

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Dokumentácia k riadeniu

Tím BAREKO

Bc. Michal Bañas

Bc. Šimon Harvan

Bc. Ernest Loureiro

Bc. Daniel Lukáč

Bc. Marko Moravčík

Bc. Lukáš Rešutík

Bc. Dávid Roba

Vedúci projektu: Ing. Ivan Kapustík

Predmet: Tímový projekt I

Ročník: 2017/2018

Obsah

Obsah	2
1. Úvod	5
2. Manažérske roly	6
2.1 Identifikované úlohy.....	6
2.2 Opis úloh	6
2.2.1 Hlavný projektový manažér.....	6
2.2.2 Manažér plánovania a úloh	7
2.2.3 Manažér verziovania	7
2.2.4 Manažér testovania.....	7
2.2.5 Manažér nasadzovania	8
2.2.6 Manažér kvality	8
2.2.7 Manažér dokumentovania	9
2.2.8 Manažér rizík	9
2.2.9 Manažér komunikácie a správy dát.....	9
2.3 Rozdelenie manažérskych úloh	10
3. Aplikácie manažmentov	11
3.1 Metodika komunikácie	11
3.2 Metodika vykazovania prác v Scrumdesk.....	11
3.3 Metodika code review	11
3.4 Metodika dokumentácie	11
3.5 Metodika code conventions	11
3.6 Metodika testovania	11
3.7 Metodika rizík.....	11
3.8 Metodika commitovania a verziovania	12
4. Šprinty	13

5.	Argentínsky šprint	14
5.1	Základné informácie	14
5.2	Retrospektíva	14
5.3	Úlohy	15
5.3.1	Opis jednotlivých úloh	15
5.4	Zhodnotenie	17
5.4.1	Plnenie úloh.....	17
5.4.2	Podiel práce členov tímu	18
6.	Brazílsky šprint	19
6.1	Základné informácie	19
6.2	Retrospektíva	19
6.3	Úlohy	20
6.3.1	Opis jednotlivých úloh	20
6.4	Zhodnotenie	21
6.4.1	Plnenie úloh.....	21
6.4.2	Podiel práce členov tímu	22
7.	Kolumbijský šprint.....	23
8.	Retrospektíva po prvých troch šprintoch	24
9.	Zoznam príloh.....	25
	Príloha A: Metodika komunikácie:	26
	Príloha B: Metodika vykazovania prác v Scrumdesk	28
	Príloha C: Metodika code review	30
	Príloha D: Metodika dokumentácie	31
1.	Úvodná strana	31
1.1.	Obsah úvodnej strany.....	31
1.2.	Formátovanie úvodnej strany	31
2.	Obsah.....	32
3.	Text.....	32
3.1.	Text.....	32

3.2.	Nadpisy.....	33
3.3.	Pomenovanie a identifikovanie objektov	34
	Príloha E: Metodika code conventions.....	35
	Príloha F: Metodika testovania	37
1.	Vytváranie plánov v prostredí Bamboo.....	37
1.1	Vytváranie nového plánu	37
1.2	Vytváranie skript taskov	38
1.3	Vyhodnotenie a zobrazenie testov.....	40
1.4	Spustenie plánu.....	41
	Príloha G: Metodika rizík	42
	Identifikácia rizika	42
	Manažovanie a minimalizácia rizika	42
	Identifikácia rizík	43
	Slabý manažment tímu.....	43
	Neporozumenie zadanej práce	43
	Neschopnosť člena splniť požiadavku	43
	Zmena rozsahu a cieľa práce	44
	Nedostatok znalostí.....	44
	Nevhodné obsadenie pozícií v tíme	44
	Odchod člena z tímu.....	44
	Nevhodné rozloženie práce.....	45
	Absencia člena na zasadnutiach	45
	Stanovenie priority rizika	46
	Príloha H: Metodika commitovania a verziovania.....	47

1. Úvod

RoboCup je pravdepodobne najväčšou súťažou inteligentných autonómnych robotov na svete. Táto súťaž je určená na podporu výskumu a výučby robotiky a umelej inteligencie. Každý tím sa v rámci obmedzení, určených pravidlami hry futbal a špecifikami simulačného prostredia, snaží vytvoriť čo najlepšieho hráča. Mužstvo, vytvorené z takýchto hráčov, by malo vyhrať nad mužstvom súpera. Dnes prebieha simulácia futbalu v 3D simulačnom prostredí. Na našej fakulte FIIT sa simulácií robotického futbalu venujú tímy študentov už od roku 2000.

Na predmete Tímový projekt sa tak strieda každý rok niekoľko nových ľudí, ktorí sa snažia posunúť latku nášho robota Jima niekde vyššie. Okrem členov tímu sa tejto problematike každý rok venujú aj bakalári a diplomanti vo svojich záverečných prácach. Získané vedomosti a pokrok ktorí dosiali sa vždy zahŕňa do tímového projektu. Takýmto spôsobom sa každý rok posúvame bližšie k tomu, aby sme vytvorili spoľahlivý, prepracovaný 3D tím hráčov schopných poraziť súpera.

Takýto každoročný pokrok by ale nebol možný bez odovzdávania si skúseností medzi sebou. Skúsenosti, ktoré jednotliví členovia tímu počas roka získajú nie sú len technického zamerania, ale aj manažérskeho. Nakoľko sa pracuje v tíme je potrebné nejakým spôsobom riadiť jednotlivé úlohy. Riadenie úloh, ktoré má tím vykonať, môžeme rozdeliť medzi členov tímu tak, aby každý člen riadil úlohy určitého zamerania. Rozdelením riadenia projektu dosiahneme efektívnejšie využívaný čas, kvalitnejší kód, prehľadnejšiu dokumentáciu a predídeme rizikám.

Nájsť správne rozdelenie oblastí úloh, ktoré je treba riadiť a správny spôsob ich riadenia je netriviálne. Hľadanie správnych spôsobov rozdelenia a riadenia trvá dlho pretože sa počas riadenia tieto spôsoby skúšajú a upravujú, aby boli čo najlepšie. Práve odovzdávaním skúseností ďalším tímom môžeme čas potrebný na hľadanie týchto spôsobov skrátiť a tým zefektívniť využívanie času nasledujúceho tímu už od jeho začiatku. Táto dokumentácia obsahuje popis postupu práce na projekte, výsledky jednotlivých šprintov, ich priebeh a zhodnotenie. Vyhodnocujú sa v nej problémy s úlohami a ich splnenie.

Dokumentácia obsahuje popis úloh jednotlivých členov tímu, metodiky vytvorené k práci na projekte a zhodnotenie ich aplikovania. Dokumentácia sa nevenuje technickej časti projektu ale spomínanej manažérskej.

2. Manažérske roly

Bc. Michal Bañas, Bc. Šimon Harvan, Bc. Ernest Loureiro,
Bc. Daniel Lukáč, Bc. Marko Moravčík, Bc. Lukáš Rešutík, Bc. Dávid Roba

V tejto kapitole sa budeme venovať manažérskym úlohám, ktoré boli pridelené členom tímu a spôsoby manažmentu všetkých potrebných častí tohto projektu vrámci akademického roku 2017/2018.

2.1 Identifikované úlohy

Pre správne fungovanie tímu sa zvolilo viacero úloh, ktoré sa rozvolili medzi členov tímu.

Medzi tieto úlohy patrí:

- Hlavný projektový manažér
- Manažérka plánovania a úloh
- Manažér verziovania
- Manažér testovania
- Manažér nasadzovania
- Manažér kvality
- Manažér dokumentovania
- Manažér rizík
- Manažér komunikácie a správy dát

2.2 Opis úloh

V tejto časti sa budeme venovať jednoduchému opisu úloh, za ktoré sú zodpovední daní členovia tímu.

2.2.1 Hlavný projektový manažér

Bc. Michal Bañas

Hlavný projektový manažér by sa mal v prvom rade starať o správne a múdre vedenie tímu. Pri správne zachovanej disciplíne sa tak dokáže tím lepšie organizovať a vykazovať lepšie výsledky. Projektový manažér sa tak stará o čím vyššiu kvalitu pri manažmente zdrojov a procesov v tíme, aby bola dosiahnutá čo najvyššia úspešnosť tímu v každom šprinte.

Hlavnými úlohami projektového manažéra sú teda :

- riadi celkovú funkčnosť a výzor projektu
- stará sa o formálne aj neformálne stretnutia
- kontroluje ostatné manažérske úlohy a ich fungovanie
- kontrolovanie a správa Scrumdesku

Náš projektový manažér sa doposiaľ staral najmä o dobrú komunikáciu v tíme, veľký počet konzultácií, vrátane aj mimo školy a staral sa, aby bola práca rovnomerne rozložená a vždy načas vykonaná.

2.2.2 Manažér plánovania a úloh

Bc. Michal Bañas

Manažér plánovania a úloh sa vo veľkej miere stará o správny chod projektu, pretože spravuje najdôležitejšiu časť tímového projektu, a to Scrum. Na začiatku šprintu si stále stanoví jednoduché a dosiahnuteľné ciele, ktoré sa nám musia podariť za jeden šprint (2 týždne). Dohliada na dôsledné plnenie úloh a plánov, ktoré sa vždy stanovujú na začiatku šprintu. Stará sa o nahodenie všetkých úloh do Scrumdesku, o priradenie úloh členom tímu a iné veci, ako napr.:

- definuje presný postup pri práci so Scrumdeskom podľa *metodiky vykazovania prác v Scrumdesku*
- kontroluje správnosť pri postupoch ostatných členov tímu pri práci so Scrumdeskom
- vytvára šprinty, úlohy, uzatvára ich
- vytvára exporty zo Scrumdesku pre ďalšie spracovanie

2.2.3 Manažér verziovania

Bc. Michal Bañas

Manažér verziovania je zodpovedný za jednu z najdôležitejších úloh pri vývoji softvéru, a to udržiavanie prehľadného repozitáru a jeho histórie commitov a verzíí. Kontroluje vytváranie nových vetiev a ich správne používanie členmi tímu. Taktiež:

- vytvára *metodiku commitovania a verziovania*
- stará sa o chod BitBucketu – GIT repozitára
- kontroluje vytváranie vetiev v repozitári
- kontroluje mergovanie vetiev do hlavnej

2.2.4 Manažér testovania

Manažér testovania vytvára testy na overovanie funkčnosti softvéru počas celej dĺžky jeho vývoja. Stará sa o automatizovaný nástroj Bamboo, v ktorom sa dajú vytvárať testy spustiteľné automaticky pri každom commite do Bitbucketu. Tento spôsob často uľahčuje veľkú časť práce, no musí sa správne pripraviť a ak nepotrebujeme automatizované testy, vytvorí si klasické JUnit testy v rámci testovacieho frameworku.

