

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

# Just18

Dokumentácia k riadeniu projektu

Vedúci tímu: Ing. Peter Kapec, PhD.

Členovia tímu: Bc. Martin Gašpar

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Peter Marušin

Bc. Bence Ligárt

Bc. Miloslav Slížik

Bc. Marek Škriečka

Akademický rok: 2017/2018

# Obsah

1	Úvod .....	4
2	Členovia tímu .....	5
2.1	Manažérske úlohy.....	6
2.2	Vývojárske úlohy.....	6
2.3	Podiel práce na častiach inžinierskeho diela .....	7
3	Aplikácie manažmentov .....	8
3.1	Manažment dokumentácie.....	8
3.2	Manažment komunikácie a plánovania úloh.....	9
3.3	Manažment verziovacieho systému a prehliadok kódu .....	10
3.4	Nástroj na manažment úloh v tíme .....	12
4	Sumarizácia šprintov .....	14
4.1	Úvodné stretnutie.....	14
4.2	Šprint č.1 .....	14
4.3	Retrospektíva šprintu č. 1.....	14
4.4	Šprint č. 2 .....	15
4.5	Retrospektíva šprintu č.2.....	15
4.6	Šprint č. 3 .....	15
4.7	Retrospektíva šprintu č. 3.....	16
5	Big Picture .....	17
5.1	Motivácia pre vizualizáciu dát v podobe grafov .....	17
5.2	Ciele projektu 3DSoftviz .....	17
6	Prílohy .....	19
6.1	Export úloh TFS .....	19
6.1.1	Šprint č.1.....	19
6.1.2	Šprint č.2.....	19
6.1.3	Šprint č.3.....	20
6.2	Zápisnice zo stretnutí.....	20
6.2.1	Zápisnica zo stretnutia č. 1 .....	20
6.2.2	Zápisnica zo stretnutia č. 2 .....	22
6.2.3	Zápisnica zo stretnutia č. 3 .....	23

6.2.4	Zápisnica zo stretnutia č. 4 .....	24
6.2.5	Zápisnica zo stretnutia č. 5 .....	25
6.2.6	Zápisnica zo stretnutia č. 6 .....	26
6.2.7	Zápisnica zo stretnutia č. 7 .....	27
6.2.8	Zápisnica zo stretnutia č. 8 .....	29
6.3	Motivačný dokument.....	30
6.4	Metodiky.....	32
6.4.1	Gitflow metodika .....	32
6.4.2	TFS metodika .....	34

# 1 Úvod

Predkladaná dokumentácia popisuje prácu na tímovom projekte s názvom Vizualizácia informácií v rozšírenej realite. Práca je náplňou predmetu Tímový projekt, ktorého hlavnou náplňou je naučiť absolventov pracovať v tíme na spoločnom projekte. Okrem programátorských zručností si teda máme na predmete osvojiť aj prácu s kolaboratívnymi nástrojmi, spôsoby delby práce, manažment úloh či odhad ich náročnosti.

Vedúcim tímu je Ing. Peter Kapec PhD., ktorý je zároveň aj tvorcom a mentorom všetkých študentov pracujúcich na projekte 3DSoftviz. Na projekte sa snažíme pracovať podľa pravidiel agilnej metodiky Scrum. Vedúci tímu plní v rámci predmetu rolu product ownera a jeho hlavnou úlohou je sprostredkovať vývojárom svoje požiadavky na funkcionality produktu. Dĺžka jedného šprintu je dva týždne.

Úlohy, ktoré sme si rozdelili v kontexte riadenia projektu či jeho vývoja, sú popísané v nasledujúcej kapitole. Dokument tiež obsahuje sumarizáciu jednotlivých šprintov, metodiky používané pri práci na projekte, aplikácie manažmentov v rámci tímovej práce a zápisnice zo spoločných stretnutí tímu.

Kapitola	Autor
Úvod, manažment dokumentácie, manažment verziovaného systému a prehľad kódu, zápisnice zo stretnutí, Big picture	Peter Marušin
Členovia tímu, manažment komunikácie, nástroj na evidenciu úloh, export šprintov, formát	Marek Škriečka

## 2 Členovia tímu

V tejto kapitole sú popísané jednotlivé manažérske a vývojárske úlohy členov tímu. Na jednotlivých úlohách každého člena sme sa dohodli na 1. stretnutí a až po koniec 3. šprintu sme ich nemenili. Niektoré úlohy však počas semestra a práce na projekte pribudli.

### **Bc. Marek Škriečka**

Marek bol prvé tri šprinty Scrum Master, staral sa o plnenie úloh, manažoval úlohy v TFS a facilitoval stretnutia. Vytvoril a spravuje webovú prezentáciu tímu. Z pohľadu vývoja produktu sa zaoberal spoznaním projektu na platforme Windows Subsystem for Linux.

### **Bc. Peter Marušin**

Peto je zodpovedný za dokumentáciu a tiež spravuje git repozitár projektu. Písal zápisnice počas stretnutí tímu. Udržiaval Slack, vytváral či revidoval dokumentáciu a snažil sa oboznámiť nielen s dokumentačnými nástrojmi využívanými v rámci projektu. Na projekte tiež plní úlohu macOS integrátora a spávcu git repozitára, dohliada na správnosť commitov a údržbu branches.

### **Bc. Bence Ligárt**

Bence je windows integrátor a tester. Z tímu len dvaja členovia: Bence a Marek pracujú na operačnom systéme windows, a keďže Marek sa rozhodol pracovať so systémom WSL, s ktorým staršie tímy sa doteraz nezaoberali, jedine Bence sa hodil na túto pozíciu. Úloha windows integrátora zahŕňa: testovanie kompatibility implementovaných častí na operačnom systéme windows, podpora ovládačov.

### **Bc. Martin Gašpar**

Martin sa spolupodieľa na vývoji a testovaní pre platformu macOS. Pomáha pri vytváraní a manažovaní úloh cez nástroj TFS

### **Bc. Miloslav Slížik**

Milo sa spolupodieľa na vývoji a testovaní pre platformu Linux a spravuje tímový server.

### **Bc. Tomáš Krupa**

Tomáš sa podieľa na refaktorizácii a automatizácii zostavovania zdrojového kódu. Primárnym zameraním je C++ a CMake na platforme Linux.

### **Bc. Michal Knapík**

Michal je OS X integrátor, podieľa sa na správe projektu z pohľadu zostavovacieho nástroja cmake a na zjednodušení jeho používania vrámci projektu.

## **2.1 Manažérske úlohy**

V nasledujúcej tabuľke sú uvedené jednotlivé manažérske úlohy s uvedením členov, ktorí ich zastávali:

Úloha	Člen
Web vývojár	Marek Škriečka, Martin Gašpar
Server administrátor	Miloslav Slížik
Scrum Master	Marek Škriečka
Dokumentácia	Peter Marušin
Git Master	Peter Marušin
Slack administrátor	Peter Marušin

## **2.2 Vývojárske úlohy**

V nasledujúcej tabuľke sú uvedené jednotlivé vývojárske úlohy s uvedením členov, ktorí ich zastávali:

Úloha	Člen/Členovia
-------	---------------

MacOs integrátor	Peter Marušin, Martin Gašpar, Michal Knapík
Windows integrátor	Bence Ligárt
Windows Subsystem For Linux integrátor	Marek Škriečka
Linux integrátor	Tomáš Krupa, Miloslav Slížik

## 2.3 Podiel práce na častiach inžinierskeho diela

V tabuľke je uvedený podiel členov tímu na inžinierskom diele v zimnom semestri:

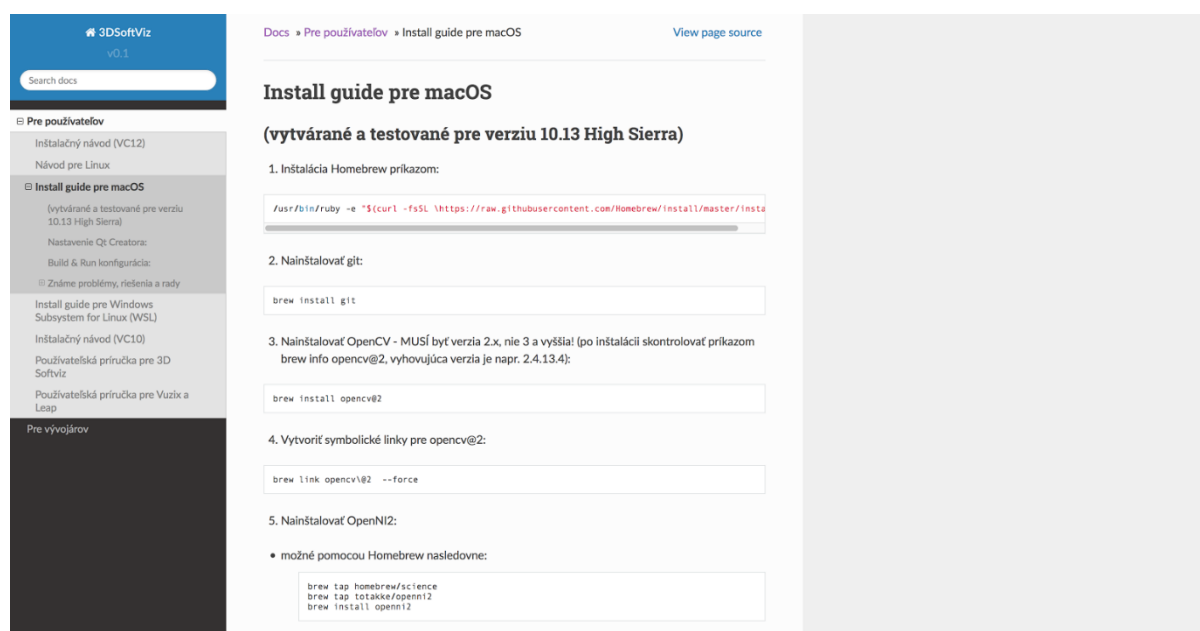
Časť diela	Autor/Autori
Build projektu na platforme WSL	Marek Škriečka
Webová prezentácia	Marek Škriečka
Aktualizácia použ. príručky	Peter Marušin, Marek Škriečka
Build projektu na platforme macOS	Martin Gašpar, Michal Knapík, Peter Marušin
Build projektu na platforme Linux	Tomáš Krupa, Miloslav Slížik
Revízia dokumentácie	Peter Marušin
Správa tímového servera	Miloslav Slížik
Vagrant	Miloslav Slížik
RefaktORIZÁCIA CMake	Michal Knapík, Tomáš Krupa

## 3 Aplikácie manažmentov

### 3.1 Manažment dokumentácie

Súčasťou práce na projekte bolo aj vytváranie, resp. údržba už existujúcej dokumentácie. Keďže sa jedná o roky existujúci projekt, snažili sme sa pokračovať v zabehnutých metódach dokumentovania, či už sa jedná o dokumentáciu pre vývojárov alebo používateľov 3DSoftvizu.

