom Product backlog a priradí sa mu kategória Milestone.

Postup založenia úlohy Product backlog (Obrázok 14-3)

1. Otvoriť záložku „New Issue“
2. Zvoliť typ úlohy: Tracker → „Task“
3. Zadať názov: Subject → „Product backlog“
4. Zvoliť kategóriu: Category → „Milestone“
5. Vyplnenie zvyšných údajov popisuje metodika vytvárania úloh [2].

## 14 | Manažment iterácií projektu (Príloha B)

The screenshot shows the 'New issue' form in Redmine. The 'Tracker' is set to 'Task'. The 'Subject' field contains 'ProductBacklog' and is circled in orange. Below the subject field is the 'Parent task' field. The 'Description' field has a rich text editor with various formatting options. The 'Status' is set to 'New', 'Priority' to 'Normal', and 'Assignee' is empty. The 'Category' field is also circled in orange. The 'Target version' is set to 'Milestone'. The 'Start date' is 2012-10-20 and the 'Due date' is 2012-12-14. The 'Estimated time' is in hours and the '% Done' is 0%.

Obrázok 14-3: Založenie úlohy

### 14.6.2 Vytvorenie šprintu

Šprint sa v Redmine simuluje použitím verzie. Zakladá sa vlastná verzia a označí sa zvoleným názvom, napríklad Sprint 1.

Postup na vytvorenie vlastnej verzie Šprint (Obrázok 14-4)

1. Otvoriť záložku „Settings“
2. Otvoriť záložku „Versions“
3. Zvoliť „New Version“
4. Vyplniť názov verzie: Name → „Sprint <poradové číslo šprintu>“
5. Potvrdiť tlačidlom „Create“

The screenshot shows the 'New version' form in Redmine. The 'Name' field contains 'Sprint 3' and is circled in orange. The 'Description' field contains 'šprint 3= 3.release'. The 'Status' is set to 'open'. The 'Wiki page' field is empty. The 'Date' is 2012-10-20 and the 'Sharing' is set to 'Not shared'. A 'Create' button is located at the bottom left of the form.

Obrázok 14-4: Vytvorenie šprintu

## 14.7 Vytváranie úloh v Redmine

Vytvorenie úlohy je proces pozostávajúci z viacerých krokov (Tabuľka 14-3) pričom tieto kroky stanovujú atribúty úlohy. Tieto atribúty je potrebné vyplniť v Redmine ako parametre.

<i>Kapitola</i>	<i>Krok</i>
7.1	Zaradenie úlohy do šprintu
7.2	Vyplnenie parametrov úlohy

Tabuľka 14-3: Vytvorenie úlohy

### 14.7.1 Zaradenie úlohy do šprintu

Šprint obsahuje viacero úloh a požiadaviek vybraných z Product backlog. Tieto zvolené úlohy sa rozdelia na najmenšie vykonateľné podúlohy a tvoria Sprint backlog. V rámci plánovania úloh v Redmine je potrebné tieto úlohy založiť a zaradiť ich do príslušného šprintu. Zámerom je teda úlohy prideliť vytvoreným verziám.

Aby sa zabezpečila príslušnosť úlohy aj do Product backlog, je potrebné uviesť ho ako rodičovskú úlohu - „Parent task“. Každá úloha však nemusí byť priamo podúlohou Product backlog, stačí ak má vo svojej hierarchii úloh na začiatku hierarchického stromu Product backlog.

Postup vytvorenia úlohy zaradenej do verzie prvého šprintu a Product backlogu (Obrázok 14-4):

1. Otvoriť záložku „New Issue“
2. Zvoliť rodičovskú úlohu: Parent task → „<číslo úlohy Product backlog>“
3. Zvoliť verziu: Target version → „Sprint 1“
4. Vyplniť ostatné údaje podľa metodiky [2] a potvrdiť tlačidlom „Create“

New issue

Tracker \* Task

Subject \* implementovať komunikačnú knižnicu

Parent task 4004

Description

Status \* New

Priority \* Normal

Assignee

Category

Target version Sprint 1

UserStoryPoints

Start date 2012-10-20

Due date 2012-10-24

Estimated time Hours

% Done 0%

Obrázok 14-5: Zaradenie úlohy do šprintu

## 14 | Manažment iterácií projektu (Príloha B)

### 14.7.2 Vyplnenie parametrov úlohy

Na obrázku (Obrázok 14-5) je viditeľné, že vytvorenie úlohy má okrem „Parent task“ a „Target version“ ďalšie parametre na vyplnenie. Podrobný postup nastavenia jednotlivých parametrov úlohy sa nachádza v metodike manažmentu úloh [2].

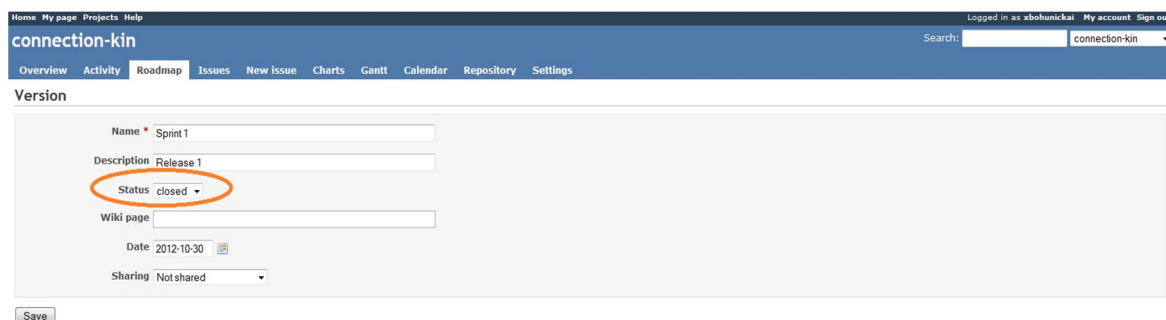
## 14.8 Ukončenie iterácií projektu v Redmine

### 14.8.1 Ukončenie šprintu

Koniec šprintu a jeho uzavretie sa nastavuje aj v Redmine. Šprint je simulovaný verziou, to znamená že sa zatvorí verzia reprezentujúca šprint.

Postup pre zatvorenie vlastnej verzie šprint (Obrázok 14-5):

1. Otvoriť záložku „Settings“
2. Otvoriť záložku „Versions“
3. V zozname ponúkaných verzii nájsť požadovanú verziu a zvoliť pri nej možnosť: „Edit“
4. Zvoliť stav: Status → „Closed“
5. Potvrdiť tlačidlom „Save“



Obrázok 14-5: Ukončenie šprintu

### 14.8.2 Ukončenie Product backlog

Na konci projektu, po splnení celého Product backlog sa zatvorí Product backlog v Redmine. Product backlog bol vytvorený ako úloha - „Task“, od ktorej dedili všetky ostatné úlohy. Tieto úlohy sú na konci uzavreté tak ako popisuje metodika [2]. Potom je nutné zatvoriť aj Product backlog.

Postup pre zatvorenie úlohy Product backlog (Obrázok 14-6):

1. Otvoriť záložku „Issues“
2. Otvoriť úlohu „Product backlog“
3. Zvoliť možnosť „Update“

## 14 | Manažment iterácií projektu (Príloha B)

4. Zmeniť nastavenie stavu na: Status → „Closed“
5. Potvrdiť tlačidlom „Submit“

Update Update Log time Watch Duplicate Copy Move Delete

Change properties (More)

Status \* Closed

Priority \* High

Assignee

Category Milestone

Target version

UserStoryPoints

Start date 2012-10-18

Due date 2012-12-12

Estimated time 37.5 Hours

Log time

Spent time Hours

Activity --- Please select ---

Comment

Notes

**B** *I* U ~~S~~ **C** H1 H2 H3

Text formatting: Help

Files

Optional description

Add another file (Maximum size: 24.5 MB)

Obrázok 14-6: Ukončenie product backlog

## 15 Manažment úloh v jednotlivých etapách projektu (Príloha C)

---

Zodpovedná osoba: Zuzana Ujhelyiová

Účelom tejto metodiky je presne zadefinovať postup manažmentu úloh v rámci jednotlivých etáp softvérového projektu s využitím nástroja Redmine. Dokument je určený vedúcemu tímu, manažérovi monitorovania, programátorom a testerom.