Medzi jeho úlohy patrí:

- plánuje a vytvára testy na vyhodnocovanie funkčnosti softvéru
- spracuje *metodikú testovania*
- spravuje Unit testy na kontrolu projektu
- stará sa o automatizovaný nástroj na testovanie (Bamboo)

2.2.5 Manažér nasadzovania

Úlohou manažéra nasadzovania je rozbehať virtuálny stroj a všetky potrebné komponenty k nemu. Tieto kroky prebehli začiatkom semestra, kedy bol rozbehaný náš unixový server, webová stránka a v neposlednom rade aj tímová wiki.

Jeho hlavnými úlohami sú:

- nasadzuje a spravuje webovú stránku tímu
- nasadzuje a spravuje tímovú wiki stránku
- stará sa o produkčný unixový server
- spravuje virtuálny stroj, stará sa o jeho chod a všetky potrebné súčasti

2.2.6 Manažér kvality

Manažér kvality má za úlohu starať sa o čím kvalitnejší výstup softvérového produktu. Pravidelne kontroluje softvér a zdrojové kódy, definuje procesy na kontrolu a lepšie overenie správnosti a podobne.

- vytvára *metodikú code review* a využíva *metodikú code conventions*
- kontroluje dodržiavanie pravidiel code review
- overuje správnosť a funkčnosť softvéru
- podľa metodík kontroluje všetky potrebné postupy pri tvorbe softvéru
- pravidelne overuje správnosť kódu na repozitári

Počas počiatočných šprintov manažér kvality viedol ostatných členov tímu k správne mu a kvalitnému pracovaniu na svojich úlohách a poctivému prístupu k práci.

2.2.7 Manažér dokumentovania

Bc. Michal Bañas

Hlavnou úlohou manažéra dokumentovania je vytvoriť jednotný spôsob tvorby dokumentov a starať sa o udržiavanie tohto štandardu naprieč celým semestrom. Pre jednoduché objasnenie, na čom tím celý čas pracoval je potrebná prehľadná a konzistentná dokumentácia. Preto je potrebné všetky dokumenty od ľudí z tímu prejsť a opraviť prípadné nezhody a nezrovnalosti.

Manažér dokumentovania:

- vytvára *metodiku dokumentácie*
- stará sa o stav dokumentácie a jej prehľadnosť a konzistentnosť
- posudzuje správnosť a opravuje dokumentácie pred finálnym odovzdaním
- posúva dokumenty na zavesenie na web

Kontrolujú sa veci ako štruktúra dokumentu, titulné strany, veľkosti písma, nadpisov a pod. Ak sú nedostatky vo väčšom meradle, daný manažér upozorní kolegov na pochybenie a požiadava, aby si v budúcnosti dávali pri tvorbe dokumentov väčší pozor.

2.2.8 Manažér rizík

Bc. Michal Bañas

Hlavnou náplňou práce manažéra rizík je odhadnúť, aké problémy môžu nastať pri našom projekte. Ak sa príde na to, že hrozí problém, manažér rizík by mal prísť s riešením pre tento problém, no hlavne by sa mal snažiť týmto problémom predísť za každú cenu.

Hlavnými bodmi sú teda:

- vytvára *metodiku rizík*
- stará sa o zisťovanie možných rizík a problémov, ktoré môžu nastať v projekte
- snaží sa zabraňovať vzniku problémov a ak vzniknú, pracovať na ich odstránení

2.2.9 Manažér komunikácie a správy dát

Bc. Michal Bañas

Úloha manažéra komunikácie je jedna z tých ľahších, a zahŕňa hlavne správu mailového klienta pre tím. Taktiež pre dostatočnú zálohu všetkých potrebných dát sa stará o cloudové úložisko, ktoré patrí mailovému klientovi (v našom prípade Gmail). V neposlednom rade sa stará o všetky

ostatné komunikačné kanály, ako napríklad komunikácia cez Slack, Skype, alebo telefonicky.

Jeho úlohami sú teda:

- vytvorenie *metodiky komunikácie*
- spravuje tímový email, Scrumdesk a cloudové úložisko
- spravuje tímové komunikačné kanály na Slacku a Skype
- stará sa o aktualizáciu webovej stránky tímu

2.3 Rozdelenie manažérskych úloh

Bc. Michal Bañas

V období od začiatku semestra po 9. týždeň bolo rozdelenie úloh v tíme nasledovné:

Úloha	Člen tímu
Hlavný projektový manažér	Bc. Dávid Roba
Manažér plánovania a úloh	Bc. Lukáš Rešutík
Manažér verziovania	Bc. Ernest Loureiro
Manažér testovania	Bc. Dávid Roba
Manažér nasadzovania	Bc. Šimon Harvan
Manažér kvality	Bc. Marko Moravčík
Manažér dokumentovania	Bc. Michal Bañas
Manažér rizík	Bc. Daniel Lukáč
Manažér komunikácie a správy dát	Bc. Lukáš Rešutík

Tab.1.: Rozdelenie úloh v tíme

3. Aplikácie manažmentov

Bc. Michal Baňas

Pri našom postupe semestrom sme si zaviedli viacero metódik, podľa ktorých postupujeme, či už pri tvorbe softvéru, testovaní alebo komunikácii.

3.1 Metodika komunikácie

Bc. Michal Baňas

3.2 Metodika vykazovania prác v Scrumdesk

Bc. Michal Baňas

3.3 Metodika code review

Bc. Michal Baňas

3.4 Metodika dokumentácie

Bc. Michal Baňas

3.5 Metodika code conventions

Bc. Michal Baňas

3.6 Metodika testovania

Bc. Michal Baňas

3.7 Metodika rizík

Bc. Michal Baňas

3.8 Metodika commitovania a verziovania

Bc. Michal Bañas

4. Šprinty

Bc. Michal Baňas

Pre originálne a jednoduché rozlíšenie šprintov sme si vybrali pomenovania podľa štátov

Južnej Ameriky. Názvy šprintov sú nasledovné:

1. Argentínsky šprint
2. Brazílsky šprint
3. Kolumbijský šprint
4. Bolívijský šprint
5. Uruguajský šprint
6. Venezuelský šprint
7. Ekvádorský šprint
8. Čilský šprint
9. Paraguajský šprint
10. Panamský šprint
11. Guyanský šprint
12. Falklandský šprint

5. Argentínsky šprint

Bc. Michal Bañas

5.1 Základné informácie

Začiatok: 2.10.2017

Koniec: 16.10.2017

Tím: Bc. Bañas Michal
Bc. Harvan Šimon
Bc. Loureiro Ernest
Bc. Lukáč Daniel
Bc. Moravčík Marko
Bc. Rešutík Lukáš
Bc. Roba Dávid

5.2 Retrospektíva

Dňa 16.10.2017 sa v rámci formálneho stretnutia uskutočnila retrospektíva Argentínskeho šprintu. Stretnutia sa zúčastnili všetci členovia. Každý zo zúčastnených dostal možnosť vyjadriť svoj názor (čo sa mu páčilo/nepáčilo, čo by zmenil).

Vyjadrenia:

Ernest: - špecifikácia taskov by mala byť podľa neho presnejšia.

Michal: - je nadmieru spokojný so všetkým

Šimon: - nadmieru spokojný so všetkým

Lukáš: - je spokojný s telefonátmi
- chcel by viac komunikácie cez slack

Dávid: - je spokojný s komunikáciou, bola ukážková

Daniel: - spokojný so všetkým

Marko: - spokojný so všetkým

5.3 Úlohy

Úlohy k šprintu boli stanovené na formálnom stretnutí dňa 2.10.2016

Key	Summary	Assignee	Priority
433435	Analýza UT	Marko Moravčík	High
433436	Analýza 2016	Lukáš Rešutík	High
433457	Analýza 2015	Michal Bañas	High
433458	Analýza 2014	David Roba	High
433443	Návrh UI webu	Šimon Harvan	Medium
433444	Grafické prvky pre web	Šimon Harvan	Medium
433445	Programovanie webu	Šimon Harvan	Medium
434321	Inštalácia serveru	Šimon Harvan	High
436141	Zavesenie dokumentácie na web	Šimon Harvan	Medium
433459	Návrh a vytvorenie pomocou PS	Daniel Lukáč	Medium
433460	Tlač	Šimon Harvan	High
434975	Metodika code review	Marko Moravčík	High
434976	Metodika testovania	David Roba	High
434978	Metodika code conventions	Daniel Lukáč	High
435091	1.zápisnica	Lukáš Rešutík	Medium
435093	2.zápisnica	Lukáš Rešutík	Medium
435096	Metodika dokumentácie	Michal Bañas	High
435098	Metodika vykazovania prác v Scrumdesk	Lukáš Rešutík	High
435646	Metodika komunikácie	Lukáš Rešutík	High
436139	3. zápisnica	Simon Harvan	Medium
435103	Príprava prostredia	Všetci	High
436127	Vybuildit UT austin - zistiť čo to robí	Ernest Loureiro	Medium

Tab. 2.: Úlohy v Argentínskom šprinte

5.3.1 Opis jednotlivých úloh

Úlohy sme si zanalyzovali, následne vytvorili Story, ktoré obsahovali menšie podúlohy a tie boli prideľované jednotlivým členom tímu.