Používateľský manuál, metodiky, inštalačné manuály a ďalšia ručne písaná dokumentácia sú vytvárané pomocou nástroja Sphinx. Jedná sa nástroj napísaný v Pythone používaný Python komunitou, ktorý dokáže z RST dokumentov vygenerovať dokumentáciu napríklad vo formátoch HTML, PDF či LaTeX.



Obrázok 1: Ukážka vygenerovaného dokumentu pomocou nástroja Sphinx vo formáte HTML

Ďalším nástrojom používaným na generovanie UML diagramov je PlantUML. Ten vytvára z textových súborov so zadanými entitami a vzťahmi medzi nimi vo vlastnom application-specific jazyku plnohodnotné UML diagramy vo formátoch .svg či .png. Ku generovaniu UML diagramov v projekte dochádza v rámci generovania Sphinx dokumentácie.



Ďalším nástrojom používaným v projekte je Doxygen, ktorý slúži na automatické generovanie dokumentácie priamo zo zdrojového kódu. Ten plánujeme viac používať v letnom semestri, keďže v zimnom sme sa venovali skôr analýze ako tvoreniu novej funkcionality

Na začiatku semestra sme sa snažili analyzovať všetku existujúcu dokumentáciu. Mnoho materiálov si vyžadovalo, prípadne stále vyžaduje aktualizáciu a revíziu, keďže na projekte sa neustále pracovalo a pribúdala nová funkcionality.

Na začiatku semestra sme sa rozhodli pre vytvorenie nových inštalačných manuálov osobitne pre každú platformu, keďže podľa už vytvorených sa nám často nepodarilo nakonfigurovať vývojárske prostredie. Nové inštalačné manuály sú súčasťou dokumentácie k produktu.

Nástroj PlantUML bol pridaný ako závislosť 3DSoftvizu a je distribuovaný ako súčasť repozitára (nie je potrebné jeho dodatočné získavanie). Po prejení prác venovaných rozšíreniu 3DSoftvizu bol aktualizovaný používateľský manuál. Dokumentácia je a počas zvyšku tímového projektu plánuje byť priebežne revidovaná a aktualizovaná.

## 3.2 Manažment komunikácie a plánovania úloh

Najväčšia časť komunikácie prebieha cez stretnutia, ktoré máme každý štvrtok od 10:00 do 13:00. Priebeh stretnutí facilituje Scrum Master.

V úvode je stretnutie je stand up, kedy každý člen tímu povie, čo od predchádzajúceho stretnutia spravil a s čím mal problém.

Ak je stretnutie v čase dokončenia šprintu na stretnutí sa ešte prezentuje retrospektíva, pričom každý člen tímu má možnosť vyjadriť sa k jej jednotlivým bodom.

Stretnutie pokračuje buď skupinovú alebo individuálnu konzultáciou problémov s vedúcim tímu, na ktoré každý člen pri plnení úloh narazil.

Ak stretnutie prebieha v čase, kedy sa začína nový šprint, v tíme diskutujeme a podľa priorit vyberáme úlohy z nástroja na manažment úloh, ohodnocujeme ich náročnosť a prideliujeme ich členom.

Retrospektívu vytvárame deň pred stretnutím ukončenia šprintu, kedy sa stretneme. Diskutujeme, čo sme spravili za daný šprint dobre, v čom určite budeme pokračovať a naopak, kde sme spravili chybu a čoho sa v ďalšom šprinte vyvarovať. Výstup z retrospektívy má formu bodov 3 bodov: *start doing*, *stop doing*, *continue doing*, ktoré sú potom na stretnutí prezentované.

V tíme je potrebné komunikovať aj online. Na to používame nástroj Slack. Tento nástroj umožňuje komunikovať s členmi tímu buď individuálne alebo formou vlákien, ktoré sú vytvorené pre určitý okruh tém diskutovania. V tíme sú najviac využívané nasledujúce kanály:

- **git** - tu diskutujeme problémy pri verziovaní zdrojového kódu. V kanáli sú tiež integrované notifikácie z GitHub-u oznamujúce zmeny v repozitári projektu.
- **build** - na tomto kanáli komunikujeme chyby, ktoré sa vyskytli pri builde projektu
- **tfs-scrum** - tento kanál slúži na komunikáciu ohľadom nástroja tfs, v ktorom spravuje tok práce

### 3.3 Manažment verziovacieho systému a prehliadok kódu

Pri vývoji je používaný verziovací systém git a projekt je uložený vo vzdialenom repozitári na GitHub-e. Keďže sa jedná o projekt s už zabehnutými pravidlami vývoja z hľadiska vytvárania branches, označovania commitov či mergovania novovytvorenej funkcionality do spoločných branches, náš tím si osvojil už zaužívané praktiky s minimálnymi modifikáciami. Tie sú spísané v GitFlow metodike, ktorá je prílohou tohto dokumentu.

V projekte sú udržiavané dve hlavné branches:

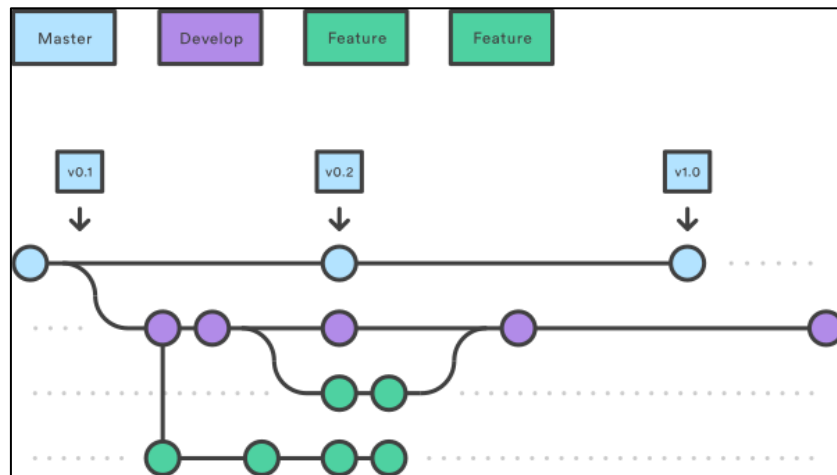
- **master** - hlavná branch obsahujúca otestovaný funkčný kód, v prípade release-u použiteľný. V čase nášho príchodu do projektu bol master už dlhšiu dobu neaktuálny pre množstvo úprav potrebných v existujúcom kóde na develop-e.
- **develop** - spoločná pracovná branch pre všetkých vývojárov. Do nej sa mergujú všetky zmeny vykonané vo vedľajších branches. Mergnutiu zmeny do develop branch musí ale predchádzať code review aspoň od jedného člena tímu (dôležité hlavne pri zmenách kódu či build procesu 3DSoftvizu). Zmeny sa mergujú prostredníctvom pull requestov. Ku prehliadkam kódu dochádzalo v zimnom semestri tiež formou pridelenia reviewers na daný pull request, čo je možnosť poskytovaná GitHub-om.

Okrem dvoch hlavných používame ešte vedľajšie branches, bližšie popísané v metodike:

- **feature** - nová funkcionality
- **hotfix** - oprava chýb

Vedľajšie branches momentálne aj po mergnutí ostávajú v projekte, nemažú sa. Dôvodom je ľahšia možnosť návratu v prípade znefunkčnenia niektorej hlavnej branch a tiež ich malý počet (ich množstvo zatiaľ výraznejšie nesťažuje orientáciu v projekte).

Minulé tímy vykonávali prehliadky kódu pomocou nástrojov Cppcheck, CppLint a AStyle. My sme sa rozhodli pre pridelenie reviewers na GitHub-e, keďže väčšina práce vykonanej v ZS sa netýkala priamo zásahov do kódu. No vyššie spomínané nástroje chceme určite používať v letnom semestri.



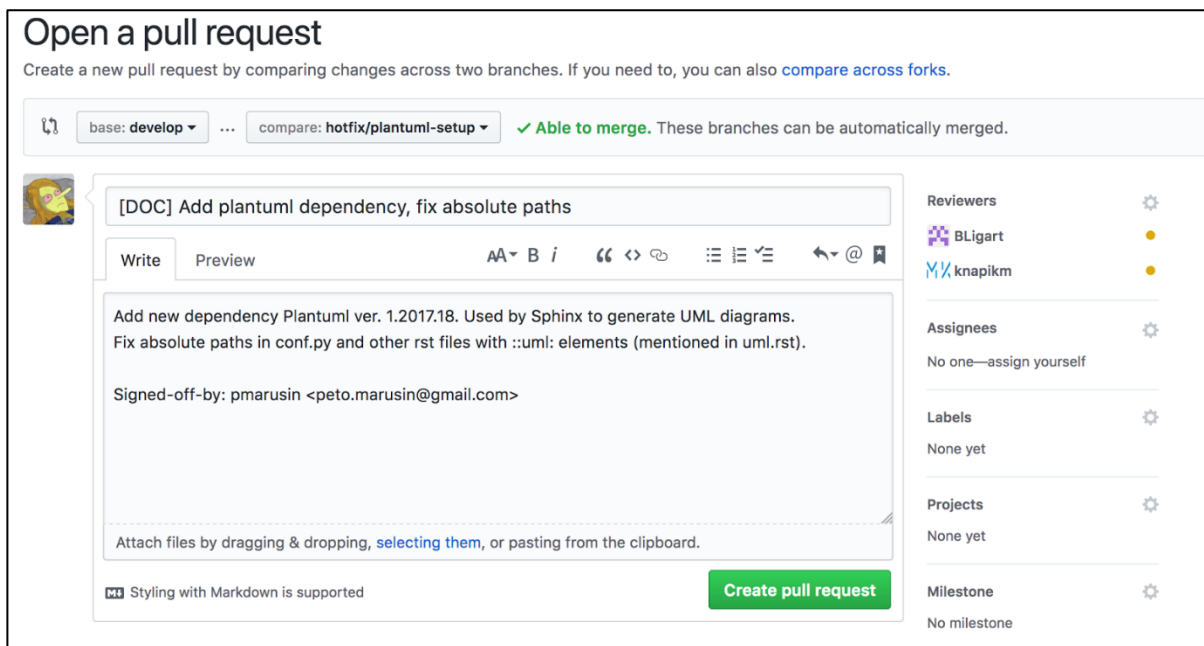
Obrázok 2: Ukážka vytvárania nových branch podľa Git metodiky<sup>1</sup>

**Workflow** členov tímu pri práci bol nasledovný:

1. Je mi pridelený task z backlog-u, čo zistím v TFS
2. Pull-nem si z zo vzdialeného repozitára develop, čím sa uistím, že mám u seba aktuálnu vetvu.
3. Vytvorím si novú branch s názvom *hotfix/nazov\_vetvy*, resp. *feature/nazov\_vetvy* podľa toho, na čom idem v danej vetve pracovať.
4. Prepnem sa do danej vetvy
5. Vykonám zmeny, ktoré sú pripravené na commit.
6. Zmenené súbory pridám do staging area.
7. Commitnem zmeny, pričom pri písaní commit message sa pridrižiam GitFlow metodiky - commit message je deskriptívna, stručná a obsahuje vhodný tag označujúci, čoho sa týka daný commit.
8. Zmeny pushnem do vedľajšej vetvy.
9. Vytvorím pull request požadujúci merge s develop-om. Pri vytváraní pull requestu mu zadám reviewers, ktorí musia mnou vykonané zmeny skontrolovať a otestovať ich funkčnosť. Snažím sa vytvárať osobitné pull requesty pre rôzne druhy úprav v projekte, napr. osobitný pull request pre modifikáciu dokumentácie, pre novú feature

<sup>1</sup> <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

či hotfix. Ostatní členovia tímu sú notifikovaní na Slacku, prípadne mailom.