### 15.1 Vymedzenie pojmov

Úloha – aktivita, ktorá má byť vykonaná vo vopred definovanom čase.

Manažment úloh – proces riadenia úloh počas životného cyklu softvéru. Zahŕňa plánovanie, testovanie, sledovanie a vykazovanie.

Nástroj na podporu plánovania – softvérový nástroj, ktorý umožňuje zlepšiť produktivitu a kvalitu výstupu projektu. Dovoľuje vytvárať, zaznamenávať či plánovať úlohy a taktiež i vykazovať činnosti s nimi spojené.

Začiatkový dátum – je dátum, ktorý definuje začiatok práce na pridelenej úlohe

Dátum ukončenia – predpokladaný dátum, dokedy má byť úloha ukončená.

Odhadovaný čas – je odhad času, ktorý je potrebný na splnenie danej úlohy.

Vykazovanie práce – je aktivita, počas ktorej riešiteľ zaznamenáva čas, ktorý strávil pri plnení úlohy.

Priorita úlohy – určuje dôležitosť danej úlohy. Je vyjadrená hodnotami *okamžite* (immediate), *urýchlene* (urgent), *vysoká* (high), *stredná* (normal) a *nízka* (low). Platí, že úlohy označené hodnotou *okamžite* sú najdôležitejšie.

Typ úlohy – väčšina softvérových nástrojov definuje viacero typov úloh, ktoré môžu byť vytvorené. Medzi základné typy patria *úloha* (task), *funkcionalita* (feature), *chyba* (bug) a *požiadavka na zmenu* (change request).

Stav – každá vytvorená úloha sa má v danom čase vždy jeden konkrétny stav. Môže ísť o stavy *otvorená* (open), *priradená* (assignee), *riešená* (in progress), *vyriešená* (resolved), *testovaná* (under test), *uzavretá* (closed), *znovu otvorená* (reopened), *zamietnutá* (rejected). Stav úlohy je možné meniť podľa potreby.

Riešiteľ – osoba, ktorej je úloha pridelená a je zodpovedná za jej vyriešenie.

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

Cieľová verzia – verzia produktu, v ktorej má byť daná úloha zahrnutá.

### 15.2 Súvisiace metodiky

- [1] Metodika o plánovaní šprintov v rámci riadenia projektu metódou SCRUM.
- [2] Metodika tvorenia a realizácie testov pre jednotlivé scenáre.
- [3] Metodika inštalácie a konfigurácie systému Redmine.

### 15.3 Roly vystupujúce v rámci manažmentu úloh

Rola	Zodpovednosť
Zákazník	<ul style="list-style-type: none"><li>- zadefinovanie požiadaviek</li><li>- prioritizácia požiadaviek</li></ul>
Vedúci tímu	<ul style="list-style-type: none"><li>- komunikácia so zákazníkom ohľadom požiadaviek</li><li>- rozdelenie funkcionality do modulov a vytvorenie štruktúry</li></ul>
Manažér plánovania	<ul style="list-style-type: none"><li>- vytvorenie úloh v rámci projektu a priradenie do príslušnej cieľovej verzie</li><li>- stanovenie priorít, začiatkový dátum, dátum ukončenia, odhadovaného času a priradenie úlohy riešiteľovi</li><li>- komunikácia s manažérom monitorovania</li></ul>
Programátor	<ul style="list-style-type: none"><li>- rieši priradenú úlohu v odhadovanom čase</li><li>- odovzdá úlohu najneskôr v deň kedy má byť splnená</li><li>- opravuje objavené chyby</li></ul>
Manažér kvality	<ul style="list-style-type: none"><li>- testuje funkcionality jednotlivých úloh</li><li>- zaznamená nájdené chyby a oznámi ich programátorovi</li><li>- zabezpečí odovzdanie projektu v bezchybnom stave</li></ul>
Manažér monitorovania	<ul style="list-style-type: none"><li>- kontrola postupu práce na projekte</li><li>- komunikácia o postupe prác s manažérom plánovania</li></ul>

Tabuľka 15-1: Roly a zodpovednosti

## 15.4 Vymedzenie typov úloh a priorit

Typ úlohy	Použitie
Úloha (task)	<ul style="list-style-type: none"> <li>- všeobecný typ úlohy</li> <li>- najčastejšie používaný typ úlohy</li> </ul>
Funkcionalita (feature)	<ul style="list-style-type: none"> <li>- používa sa na zadefinovanie úloh, ktorých výsledkom je nová funkcionalita</li> </ul>
Chyba (bug)	<ul style="list-style-type: none"> <li>- vytvára sa v prípade nájdania chýb v procese testovania</li> </ul>
Zmenová požiadavka (change request)	<ul style="list-style-type: none"> <li>- používa sa na zadefinovanie zmenových požiadaviek zákazníka, ktoré nie definované v dohodnutej funkcionalite a spravidla predstavujú zvýšenie ceny projektu</li> </ul>

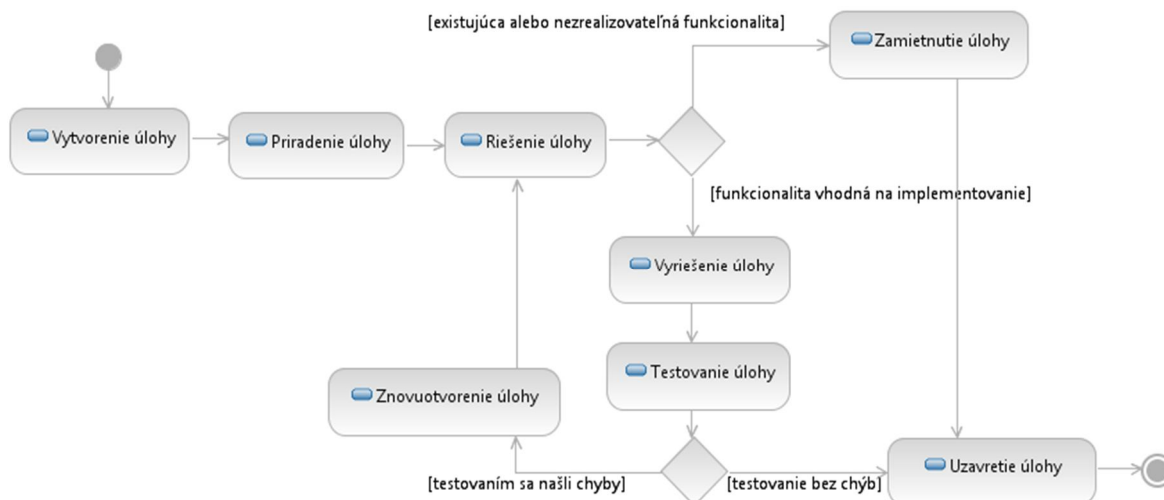
Tabuľka 15-2: Typy úloh a použitie

Priorita úlohy	Použitie
Nízka (low)	<ul style="list-style-type: none"> <li>- úlohy nízkej priority</li> <li>- nemajú dopad na celkový výsledok projektu</li> </ul>
Stredná (normal)	<ul style="list-style-type: none"> <li>- štandarde používaný typ úlohy</li> </ul>
Vysoká (high)	<ul style="list-style-type: none"> <li>- úlohy s vysokou prioritou</li> </ul>
Urýchlene (urgent)	<ul style="list-style-type: none"> <li>- úlohy, ktoré treba riešiť čo možno najskôr</li> </ul>
Okamžite (immediate)	<ul style="list-style-type: none"> <li>- úlohy, ktoré je nutné riešiť okamžite predstavujú prekážku v pokračovaní projektu</li> </ul>

Tabuľka 15-3: Priorita úloh a použitie



### 15.5 Proces manažmentu úloh



Obrázok 15-1: Manažment úloh

### 15.6 Opis jednotlivých krokov v rámci manažmentu úloh

V tejto kapitole sú zadané jednotlivé kroky v procese plnenia úlohy od jej vytvorenia až po jej uzavretie.