- **Task 433435 Analýza UT** - Analyzovať štruktúru kódu UT. Porovnať ju s naším kódom. Pozrieť sa na chôdzu hráča UT.
- **Task 433436 Analýza 2016** - Analyzovať prácu tímu sixPack. Nájsť to najdôležitejšie čo pridali/upravili. Pozrieť sa na metodiky.
- **Task 433457 Analýza 2015** - Analyzovať prácu tímu Team Bender. Nájsť to najdôležitejšie čo pridali/upravili. Pozrieť sa na metodiky.
- **Task 433458 Analýza 2014** - Analyzovať prácu tímu Infinity. Nájsť to najdôležitejšie čo pridali/upravili. Pozrieť sa na metodiky.
- **Task 433443 Návrh UI webu** – návrh webovej stránky a rozhrania pre webovú stránku tímu
- **Task 433444 Grafické prvky pre web** – výber grafických prvkov na webovú stránku
- **Task 433445 Programovanie webu** – naprogramovanie webovej stránky tímu
- **Task 434321 Inštalácia serveru** – proces inštalácie serverovej časti projektu
- **Task 436141 Zavesenie dokumentácie na web** - Zavesenie potrebnej dokumentácie na web
- **Task 433459 Návrh a vytvorenie pomocou PS** – photoshopový návrh plagátu tímu
- **Task 433460 Tlač** – tlač plagátu
- **Task 434975 Metodika code review** – vytvorenie metodiky, podľa ktorej sa celý tím riadi pri procese revízie kódu
- **Task 434976 Metodika testovania** – vytvorenie metodiky, podľa ktorej budú všetci členovia tímu testovať softvér 3D futbalu
- **Task 434978 Metodika code conventions** – vytvorenie metodiky zodpovednej za udržiavanie prehľadného a čistého kódu
- **Task 435091 1.zápisnica** – vytvorenie zápisnice z prvého stretnutia
- **Task 435093 2.zápisnica**– vytvorenie zápisnice z druhého stretnutia
- **Task 435096 Metodika dokumentácie** – metodika k písaniu všetkých potrebných dokumentov
- **Task 435098 Metodika vykazovania prác v Scrumdesk** - špecifikácia pravidiel a spôsobov vykazovania práce v nástroji Scrumdesk
- **Task 435646 Metodika komunikácie** – vytvorenie metodiky pre komunikáciu v rámci tímu
- **Task 436139 3. zápisnica**– vytvorenie zápisnice z tretieho stretnutia
- **Task 435103 Príprava prostredia** – nastavenie prostredia a rozbehanie všetkých potrebných akcií
- **Task 436127 Vybuildit UT austin** - zistiť čo to robí

5.4 Zhodnotenie

V tomto šprinte sa nám podarilo ukončiť všetky úlohy okrem jednej, a to rozbehania prostredia. Nie všetkým členom tímu sa podarilo rozbehať požadované prostredie, no naplnia to v ďalšom šprinte.

5.4.1 Plnenie úloh

Key	Summary	Worked	Status	Logged Work
433435	Analýza UT	Marko	Done	3 hodiny
433436	Analýza 2016	Lukáš (2h 30m)	Done	2 hodiny 30 minút
433457	Analýza 2015	Michal (2h)	Done	2 hodiny
433458	Analýza 2014	David (3h)	Done	3 hodiny
433443	Návrh UI webu	Šimon (2h)	Done	2 hodiny
433444	Grafické prvky pre web	Šimon (1h)	Done	1 hodina
433445	Programovanie webu	Šimon (5h)	Done	5 hodín
434321	Inštalácia serveru	Šimon (3h)	Done	3 hodiny
436141	Zavesenie dokumentácie na web	Šimon (1h)	Done	1 hodina
433459	Návrh a vytvorenie pomocou PS	Daniel (3h)	Done	3 hodiny
433460	Tlač	Šimon (1h)	Done	1 hodina
434975	Metodika code review	Marko	Done	
434976	Metodika testovania	David (3h)	Done	3 hodiny
434978	Metodika code conventions	Daniel (3h)	Done	3 hodiny
435091	1.zápisnica	Lukáš (45m)	Done	45 minút
435093	2.zápisnica	Lukáš (1h)	Done	1 hodina
435096	Metodika dokumentácie	Michal (2h)	Done	2 hodiny
435098	Metodika vykazovania prác v Scrumdesk	Lukáš (3h)	Done	3 hodiny

435646	Metodika komunikácie	Lukáš (2h)	Done	2 hodiny
436139	3. zázpisnica	Šimon (1h)	Done	1 hodina
435103	Príprava prostredia	Lukáš (1h 30m) Michal (1h 30m) Marko (1h 30m) Dávid (1h 30m)	In progress	6 hodín
436127	Vybuildit UT austin - zistiť čo to robí	Ernest (4h)	In progress	4 hodiny

Tab. 3.: Ukážka plnenia úloh v tíme

5.4.2 Podiel práce členov tímu

Na základe vykazovania práce v Scrumdesk bola vytvorená tabuľka sumarizácie vykonanej práce jednotlivými členmi tímu.

Člen tímu	Úlohy	Formálne stretnutia	Neformálne stretnutia	Spolu	Spolu %
<i>Bc. Michal Bañas</i>	<i>4:00</i>	<i>5h</i>	<i>1:30</i>	<i>10,50</i>	<i>11,98%</i>
<i>Bc. Šimon Harvan</i>	<i>14:00</i>	<i>3h</i>	<i>1:30</i>	<i>18,50</i>	<i>21,11%</i>
<i>Bc. Ernest Loureiro</i>	<i>0:00</i>	<i>5h</i>	<i>1:30</i>	<i>6,50</i>	<i>7,42%</i>
<i>Bc. Daniel Lukáč</i>	<i>6:00</i>	<i>5h</i>	<i>1:30</i>	<i>12,50</i>	<i>14,26%</i>
<i>Bc. Marko Moravčík</i>	<i>5:00</i>	<i>5h</i>	<i>1:30</i>	<i>11,50</i>	<i>13,12%</i>
<i>Bc. Lukáš Rešutík</i>	<i>9,35</i>	<i>5h</i>	<i>1:30</i>	<i>15,85</i>	<i>18,08%</i>
<i>Bc. Dávid Roba</i>	<i>6:00</i>	<i>5h</i>	<i>1:30</i>	<i>12,30</i>	<i>14,03%</i>

Tab. 4.: Podiel práce členov tímu

6. Brazílsky šprint

Bc. Lukáš Rešutík

6.1 Základné informácie

Začiatok: 16.10.2017

Koniec: 30.10.2017

Tím: Bc. Baňas Michal
Bc. Harvan Šimon
Bc. Loureiro Ernest
Bc. Lukáč Daniel
Bc. Moravčík Marko
Bc. Rešutík Lukáš
Bc. Roba Dávid

6.2 Retrospektíva

Dňa 30.10.2017 sa v rámci formálneho stretnutia uskutočnila retrospektíva Brazílskeho šprintu. Stretnutia sa zúčastnili všetci členovia. Každý zo zúčastnených dostal možnosť vyjadriť svoj názor (čo sa mu páčilo/nepáčilo, čo by zmenil). Svoj názor taktiež vyjadril aj vedúci tímu Ing. Ivan Kapustík.

Vyjadrenia:

Ing. Ivan Kapustík: *“Nové úlohy, ktoré vznikajú z aktuálnych by si bolo dobré predpripraviť. Taktiež by sme si úlohy nemali vytvárať len tak, ale len počas formálnych stretnutí a po dohode v tíme“.*

Ernest: *“Som spokojný s priebehom tohto šprintu“.*

Michal: *“Dobre sme zvládli implementáciu, funguje dobrá pomoc medzi nami ak niekto niečo nevie“.*

Šimon: *“Rád by som pracoval aj na tímových taskoch, nie len na úlohách spojených s webom“.*

Lukáš: *“Páčilo sa mi, že sme dokázali spraviť niektoré veci s predstihom na termíne, ktorý sme dohodli“.*

Dávid: *“Myslím, že v ďalšom šprinte by sme sa mali viac venovať implementačným úlohám a robiť analýzy rýchlejšie“.*

Daniel: *“Myslím, že už bolo povedané všetko“.*

6.3 Úlohy

Úlohy k šprintu boli stanovené na formálnom stretnutí dňa 26.10.2016

Key	Summary	Assignee	Priority
242203	JIM		High
242184	Dokumentácia		High
242180	UT Austin		High
442003	Implementácia pomalých funkcií (sin/cos)	Michal Bañas	High
439491	Analýza optimalizácie pomalých funkcií (sin/cos)	Michal Bañas	High
439479	Vybaviť licenciu k atlassianu	Ernest Loureiro	Medium
439483	Analýza implementácie kalmana v C++ (UT)	Marko Moravčík	Medium
436127	Vybuildit UT Austin - zistiť čo to robí	Ernest Loureiro	Medium
439484	Analýza starších tímov FIIT práce s kalmanom	Dávid Roba	High
439482	Analýza implementácie kalmana v C++ (UT)	Dávid Roba	High
436097	Porovnanie kalman UT - JIM	Marko Moravčík	High
435103	Rozbehánie prostredia	Všetci	High
436003	Wiki návod inštalácie projektu - MAC	Ernest Loureiro	Medium
442017	Rozbehánie media wiki	Šimon Harvan	Medium
439508	Zavesenie návodu na web	Ernest Loureiro	Low
439446	4. zápisnica	Michal Bañas	medium
439499	Retrospektíva/sumarizácia 1.šprintu	Lukáš Rešutík Michal Bañas	High
439450	Metodika rizík	Daniel Lukáč	High
439505	Dokument k názvom šprintov	Lukáš Rešutík	Low
441994	5. zápisnica	Daniel Lukáč	Medium
439488	Analýza vykresľovania čiar	Daniel Lukáč	High

Tab.5.: Úlohy v Brazílskom šprinte

6.3.1 Opis jednotlivých úloh

Úlohy sme si zanalyzovali, následne vytvorili Story, ktoré obsahovali menšie podúlohy a tie boli pridelované jednotlivým členom tímu.