Obrázok 3: Vytváranie žiadosti o pull request s pridaniami kontrolórmí

10. Ak je pull request schválený všetkými reviewers, môže byť vykonaný merge s develop-om. Ak nie, musím commitnúť zmeny, ktoré opravujú prípadné nedostatky. Merge pritom môžem vykonať sám alebo ktokoľvek ďalší, kto zbadá, že pull request prešiel prehliadkou. Ostatní členovia tímu sú notifikovaní na Slacku, prípadne mailom.
11. V TFS označím task za splnený.

### 3.4 Nástroj na manažment úloh v tíme

Pre manažment úloh v tíme sme používali Microsoft Team Foundation Server (TFS). Tento nástroj umožňuje nastaviť rôzne šablóny pre spravovanie toku práce. Náš tím si vybral šablónu Scrum. V tejto šablóne je možné spravovať úlohy a vývoj celého produktu v nasledujúcej hierarchii:

- **EPICS** - táto časť združuje informácie o komplexnej funkcionalite časti produktu, ktorá je pre lepšiu organizovanosť rozdelená na niekoľko častí nazývaných FEATURES
- **FEATURES** - táto časť umožňuje zjednotiť viacero podobných BACKLOG ITEMS
- **BACKLOG ITEMS** - pozostávajú z úloh, ktoré sú podobné a súvisia s podobnou funkcionalitou. Na tejto úrovni tiež určujeme náročnosť kladenú na úspešné ukončenie všetkých podúloh, ktoré s daným BACKLOG ITEM-om súvisia.
- **TASKS** - na tejto úrovni sa nachádzajú jednotlivé úlohy, ktoré si určujeme pre úspešné beh celého vývoja produktu. Každá úloha má svojho autora, ktorý si ju spravuje a

nastavuje jej proces riešenia. Príklad hierarchie sa nachádza na nasledujúcom obrázku:

3	Epic	▼ 🏰 Support all plaforms	● In Progress		Business
	Feature	▼ 🏆 Provide development environment on all supported platf...	● In Progress		Business
	Product Backl...	> 📄 [LINUX] Project setup	● Done	20	Business
	Product Backl...	> 📄 [WINDOWS] Project setup	● Committed	20	Business
+	Product Backl...	▼ 📄 [WSL] Project setup	*** ● Done	20	Business
	Task	📄 Installation of WSL	● Done		
	Task	📄 Installation of dependencies 3DSofTViz and IDE	*** ● Done		
	Task	📄 Build and run project	● Done		
	Task	📄 Install guide	● Done		
	Product Backl...	> 📄 [macOS] Project setup	● Done	20	Business

Obrázok 4: Ukážka hierarchie úloh v nástroji TFS

## 4 Sumarizácia šprintov

### 4.1 Úvodné stretnutie

Na úvodnom stretnutí sme diskutovali rozdelenie manažérskych úloh v tíme, voľby nástrojov pre manažovanie toku práce a tiež webovú prezentáciu tímu. Vytvorili sme nasledujúce úlohy – správa git repozitára, správa servera, osoba zodpovedná za dokumentáciu a reporty a správcu webovej stránky.

Na komunikácií sme sa zhodli na mailing liste a Slack a ako manažment pracovného toku sme zvolili Microsoft Team Foundation Server.

### 4.2 Šprint č.1

V šprinte č. 1 sme si stanovili za prioritný cieľ build projektu a jeho spustenie na s najnovšími knižnicami a dostupnými závislosťami na platforme Windows, MacOS a Unix. Medzi úlohami tiež figurovalo vytvorenie webovej prezentácie tímu a nastavenie prideleného servera.

Rozbehnutie projektu sa na niektorých platformách nepodarilo alebo podarilo len so starými knižnicami a závislosťami. Príčinou bolo nesprávne ohodnotenie úsilia potrebného pre dané úlohy spustenia projektu. V ďalšom šprinte č. 2 sme si preto tieto úlohy rozčlenili na menšie podúlohy typické pre každý operačný systém a prísnejšie ohodnotili úsilie rozbehnutia projektu na jednotlivých platformách. V šprinte č. 1 sa nám však podarilo vytvoriť webovú prezentáciu tímu.

### 4.3 Retrospektíva šprintu č. 1

Medzi diskutované vylepšenia, ktoré by sme mali začať dodržiavať patrilo väčšia viditeľnosť procesov – veľa členom tímu unikali informácie, keďže sa šírili len individuálne a neboli nikdy publikované. S tým súvisí aj ďalšie vylepšenie a to komunikácia o problémoch s projektom výhradne vo verejných vláknoch, ktoré Slack poskytuje alebo cez mailing list. Uľahčenie manažmentu práce tiež môže vylepšiť správne značenie úloh v TFS podľa metodiky.

Medzi praktiky, ktoré by sme mali prestať robiť patrilo súkromná komunikácia o problémoch v projekte a celkovo tímu. Medzi zlé praktiky patrilo aj spomínané zle ohodnotené úlohy voči úsiliu a nedostatočne rozčlenenie na podúlohy.

V retrospektíve sme tiež usúdili, že by sme mali pokračovať v kladení otázok buď vedúcemu projektu alebo aj jednotlivým členom tímu, pretože nám to výrazne pomáha.

## 4.4 Šprint č. 2

V šprinte č. 2 bolo hlavným cieľom dokončiť prepracované úlohy zo šprintu č. 1, ktoré sa týkali spustenia projektu na každej platforme s dohodnutými knižnicami. Výstup bolo treba prezentovať formou aktualizovaného manuálu na inštaláciu projektu. Úlohou tiež bolo spraviť export a oprava chýb, ku ktorým sme počas spúšťaniu projektu prišli, tak aby ich bolo možné zlúčiť so zdrojovým kódom projektu. Samostatnou úlohou tiež bolo prejsť minuloročné diplomové a bakalárske práce, výstupy tímového projektu a aktualizovať podľa používateľskú príručku.

Rozbehanie projektu sa úspešne podarilo na všetkých platformách okrem operačného systému Windows, kde pretrvávali rôzne chyby. Všetky inštalčné príručky boli aktualizované a boli opravené chyby pri builde projektu. Aktualizovaná bola tiež používateľská príručka.

## 4.5 Retrospektíva šprintu č.2

Medzi praktiky, ktoré by sme mali zaviesť bola diskutovaná téma ohľadom práce na úlohách s pridanou hodnotou pre projekt – refaktORIZÁCIA, optimalizácia alebo úlohy, ktoré by pridávali novú funkcionálnosť systému. Potrebne je tiež dávať si pozor pri písaní zdrojového kódu na white-spaces. Diskutované tiež bolo lepšia responzivita na pull requesty a potvrdzovanie funkčnosti zmien.

Medzi dobré praktiky, v ktorých chceme pokračovať patrí spolupráca – riešiť problém vo dvojiciach a lepšie značenie toku práce v TFS.

## 4.6 Šprint č. 3

V 3. šprinte bolo treba nutne dokončiť Windows build, keďže nebol v 2. šprinte úspešný. Pri súčasnej konfigurácii je build vykonaný úspešne avšak aplikácia padne pri spustení. V novom šprinte je potrebné spustiť projekt s aktualizovanými verziami debugger-a a 64-bitovými verziami závislostí. V šprinte bola tiež naplánovaná analýza knižnice OpenPose, ktorá sa zaoberá detekciou kľúčových bodov ľudského tela, refaktORIZÁCIA súboru Cmake, ktorý je používaný pri build-ovaní projektu a vytvorenie prostredia pre CI. Významnou úlohou v 3. šprinte bolo tiež vytvorenie dokumentácie ku kontrolnému bodu č.1.

Rozbehanie na platforme Windows sa nepodarilo, zistilo sa, že s novým debugger-om pravdepodobne nebude build možný a tak je treba skúsiť build s pôvodným na inej stanici. Problém bol tiež s úlohami ohľadom vytvorenia prostredia pre CI, dôvodom bol nedostatok času. Ostatné úlohy sa podarilo splniť.

## 4.7 Retrospektíva šprintu č. 3

V šprinte č.3 boli zistené nedostatky ohľadom plnenie malých úloh , ktoré trvajú max. 2 minúty (napr. doplniť si niečo alebo zmeniť v TFS). Členovia si ich často nechávajú na neskôr a potom sa na ne zabúda. Vylepšenie bolo navrhnuté tak, že ak niekto dostane takýto typ úlohy musí ju bezodkladne vyriešiť. Chyba bola taktiež v komunikácií, bolo navrhnuté aby niektorí členovia dávali na správy dávali vždy odozvu. Navrhnuté bolo tiež začať pracovať na novej funkcionalite projektu a menovať branch-e podľa čísla úloh z TFS. Na týchto zlepšeniach sme sa zhodli.

Medzi zlé praktiky patrilo miešanie slovenského a anglického jazyka v commit správach a tiež, že niektorí členovia plnili úlohy za iných, bez ich súhlasu alebo oznámenia.

Praktiky, ktoré sa nám overili bola git-flow metodika vo forme vetiev, ktorú sme začali v 3. šprinte naplno využívať.



## 5 Big Picture

V tejto kapitole opisujeme projekt 3DSoftviz, na ktorom sme sa podieľali v rámci tímového projektu a predstavuje náš produkt. Budeme sa snažiť opísať 3DSoftviz z hľadiska možného využitia, špecifikovať cieľového zákazníka a opísať zámery jeho tvorcov v čase vzniku projektu. Tiež sa budeme snažiť systém popísať z technického hľadiska - aktuálne poskytovaná funkcionálnosť, podpora viacerých periférnych zariadení či väčšie zmeny vykonané predošlými študentami v rámci svojich diplomových a bakalárskych prác či tímových projektov.