#### 15.6.1 Vytvorenie úlohy

Vstup: špecifikácia požiadaviek zákazníka

Výstup: vytvorená úloha

Identifikované roly: manažér plánovania

Základné kroky procesu:

1. Manažér plánovania sa oboznámi s požiadavkami zákazníka
  - naštuduje si pripravenú analýzu produktu, ktorý má byť vytváraný,
  - navrhne rozdelenie do modulov.
2. Na základe priorit rozdelí požiadavky do jednotlivých cieľovej verzie [1].
3. Vytvorí jednotlivé úlohy pre príslušné cieľové verzie:
  - definuje predmet úlohy a očakávaný výstup z nej, ako aj príznačný názov (z popisu úlohy musí byť jasné, čo má riešiteľ urobiť a ako má vyzerat' výsledok),
  - zdefiniuje odhadovaný čas,
  - nastaví začiatkový dátum, dátum ukončenia,
  - zadá verziu produktu, do ktorej patrí funkcionálnosť danej úlohy.

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

### 15.6.2 Pridelenie úlohy

Vstup: vytvorená úloha

Výstup: priradená úloha

Identifikované úlohy: manažér plánovania

Základné kroky procesu:

1. Manažér plánovania sa oboznámi s požiadavkami úlohy.
2. Na základe zručností, skúseností a voľných kapacít vyberie programátora, ktorému pridelí úlohu.

### 15.6.3 Riešenie úlohy

Vstup: priradená úloha, znovuo tvorená úloha

Výstup: implementovaná funkcionálna špecifikovaná v úlohe

Identifikované roly: programátor

Základné kroky procesu:

1. Programátor začne pracovať na úlohe
  - zmení stav úlohy na otvorená.
2. Programátor realizuje kroky smerujúce k vyriešeniu úlohy
  - na dennej báze vykazuje čas strávený na úlohe,
  - implementuje potrebnú funkcionálnu (ak ide o implementačnú úlohu), píše dokumentáciu špecifikovanú v úlohe napr. návrh riešenia, popis architektúry systému (ak ide o dokumentačnú úlohu),
  - k funkcionálnosti tvorí automatické testy.

### 15.6.4 Vyriešenie úlohy

Vstup: splnená úloha (implementovaná funkcionálna ak ide o implementačnú úlohu, napísaná špecifikovaná dokumentácia ak ide o dokumentačnú úlohu)

Výstup: vyriešená úloha

Identifikované úlohy: programátor, vedúci projektu

Základne kroky procesu:

1. Riešiteľ splnil úlohu.
2. Implementačná úloha: spustí akceptačné testy
  - riešenie spĺňa všetky akceptačné testy,
  - riešenie nespĺňa akceptačné testy,
    - programátor opraví danú chybu,

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

- vráti sa na bod 2.
- 3. Programátor nastaví úlohu stav vyriešená.
- 4. Programátor priradí úlohu manažérovi kvality, ktorý bude vykonávať testovanie funkcionality riešenia.
- 2. Dokumentačná úloha: vedúci projektu si prezrie dokumentáciu vytvorenú riešiteľom.
- 3. Zintegruje noavovytvorenú dokumentáciu s existujúcou dokumentáciou.

### 15.6.5 Testovanie úlohy

Vstup: vyriešená úloha

Výstup: úloha bez chýb, úloha obsahujúca chyby

Identifikované roly: manažér kvality

Základné kroky procesu:

1. Manažér kvality pretestuje implementovanú funkcionality
  - testuje pozitívne aj negatívne scenáre funkcionality [2],
  - porovná vytvorenú funkcionality so špecifikáciou projektu.
2. Pri testovaní
  - boli objavené chyby alebo nesúlad so špecifikáciou:
    - pokračuje staťou 15.6.6Znovuotvorenie úlohy
  - neboli objavené chyby:
    - pokračuje staťou 15.6.7Uzavretie úlohy

### 15.6.6 Znovuotvorenie úlohy

Vstup: pri testovaní úlohy boli objavené chyby

Výstup: znovuoťvorenie úlohy

Identifikované úlohy: manažér kvality, programátor

Základne kroky procesu:

1. Manažér kvality spíše zoznam objavených chýb aj s príkladom ich vyvolania.
2. Znovuoťvorí otestovanú úlohu a pridá k nej zoznam objavených chýb.
3. Pridá danú úlohu programátorovi.
4. V prípade nejasností komunikuje s programátorom o objavených chybách.

### 15.6.7 Uzavretie úlohy

Vstup: otestovaná úloha bez chýb, zamietnutá úloha

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

Výstup: uzavretá úloha

Identifikované úlohy: manažér kvality, vedúci tímu

Základne kroky procesu:

1. Otestovaná úloha bez chýb:
  - manažér kvality skontroluje, či overil všetky pozitívne aj negatívne scenáre funkcionality,
  - manažér kvality uzavrie otestovanú úlohu.
2. Zamietnutá úloha:
  - vedúci tímu uvedie dôvod uzavretia úlohy,
  - vedúci tímu následne uzavrie úlohu.

### 15.6.8 Zamietnutie úlohy

Vstup: riešená úloha (požadovaná funkcionality už existuje prípadne nie je možné ju implementovať)

Výstup: zamietnutá úloha

Identifikované roly: programátor, vedúci tímu

Základné kroky procesu:

1. Programátor rieši úlohu, ktorá mu bola pridelená.
2. V procese riešenia zistí, že funkcionality, ktorú má implementovať duplikuje existujúcu resp. nie je realizovateľná.
3. Úlohu odkomunikuje s vedúcim tímu a dá do stavu zamietnutá.

## 15.7 Činnosti spojené s manažmentom úloh

### 15.7.1 Vykazovanie práce na úlohe

Vstup: riešená úloha

Výstup: riešená úloha s vykázaným časom práce

Identifikované roly: programátor

Základné kroky procesu:

1. Programátor pracoval na úlohe a strávil jej plnením určitý čas.
2. Zmení množstvo zostávajúceho času vyhradeného na splnenie danej úlohy.

### 15.7.2 Kontrola postupu práce počas aktuálnej cieľovej verzie

Vstup: realizovaný projekt

Výstup: analýza odovzdanej verzie projektu

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

Identifikované roly: manažér monitorovania, manažér plánovania

Základné kroky procesu:

1. Manažér monitorovania:
  - porovnáva vykázanú prácu so skutočným stavom,
  - kontroluje množstvo vykonanej práce po ukončení aktuálnej cieľovej verzie.
2. Priebežne komunikuje s manažérom plánovania a odovzdáva mu svoje postrehy a námety na vylepšenie postupu práce.
3. Na konci každej cieľovej verzii vypracuje dokument, v ktorom zhodnotí možnosti vylepšenia práce na ďalšej cieľovej verzii.

### 15.8 Vykazovanie postupu prác na úlohe v systéme Redmine

Kapitola opisuje krokov pri vykazovaní postupu prác v systéme Redmine. Ide o aktivitu realizovanú riešiteľmi projektu, ktorý majú v systéme Redmine priradené úlohy.