- **Task 439491 Analýza optimalizácie pomalých funkcií (sin/cos)** – Cieľom analýzy je navrhnúť spôsob vlastnej implementácie funkcií sínus a kosínus.
- **Task 442003 Implementácia pomalých funkcií (sin/cos)** – Z predchádzajúcej analýzy sme sa rozhodli implementovať funkcie sínus a kosínus.
- **Task 439484 Analýza starších tímov práce s kalmanom** - K ďalšej práci je nutné zanalyzovať práce minuloročných tímov na práci pri kalmane.
- **Task 439482 Analýza implementácie kalmanu (všeobecne)** – Taktiež je nutná analýza princípov Kalmanovho filtra. Ako funguje, kde sa využíva, ako sa implementuje v rôznych programovacích jazykoch.
- **Task 436003 Wiki návod inštalácie projektu na MAC** – Na základe problémom pri rozbehaní

prostredia na MAC sme sa rozhodli použiť vlastné skúsenosti a napísať návod.

- **Task 439508 Zavesenie návodu na web** – Návod je potrebné zverejniť na webe.
- **Task 439446 4. Zápisnica** - Po každom formálnom stretnutí je vytvorená zápisnica, v ktorej sú spomenuté všetky dôležité veci, ktoré boli prebraté alebo spomenuté na tomto stretnutí.
- **Task 441994 5. Zápisnica** - Po každom formálnom stretnutí je vytvorená zápisnica, v ktorej sú spomenuté všetky dôležité veci, ktoré boli prebraté alebo spomenuté na tomto stretnutí.
- **Task 439450 Metodika rizík** - Každý projekt má svoje riziká. Preto je nutné tieto riziká spísať a pripraviť riešenia ako sa im vyhnúť.
- **Task 439505 Dokument k názvom šprintov** – Rozhodli sme sa pre lepšie označenie šprintov vytvoriť dokument s názvom šprintom.

6.4 Zhodnotenie

V tomto šprinte sa nám nepodarilo ukončiť všetky úlohy. Analýzu vykresľovania čiar sa nám nepodarilo dobre odhadnúť obťažnosť a komplexnosť úlohy. Preto sme sa ju rozhodli preniesť do ďalšieho šprintu a rozložiť na menšie úlohy Získali sme 13 Story pointov z 21.

6.4.1 Plnenie úloh

Key	Summary	Worked	Status	Logged Work
442003	Implementácia pomalých funkcií (sin/cos)	Michal (5h)	Done	5 hodín
439491	Analýza optimalizácie pomalých funkcií (sin/cos)	Michal (2h)	Done	2 hodiny
439479	Vybavit licenciu k atlassianu	Ernest (2h)	Done	2 hodiny
439483	Analýza implementácie kalmana v C++ (UT)	Marko (5h)	Done	5 hodín
436127	Vybuildit UT Austin - zistiť čo to robí	Ernest (2h)	Done	2 hodiny
439484	Analýza starších tímov FIIT práce s kalmanom	Dávid (3h)	Done	3 hodiny
439482	Analýza implementácie kalmanu (všeobecne)	Dávid (5h)	Done	5 hodín
436097	Porovnanie kalman UT - JIM	Marko (4h)	Done	4 hodiny
435103	Rozbehovanie prostredia	Ernest (1h 30m) Šimon(1h 30m)	Done	3 hodiny
436003	Wiki návod inštalácie projektu - MAC	Ernest (5h)	Done	5 hodín
442017	Rozbehovanie media wiki	Šimon (2h 30m)	Done	2 hodiny 30 minút
439508	Zavesenie návodu na web	Ernest (2h)	Done	2 hodiny
439446	4. zápisnica	Michal (1h)	Done	1 hodina
439499	Retrospektíva/sumarizácia 1.šprintu	Lukáš (2h 30m) Michal (1h 30m)	Done	4 hodiny
439450	Metodika rizík	Daniel (3h)	Done	3 hodiny
439505	Dokument k názvom šprintov	Lukáš (1h)	Done	1 hodina

441994	5. zápisnica	Daniel (1h 30m)	Done	1 hodina 30 minút
439488	Analýza vykresľovania čiar	Daniel	Prenesenie do ďalšieho šprintu	0 hodín

Tab. 6.: Ukážka plnenia úloh v tíme

6.4.2 Podiel práce členov tímu

Na základe vykazovania práce v SCRUMDESK bola vytvorená tabuľka sumarizácie vykonanej práce jednotlivými členmi tímu.

Člen tímu	Úlohy	Formálne stretnutia	Neformálne stretnutia	Spolu	Spolu (%)
Bc. Michal Bañas	9:30	6h	2h	17:30	15,6
Bc. Šimon Harvan	4:00	6h	2h	10:00	11,7
Bc. Ernest Loureiro	10:30	6h	2h	18:30	17,1
Bc. Daniel Lukáč	4:30	6h	2h	12:30	9,8
Bc. Marko Moravčík	9:00	6h	2h	17:00	18,1
Bc. Lukáš Rešutík	4:00	6h	2h	12:00	12,2
Bc. Dávid Roba	8:00	6h	2h	16:00	16,6
				103:00	

Tab. 7.: Podiel práce členov tímu

7. Kolumbijský šprint

LACOOOOOOOOO MUHAHAHAHAHA

8. Retrospektíva po prvých troch šprintoch

9. Zoznam príloh

Príloha A: Metodika komunikácie

Príloha B: Metodika vykazovania prác v Scrumdesk

Príloha C: Metodika code review

Príloha D: Metodika dokumentácie

Príloha E: Metodika code conventions

Príloha F: Metodika testovania

Príloha G: Metodika rizík

Príloha H: Metodika commitovania a verziovania

Príloha A: Metodika komunikácie:

1. Základné informácie

Cieľom tohto dokumentu je špecifikovať pravidlá a priebeh komunikácie počas práce tímu BAREKO na tímovom projekte v akademickom roku 2017/2018.

Komunikačné nástroje:

- Formálny komunikačný kanál: Osobné stretnutia, Slack, Gmail
- Neformálny komunikačný kanál: Skype, Facebook, mobil
- Dátové úložisko: Scrumdesk, Google Drive

2. Formálny komunikačný kanál

2.1 Osobné stretnutia

Osobné stretnutia sa riadia metodikou priebehu formálnych stretnutí.

2.2 Slack

Služi ako hlavný komunikačný kanál medzi členmi tímu. V rámci Slacku sme si vytvorili niekoľko špecifických kanálov:

- General - základný kanál, ktorý slúži na bežnú (pracovnú) komunikáciu
- Urgent - kanál, ktorý slúži pri krízových situáciách
- Zábava – kanál, ktorý slúži na pobavenie tímu

2.3 Gmail

Gmail funguje ako oficiálna tímová e-mailová schránka – barekofiit@gmail.com. Služi na posielanie, prijímanie e-mailov a všetku oficiálnu komunikáciu za tím BAREKO.

Pravidlá pre písanie e-mailov:

- Každý oficiálny mail je nutné napísať z tejto adresy
- Obsah správy by mal byť štruktúrovaný a gramaticky správny
- Predmet by mal byť krátky a výstižný

3. Neformálny komunikačný kanál

3.1 Skype

Skype je platforma zjednotenej komunikácie, ktorá umožňuje odosielať okamžité správy, uskutočňovať audiokonferencie. Tento kanál je využívaný v prípade nejasností, problémov, pomoci medzi členmi tímu. Ak sa člen fyzicky nemôže zúčastni tímového stretnutia, môže tento kanál použiť.

3.2 Facebook

Na adrese www.facebook.com sme si zriadili tímový chat. Tento chat sa používa sporadicky, prípadne ako zábava pre tím.

3.3 Mobil

Mobilnú komunikáciu používame len v prípade nutných prípadov (dohodnutie neformálneho

tímového stretnutia, výpadok člena tímu). Telefónne čísla si členovia tímu vymenia na tímovom stretnutí.

2

4. Dátové úložisko

4.1 Scrumdesk

Primárne úložisko predstavuje Scrumdesk. Každý vzniknutý dokument pripadá konkrétnej úlohe (tasku) v Scrumdesku.

4.2 Google Drive

Prestavuje sekundárne úložisko, ktoré slúži na zálohovanie dát.

Lukáš Rešutík

Príloha B: Metodika vykazovania prác

v Scrumdesk

1. Základné informácie

Cieľom tohto dokumentu je špecifikovať pravidlá a spôsob vykazovania práce v nástroji Scrumdesk počas práce tímu BAREKO na tímovom projekte v akademickom roku 2017/2018. Účelom tohto nástroja je manažovanie projektu - vytváranie šprintov, úloh, sledovanie stavu plnenia úloh, preto k tomu treba pristupovať jednotne.

1.1 Životný cyklus v nástroji Scrumdesk

V rámci agilného vývoja a nástroja na vykazovanie práce Scrumdesk, sa môže identifikovaná úloha nachádzať práve v jednom z týchto stavov:

- Todo – vytvorená úloha, ktorú je potrebné vypracovať,
- In progress – na vyriešení úlohy už niekto pracuje,
- Done – úloha bola úspešne spravená (zahŕňa vykonania revízie kódu, úspešným testovaním a gramatickou kontrolou).