### 5.1 Motivácia pre vizualizáciu dát v podobe grafov

Človek je v dnešnej dobe zahltený čoraz väčším množstvom informácií a je pre neho obtiažne si dané informácie zapamätať. Z nedávnych poznatkov je zrejmé, že vizualizácia informácií nám pomáha si ich ľahšie zapamätať. Je to spôsobené zapojením oboch hemisfér mozgu narozdiel od zapojenia iba ľavej hemisféry pri spracovaní informácií vo forme čísel alebo písaného textu.

Ďalším dôležitým faktorom je charakter dnešných dát. Tie bývajú často multidimenziálne, pričom pod dimenziou chápeme počet atribútov, ktoré majú jednotlivé položky datasetu. Často nás zaujímajú iba niektoré dimenzie a tie ostatné vnímame ako šum, ktorý nám sťažuje vnímanie vlastností a vzťahov v datasete. Neprehľadnosť môže byť tiež umocnená veľkosťou analyzovaných dát. V takom prípade tiež výrazne pomôže vizualizácia dát vo forme grafu. Tá nám umožní vnímať vzťahy či vzory, ktoré predtým neboli viditeľné.

### 5.2 Ciele projektu 3DSoftviz

3DSoftviz je už niekoľko rokov vyvíjaný softvér na Fakulte informatiky a informačných technológií STU. Na začiatku bol vytvorený s úmyslom stať sa general-purpose nástrojom na vizualizáciu informácií pomocou grafových štruktúr. Neskôr sa zamerlal na vizualizáciu softvéru. Napriek neskoršiemu vymedzeniu sa je ho ale po pomerne malých modifikáciách možné prispôbiť na vizualizáciu informácií aj z iných oblastí.

Aktuálne sa využíva na vizualizáciu statických aspektov softvéru vo forme orientovaných či neorientovaných grafov. Datasets s informáciami o uzloch a hranách dokáže načítať z XML súboru, databázy či git repozitára. Okrem iného je tiež možné pridávanie uzlov či hrán, viacero možností rozmiestnenia uzlov grafu, manipulácia s grafom pomocou myši, zhlukovanie uzlov či vizualizácia evolúcie softvéru - zmien softvéru v čase zachytených v trojrozmernom priestore.

Projekt tiež obsahuje moduly pre spoluprácu s nižšie uvedenými periférnymi zariadeniami:

- Kinect

- 3D myš 3DConnexion
- AR Okuliare Vuzix STAR 1200XL
- Okuliare Nvidia 3D Vision Pro
- Leap senzor

## 6 Prílohy

### 6.1 Export úloh TFS

#### 6.1.1 Šprint č.1

Project: Just18 Server: tfs.fiit.stuba.sk\StudentsProjects Query: Just18 Team - Sprint 1 - Backlog_new List type: Tree						
ID	Work Item Type	Title 1	Title 2	State	Effort	Assigned To
6466	Product Backlog Item	Run and build project on all platforms		Done	10	
6437	Task		Run and build on macOS	Done		Bc. Peter Marusin
6359	Task		Run and build on UNIX	Done		Tomas Krupa
6491	Task		Run and build on windows	Done		Bc. Bence Ligart
6725	Task		Run and build on WSL	Done		Bc. Marek Skriecka
6671	Product Backlog Item	Team website		Done	3	
6660	Task		Creation of web presentation of tea	Done		Bc. Marek Skriecka
6672	Task		Setup of server for web presentation	Done		Bc. Miloslav Slizik
6727	Task		Add images of team members	Done		Bc. Martin Gaspar

Obrázok 5: Export úloh zo šprintu 1

#### 6.1.2 Šprint č.2

Project: Just18 Server: tfs.fiit.stuba.sk\StudentsProjects Query: Just18 Team - Sprint 2 - Backlog List type: Tree						
ID	Work Item Type	Title 1	Title 2	State	Effort	Assigned To
6764	Product Backlog Item	Make export with all bugs		Done		
6765	Task		[WSL] Add errors	Done		Bc. Marek Skriecka
6766	Task		[macOS] Add erros to doc	Done		Bc. Peter Marusin
6767	Task		[LINUX] Add erros	Done		Bc. Miloslav Slizik
6763	Product Backlog Item	[LINUX] Project setup		Done	20	Bc. Miloslav Slizik
6775	Task		Installation of Dependencies and IDI	Done		Bc. Miloslav Slizik
6776	Task		Succesfully build and run project	Done		
6777	Task		Install guide	Done		
6762	Product Backlog Item	[WINDOWS] Project setup		Committed	20	Bc. Bence Ligart
6798	Task		Installation of Microsoft Visual Stud	Done		Bc. Bence Ligart
6799	Task		Installation of dependencies and IDI	Done		Bc. Bence Ligart
6741	Product Backlog Item	[WSL] Project setup		Done	20	
6742	Task		Installation of WSL	Done		Bc. Marek Skriecka
6743	Task		Installation of dependencies 3DSoftvi	Done		Bc. Marek Skriecka
6744	Task		Build and run project	Done		
6745	Task		Install guide	Done		Bc. Marek Skriecka
6746	Product Backlog Item	[macOS] Project setup		Done	20	
6747	Task		[macOS 10.13] Install dependencies	Done		Bc. Peter Marusin
6748	Task		[macOS 10.13] Set up build & run en	Done		Bc. Peter Marusin
6673	Product Backlog Item	Update user manual		Done	3	
6652	Task		Remake user guide - bachelor thesis	Done		Bc. Marek Skriecka
6674	Task		Remake user guide - diploma thesis	Done		Bc. Peter Marusin
6802	Bug	Build error: src/Mouse3d/LibMouse3d/Mac/Mou		Done		Bc. Peter Marusin

Obrázok 6: : Export úloh zo šprintu 2

### 6.1.3 Šprint č.3

Project: Just18 Server: tfs.fiit.stuba.sk\StudentsProjects Query: Just18 Team - Sprint 3 - Backlog List type: Tree					
ID	Title 1	Title 2	State	Effort Iteration Path	Assigned To
6886	Preparation for CI		Committed	8 \Sprint 3	Bc. Miloslav Slizik
		Setup Vagrant machine with Ansible provisioning - Ubuntu 16.04	Done	\Sprint 3	Bc. Miloslav Slizik
6887					
6888		Create Docker container	In Progress	\Sprint 3	Bc. Miloslav Slizik
		Write documentation for Docker, Vagrant and Ansible	Done	\Sprint 3	Bc. Miloslav Slizik
6938					
6934	OpenPose library (analysis)		Done	20 \Sprint 3	
6933		OpenPose - report	Done	\Sprint 3	Bc. Martin Gaspar
6935		OpenPose - perform test	Removed	\Sprint 3	Bc. Bence Ligart
6932	Document for assignment n. 1 (AIS)		Done	5 \Sprint 3	Bc. Peter Marusin
6981		Product description	Done	\Sprint 3	Bc. Peter Marusin
7101		Management description	Done	\Sprint 3	Bc. Marek Skriecka
7102		Architecture description	Done	\Sprint 3	Bc. Bence Ligart
6762	[WINDOWS] Project setup		Committed	20 \Sprint 3	Bc. Bence Ligart
6798		Installation of Microsoft Visual Studio	Done	\Sprint 2	Bc. Bence Ligart
6799		Installation of dependencies and IDE	Done	\Sprint 2	Bc. Bence Ligart
6800		Build and run project	In Progress	\Sprint 3	Bc. Bence Ligart
6801		Install guide	In Progress	\Sprint 3	Bc. Bence Ligart
6666	Refactor CMakeLists.txt		Committed	34 \Sprint 3	
		Split main CMakeLists.txt (subdirectories)	Done	\Sprint 3	Bc. Michal Knapik
6668					
6667		Loops for repeated commands	Done	\Sprint 3	Bc. Peter Marusin
6913		Add options to CMake	Done	\Sprint 3	Tomas Krupa

Obrázok 7: Export úloh šprint 3

## 6.2 Zápisnice zo stretnutí

### 6.2.1 Zápisnica zo stretnutia č. 1

**Téma stretnutia:** Úvodné stretnutie, rozdelenie úloh v tíme, voľba nástrojov, webstránka, vytvorenie spoločnej fotografie na plagát tímu

**Dátum:** 28. 10. 2017

**Čas:** 10:00-13:00

**Miestnosť:** 4.46

**Prítomní:**

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

### Priebeh stretnutia:

- Diskusia o vzhľade a obsahu webstránky tímu. Marek a Martin si vzali na starosti tvorbu webstránky. Marek si zapisoval pripomienky členov a pokyny zo stránky predmetu a vyžiadal si fotografie členov.
- V tíme sme si rozdelili nižšie vypísané roly. Na každú z nich sa dotýčný člen prihlásil sám. Správcu git repozitára sme zvolili, lebo predpokladáme, že pri vývoji je použitý git. Zistenie konkrétnych detailov bude náplňou ďalšieho stretnutia.
- správa git repozitára: Martin
- správa servera: Milo
- dokumentácia, reporty: Peťo
- správa webstránky: Marek
- Zhodli sme sa komunikácii prostredníctvom mailing listu, ako možnosť padol aj Slack. Je potrebné zistiť, prečo sa minulým tímom neosvedčil. Mailing list sme na stretnutí nevytvorili, konfiguráciu si zobral na starosti Mišo.
- Tvorbu tímového plagátu si zobral na starosti Bence. Počas stretnutia sme vytvorili spoločnú fotografiu tímu
- Zhodli sme sa na používaní systému TFS na SCRUM management. Oboznamovali sme sa spoločne so systémom.
- Reporty budú dostupné pre všetkých členov tímu na GDrive tímu a na jeho webe.

### Úlohy:

Do ďalšieho stretnutia bude potrebné vyriešiť nižšie uvedené úlohy:

Č. úlohy	Popis	Zodpovedný
1	Oboznámenie sa s TFS	všetci
2	Konfigurácia mailing listu	Mišo

3	Tvorba tímového plagátu	Bence
4	Oboznámenie sa s hostingom	Milo
5	Poslať fotografiu a bio správcovi websráanky	všetci
6	Začať s tvorbou webstránky	Marek, Martin

### 6.2.2 Zápisnica zo stretnutia č. 2

**Téma stretnutia:** Stretnutie s vedúcim tímu, definovanie cieľov projektu a motivácie tímu

**Dátum:** 5. 10. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

#### **Priebeh stretnutia:**

- Vedúci tímu nás oboznámil s prvotnými organizačnými pokynmi. Ako tím sme si stanovili ciele a odpovedali na 3 otázky:
  - a. Ako sme spokojní s projektom?
  - b. Čím by sme chceli my prispieť do projektu?
  - c. Čo očakávame, že sa ako tím naučíme?
- Migrácia na GitLab je v stave riešenia.