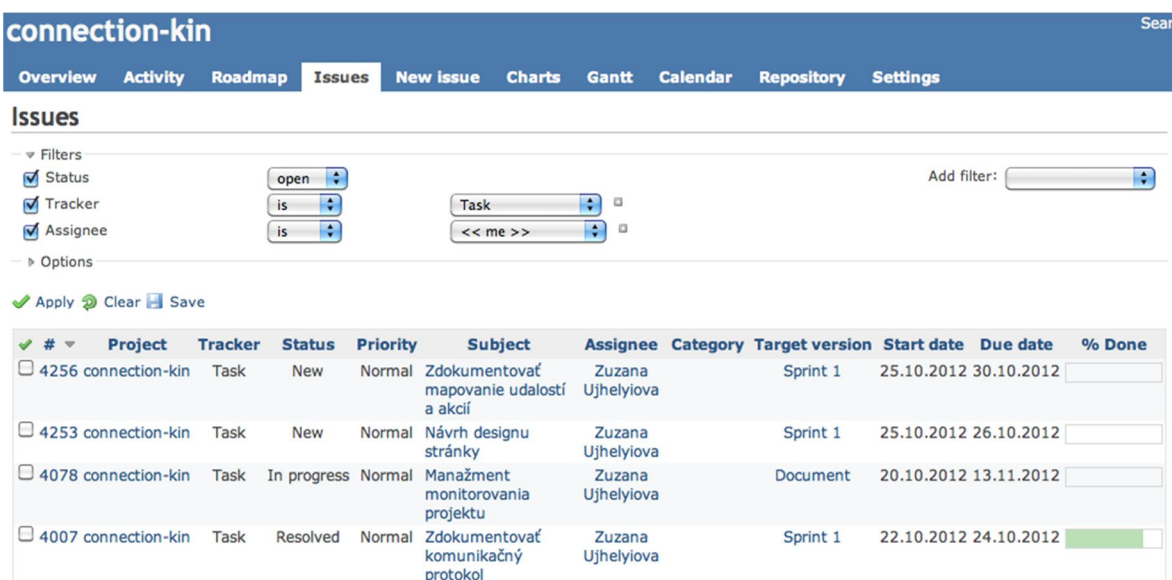
Predpoklad pre nasledujúce: používateľ má konto v systéme Redmine, je priradený na projekt a vie sa do systému prihlásiť.

#### 15.8.1 Postup práce na úlohe z pohľadu využitia systému Redmine

##### 15.8.1.1 Začiatok práce na úlohe

Predpoklad: vytvorená úloha na každú činnosť, ktorú ide riešiteľ vykonávať.

1. Z ponuky vyberie projekt, na ktorom pracuje.
2. Zobrazí si zoznam úloh, ktoré má priradené.



#	Project	Tracker	Status	Priority	Subject	Assignee	Category	Target version	Start date	Due date	% Done
4256	connection-kin	Task	New	Normal	Zdokumentovať mapovanie udalostí a akcií	Zuzana Ujhelyiova		Sprint 1	25.10.2012	30.10.2012	
4253	connection-kin	Task	New	Normal	Návrh designu stránky	Zuzana Ujhelyiova		Sprint 1	25.10.2012	26.10.2012	
4078	connection-kin	Task	In progress	Normal	Manažment monitorovania projektu	Zuzana Ujhelyiova		Document	20.10.2012	13.11.2012	
4007	connection-kin	Task	Resolved	Normal	Zdokumentovať komunikačný protokol	Zuzana Ujhelyiova		Sprint 1	22.10.2012	24.10.2012	

Obrázok 15-2: Príklad úloh

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

3. Vyberie úlohu, na ktorej chce začať pracovať.
4. Vyberie možnosť update a zmení stav úlohy z New na In progress.

### 15.8.1.2 Vykazovanie práce na úlohe

*Predpoklad:* používateľ má vytvorenú úlohu a tá je v stave In progress (riešená).

1. Vyberie úlohu, ktorej realizáciou sa zaoberal.
2. Vyberie možnosť Log time.
3. Zaznamená dobu, ktorú strávil plnením úlohy (hodiny, príp. minúty), pridá komentár, v ktorom presne špecifikuje čo v danom čase vykonával (napr. analyzoval som existujúce knižnice pre spracovanie hlasu) a vyberie typ aktivity, ktorú vykonával.

### 15.8.1.3 Zmena stavu úlohy po jej dokončení

*Predpoklad:* používateľ ukončil prácu na úlohe, ktorá mu bola pridelená vrátane testovania.

1. Vyberie úlohu, ktorú chce zmeniť.
2. Vyberie možnosť Update úlohy.
3. Zmení stav úlohy z In progress na Resolved a počet percent na 100%, pridá komentár, v ktorom špecifikuje v čom spočívalo riešenie úlohy, prípadne nejaké špecifiká použitého riešenia.

Update

Change properties (More)

Status *	Resolved	Start date	2012-10-25
Priority *	Normal	Due date	2012-10-26
Assignee	Zuzana Ujhelyiova	Estimated time	Hours
Category		% Done	100 %
Target version	Sprint 1		
UserStoryPoints			

Log time

Spent time	Hours	Activity	--- Please select ---
Comment			

Notes

B I U S C H1 H2 H3 [List Icons] pre [Image Icons] Text formatting: Help

Files

Choose File no file selected Optional description

Add another file (Maximum size: 24.5 MB)

Obrázok 15-3: Ukážka úlohy

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

4. Zaznamená čas strávený na úlohe.
5. Pridá príslušný komentár a zmení riešiteľa úlohy na manažéra kvality.

### 15.9 Kontrola postupu prác na úlohe v systéme Redmine

Kapitola sa zaoberá kontrolou postupu prác na úlohe z pohľadu manažéra monitorovania.

Predpoklad pre nasledujúce: manažér plánovania, resp. manažér monitorovania má konto v systéme Redmine, má na starosti daný projekt a vie sa do systému prihlásiť.

#### 15.9.1 Monitorovanie stavu úloh v rámci aktuálne riešenej cieľovej verzie

Úlohou manažéra monitorovania je v priebehu aktuálnej verzie kontrolovať stav úloh a zabezpečiť tak ich dokončenie v stanovenom čase (ku koncu verzie).

##### 15.9.1.1 Porovnanie vykázanej práce so skutočným stavom

Predpoklad: existencia a využívanie nástroja na manažment verzii v rámci projektu, vyriešená úloha.

1. Manažér monitorovania si pozrie si stav jednotlivých úloh [3].
2. Porovná množstvo vykazaného času s množstvom vložených súborov (ak boli pridané, resp. zmenené).
3. Pozrie si v úlohe komentár manažéra kvality, či bola funkcionálna pretestovaná, a následne úlohu uzavrie.

##### 15.9.1.2 Kontrola množstva vykázanej práce po skončení aktuálnej cieľovej verzie

Predpoklad: verzia bol skončená a všetky úlohy definované pre danú verziu boli vyriešené.

1. Manažér monitorovania skontroluje, či sú všetky úlohy danej verzie uzavreté (stav closed).
2. Zobrazí si príslušnú cieľovú verziu projektu, ktorá skončila (šprint).

The screenshot shows the Redmine interface for a sprint named "SPRINT 1 - káva". At the top right, there is an "Edit" button. The main content area is divided into several sections:

- Progress:** A green progress bar is shown at 100% completion. Below it, the text reads "35 closed (100%) 0 open (0%)".
- Time tracking:** A box on the right displays "Estimated time" as 220.34 hours and "Spent time" as 223.17 hours.
- Related issues:** A list of issues is shown, including Bug #912 (zmeniť v mysl encoding) and several tasks (Task #853 to #868) related to task distribution, communication, and meetings.
- Issues by Category:** A dropdown menu is set to "Category", and the text "No data to display" is shown below it.