Keďže Scrumdesk neponúka v rámci stavov fázu testovania, tak fáza In progress zahŕňa tieto činnosti. Nasledujúci obrázok predstavuje skutočný cyklus úlohy.

Obrázok 1: Životný cyklus úlohy

2

2. Vytváranie úloh

Primárne sa úlohy vytvárajú na formálnych stretnutiach s vedúcim projektu, ktorý zároveň predstavuje vlastníka produktu (Ing. Ivan Kapustík). Úlohy tvorí vlastník produktu, v druhom prípade ich tvoria členovia tímu BAREKO. Úlohy, ktoré boli identifikované na stretnutiach, vytvára do nástroja Scrumdesk náš scrum master Bc. Rešutík Lukáš. Vo výnimočných situáciách (nepítomnosť scrum mastra) ich môže pridať každý člen po skonzultovaní so scrum mastrom, prípadne celým tímom.

2.1 Typy úloh a atribúty úloh

V nástroji Scrumdesk poznáme tri typy úloh:

- Epic – úloha, ktorá je svojou zložitosťou veľmi veľká, obsahuje viacero stories a veľmi často môže trvať viac ako jeden šprint,
- Story – úloha, ktorá predstavuje príspevok z hľadiska funkcionality, typicky by mala byť ukončená v rámci jedného šprintu a zložitosť úlohy je na prostrednej úrovni,
- Task – úloha, ktorá je zložitosťou najjednoduchšia a zároveň je definovaná s očakávaným výsledkom.

Atribúty taskov v nástroji Scrumdesk, ktoré je potrebné vyplniť:

- Type – typ úlohy:
 - o Task – defaultný typ úlohy,
 - o Analýza – úloha zameraná na analýzu,
 - o Vývoj – úloha zameraná na vývoj,
 - o Test – testovacia úloha,
 - o Document – úloha zameraná na tvorbu dokumentu.
- Title – názov úlohy, ktorý by mal byť stručný a výstižný,
- Estimated time – odhadovaný čas potrebný na ukončenie úlohy (v hodinách),
- Description – podrobný popis úlohy, ktorá sa má spraviť.

Pre story je potrebné nastaviť ďalšie atribúty:

- Release – zaradenie úlohy do konkrétneho releasu,
- Sprint – zaradenie úlohy do konkrétneho šprintu.

3. Vykazovanie práce

V rámci každej úlohy v nástroji Scrumdesk existuje možnosť Worklog, ktorá umožňuje zalogovať čas strávený vypracovávaním úlohy. Za účelom získavania čo najpresnejšej štatistiky

sme si určili tieto pravidlá:

- Vykazovať všetku prácu, ktorá bola vykonaná pri úlohe,
- Čas sa vykazuje v hodinách,
- Snažiť sa vykazovať čo najpresnejší časový úsek, ktorý bol skutočne odpracovaný,
- Do komentáru zapíše, čo konkrétne spravil,
- V prípade, že sa jedná o úlohu, ktorej výsledok je dokument, uploadne tento dokument k tasku.

Čas sa vykazuje v stave In Progress alebo po ukončení úlohy.

3

4. Ďalšie informácie

4.1 Prílohy

Scrumdesk umožňuje ku každej úlohe priložiť súbor. Súbory sa prikladajú v úlohách, ktoré vytvoria dokument a ten je následne do tohto miesta zavesený. V prípade úlohy, ktorá predstavuje chybu je vhodné priložiť obrázky zachytávajúce chybu.

Príloha C: Metodika code review

Tento dokument opisuje postup, ktorým sa celý tím riadi pri procese revízie kódu (code review). Revízia kódu spočíva na princípe komentovania kódu za účelom maximalizovania kvality, funkčnosti, prehľadnosti a zrozumiteľnosti kódu.

1 Základné informácie

Máme k dispozícii kód projektu z minulých rokov, na ktorom teda budeme pokračovať. Nachádza sa na repositároch služby Bitbucket. Preto budeme používať rozhranie tohto nástroja pre proces code review.

Tento proces zahŕňa dve roly:

👤 **Riešiteľ úlohy** - Každá implementačná úloha musí mať prideleného člena tímu, ktorý je zodpovedný za jej riešenie. Riešiteľ je autor kódu prislúchajúceho danej úlohe.

👤 **Posudzovateľ (reviewer)** - Každý člen tímu má prideleného posudzovateľa, ktorý mu prislúcha počas celého projektu. Reviewer je člen tímu, ktorého úlohou je skontrolovať kód, ktorého autorom je riešiteľ.

2 Priebeh code review

Code review bude prebiehať podľa nasledujúcich krokov:

1. Riešiteľ odovzdá kód s riešením tasku do príslušnej vetvy repositára.
2. Riešiteľ vytvorí „pull request“ prostredníctvom rozhrania nástroja Bitbucket, do poľa „Reviewers“ vloží meno svojej dvojice a do časti „Description“ napíše krátky opis riešenia z vyzdvihnutím nejakých hlavných častí + názov tasku zo ScrumDesku.
3. Po vytvorení „pull requestu“ riešiteľ v ScrumDesku pridá k tasku komentár s informáciou, že kód bol odoslaný na revíziu, zmení stav úlohy na „Done“ a označí task ako „Mark as waiting“.
4. Reviewer si stiahne verziu kódu s vyriešenou úlohou a preverí kvalitu kódu, napríklad: dodržiavanie konvencií, prehľadnosť kódu, funkčnosť riešenia. Ak boli vytvorené príslušné testy, reviewer ich spustí a vyhodnotí.
5. V prípade, že reviewer posúdil kód ako vyhovujúci a nemá žiadne výhrady, potvrdí spojenie vetiev výberom voľby „Merge“. Následne okomentuje daný task v ScrumDesku, že prebehol code review a všetko je v poriadku a odznačí flag „Mark as waiting“ z tasku. Týmto sa proces code review končí.
6. V prípade nájdených nedostatkov v kóde, reviewer vyjadrí svoje pripomienky prostredníctvom komentárov. Komentáre sa pridávajú ku riadku kódu, ktorého sa týkajú. Pull request sa zamietne voľbou „Decline“. V ScrumDesku označí task ako „Impeded“ a okomentuje ho.
7. Riešiteľ opraví identifikované nedostatky, vloží opravený kód do repositára a v ScrumDesku odznačí flag „Impeded“.
8. Pokračuje sa krokom 3

Poznámka: prípadné konflikty pri spájaní vetiev musí vyriešiť autor kódu, teda riešiteľ

Obrázok 1: Ukážka obrazovky pre vytvorenie "pull requestu"

Vypracoval: Marko Moravčík

Príloha D: Metodika dokumentácie

Táto metodika popisuje zjednocuje štýl písania dokumentov tímu BAREKO počas práce na tímovom projekte v akad. roku 2017/2018.

Na vytváranie dokumentov sa používa Microsoft Word vo verzii podporujúcej .docx formát. V celých dokumentoch sa používa čierne písmo, modrý text je pre hyperlinky.

1. Úvodná strana

Úvodná strana neobsahuje číslo strany.

1.1. Obsah úvodnej strany

Úvodná strana obsahuje nasledujúce údaje v danom poradí :

1. názov univerzity – „Slovenská technická univerzita v Bratislave“,
2. názov fakulty - „fakulta informatiky a informačných technológií“,
3. šesť prázdnych riadkov,
4. názov dokumentu,
5. názov tímu - „Tím BAREKO“
6. Prázdne riadky
 - a. Ak má názov dokumentu jeden riadok – 19
 - b. Ak má názov dokumentu dva riadky - 16
7. Abecedne zoradený zoznam členov tímu podľa priezviska v tvare titul meno priezvisko, oddelené jednou medzerou.
8. Prázdny riadok
9. Text „Vedúci projektu:“ a meno vedúceho projektu – „Ing. Ivan Kapustík“ oddelený jedným znakom tab.
10. Text „predmet: Tímový projekt I:“. Text je oddelený dvoma znakmi tab, medzi znakom dvojbodky a začiatkom slova Tímový
11. Text „Akademický rok: 2016 / 2017“. Text je oddelený dvoma znakmi tab, medzi znakom dvojbodky a 2016 / 2017. Pred a za lomkou sa nachádza jedna medzera.

1.2. Formátovanie úvodnej strany

Formátovanie úvodnej strany dokumentov, podľa bodov uvedených v kapitole „1.1.1. Obsah úvodnej strany“:

1,2 (názov univerzity a fakulty):
Typ písma: TimesNewRoman,
Veľkosť písma: 14,
Medzera pred: 0b,
Medzera po: 8b
Zarovnanie na stred

3,6,8 (Prázdne riadky)
Typ písma: TimesNewRoman,
Veľkosť písma: 12,
Medzera pred: 0b,
Medzera po: 0b

4 (Názov dokumentu)
Typ písma: TimesNewRoman,
Veľkosť písma: 26,
Medzera pred: 0b,
Medzera po: 8b
Zarovnanie na stred

5 (Tím sixPack)
Typ písma: TimesNewRoman,
Veľkosť písma: 20,
Medzera pred: 0b,
Medzera po: 8b
Zarovnanie na stred

7 (zoznam členov tímu)
Typ písma: Calibri,
Veľkosť písma: 11,
Medzera pred: 0b,
Medzera po: 8b
Zarovnanie doľava
Kurzíva

9,10,11
Typ písma: Calibri,
Veľkosť písma: 11,
Medzera pred: 0b,
Medzera po: 8b
Zarovnanie doľava

2. Obsah

Obsah sa nachádza na prázdnej strane, nasledujúcej po úvodnej strane. Formát číslovania strán obsahu je nasledujúci „i“, „ii“, „iii“, ...

Prvý riadok obsahu je prázdny, nasleduje text „Obsah“ ako nadpis prvej úrovne bez čísla (vid' [kapitola 3](#)) a opäť prázdny riadok.