- Organizačné pokyny a rady ohľadom fungovania SCRUM-u a agilného vývoja. Prehodnotenie rozsahu dokumentácie. Dokumentácia by mala byť preusporiadaná a prečistená.
- Bude potrebné si definovať user stories/system stories (ohľadom podpory a rozbehnutia produktu na rôznych platformách).
- Projekt by mal byť funkčný na 3 podporovaných platformách (Windows, Linux, macOS).
- Možná system story - aktualizácia modulov a knižníc na najnovšie verzie (hlavne QT) a prispôbenie inštalačnej príručky. Taktiež aktualizované OpenCV má iné API ako nami používané (možný task pre budúcnosť).
- Zvážiť Vagrant & Ansible a ich konfiguráciu ako task pre prvý šprint. Doxygen - automatické generovanie dokumentácie.
- Prioritou pre začiatok by malo byť vytvorenie backlog-u.
- Dĺžka šprintu 2 týždne, stand-up každý týždeň.
- Retrospektíva - čo nám nešlo, čo nám šlo a s čím by sme vôbec mali začínať.

### Úlohy:

Do ďalšieho stretnutia bude potrebné vyriešiť nižšie uvedené úlohy:

Č. úlohy	Popis	Zodpovedný
1	Testovanie Linux subsystému (report)	Marek
2	Oboznámenie sa s Vagrantom (report)	Milo (a hocikto ďalší, kto má záujem a čas)
3	Oboznámienie sa s gitflow metodikou	všetci
4	Oboznámenie sa s Doxygen, Sphinx, štruktúrou dokumentácie (report)	Peťo (a hocikto ďalší, kto má záujem a čas)

### 6.2.3 Zápisnica zo stretnutia č. 3

**Téma stretnutia:** Vzájomná diskusia o problémoch s buildom projektu, spoločná práca

**Dátum:** 12. 10. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

**Priebeh stretnutia:**

- Snažili sme sa vzájomne si pomôcť s buildom projektu 3dsoftviz.
- Vedúci projektu nám radil ohľadom opráv v CMakeLists.txt
- Úspešný build sa počas stretnutia podaril viacerým (Milo a Peťo), no ako problém sa ukázal cotire, ktorý nie je pod stálym vývojom a neustále aktualizácie viacerých knižníc (aktualizácia knižnice Qt bola vydaná dve hodiny pred začiatkom stretnutia)
- Do budúceho stretnutia by bolo najlepšie, keby každý vedel projekt buildnúť a pustiť
- Pripravujeme sa na buducotýžňovú retrospektívu
- Bol zriadený Slack na urýchlenie komunikácie medzi členmi tímu
- Tasky sa budú nahadzovať už iba do TFS

#### 6.2.4 Zápisnica zo stretnutia č. 4

**Téma stretnutia:** Retrospektíva, stand-up, vytvorenie nového backlogu

**Dátum:** 19. 10. 2017

**Čas:** 10:00 - 13:00



**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

**Priebeh stretnutia:**

- Stretnutie začalo retrospektívou, scrum-master (Marek) menežoval priebeh retrospektívy.
- Marek splnil task zostavenia a spustenia projektu na platforme Windows s Windows Subsystem for Linux. Projekt beží s QT5, testy fungujú.
- Peťo projekt zostavil a spustil na platforme macOS, verzii 10.13 (High Sierra). Vizualizácie grafov však nemajú hrany.
- Milo zostavil predpripravené vývojové prostredie pre 3dsoftviz vo Vagrant, projekt beží s QT 4.
- Tomáš projekt rozbehal vo virtuálnej mašine na Linuxe.
- Hlavnou náplňou stretnutia bolo vytváranie backlogu, ktorý je obsahom priloženého textového súboru. Počas brainstormingu sme si v ňom zadefinovali množinu úloh, ktorých rozsah pravdepodobne vystačí na viacero budúcich šprintov.
- Backlog a rozpracované tasky je k priložený v samostatnom textovom súbore.
- Vyskúšali sme si planning poker, praktika sa osvedčila.

## 6.2.5 Zápisnica zo stretnutia č. 5

**Téma stretnutia:** Diskusia o migrácii projektu na GitLab, diskusia o SCRUM-e, prvé live zadávanie taskov do TFS

**Dátum:** 26. 10. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

**Priebeh stretnutia:**

- Hlavnou náplňou stretnutia po úvodnom stand-upe bola diskusia ohľadom správnej granularity taskov v TFS a celkovo o manažmente nášho projektu v rámci SCRUM-u.
- Zhodli sme sa na adaptácii gitflow metodiky dostupnej v remote repozitári, pričom sme ju rozšírili o pravidlo v rámci commit messages používať angličtinu.
- Diskutovalo sa taktiež o bugoch nájdených počas uplynulého týždňa.
- Hlavný CMakeLists.txt bude optimálne “rozbiť” do viacerých menších v subadresároch projektu, možný task na ďalší šprint. K tasku sa možno prihlási Michal
- Vyskúšali sme si vytvorenie backlog itemov priamo v TFS, pričom to chceme robiť aj naďalej už iba takto.
- Platforma Windows je stále problémová, Bencemu sa nepodarilo 3dsoftviz spustiť.

## 6.2.6 Zápisnica zo stretnutia č. 6

**Téma stretnutia:** Retrospektíva druhého šprintu, zadefinovanie epicov a iných náležitostí pre 1. odovzdanie, plánovanie 3. šprintu

**Dátum:** 2. 11. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

#### **Priebeh stretnutia:**

- Hlavnou náplňou stretnutia bola retrospektíva druhého šprintu a príprava na tretí.
- Radi by sme od tretieho šprintu pokračovali v manažmente TFS načrtnutom na minulom stretnutí.
- Nastavenie notifikácií na GitHub-e pre lepší prehľad o zmenách
- Správna granularita commitov a pull requesty pre navzájom súvisiace commity (osobitný pull request pre úpravy dokumentácie, osobitný pre fixy apod).
- Pokračuje analýza a úpravy dokumentácie projektu, aktualizovaný user manual.
- Zadefinovanie nového epicu (Interaktívna manipulácia s ulzami grafov v AR) a viacerých features s ním súvisiacich (priamo do TFS).
- Zadefinovanie zákazníka pre dokumentáciu k produktu. V odovzdanom dokumente treba rozobrať projekt podrobnejšie, spomenúť jeho chystanú migráciu a zavedenie CI/CD.
- súčasťou odovzdaného dokumentu môže byť aj analýza knižnice OpenPose na real-time detekciu kľúčových bodov postavy (popis knižnice, aktuálny stav vývoja, prípadne či by bola možná jej integrácia do projektu). Report si vzal na starosti Martin.

#### **6.2.7 Zápisnica zo stretnutia č. 7**

**Téma stretnutia:** Rozvrhnutie práce na dokumente, zmena scrum-mastera, diskusia ohľadom migrácie

**Dátum:** 9. 11. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

#### **Priebeh stretnutia:**

- Začali sme stand-upom. So začiatkom 4. šprintu sme sa zhodli na zmene scrum-mastera.
- Martin spracováva analýzu OpenPose.
- Vzhľadom k zvolenému template-u systému TFS sme sa zhodli, že pravdepodobne nebude pre nás smerodajným ukazovateľom burn-down chart. Bolo by totiž potrebné odznova nakonfigurovať TFS.
- Snažili sme sa rozbehnúť nástroj PlantUML zavedený predchádzajúcim tímovým projektom, na generovanie UML diagramov v rámci generovania Sphinx dokumentácie. PlantUML bude pridaný ako závislosť projektu.
- Bence sa naďalej snažil projekt rozbehať na platforme Windows.
- Počas uplynulého týždňa došlo k vykonaniu tasku Peťom napriek tomu, že ho mal pridelený Mišo. K podobným omylom už nesmie dochádzať.
- Naďalej sa pracuje na dokumente k riadeniu a produktu na prvý kontrolný bod(Peťo a Marek)
- Tomáš s Michalom refaktorujú CMake.
- Všetci sme začali pracovať na Peťovom forku ako BergiSK, na Slack sa zavesila dočasná metodika na code review využívajúca reviews pull requestov na GitHub-e. V letom semestri chceme používať už aj Astyle a cppcheck.

## 6.2.8 Zápisnica zo stretnutia č. 8

**Téma stretnutia:** Retrospektíva tretieho šprintu, planning poker pre 4. šprint, diskusia o estimácii storypointov, GitLab runneri, snaha o rozbehanie Windowsu

**Dátum:** 16. 11. 2017

**Čas:** 10:00 - 13:00

**Miestnosť:** 4.46

**Prítomní:** Ing. Peter Kapec

Bc. Martin Gašpar

Bc. Bence Ligárt

Bc. Michal Knapík

Bc. Tomáš Krupa

Bc. Miloslav Slížik

Bc. Peter Marušin

Bc. Marek Škriečka

**Priebeh stretnutia:**

- Po úvodnom stand-upe sme začali retrospektívou 3. šprintu. Ukázalo sa, že nevieme ako tím efektívne komunikovať. Čas odozvy niektorých členov tímu je dlhší ako 24 hodín, čo brzdí dianie v tíme a rozhorčuje scrum-mastera.
- Mišo navrhol do označenia commitov pridávať aj číslo tasku z TFS. Tiež bude mierne aktualizovaná gitflow metodika
- Windows setup stále neúspešný. Pre budúcnosť projektu je však veľmi dôležitý, keďže na ňom budú prebiehať ďalšie bakalárske a diplomové práce.
- Znova sa prizvukovalo, že situácia, keď Peťo urobil Michalov task, sa už nesmie zopakovať.
- Planning poker pre 4. šprint, tasky v TFS (tu spomíname iba niektoré):
  - pokračovanie v refaktore CMake-u (Mišo a Tomáš).
  - zapracovanie "big picture" opisov funkcionality 3DSoftvizu z bakalárskych a diplomových prác a zapracovanie do Sphinx dokumentácie
  - ošetrovanie warningov (Martin)
  - GitLab CI Runner a Docker (Milo)

- Dobrý spôsob na uľahčenie odhadu zložitosti taskov pre budúcnosť je vytvorenie zdieľaného dokumentu, kde by sme si značili doteraz odhadnuté úlohy. Tie poslúžia ako odrazový mostík pri ďalšom odhadovaní.

## 6.3 Motivačný dokument

Predkladaným motivačným listom sa budeme snažiť vyzdvihnúť hlavné kvality nášho tímu, ktoré hrali úlohu pri výbere tém, no tiež nám pomôžu pri práci na projekte.