Obrázok 15-4: Ukážka úloh šprintu

## 15| Manažment úloh v jednotlivých etapách projektu (Príloha C)

3. Porovná si predpokladanú dobu vyhradenú na danú verziu so skutočne stráveným časom.
4. Zistí, ktoré úlohy spôsobili odklonenie od odhadov. Analyzuje možnosti vylepšenia v budúcej verzii.



## 16 Manažment testovania (Príloha D)

---

Zodpovedná osoba: František Nagy

### 16.1 Úvod

Účelom tejto metodiky je určenie procesov, ich vstupov a výstupov, zodpovedných osôb a ich povinností počas testovania. Táto metodika je teda určená pre členov tímu vystupujúcich vo fáze testovania.

Ďalej sa metodika zaoberá testovaním pomocou nástroja JUnit vo vývojovom prostredí Eclipse. Určuje postupy, ktorými sa členovia tímu musia počas testovania riadiť.

### 16.2 Použité pojmy

- testovacia jednotka (test unit) - trieda vykonávajúca testy pre jednu triedu
- súbor testov (test suite) - trieda vykonávajúca testy viacerých testovacích jednotiek
- testovaná trieda - trieda, ktorej metódy majú byť otestované
- testovaná metóda – metóda testovanej triedy, ktorá má byť otestovaná
- testovacia metóda – metóda testovacej jednotky, ktorá testuje danú testovanú metódu

### 16.3 Roly a zodpovednosti

Tester	<ul style="list-style-type: none"> <li>• analýza špecifikácie</li> <li>• návrh testov</li> <li>• vytvorenie testov</li> <li>• vykonanie testov</li> <li>• spracovanie výsledkov</li> </ul>
Hlavný tester	<ul style="list-style-type: none"> <li>• analýza špecifikácie</li> <li>• návrh testov</li> <li>• revízia testov</li> <li>• kontrola a postúpenie výsledkov testov</li> </ul>
Manažér testovania	<ul style="list-style-type: none"> <li>• začatie testovania</li> <li>• postúpenie výsledkov testov</li> </ul>
Vedúci tímu	<ul style="list-style-type: none"> <li>• naplánovanie testovania</li> <li>• začatie testovania</li> </ul>

Tabuľka 16-1: Roly a zodpovednosti

### 16.4 Použité procesy

#### 16.4.1 Začatie procesu testovania

Pred začatím samotnej testovacej fázy sa zanalyzuje stav projektu podľa daného plánu a ak je projekt pripravený, rozhodne sa o začiatku testovania.

	<b>Krok</b>	<b>Kapitola</b>
1.	Analýza projektového plánu	16.5.1
2.	Analýza stavu projektu	16.5.2
3.	Rozhodnutie o začiatku testovania	16.5.3

Tabuľka 16-2: Proces 1

## 16 | Manažment testovania (Príloha D)

### 16.4.2 Návrh stratégie testovania

Pred návrhom konkrétnych testov sa najprv navrhne všeobecná stratégia testovania. Tá sa vytvorí na základe analýzy špecifikácie požiadaviek a projektovej dokumentácie.

	<b>Krok</b>	<b>Kapitola</b>
1.	Analýza špecifikácie požiadaviek	16.5.4
2.	Analýza projektovej dokumentácie	16.5.5
3.	Návrh stratégie testovania	16.5.6
4.	Schválenie stratégie testovania	16.5.7

Tabuľka 16-3: Proces 2

### 16.4.3 Návrh jednotlivých testov

Pri návrhu konkrétnych testov sa vychádza z dokumentácie testovanej časti s možným prihliadnutím na zdrojový kód testovanej časti.

	<b>Krok</b>	<b>Kapitola</b>
1.	Analýza dokumentácie testovanej časti	16.5.8
2.	Vytvorenie testovacích jednotiek a súborov	16.5.9
3.	Revízia a schválenie testov	16.5.10

Tabuľka 16-4: Proces 3

### 16.4.4 Vykonanie testov

Vytvorené testy sú vložené do testovacieho nástroja, ktorý je najprv potrebné správne nakonfigurovať. Výsledky testov sú uložené.

	<b>Krok</b>	<b>Kapitola</b>
1.	Nastavenie testovacieho nástroja	16.5.11
2.	Samotné vykonanie testov	16.5.12
3.	Uloženie výsledkov testov	16.5.13

Tabuľka 16-5: Proces 4

### 16.4.5 Spracovanie úspešných výsledkov testov

Testy sú vyhodnotené. V prípade, že všetky testy prebehli úspešne, je to oznámené zvyšku tímu a fáza testovania je ukončená.

	Krok	Kapitola
1.	Vyhodnotenie výsledkov testov	16.5.14
2.	Oznámenie o úspešnom výsledku testov	16.5.15
3.	Ukončenie testovania	16.5.16

Tabuľka 16-6: Proces 5

### 16.4.6 Spracovanie neúspešných výsledkov testov

Testy sú vyhodnotené a v prípade, že je niektorý z testov vyhodnotený ako neúspešný je identifikovaný pracovník zodpovedný za danú časť projektu, je informovaný o výsledku testu a daná časť projektu sa vracia do fázy vývoja.

	Krok	Kapitola
1.	Vyhodnotenie výsledkov testov	16.5.17
2.	Oznámenie o neúspešnom výsledku zodpovednej osobe	16.5.18
3.	Návrat do fázy vývoja	16.5.19

Tabuľka 16-7: Proces 6

## 16.5 Podrobný popis krokov

### 16.5.1 Analýza projektového plánu

Vstup: Plán projektu

Výstup: Otvorená dokumentácia testovania

Zodpovedná osoba: Vedúci tímu, manažér plánovania

Vedúci tímu spolu s manažérom plánovania zanalyzujú plán projektu vytvorený podľa metodiky [1]. Keď podľa plánu nastane čas fázy testovania, založia dokumentáciu testovania.

## 16 | Manažment testovania (Príloha D)

### 16.5.2 Analýza stavu projektu

Vstup: Otvorená dokumentácia testovania, projektová dokumentácia

Výstup: Aktualizovaná dokumentácia testovania

Zodpovedná osoba: Manažér plánovania, manažér monitorovania

Manažér plánovania a manažér monitorovania zanalyzujú stav projektu podľa projektovej dokumentácie vytvorenej podľa metodiky [2].

### 16.5.3 Rozhodnutie o začiatku testovania

Vstup: Aktualizovaná dokumentácia testovania

Výstup: Schválená a uzavrená dokumentácia testovania

Zodpovedná osoba: Vedúci tímu

V prípade, že je projekt pripravený na fázu testovania, uzavrie a schváli sa dokumentácia testovania.

### 16.5.4 Analýza špecifikácie požiadaviek

Vstup: Špecifikácia požiadaviek

Výstup: Prvý návrh stratégie testovania

Zodpovedná osoba: Hlavný tester, testeri

Urobí sa analýza špecifikácie testovania a určí sa ktoré časti produktu potrebujú byť akým spôsobom otestované.

### 16.5.5 Analýza projektovej dokumentácie

Vstup: Projektová dokumentácia, prvý návrh stratégie testovania

Výstup: aktualizovaný návrh stratégie testovania

Zodpovedná osoba: Hlavný tester, testeri

Po určení testovaných častí podľa špecifikácie požiadaviek sa bližšie určí spôsob testovania. Vhodný spôsob sa vyberie pomocou projektovej dokumentácie a analýzy, ako je testovaná časť systému programovaná.