Telo obsahu obsahuje kapitoly maximálne po tretiu úroveň. Každá úroveň je v obsahu odsadená. Za názvom kapitoly poprípade podkapitoly sa nachádzajú znaky bodky pričom čísla strán sú zarovnané vpravo.

Prázdne riadky a telo obsahu sú normálnym štýlom.

3. Text

3.1. Text

Normálny text:
Typ písma: Calibri,
Veľkosť písma: 11,
Medzera pred: 0b,
Medzera po: 0b

Nadpis prvej úrovne:

Typ písma: Calibri light,
Zarážka vľavo: 1,23 cm,
Veľkosť písma: 24,
Medzera pred: 0b,
Medzera po: 0b
Zarovnanie doľava

Nadpis druhej úrovne:

Typ písma: Calibri,
Zarážka vľavo: 1,23 cm
Veľkosť písma: 20,
Medzera pred: 0b,
Medzera po: 0b
Zarovnanie doľava

Nadpis tretej úrovne:

Typ písma: Calibri,
Zarážka vľavo: 1,27 cm
Veľkosť písma: 12,
Medzera pred: 0b,
Medzera po: 0b
Zarovnanie doľava

3.3. Pomenovanie a identifikovanie objektov

Pod každým obrázkom, grafom, diagramom, tabuľku, ... sa musí nachádzať identifikátor objektu a stručný opis. Normálny text zarovnaný na stred.

Identifikátor objektu:

- Obr.[číslo] – pre obrázky
- Graf[číslo] – pre grafy
- Tab.[číslo] – pre tabuľku
- Diagram[číslo] – pre diagram

Číslo je číslo kapitoly/ podkapitoly, kde sa objekt nachádza. Ak je viac objektov rovnakého typu v danej kapitole/ podkapitole, za číslom sa nachádza malé písmeno (začínajúc od a pre prvý výskyt objektu daného typu napr. obr1.1a (obrázok je prvý v prvej podkapitole prvej kapitoly)).

Vypracoval: Michal Bañas

Príloha E: Metodika code conventions

1. Úvod

Tento dokument sa zaoberá problematikou konvencie písania kódu pre projekt RoboCup. Obsahuje opis požiadaviek pre vytváranie zdrojového kódu, ktorý má byť každý zúčastnený programátor a člen tímu riadiť. Projekt je vytváraný v jazyku Java, ktorého konvencie sú už známe a zaužívané. Aj napriek už zaužívaným konvenciám písania Java kódu sa vylučujú potrebné zmeny niektorých členov z tímu v istej miere.

2. Komentáre

Komentáre sa používajú pre vysvetlenie v prípade nejasností v kóde. Komentár je potrebné umiestniť na správne miesto. V prípade funkcií je vhodné umiestniť komentár pred deklaráciu funkcie; hlavičku, kde je možné uviesť v skratke k čomu daná funkcia slúži, uviesť vysvetlenie návratového typu, vstupných parametrov a pododiel. Tak isto je vhodné komentovať deklaráciu jednotlivých premenných, vytvorené podmienky v kóde a podobne. V jave rozlišujeme hlavne dva typy komentárov a to:

■ jednoriadkové

o začínajú sa znakmi //

o majú vplyv na celý riadok (komentár nepozná uzatvárací znak)

//jednoriadkový komentár

■ viacriadkové

o sú uzavreté medzi znaky /* a */, pričom medzi týmito znakmi je potrebné uviesť na začiatku každého riadku znak *

```
/*
```

```
* viacriadkový
```

```
* komentár
```

```
*/
```

Príklady použitia:

```
/**
```

```
* Returns lower limit of Joint angle if specified angle parameter is less
```

```
* than lower limit or upper limit of Joint angle if specified angle
```

```
* parameter is greater than upper limit. Otherwise returns specified angle.
```

```
*
```

```
* @param value
```

```
* angleInXml as double
```

```
* @return value angleInXml as double
```

```
*/
```

```
public double getNextRecord(double angleInXml) {
```

```
if (angleInXml < lowerLimit){
```

```
return lowerLimit;
```

```
}
```

```
if (angleInXml > upperLimit){
```

```
return upperLimit;
```

```
}
```

```
return angleInXml;
```

```
}
```

3. Formátovanie kódu

Dôležitým aspektom čitateľnosti a porozumeťia zdrojového kódu je jeho formátovanie. V zásade má valovať, že dĺžka riadku má byť presiahnuť počet znakov 80, ak je tomu tak, riadok sa správne odsadí a rozdelí do ďalšieho riadku. V súčasnosti sú dostupné rôzne nástroje pre automatické formátovanie, no v dokumente uvádzame tie hlavné z nich.

Pre deklaráciu premenných platí, že každá premenová musí byť deklarovaná vždy na svojom

riadku. Nešlo by sa stať až v tomto prípade každý riadok obsahoval viač ako jeden znak ' ; ' .

```
int count = 1;
```

```
float[] temperatures = new float[1000];
```

Pre príkazy, podobne ako premenné, v zásade platí, čo nový príkaz, to nový riadok.

```
count = count + progressNo;
```

```
count++;
```

Pre bloky platí viacero pravidiel. V rámci bloku musí byť každý riadok odsadený tabulátorom.

V rámci podmienok(if-else) alebo cyklov(for) je vhodné uviesť do bloku aj jeden príkaz, ktorý nasleduje za danou konštrukciou.

```
{
```

```
count++;
```

```
}
```

```
for(i = 0; i < 10; i++) {
```

```
count += i;
```

```
}
```

```
if(count == 10) {
```

```
count = 0;
```

```
}
```

4. Deklarácie

Pri deklaráciách platia v zásade určité pravidlá pre pomenovanie tried, rozhraní a metód. Každé pomenovanie má byť stručné a jasné, čo dáva predstavu o triede alebo metóde. Okrem toho je dôležité dodržiavať aj konvencie dodržiavať veľké a malé písmená pre jednotlivé názvy. Pre konštanty platí, že majú byť zložené výlučne veľkými písmenami.

Pre premenné platí, že majú sa začínať veľkým písmenom. Ak je pomenovanie zložené z viacerých slov, ďalšie slovo má začínať veľkým písmenom.

```
string encodedName;
```

```
int numberOfRecords;
```

Pre triedy platí, že majú sa začínať veľkým písmenom a majú byť podstatné meno. Ak je pomenovanie zložené z viacerých slov, ďalšie slovo má začínať veľkým písmenom.

```
public class LowSkill{}
```

```
public class EffectorData{}
```

```
public class Actor{}
```

Pre rozhrania platí, že majú sa začínať veľkým písmenom a majú byť prídavné meno. Ak je pomenovanie zložené z viacerých slov, ďalšie slovo má začínať veľkým písmenom.

```
public interface ActionListener{}
```

```
public interface Runnable{}
```

```
public interface Remote{}
```

Pre metódy platí, že majú sa začínať veľkým písmenom a majú byť sloveso. Ak je pomenovanie zložené z viacerých slov, ďalšie slovo má začínať veľkým písmenom.

```
public void actionPerformed()
```

```
public string readNextRecord()
```

Zdroje

[1] <https://www.javatpoint.com/java-naming-conventions>

[2] <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

[3] <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-141999.html#216>

[4] <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-142311.html#430>

[5] <http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-141270.html#2991>

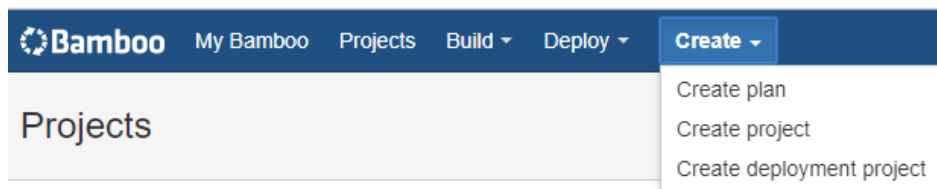
Príloha F: Metodika testovania

Obsahom tohto dokumentu je postup, ktorým sa bude celý tím riadiť pri procese testovania softvéru 3D futbalu. Primárne je táto metodika určená pre manažéra testovania ale budú ju využívať tiež vývojári pri vytváraní testov pretože opisuje postupy pre vytváranie plánov pre automatické testovanie a buildovanie v prostredí Bamboo.

1. Vytváranie plánov v prostredí Bamboo

1.1 Vytváranie nového plánu

Prostredníctvom tlačidla Create sa zobrazí ponuka, z ktorej si pre vytvorenie nového plánu zvolí Create plan (Obr. 1.1a).



Obrázok 1.1a: Vytvorenie nového plánu

Následne je potrebné vyplniť základný popis projektu, pre ktorý sa daný plán vytvára a taktiež popis samotného plánu (Obr. 1.1b).

The image shows a screenshot of the Bamboo form for creating a new project and build plan. The form is titled 'Project and build plan name' and contains several input fields. The 'Project name' field is filled with 'BAREKO'. The 'Project key' field is filled with 'BAR', with a note below it: 'For example AT (for a project named Atlassian)'. The 'Project description' field is filled with '3D footbal'. The 'Plan name' field is filled with 'Test Plan'. The 'Plan key' field is filled with 'TP', with a note below it: 'For example WEB (for a plan named Website)'. The 'Plan description' field is filled with 'Metodika testovania'.

Obrázok 1.1b: Popis plánu

Po vyplnení základného popisu je potrebné nastaviť pripojenie na Bitbucket. Pomocou dropdown menu sa vyberie možnosť Bitbucket Cloud a zvolí sa pomenovanie pre daný repozitár. Spôsob prístupu sa vyberie Private a ako prihlasovacie meno a heslo sa použijú údaje, ktorými sa prihlasuje na Bitbucket. Následne sa na základe prihlasovacieho mena načíta príslušný repozitár a ako spôsob autentifikácie sa vyberie prihlasovacie meno a heslo. Na záver sa zvolí Branch, ktorého zdrojový kód chceme otestovať a overí sa spojenie. Ak je všetko v poriadku je možné pokračovať k pridávaniu taskov (Obr. 1.1c).