Tím obsahuje jednotlivcov so skúsenosťami s tvorbou webových aplikácií (Marek, Martin, Bence). Niektorí sa zaujímame o oblasť dátovej analýzy (Milo, Peťo), iní o vnorené systémy a siete (Mišo a Tomáš) či o architektúru a softvérové inžinierstvo (Marek). Máme aj členov so skúsenosťami s prácou v tíme ktoré získali v rámci brigád, resp. stáži vo firmách (Martin, Mišo, Tomáš).

Práve spomínaná rôznorodosť je našou najväčšou devízou. Schopnosť adaptovať sa na nové doteraz nepoznané technológie je ale vlastnosť, ktorá nás všetkých spája.

Aj pri voľbe rozvrhu sme sa každý zariadili podľa vlastných preferencií, no kombinácia predmetov hrala úlohu aj pri výbere tém. Niektoré nami zvolené predmety sú: Aspektovo-orientovaný vývoj softvéru, Pokročilé databázové technológie, Spracovanie obrazu, grafika a multimedია, Vizualizácia dát a Neurónové siete.

Takmer všetci sa medzi sebou dlho poznáme a vieme navzájom odhadnúť svoje kvality. Všetci bývame na internáte alebo v okolí Bratislavy, čiže ani osobné stretnutia tímu kedykoľvek v prípade potreby nepredstavujú žiadny problém. Veľkou výhodou sú pestré záujmy a zameranie jednotlivých členov. Traja z nás sa na inžinierskom stupni rozhodli pre štud. program Internetové technológie a môžu tak prispieť ďalšími znalosťami či iným uhlom pohľadu.

### **Motivácia pre projekt Inteligentný bazár [IBazar]**

Projekt IBazar nás oslovil najviac. Viacerí sme sa v rámci svojich bakalárskych projektov už venovali práci s väčšími množinami rôzne štruktúrovaných dát (Mišo, Peťo a Tomáš) a viacerí by sa chceli „veľkým dátam“ a ich spracovaniu venovať aj v ďalšom štúdiu a následne v práci. Pre Peťu je motiváciou tiež získať skúsenosť s jazykom Python či konečne si poriadne „ohmatať“ svet webu a projekt sa mu javí ako ideálna voľba pre oba spomínané ciele. Mišo a Tomáš už majú s Pythonom skúsenosti. Všetkých nás zaujali odporúčané technológie. Keďže všetci denne používame bazáry a weby ponúkajúce „tovar z druhej ruky“, vieme, akými neduhmi trpia najznámejšie portály. Hlavne na portáli bazos.sk je situácia miestami naozaj zúfalá, čo ale vyplýva aj z jeho všeobecného zamerania. Umožňuje predaj širokého spektra druhov tovaru a nie je možné jeho zaradenie až do toľkých podkategórií ako pri iných

portáloch špecializujúcich sa napríklad iba na oblečenie. Na bazos.sk navyše musia byť všetky kľúčové slová súčasťou inzerátu. Dohľadateľnosť inzerátu preto závisí od jeho sformulovania a často aj od faktorov ako preklepy či znalosť gramatiky zadávateľa.

Chceme zlepšiť možnosti indexovania položiek v novom modeli bazáru a zvýšiť relevanciu inzerátov. Prieskum podobných riešení ako aplikácia LetGo či český portál vinted.cz už máme za sebou a poučení ich nedostatkami by sme radi navrhli vlastné riešenie.

## **Príloha A – Zoradenie všetkých tém podľa priority**

4. Inteligentný bazár [IBazar]
9. Odporúčanie pre e-biznis (Recommendation for eCommerce) [reCommers]
18. Rozpoznávanie cloudových služieb [OntoSEC]
20. Behaviorálna biometria na mobilných zariadeniach [Behametrics]
17. Vnorený systém monitorovania osôb [Breyslet 2.0]
26. Inteligentné parkovanie [SmartParking]
15. DeepSearch, alebo nájdeme to, čo práve potrebujete [DeepSearch]
24. Investment Portal [Invest]
25. Artificial Intelligence: Voice Channel [VirtualAsist]
3. Otvorené zmluvy: Budovanie prepojení vo verejných dátach [Zmluvy]
2. Group de'Cider [Group]
13. Analýza správania sa používateľa v mobilných aplikáciách [Mob-UX]
14. 3D UML, optimized version [3D-UML]
11. Databanka otázok a úloh [FIIT-DU]
5. Monitorovanie a vyhodnocovanie fyziologických procesov človeka [StresMonitor]
19. Pohlcujúci Web [iWeb]
12. Kolaboratívne prototypovanie používateľských rozhraní [Collab-UI]
7. Vizualizácia informácií v rozšírenej realite [VizReal]
8. Vzdelávanie vo Virtuálnej realite [EduVirtual]
10. Podpora diagnostiky [Look-Inside-Me]
1. Importér verejných datasetov [PubDatasets]
23. Návrh systému MOD [Future MOD]
16. Softvérovo definované siete pre budúci Internet [SDN4FutI]
22. Softvérovo riadené siete rozšírené o WiFi štandard [SDWN]
21. 3D simulovaný robotický futbal [3D futbal]
6. Internet vecí v našich životoch [IoT]

## **Príloha B**

	8.00 - 9.00	9.00 - 10.00	10.00 - 11.00	11.00 - 12.00	12.00 - 13.00	13.00 - 14.00	14.00 - 15.00	15.00 - 16.00	16.00 - 17.00	17.00 - 18.00	18.00 - 19.00	19.00 - 20.00	20.00 - 21.00	21.00 - 22.00
Pondelok														
Utorok														
Streda														
Štvrtok														
Piatok														

	ŠKOLA
	VOLNO

Obrázok 8: Rozvrh tímu s farebne odlišenými voľnými a obsadenými časovými úsekmi v rámci jednotlivých dní

## 6.4 Metodiky

Nižšie uvedené metodiky sme po analýze revidovali a uplatnili do praxe, aj keď pôvodné verzie boli vytvorené týmami pred nami. Snažili sme sa zachovať zaužívaný workflow, no zároveň si prispôbiť časti, ktoré sa pri práci na projekte v našom tíme neosvedčili.

### 6.4.1 Gitflow metodika

#### Forkovanie na GitHub-e

Fork na GitHub-e neprenesie tag-y do forknutého repozitára, treba ich ručne preniesť, v novom repozitári:

- `git remote add povodny-repozitar git@github.com:povodny-repozitar/nazov_repo.git`
- `git fetch povodny-repozitar 'refs/tags/:refs/tags/'`
- `git push --tags`

#### Vetvy

- Master - hlavný projekt
- Develop - branchnuta z mastra, každý sprint má vlastnú Develop vetvu, na konci sprintu sa merge späť do mastra,
- **!!! pred mergom treba spraviť komplet build (nie len unity)**
- Feature - branchnuta z developu, každý nový kus funkcionality (task v issue tracking nástroji), ktorý sa kódi musí mať vlastnú
  - Feature vetvu... po dokončení a validácii kódu sa merge späť do Developu, NEINTERAGUJE S MASTER VETVOU
- Hotfix - vetva na rýchly fix priamo z mastra, merge sa do mastra AJ developu, navyšuje aktuálnu verziu

Vždy mergeujeme cez Shell a s prepínačom **--no-ff**

Cheat sheet so všetkými základnými commandmi:

<https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>

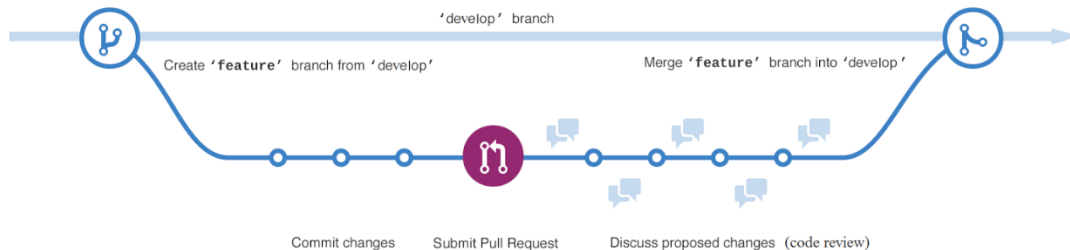


## Pull requesty

Po dokončení práce, keď sme ready to review sa dáva pull request na vetvu, do ktorej sa bude mergovať.

Pull request sa robí z GUI GitHubu (pravy horný roh), alebo `$ git request-pull {meno_commitu} {URL}` (doporučujem robiť cez GUI)

Po odsúhlasení Pull requestu sa potom pristúpi k mergu.



Obrázok 9: Práca na novej funkcionalite z pohľadu verziovacieho systému podľa Git metodiky

Obrazok ilustruje vytvorenie feature branch z develop vetvy, implementáciu rozdelenú do znázornených commitov, následný pull request predstavujúci žiadosť o code review a finálny merge do develop vetvy daného sprintu.

## Commit messages

v Commit messages používame tagy a ID ulohy na začiatok:

- [FIX] - fixli sme nejakú chybu z minula, bugfix, hotfix a podobne
- [ADD] - pridali sme novú funkčnosť, súbor, ...
- [DOC] - pridali sme dokumentáciu, komentý...
- [REF] - pre refactoring
- [FMT] - formátovanie textu, úprava
- [TEST] - pre testy
- [BUILD] - aktualizácia CMake build systému, modulov

Za tým veľmi stručne (a výstižne) opiseme, aké zmeny sme spravili. Message by mali byť krátke, no pokrývať všetko, čo sme v commitu spravili. **!!! všeobecný tvar: "[tag] #taskId**

**Popis vykonanej zmeny"** Napr. `[DOC] #3654 Pridanie uvádzania ID ulohy do gitflow metodiky`

## Useful commands

- `$ git submodule update --init --recursive`
  - update submodule (dependencies)
- `$ git checkout -f meno_branch`
  - checkout branch aj napriek lokálnym zmenám, budú zahodené
- `$ git status`
  - vypíše všetky vykonané zmeny
- `$ git stash / $ git stash pop`

- uloži stav projektu do stashu, z ktorého sa da potom tento stav pop-nut, dobre na prenos zmien medzi vetvami

#### **Tvorba feature branch-u:**

- \$ git checkout -b "feature/meno-feature" develop //Switched to a new branch "feature/meno-feature"

#### **Mergovanie hotoveho feature:**

- \$ git checkout develop //Switched to branch 'develop'
- \$ git merge --no-ff meno-feature
- \$ git push origin develop

#### **Tvorba hotfix branch-u:**

- \$ git checkout -b "hotfix/nazov-co-fixujem" master //Switched to a new branch "hotfix-{cislo\_verzie}"
- \$ git commit -m "sprava, co som spravil"

#### **Uzatvorenie Hotfix branchu:**

- \$ git checkout develop //Switched to branch 'develop'
- \$ git merge --no-ff "hotfix/nazov-co-fixujem"

## **6.4.2 TFS metodika**

### **Všeobecná metodika na manažment úloh v tíme**

#### **Pridanie novej úlohy**

- Pri každej úlohe je potrebné uviesť opis. Opis si k úlohe zadáva ten, kto si ju vyberie.
- Uviesť odhadovaný čas dokončenia.
- Opis musí byť podrobný, aby každému členovi bolo jasné, čo ma vykonať po pridelení úlohy.