### 16.5.6 Návrh stratégie testovania

Vstup: Aktualizovaný návrh stratégie testovania

Výstup: Finálny návrh stratégie testovania

Zodpovedná osoba: Hlavný tester, testeri

Na základe predchádzajúcich analýz sa dokončí návrh stratégie testovania.

### 16.5.7 Schválenie stratégie testovania

Vstup: Finálny návrh stratégie testovania

Výstup: Schválená stratégia testovania

## 16 | Manažment testovania (Príloha D)

Zodpovedná osoba: Hlavný tester, manažér testovania

Hlavný tester a manažér testovania skontrolujú finálny návrh stratégie testovania a ak nenájdu nedostatky, schvália ho.

### 16.5.8 Analýza dokumentácie testovanej časti

Vstup: Projektová dokumentácia, zdrojový kód

Výstup: Prvý návrh testov

Zodpovedná osoba: Tester

Tester na základe dokumentácie testovanej časti a prípadnej analýze zdrojového kódu navrhne testy.

### 16.5.9 Vytvorenie testovacích jednotiek a súborov

Vstup: Prvý návrh testov

Výstup: Vytvorené testovacie jednotky a súbory testov

Zodpovedná osoba: Tester

Tester vytvorí testovacie jednotky a súbory testov podľa predchádzajúceho návrhu.

### 16.5.10 Revízia a schválenie testov

Vstup: Vytvorené testovacie jednotky a súbory testov

Výstup: Schválené testovacie jednotky a súbory testov

Zodpovedná osoba: Hlavný tester, testeri

Vytvorené testy sú kontrolované ostatnými testerami (peer review) a hlavným testerom.

### 16.5.11 Nastavenie testovacieho nástroja

Vstup: Testovací nástroj

Výstup: Nastavený testovací nástroj

Zodpovedná osoba: Tester

Pred samotným vykonaním testov tester nastaví testovací nástroj podľa potrieb daného projektu a testovanej časti.

### 16.5.12 Samotné vykonanie testov

Vstup: Testovacie jednotky a súbory testov, zdrojový kód

Výstup: Výsledky testov

Zodpovedná osoba: Tester

Tester spustí súbory testov.

### 16.5.13 Uloženie výsledkov testov

Vstup: Výsledky testov

## 16 | Manažment testovania (Príloha D)

Výstup: Uložené výsledky testov

Zodpovedná osoba: Tester

Po ukončení všetkých testov sa výsledky uložia podľa metodiky [3].

### 16.5.14 Vyhodnotenie výsledkov testov

Vstup: Uložené výsledky testov

Výstup: Vyhodnotené výsledky testov

Zodpovedná osoba: Hlavný tester

Hlavný tester vyhodnotí uložené výsledky testov.

### 16.5.15 Oznámenie o úspešnom výsledku testov

Vstup: Testy vyhodnotené ako úspešné

Výstup: Správa o výsledkoch testov

Zodpovedná osoba: Hlavný tester

Po vyhodnotení výsledkov testov o tom vypracuje hlavný tester správu.

### 16.5.16 Ukončenie testovania

Vstup: Správa o výsledkoch testov

Výstup: Ukončenie testovania

Zodpovedná osoba: Manažér testovania

Na základe správy o výsledkoch testov ukončí manažér testovania testovaciu fázu.

### 16.5.17 Vyhodnotenie výsledkov testov

Vid' 14.5.14.

### 16.5.18 Oznámenie o neúspešnom výsledku zodpovednej osobe

Vstup: Testy vyhodnotené ako neúspešné

Výstup: Správa o výsledkoch testov

Zodpovedná osoba: Hlavný tester, manažér komunikácie

Po zistení neúspešného výsledku testu odovzdá hlavný tester správu o výsledkoch testov manažérovi komunikácie, ktorý odovzdá správu vedúcemu tímu vývojárov a zodpovednému vývojárovi.

### 16.5.19 Návrat do fázy vývoja

Vstup: Správa o výsledkoch testov

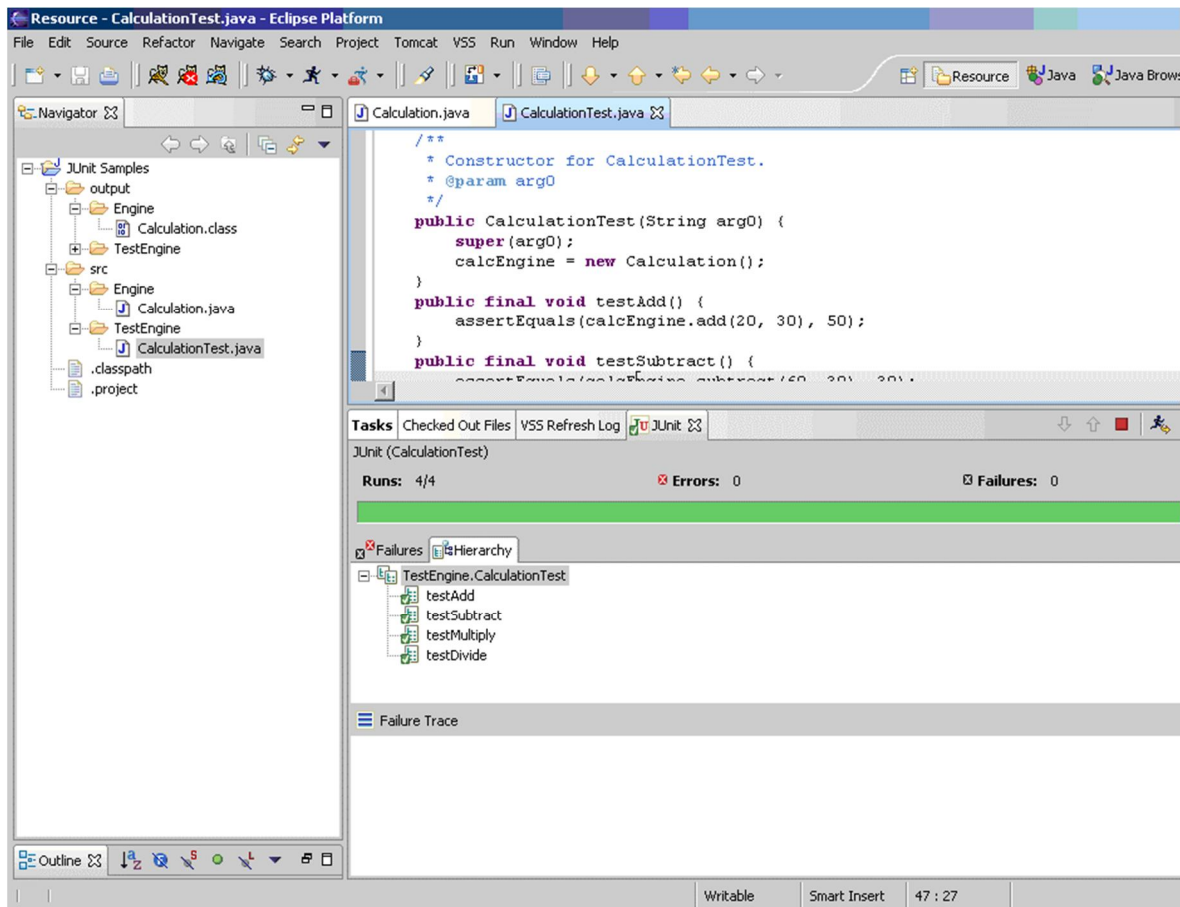
Výstup: Návrat do fázy vývoja

Zodpovedná osoba: Manažér testovania, vedúci tímu

## **16 | Manažment testovania (Príloha D)**

Po vyhodnotení neúspešných testov sa projekt v daných častiach vráti do fázy vývoja. Následne už stačí pokračovať od bodu 16.5.4.