Link repository to new build plan

Repository host* Bitbucket Cloud ▾

Display name* Bareko (first branch)

Bitbucket Cloud details

Repository access Private ▾

Select a private repository that a particular Bitbucket Cloud user can access.

Use shared credentials to authenticate

Provide username and password

Reuse predefined shared credentials or provide custom username/password pair for authentication.

Username* davidroba@gmail.com

The Bitbucket Cloud user required to access the repository.

Password*

The password required by the Bitbucket Cloud username.

Repository robocup_tp09/agent (git) ▾ Load Repositories

Select the repository you want to use for your Plan.

Authentication type Username and password ▾

Choose how Bamboo should authenticate when connecting to the repository. You can reuse the username to a different authentication type.

Branch master ▾

Choose a branch you want to check out your code from.

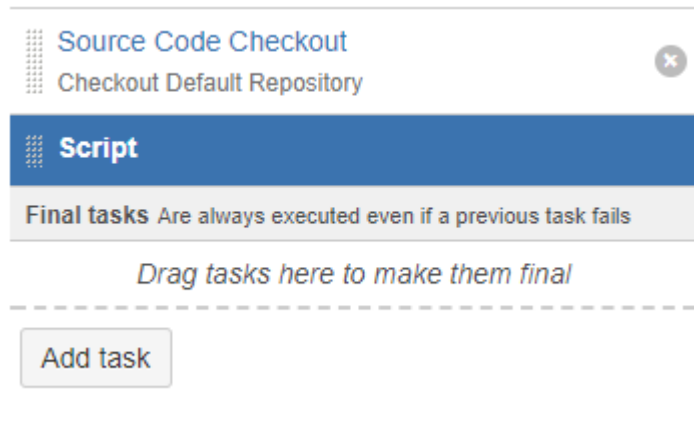
Test connection

✓ Connection successful

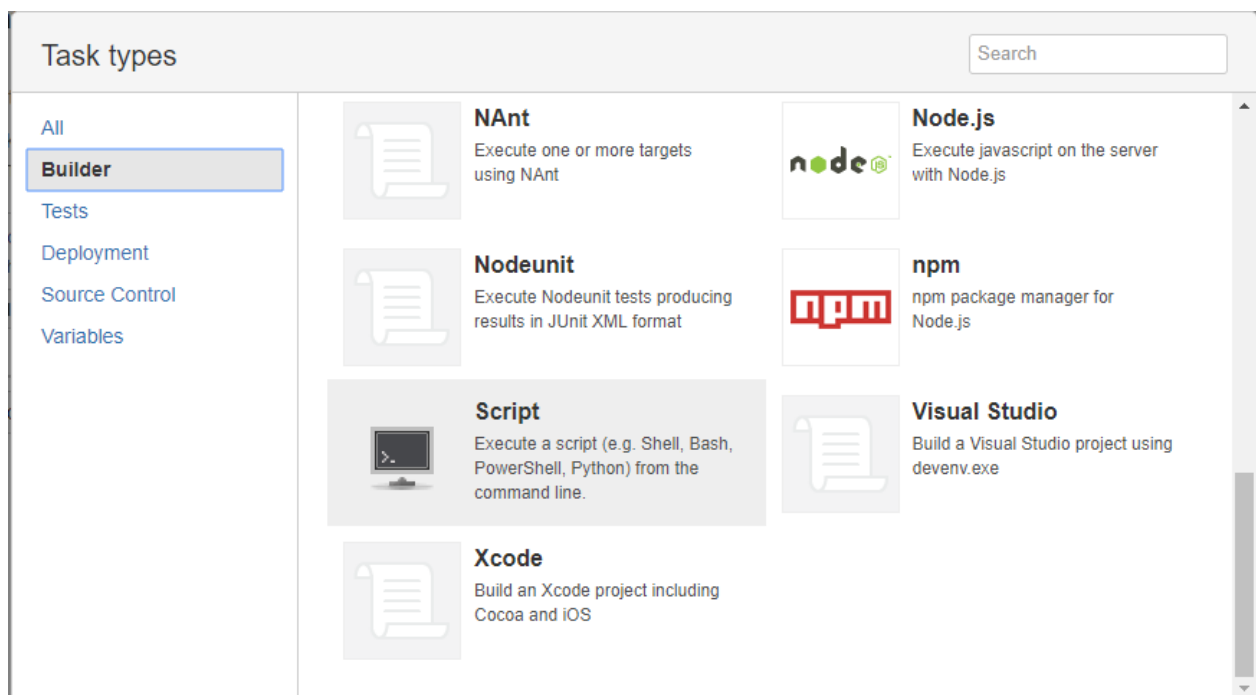
Obrázok 1.1c: Nastavenie pripojenia na repositár

1.2 Vytváranie skriptov

Nový task je možné vytvoriť pomocou tlačidla Add Task (Obr. 1.2a). Pre vytvorenie jednoduchého skriptu, ktorý slúži na zbuildovanie projektu sa vyberie z časti Builder úloha Script (Obr. 1.2b).



Obrázok 1.2a: Pridanie nového tasku



Obrázok 1.2b: Výber úlohy Script z časti Builder

Samotný script sa skladá z dvoch jednoduchých príkazov:

```
cd TestFramework
ant -f Build3.xml
```

Prvým príkazom sa script nastaví na priečinok TestFramework a druhým príkazom sa spustí súbor Build3.xml vo formáte Ant. Prepínač `-f` umožňuje vybrať súbor (Obr. 1.2c).

Script configuration

How to use the Script task

Task description

Prvý skript Test

Disable this task

Interpreter

Shell

An interpreter is chosen based on the shebang line of your script.

Script location

Inline

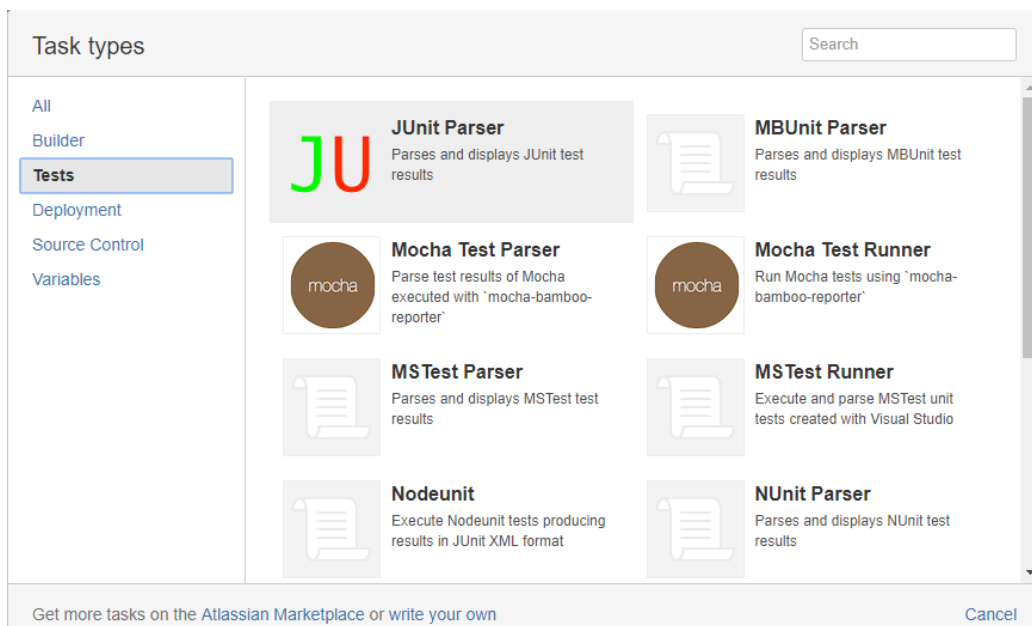
Script body*

```
1 cd TestFramework
2 ant -f Build3.xml
3 |
```

Obrázok 1.2c: Pridanie skriptu

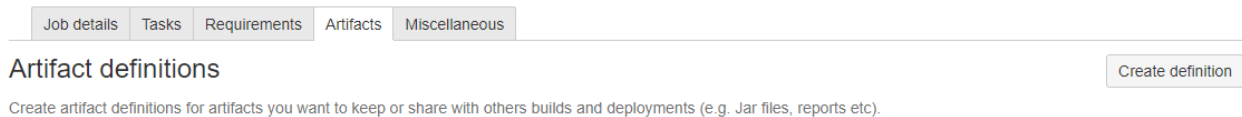
1.3 Vyhodnotenie a zobrazenie testov

Na reálne testovanie je potrebné naplniť tasky JUnit testmi. Tie je možné pridať pomocou tlačidla Add Task, kde sa v časti Tests vyberie úloha JUnit Parser, ktorá zanalyzuje testy a umožní ich následne zobrazenie (Obr. 1.3a).



Obrázok 1.3a: Výber úlohy JUnit Parser z časti Tests

Následne je potrebné vytvoriť artefakt, pomocou ktorého je možné zobraziť výsledky testov. Artefakt sa vytvorí pomocou tlačidla Create definition (Obr. 1.3b).



Obrázok 1.3b: Pridanie definície artefaktu

V definícií je potrebné zadať výstižný názov, cestu kam má byť umiestnený artefakt a Copy pattern, ktorý má v našom prípade hodnotu `**/*.*` (Obr. 1.3c).

Obrázok 1.3c: Vytváranie artefaktu

1.4 Spustenie plánu

Po prepnutí na Dashboard vidíme všetky projekty spolu s plánmi, ktoré im boli vytvorené a priradené. Plán je možné zapnúť pomocou tlačidla Enable v pravej časti obrazovky (Obr. 1.4a).