#### **Rozdeľovanie úloh**

- Každý si vyberie (potiahne) úlohu/úlohy, ktorá/é majú najvyššiu prioritu.
- Ak ostanú nepridelené úlohy, študentský vedúci tímu pridelí členom zvyšné úlohy.

#### **Kedy je úloha hotová**

- Dokumentácia: Keď je znovu vygenerovaná.
- Kód: Potrebné spraviť code review a vykonať pull request do vetvy, ktorá sa bude mergevať.
- Testy: Keď je spravený report z testu.
- Zápisnica: Keď je nahratá vo formáte pdf na stránke tímu.

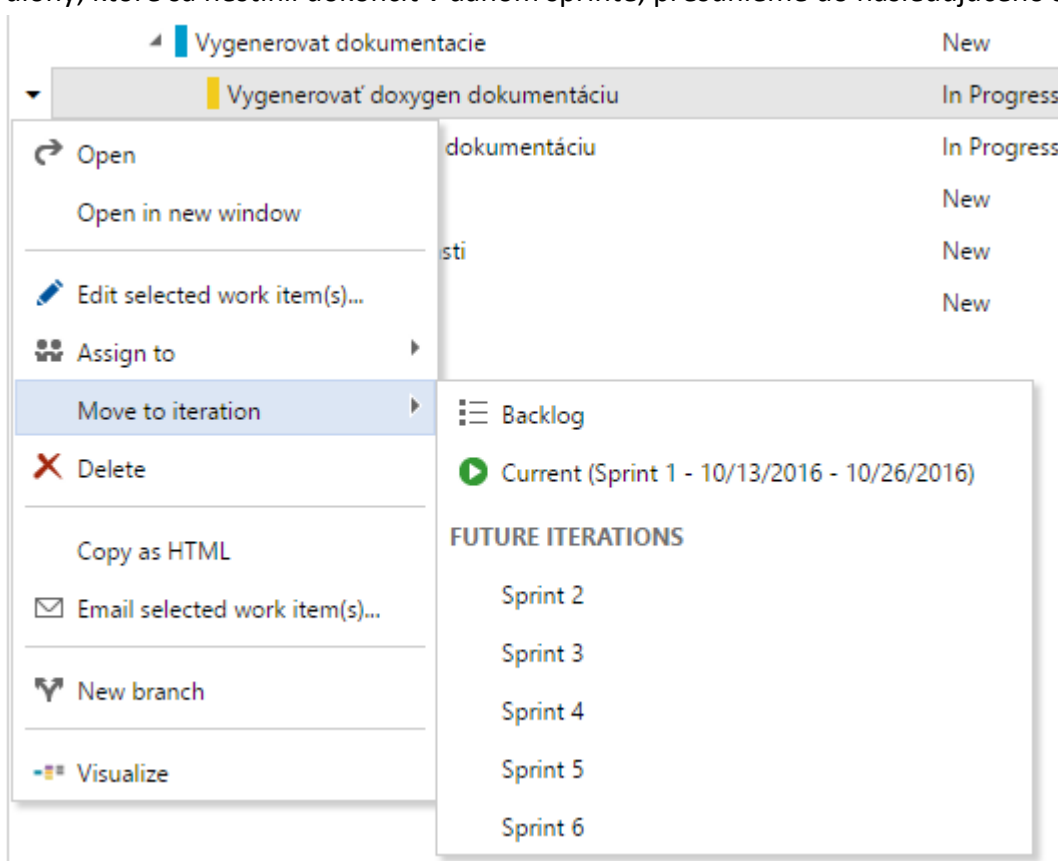
## TFS metodika

- Adresa TFS: <https://tfs.fiit.stuba.sk:8443/tfs/> (Potrebné sa lognúť 2x)
- Projekt: Just18

## Pridávanie úloh

- Úlohy podobného charakteru priradíme do spoločného backlog itemu.
- Pri pridávaní úlohy sa automaticky nastaví stav 'To Do'.
- V prípade objavenia chyby, je potrebné vytvoriť novú úlohu typu Bug (Chyba)
- Odhadovaný čas dávame na backlog itemy, rovnako ako aj Acceptance Criteria (predpripravený field pri rozkliknutí backlog itemu v TFS).
- POZOR: TFS podporuje estimáciu času v IBA v hodinách.

Úlohy (tasky) sa môžu nachádzať v troch stavoch: \* To Do \* In Progress \* Done Nesplnené úlohy, ktoré sa nestihli dokončiť v danom šprinte, presunieme do nasledujúceho šprintu.



Obrázok 10: Priradovanie úlohy do prislúchajúceho šprintu

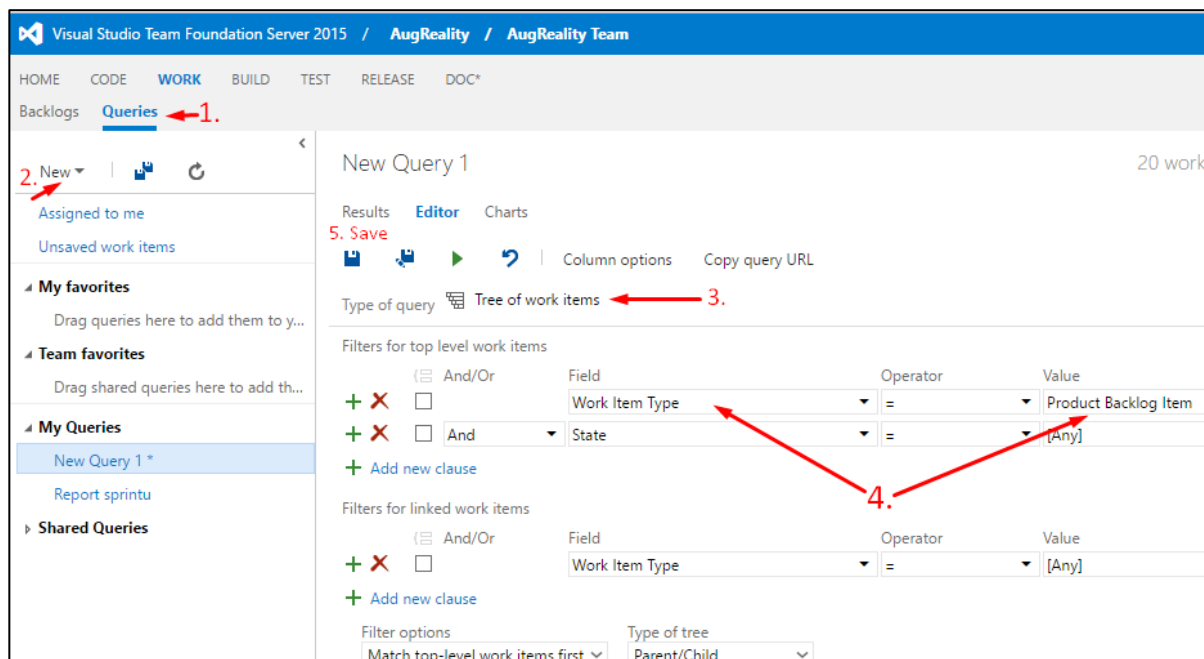
## Vytváranie exportov z TFS

Navod na stranke: [https://msdn.microsoft.com/en-us/library/dd286627\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/dd286627(v=vs.110).aspx)

URL: <https://tfs.fiit.stuba.sk:8443/tfs/>

Login: ako do AIS-u

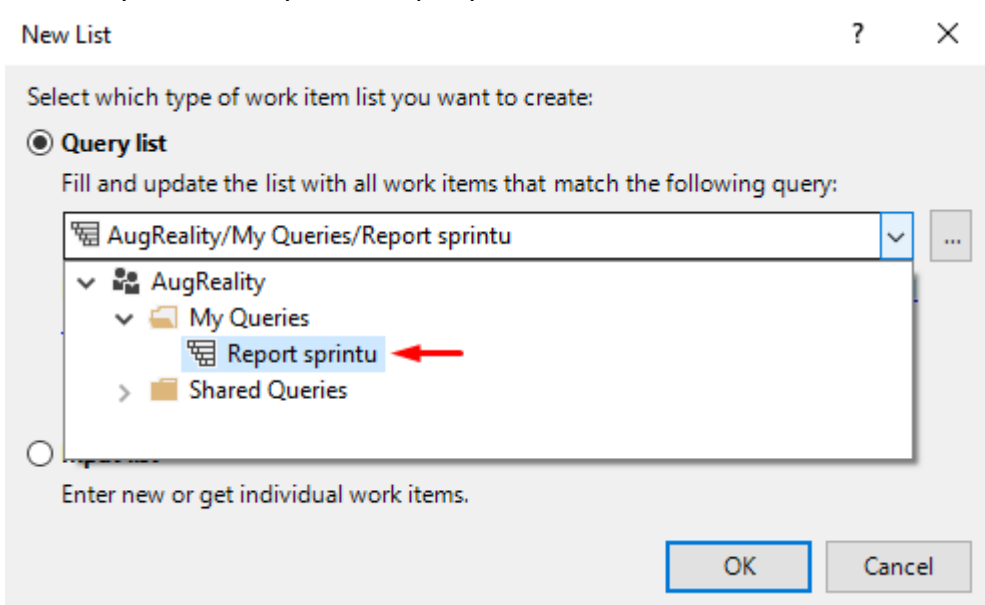
Pred prvým exportom je potrebné si vytvoriť query, ktoré vráti stav úloh v danom šprinte.  
Návod na vytvorenie query:



Obrázok 11: Vytváranie query za účelom exportu úloh zo systému TFS

## Excel

- V hlavnom menu vybrať záložku TEAM (mala by byť vpravo hore).
- Klik na New List ( umiestnená na ľavo pod záložkou File/Súbor).
- Vybrať novo vytvorené query:



Obrázok 12: Výber query pre export úloh zo systému TFS

- Hotovo

## Metodika tvorby a údržby UML diagramov prostredníctvom PlantUML

### PlantUML

[PlantUML](#) je jednoduchý program na tvorbu UML diagramov prostredníctvom ich textového opisu. K samotnému programu prislúcha aj rozsiahla [dokumentácia](#).

PlantUML je voľne dostupný na [stiahnutie](#) z oficiálnej stránky, prípadne je možné na otestovanie použiť aj jednoduchú [web aplikáciu](#).