Obrázok 16-1

## 16.6 Použitie testovacieho frameworku JUnit

### 16.6.1 Čo testovať

- Testované musia byť všetky triedy, ktoré sú verejne prístupné.
- V každej takejto triede musia byť otestované všetky jej verejné metódy.

### 16.6.2 Ako testovať

- Vytvárame jednu testovaciu triedu pre každú testovanú triedu
- V testovacej triede testujeme všetky verejné metódy testovanej triedy.
- Pre testovanie každej testovanej metódy vytvárame samostatnú testovaciu metódu
- Všetky testy zahŕňame do súboru testov, aby sa dali spustiť všetky naraz

## 16 | Manažment testovania (Príloha D)

- Jednotlivé testy musia byť nezávislé, teda výsledok testu nesmie závisieť na tom, či sa iný test vykoná, alebo na poradí vykonania testov
- Testujeme metódou bielej skrinky, teda prihliadame na štruktúru testovaného zdrojového kódu
- Testujeme aj okrajové vstupy
- Ak je to možné, testujeme náhodne generované vstupy
- Pri testovaní sa pokúšame dosiahnuť čo najväčšie pokrytie kódu (code coverage), tj. pokúšame sa, aby boli vykonané všetky časti programu. Toto je možné sledovať napríklad pomocou pluginu do prostredia Eclipse EclEmma (viď Obrázok 16-2).
- Testy, ktoré boli vyhodnotené ako neúspešné sa vždy snažíme, aby sa dali zopakovať

Element	Coverage	Covered Lines	T
java - commons-collections	79,5 %	10927	
org.apache.commons.collections	74,1 %	3842	
ArrayStack.java	86,5 %	32	
BagUtils.java	86,7 %	13	
BeanMap.java	72,4 %	155	
BinaryHeap.java	87,6 %	127	
BoundedFifoBuffer.java	93,2 %	82	
BufferOverflowException.java	55,6 %	5	
BufferUnderflowException.java	88,9 %	8	
BufferUtils.java	30,8 %	4	
ClosureUtils.java	93,9 %	31	
CollectionUtils.java	92,4 %	293	
ComparatorUtils.java	8,6 %	3	
CursorableLinkedList.java	85,4 %	444	

Obrázok 16-2: Testovanie v Eclipse

### 16.6.3 Pomenovania

- Všetky súbory používané pri testovaní (testovacie jednotky, súbory testov, pomocné objekty) musia byť v priečinku „tests“
- Testovaciu triedu pomenujeme podľa testovanej triedy takto: „<MenoTestovanejTriedy>Test“
- Testovaciu metódu pomenujeme podľa testovanej metódy: „<MenoTestovanejMetódy>Test“
- Hlavný súbor testov pomenujeme „AllTests“
- Ak máme iné súbory testov pomenujeme výstižne s príponou „Suite“
- V prípade, že vytvárame testy, ktoré netestujú konkrétnu metódu, tak ich pomenujeme podľa funkcionality, ktorú majú testovať s príponou „Test“

### 16.6.4 Pomocné objekty (mock objects)

- V prípade potreby môžeme používať pomocné objekty
- Pomocné objekty používame na simulovanie objektov, ktoré sú ťažko testovateľné
- Objekt nahradíme pomocným najmä ak:
  - má nedeterministické správanie (napr. aktuálny čas, aktuálna teplota)
  - jeho stavy je náročné napodobniť (napr. sieťová chyba)
  - jeho používanie je časovo náročné (napr. databáza, ktorá by musela byť inicializovaná)
  - ešte nie je implementovaný, alebo sa môže ešte zmeniť
  - by vytváral funkcionality len pre testovacie účely
- Pomocné objekty pomenovávame: <MenoNahrádzanéhoObjektu>Mock

## 16.7 Záver

V metodike vyššej úrovne boli rozpracované procesy, osoby, ktoré sú za ne zodpovedné, ich vstupy a výstupy, ktoré sú vykonávané vo fáze testovania vo fiktívnej softvérovej firme.

V metodike nižšej úrovne boli popísané pravidlá, ktorými sa má tím riadiť pri testovaní pomocou JUnit v prostredí Eclipse.

## 17 Šprint review ( Príloha E)

---

Zodpovedná osoba: Ivana Bohunická, Zuzana Ujhelyiová

V nasledujúcej časti je uvedený príklad vyhodnotenia šprintu, ktorý vznikol v rámci šprint review. Úvodná tabuľka (Tabuľka 17-1) predstavuje sumár času stráveného na projekte všetkými členmi tímu. Zvyšné tabuľky uvádzajú vyhodnotenie stráveného času na projekte individuálne pre každého člena tímu.

Verzia	2012-10	2012-11	2012-12	Spolu
none	24.50	24.00	2.00	50.50
Document	13.50	66.25	12.00	91.75
Sprint 1	109.62	27.25		136.87
Sprint 2		129.00		129.00
Sprint 3		184.50	2.00	186.50
Sprint 4		20.50	14.50	35.00
<b>Total</b>	<b>147.62</b>	<b>451.50</b>	<b>30.50</b>	<b>629.62</b>

Tabuľka 17-1: Šprint review jednotlivých šprintov

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Ivana Bohunicka			19.50	64.00	83.50
	none		3.00		3.00
		Task #3922: analyzovať existujúce riešenia	3.00		3.00
	Sprint 1		12.00	1.00	13.00
		Task #4006: Navrh protokolu	2.00		2.00
		Task #4245: Návrh C++ knižnice	3.00		3.00
		Task #4252: Návrh designu stránky	2.00		2.00
		Task #4258: Navrhnuť používateľské rozhranie- prototyp	5.00		5.00
		Task #4549: Vytvorenie dokumentačného a textového obsahu stránky		1.00	1.00
	Sprint 2			29.50	29.50
		Task #4441: Finálny návrh používateľského rozhrania		5.00	5.00
		Task #4443: Implementácia a vytvorenie obrazoviek používateľského rozhrania		10.00	10.00
		Task #4444: Zdokumentovať návrh používateľského rozhrania		5.00	5.00
		Task #4545: Implementácia prázdnej triedy a metód Mapper-a		2.00	2.00
		Task #4547: Revízia a kontrola dokumentácií pred odovzdaním		7.50	7.50
	Document		4.50	5.50	10.00
		Task #4076: Manažment plánovania	4.50	5.50	10.00
	Sprint 3			28.00	28.00
		Task #4634: Implementácia GUI dizajnu		10.00	10.00
		Task #4635: Implementácia GUI handling		16.00	16.00
		Feature #4703: Implementácia funkcionality vybraných metód v Mapper class		2.00	2.00

Tabuľka 17-2: Šprint review Ivana Bohunická

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Jan Greppel			17.00	62.50	79.50
	none		9.00		9.00
		Task #3920: Vytvorit' web stránku tímu, založit' tímovú wiki	9.00		9.00
	Sprint 1		8.00		8.00
		Task #4255: Pripraviť analýzu riešenia	3.00		3.00
		Task #4260: Skompletizovať Product backlog	5.00		5.00
	Sprint 2			18.50	18.50
		Task #4439: Návrh mapovania (Bussiness entity)		3.00	3.00
		Task #4445: Zdokumentovať analýzu existujúceho riešenia- knižnica Fiit Kinect		3.00	3.00
		Task #4446: Zdokumentovať komplexný návrh a architektúru nášho riešenia		1.50	1.50
		Task #4543: Implementácia connect, load, save databázy		7.50	7.50
		Task #4616: Revízia a kontrola dokumentácií pred odovzdaním		3.50	3.50
	Document			5.00	5.00
		Task #4080: Manažment komunikácie		5.00	5.00
	Sprint 3			39.00	39.00
		Task #4622: Implementácia Android Sender- komunikácia po sieti s Core		8.00	8.00
		Task #4623: Android Sender- grafické rozhranie		4.50	4.50
		Task #4689: Dokoncenie implementácie load, save metod		25.50	25.50
		Task #4826: Zdokumentovať Android Sender		1.00	1.00