Project	Plan	Build	Completed	Tests	Reason
▼ BAREKO	Test Plan	⊖ Never built			🔄 ✎ ☆

Obrázok 1.4a: Zapnutie plánu

Vypracoval: Dávid Roba

Príloha G: Metodika rizík

Identifikácia rizika

Treba identifikovať, čo môže ohroziť projekt počas jeho vývoja. Riziká treba dokumentovať a patrične analyzovať. Analýza rizík by mala prebiehať počas celej práce na projekte. Riziká majú dve základné vlastnosti:

- pravdepodobnosť, že udalosť nastane
- odhad škôd

Riziká môžeme ďalej rozdeliť na interné a externé. Interné predstavujú ovplyvniteľné veci, ako priradenie práce členovi tímu, prijatie ďalšieho člena a podobne. Medzi externé riziká patria napríklad zmeny na trhu, a ďalšie, ktoré nie je možné ovplyvniť. V našom projekte sa budeme zameriavať na analýzu a manažovanie interných rizík.

Manažovanie a minimalizácia rizika



Obr.1.: Manažment rizík

- potreba identifikácie a definície rizika
- určenie rozsahu rizika
- analýza rizika vytvorenie plánu
- vytvorenie opatrení a zlepšenie situácie
- kontrolovanie alebo akceptácia rizika

Identifikácia rizík

1 - najmenej závažné a 10 - najviac závažné

	Pravdepodobnosť v %	Závažnosť
Slabý manažment tímu	20	9
Neporozumenie zadanej práce	50	6
Neschopnosť člena splniť požiadavku	30	7
Zmena rozsahu a cieľa práce	5	8
Nedostatok znalostí	20	3
Nevhodné obsadenie pozícií v tíme	20	4
Odchod člena z tímu	10	8
Nevhodné rozloženie práce	40	7
Absencia člena na zasadnutiach	30	2

Tab.1.: Identifikované riziká

Slabý manažment tímu

Slabý manažment alebo málo aktívny scrum master môžu spôsobiť v tíme nemalé problémy. Toto riziko môžeme spojiť s nedodržiavaním termínov alebo nedostatočnou angažovanosťou jednotlivých členov pracovať na projekte. S týmto problémom sa spája mnoho problémov a preto sme tomuto riziku priradili hodnotu závažnosti 9. Avšak v našom tíme sme pravdepodobnosť vzniku tohto rizika vyhodnotili na 20 % a zvolili si správneho scrum mastera.

Neporozumenie zadanej práce

Ďalším rizikom je určite neporozumenie zadanej úlohy. Priradená úloha sa tak môže uberať iným smerom a nadobúdať nechcený charakter. S týmto problémom je spojené prekladanie a naťahovanie termínov. Toto riziko sme vyhodnotili so závažnosťou 6 a pravdepodobnosť vzniku tohto rizika sme vyhodnotili na 50 %. Preto je potrebné zadanie práce vždy odkonzultovať a porozumieť mu čo najviac. Taktiež neváhať a nebáť sa komunikovať.

Neschopnosť člena splniť požiadavku

Riziko neschopnosti člena splniť zadanú požiadavku alebo úlohu je spojené s preceňovaním svojich schopností alebo s nedostatočnou komunikáciou. Toto riziko sme vyhodnotili so závažnosťou 7 a pravdepodobnosť vzniku tohto rizika sme vyhodnotili na 30 %. Taktiež, ako aj v predošlom riziku neporozumenia zadanej práce je riešením potreba lepšej komunikácie a lepšieho zadania danej práce správnym členom tímu.

Zmena rozsahu a cieľa práce

Riziko zmeny rozsahu a cieľa práce považujeme za vážny problém so závažnosťou 8, no pravdepodobnosť vzniku tohto problému je v našom projekte takmer nulová s hodnotou 5.

Nedostatok znalostí

Riziko nedostatku znalostí je spojené s nedostatočnou pripravenosťou pre prácu na danom projekte a s nedostatočnou analýzou. Preto sme sa v našom projekte zamerali na analyzovanie potrebných znalostí na základe študovania už naprogramovaného kódu. Taktiež pre toto riziko zohráva rolu aj pasivita jednotlivých členov a záujem o danú problematiku. Závažnosť rizika sme vyhodnotili hodnotou 3 a pravdepodobnosť vzniku hodnotou 20 %. Prijali sme opatrenia pre motiváciu jednotlivých členov v podobe zasadnutí a konzultácie o projekte.

Nevhodné obsadenie pozícií v tíme

Nevhodné obsadenie jednotlivých pozícií v tíme predstavuje riziko v podobe nevyužitia znalostí a vedomostí jednotlivých členov. Tento problém sme vyhodnotili so závažnosťou v hodnote 4 a pravdepodobnosť vzniku tohto problému sme vyhodnotili na hodnotu 20 %. Na začiatku projektu bolo preto potrebné analyzovať jednotlivých členov tímu a ich schopnosti a na základe tohto im priradiť jednotlivé roly.

Odchod člena z tímu

Odchod člena z tímu predstavuje závažný problém najmä vo fáze rozpracovaného projektu, kedy je už práca rozdelená a rozdelenie zodpovedá časovému plánu pre dokončenie projektu. Preto sme závažnosť vyhodnotili hodnotou 8 s pravdepodobnosťou vzniku 10 %.

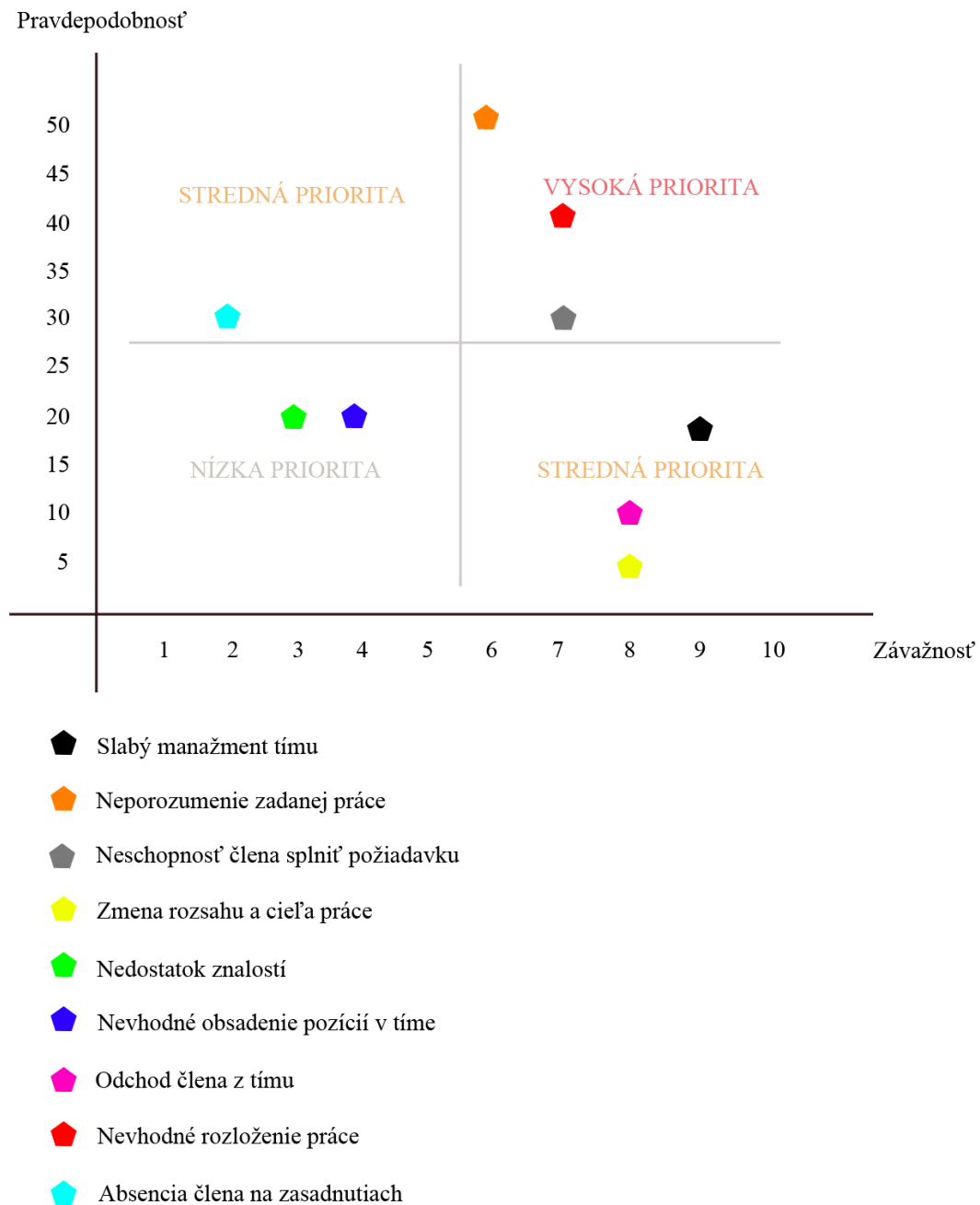
Nevhodné rozloženie práce

Dôležitou úlohou v tíme je vhodné rozloženie práce pre jednotlivých členov so splniteľným časovým plánom. Rozdelenie sa týka taktiež schopností členov tímu. Pravdepodobnosť vzniku tohto problému sme určili hodnotou 40 % a závažnosť 7. Preto tu hrá významnú úlohu správny scrum master alebo manažment tímu.

Absencia člena na zasadnutiach

Každý člen tímu by sa mal aktívne zúčastňovať stretnutí pre riešenie projektu, kde prebieha SCRUM. Mal by sa neustále zúčastňovať diskusií o jednotlivých problémoch, na ktoré narazil, poprípade pomohol iným.

Stanovenie priority rizika



Obr.2.: Priority rizika

Vypracoval: Daniel Lukáč

Príloha H: Metodika commitovania a verziovania