Pre plnohodnotné využitie je potrebné mať taktiež nainštalovaný [Graphviz](#).

Tiež ponúka možnosť [integrácie](#) s viacerými textovými editormi a wiki stránkami.

### Pravidlá pre tvorbu súborov

1. Každý diagram sa nachádza v samostatnom textovom súbore (koncovka .txt, resp .wsd pri použití integrácie so sublime text).
2. Vygenerovaný diagram má identický názov ako prislúchajúci textový súbor (koncovka .png).
3. Názvy súborov sú po anglicky.

### Užitočné príkazy a postupy

#### Odstránenie duplicity pomocou Preprocesoru

Pri písaní diagramov, ktoré obsahujú komplikované vzťahy medzi entitami môžeme naraziť na situáciu, kde budeme veľa krát za sebou písať ten istý názov triedy alebo metódy. S využitím makier preprocesoru môžeme túto duplicitu ľahko odstrániť.

@startuml

'Bez proprocessoru

```
package "class Filter representation" {
    class ObjectStructure
    class Element {
        +{abstract}register(Visitor v)
    }
    class Mapper {
        +register(Filter f)
    }
    class Klient
    class Visitor {
        +{abstract}visitMapper(Mapper m)
    }
    class Filter {
        +visitMapper(Mapper m)
    }
}
```

```

}

ObjectStructure -down-> Element
Mapper -up-|> Element

ObjectStructure <-left- Klient

Klient -down-> Visitor
Filter -up-|> Visitor
}
@enduml

@startuml
'S preprocesorom
!define o(x) ObjectStructure
!define e(x) Element
!define m(x) Mapper
!define k(x) Klient
!define v(x) Visitor
!define f(x) Filter

package "class Filter representation" {
  class o()
  class e() {
    +{abstract}register(v(x) v)
  }
  class m() {
    +register(Filter f)
  }
  class k()
  class v() {
    +{abstract}visitMapper(m(x) m)
  }
  class f() {
    +visitMapper(m(x) m)
  }

  o() -down-> e()
  m() -up-|> e()

```

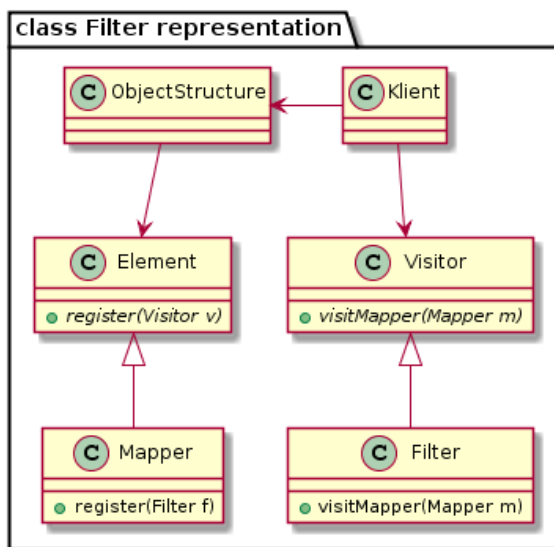
```

o() <-left- k()

k() -down-> v()
f() -up-|> v()
}
@enduml

```

*V oboch prípadoch bude výsledok nasledovný:*



V druhom prípade sa rozhodne menej napíšeme a máme možnosť meniť použité názvy tried na jednom mieste namiesto toho aby sme ich museli meniť všade. Stojí za poznámku, že každé definované makro musí mať parameter (v našom prípade x, z ktorého ajtak nečítame). Viac o Preprocesore na [tejto](#) stránke.

### Použitie aliasov v sekvenčnom diagrame

V sekvenčných diagramoch odporúčame pri definovaní volaní medzi objektami používať aliasy (skratky). Ich princíp je analogický s predchádzajúcim makrom avšak sú ešte o niečo prehľadnejšie. Aliasy nie sú však podporované v class diagrame.

```
@startuml
```

```
participant Client as c
```

```
participant Server as s
```

```
title sd Basic Server call
```

```
activate c
```

```
c -> s: sendMessage("You are awesome!")
```

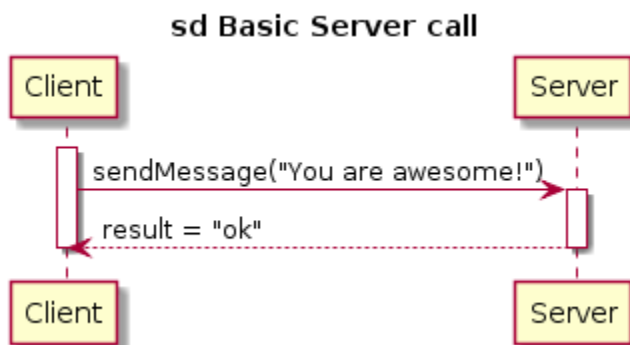
```

activate s
    s --> c: result = "ok"
deactivate s
deactivate c

```

@enduml

Výsledok:



Užitočnosť týchto skratiek (a makier) pochopiteľne narastá s narastajúcou komplexitou daného diagramu.

### Pravidlá pre súborovú štruktúru

Samotné UML diagramy je potrebné rozdeliť do prehľadnej súborovej štruktúry:

- projekt (názov projektu, napr. 3dsoftviz)
  - doc (inštalačná dokumentácia, vygenerovaná dokumentácia atď.)
  - uml
    - structural
      - class diagrams (korešpondujúce s reálnym kódom)
      - component diagrams
    - behavioral
      - activity diagrams
      - use-case diagrams
      - sequence diagrams
      - state diagrams

### Sphinx dokumentacia

### Instalacia

### Python

- Pre prácu so Sphinxom treba mať nainštalovaný [Python](#).



Python ponuka verzie 2.x a 3.x. Sphinx 1.3 moze bezat pod Python 2.6, 2.7, 3.3, 3.4, ale odporucana verzia je 2.7.

- Pre stahnutie a instalovanie externych kniznic pre Python existuje prikaz *pip*. Prikaz uz sa nachadza v oficialnych verziach Pythonu 3.4.0 alebo 2.7.9.
- Ak prikaz sa nenainstaloval automaticky, treba ho stiahnuť zo stránky <https://bootstrap.pypa.io/get-pip.py> a niekam uložiť. V prikazovom riadku treba prejsť do adresára s *get-pip.py* a spustiť nasledovný prikaz:
- `python get-pip.py`

## Sphinx

- Prejsť do priečinku s dokumentáciou (tam kde *index.rst* sa nachadza) a pomocou prikazu `pip install sphinx`
- `pip install sphinx`
  - ([sphinx-doc.org](http://sphinx-doc.org))
- Ak treba vytvoriť novú dokumentáciu, pre nastavenie zdrojového adresára a vytvorenie potrebných suborov na prácu so Sphinx treba spustiť prikaz
- `sphinx-quickstart`
- a odpovedať na otázky. Vyberte si všetky predvolené odpovede a po výzve zadajte názov, autorov a verziu projektu.
- Týmto prikazom budú vygenerované subory *Makefile*, *make.bat* a *conf.py.in*.
  - Všetky konfigurácie dokumentácie sú v *conf.py.in*.

Attention!

*Sphinx-quickstart* a vytváranie týchto suborov generujú novú dokumentáciu! Ak subory *index.rst*, *Makefile*, *make.bat* a *conf.py.in* už existovali, tak sa prepisu!

- Sphinx dokumentácia generuje výstup v rôznych formátoch zo suborov *.rst*. Podrobnejšie o [RestructuredText](#).

## HTML dokumentácia

- Subor *make.bat* povolí vygenerovať dokumentáciu v tom formáte, ktorý potrebujete
- Pre generovanie HTML dokumentácie treba v prikazovom riadku prejsť do priečinku s ReST subormi a *make.bat* suborom a spustiť prikaz
- `make html`

- Inak generovanie dokumentacie sa da spustit pomocou CMake v QtCreatore

## PDF dokumentacia

Pre generovanie PDF dokumentacie potrebujeme najprv vytvorit Latex dokumentaciju.

Note

Pre pracu s Latex treba mat [TeXlive](#)

Prikazom

make latex

vygeneruje sa Latex dokumentacia, ktora nasledne sa moze konvertovat do PDF pomocou programu [TeXstudio](#).

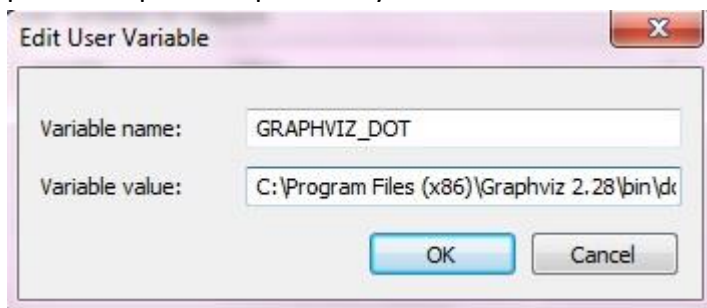
Note

PDF dokumentacia generuje len pomocou prikazoveho riadku a externeho programu, neda sa spustat cez CMake!

## PlantUML

Pre pracu s PlantUML nastrojmi v Sphinx treba:

- nainstalovat [Javu](#)
- pridať Javu do premenných prostredí (environment variable)
- nainstalovat [Graphviz](#)
  - odporucana verzia je 2.28
- pridať Graphviz do premenných



Note

Hodnota premennej ma byt do *dot.exe*

- pridať Graphviz do extensions v conf.py.in:
- extensions = ['sphinx.ext.graphviz']
- nainstalovat *sphinxcontrib-plantuml* zo [stranky](#) alebo prikazom
- pip install sphinxcontrib-plantuml
- pridať plantuml do extensions v conf.py.in:
- extensions = ['sphinxcontrib.plantuml']
- stiahnut [plantuml.jar](#)
- pridať do conf.py.in prikaz
- plantuml = 'java -jar cesta/do/plantuml.jar'

- Attention!
- Doležite je zmenit tuto cestu na spravnú, akú máte aktuálnu na Vasom počítači!
- pridávať UML do dokumentácie je možné pomocou
- .. uml::

```
!include /cesta/do/subor.wsd(txt)
```

alebo

```
@startuml
PlantUML kod
@enduml
```

### Excel tabulky

- Pre import Excel suborov do dokumentácie treba nainštalovať *exceltable* pomocou príkazu

```
pip install sphinxcontrib-exceltable
```

- Pridať *exceltable* do extensions v *conf.py.in*:  
extensions = ['sphinxcontrib.exceltable']

- Importovať tabulky pridaním do .rst suboru:

```
.. exceltable:: caption
:file: path/to/document.xls
:header: 1
:selection: A1:B2
```

- Podrobnejšie o [Options](#)