Tabuľka 17-3: Šprint review Ján Greppel

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Juraj Muransky			13.00	46.75	59.75
	Sprint 1		9.00	6.75	15.75
		Task #4006: Navrh protokolu	2.00		2.00
		Task #4254: Prerobiť stránku podľa navrhnutého designu	3.00		3.00
		Task #4259: Implementovať používateľské rozhranie- prototyp	4.00		4.00
		Task #4436: Návrh alternatívneho protokolu		6.75	6.75
	Sprint 2			12.00	12.00
		Task #4442: Implementácia a vytvorenie obrazoviek používateľského rozhrania		3.00	3.00
		Task #4541: Pripraviť databázu, databázovú schému		5.00	5.00
		Task #4542: Zdokumentovať databázu a jej fyzický model		4.00	4.00
	Document		4.00	10.00	14.00
		Task #4079: Manažment podpory vývoja	4.00	10.00	14.00
	Sprint 3			18.00	18.00
		Task #4618: Implementácia IR prijímača		11.00	11.00
		Task #4735: Zdokumentovať IR		2.00	2.00
		Task #4736: Implementácia IR vysielača		5.00	5.00

Tabuľka 17-4: Šprint review Juraj Muránsky

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Frantisek Nagy			18.50	54.00	72.50
	none		0.50		0.50
		Task #4019: Vytvorit' štruktúru Gitu	0.50		0.50
	Sprint 1		18.00	9.00	27.00
		Task #4006: Navrh protokolu	3.00		3.00
		Task #4010: Implementovat' protokol v Java	13.00	6.00	19.00
		Task #4011: Vytvorit' testovací program pre Java knižnicu		3.00	3.00
		Task #4244: Návrh Java knižnice	2.00		2.00
	Sprint 2			14.00	14.00
		Task #4536: Zdokumentovat' návrh centrálnej aplikácie		6.00	6.00
		Task #4538: Implementácia a vytvorenie obrazoviek používateľského rozhrania		8.00	8.00
	Document			2.00	2.00
		Task #4081: Manažment kvality		2.00	2.00
	Sprint 3			29.00	29.00
		Task #4627: Implementácia metód v Mapper class		19.00	19.00
		Task #4628: Unit testy- Mapper class		4.00	4.00
		Task #4733: Unit testy sieťovej komunikácie		6.00	6.00

Tabuľka 17-5: Šprint review František Nagy



17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Dominik Rerko			4.00	75.00	79.00
	Sprint 1		2.00	2.00	4.00
		Task #4243: Návrh Java knižnice	2.00	2.00	4.00
	Sprint 2			12.00	12.00
		Task #4440: Návrh mapovania (Bussiness entity)		4.00	4.00
		Task #4539: Implementácia a vytvorenie obrazoviek používateľského rozhrania		8.00	8.00
	Document		2.00	36.00	38.00
		Task #4014: Tvorba dokumentácie		27.00	27.00
		Task #4016: Projektová dokumentácia	1.00	1.00	2.00
		Task #4017: Dokumentácia riadenia	1.00		1.00
		Task #4082: Manažment dokumentovania		8.00	8.00
	Sprint 3			25.00	25.00
		Task #4624: Implementácia Android Receiver- príjem udalosti z Core app		10.00	10.00
		Task #4625: Implementácia Android Receiver- spúšťanie akcií		13.00	13.00
		Task #4743: Zdokumentovať Android Receiver		2.00	2.00

Tabuľka 17-6: Šprint review Dominik Rerko

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Matus Ujhelyi			57.12	88.00	145.12
	none		12.00	22.00	34.00
		Task #3920: Vytvoriť web stránku tímu, založiť tímovú wiki	2.00		2.00
		Task #3921: pripraviť Git a napísať návod	3.00		3.00
		Task #3922: analyzovať existujúce riešenia	7.00		7.00
		Support #4585: Správa Git-u		22.00	22.00
	Sprint 1		44.12	8.00	52.12
		Task #4006: Navrh protokolu	6.00		6.00
		Task #4007: Zdokumentovať komunikačný protokol	0.12		0.12
		Task #4009: Implementovať protokol v C++		8.00	8.00
		Task #4012: Vytvoriť testovací program pre C++ knižnicu	17.00		17.00
		Task #4020: Pripraviť existujúcu Fiit Kinect knižnicu	17.00		17.00
		Task #4246: Návrh C++ knižnice	4.00		4.00
	Sprint 2			27.00	27.00
		Task #4442: Implementácia a vytvorenie obrazoviek používateľského rozhrania		14.00	14.00
		Task #4489: Otestovať výsek minuloročného riešenia s kiniektom		5.00	5.00
		Task #4537: Vytvoriť diagram tried centrálnej aplikácie		4.00	4.00
		Task #4540: Vylepšenie a implementačné prispôsobenie FiitKinect knižnice		4.00	4.00
	Document		1.00	3.00	4.00
		Task #4077: Manažment rizík	1.00	3.00	4.00
	Sprint 3			28.00	28.00
		Task #4632: Doimplementovať obrázky do FiitKinect aplikácie		2.00	2.00
		Task #4636: Implementácia GUI handling		25.00	25.00
		Task #4734: Zdokumentovať doplnenie a vylepšenie FiitKinect knižnice		1.00	1.00

Tabuľka 17-7: Šprint review Matúš Ujhelyi

17 | Šprint review ( Príloha E)

Člen tímu	Verzia	Úloha	2012-10	2012-11	Spolu
Zuzana Ujhelyiova			18.50	45.75	64.25
	none			2.00	2.00
		Support #4621: Správa Maven-u		2.00	2.00
	Sprint 1		16.50	0.50	17.00
		Task #4006: Navrh protokolu	5.00		5.00
		Task #4007: Zdokumentovať komunikačný protokol	2.00		2.00
		Task #4253: Návrh designu stránky	1.00		1.00
		Task #4256: Zdokumentovať mapovanie udalostí a akcií	1.50	0.50	2.00
		Feature #4274: Konfigurácia java aplikácie + serializácia JSONObjektov	7.00		7.00
	Sprint 2			16.00	16.00
		Task #4438: Návrh mapovania (Bussiness entity)		2.00	2.00
		Task #4544: Implementácia- vytvorenie prázdnych mapovacích objektov		5.00	5.00
		Task #4546: Revízia a kontrola dokumentácií pred odovzdaním		9.00	9.00
	Document		2.00	4.75	6.75
		Task #4078: Manažment monitorovania projektu	2.00	3.00	5.00
		Task #4548: Vytvoriť prihlášku do TP Cup		1.75	1.75
	Sprint 3			17.50	17.50
		Task #4629: Implementácia metód mapovacích tried		2.00	2.00
		Task #4630: Unit testy-mapovacie triedy		2.00	2.00
		Task #4631: Doimplementovať obrázky do serializácie		1.50	1.50
		Task #4637: Implementácia GUI handling		12.00	12.00
	Sprint 4			5.00	5.00
		Task #4822: Revízia a oprava dokumenatacii		5.00	5.00

Tabuľka 17-8: Šprint review Zuzana Ujhelyiová

## 18 Preberací protokol (Príloha G)

---

Ja, dolu podpísaný Ing. Michal Kottman svojim podpisom potvrdzujem, že som prebral od tímu č. 11 dokumentáciu na konci zimného semestra k tímovému projektu pozostávajúcu z dvoch častí:

- dokumentácia k inžinierskemu dielu
- dokumentácia k riadeniu

V Bratislave dňa .....

odovzdal:

prebral:

.....  
Matúš Ujhelyi

.....  
Ing. Michal Kottman