



Tímový projekt

REKONŠTRUKCIA 3D SCÉNY V2

Riadenie

Vedúci projektu:

Ing. Vanda Benešová, PhD.

Členovia tímu:

Bc. Jakub Ginter (SI)
Bc. Miroslav Haščič (SI)
Bc. Mário Hunka (SI)
Bc. Viktor Košťan (IS)
Bc. Richard Pintér (IS)

Názov tímu: 3DRecon (tím č. 11)

Web: <http://team11-16.studenti.fiit.stuba.sk/>

Kontakt: teamfiit11@gmail.com

Akademický rok: 2016/2017

Dátum odovzdania: 15. 05. 2017

Obsah

1	Úvod	4
1.1	Prehľad dokumentu	4
2	Role členov tímu a podiel práce	5
2.1	Manažérske činnosti	5
2.2	Podiel práce	5
3	Aplikácie manažmentov	6
3.1	Manažment komunikácie	6
3.1.1	Tímové stretnutia	6
3.1.2	Komunikačné nástroje	7
3.2	Manažment vývoja a integrácie	7
3.3	Manažment dokumentácie	8
3.4	Manažment plánovania	8
3.5	Manažment riadenia	9
4	Sumarizácia šprintov	9
4.1	Šprint 1	9
4.2	Šprint 2	10
4.3	Šprint 3	10
4.4	Šprint 4	11
4.5	Šprint 5	11
4.6	Šprint 6	12
4.7	Šprint 7	12
4.8	Šprint 8	12
4.9	Šprint 9	13
4.10	Šprint 10	13
5	Globálna retrospektíva	13
5.1	Globálna retrospektíva za ZS	13
5.2	Globálna retrospektíva za LS	14
6	Používané metodiky	15
7	Export úloh	16
8	Preberacie Protokoly	24
9	Motivačný dokument	25
10	Metodiky	27
10.1	Metodika Zadávania úloh do TFS	27

10.1.1	Tvorba Backlogu	27
10.1.1.1	Vytvorenie Backlog Itemu	27
10.1.1.2	Schválenie Backlog Itemu Product Ownerom	28
10.1.1.3	Odhadnutie effortu Backlog Itemu	28
10.1.2	Plánovanie šprintu	29
10.1.3	Práca na taskoch počas šprintu	29
10.2	Metodika prehliadok kódu	31
10.2.1	Základné ustanovenia	31
10.2.2	Priebeh prehliadky	31
10.2.3	Testing	32
10.2.4	Ako robiť prehliadku	32
10.3	Metodika verziovania kódu	33
10.3.1	Branches	33
10.3.2	Pull-Request	34
10.3.3	Commit	34
10.3.4	Súhrn pravidiel odosielaného kódu	35
10.4	Metodika testovania	36
10.4.1	Základné pravidlá	36
10.4.2	Priebeh testovania	36
10.5	metodika písania dokumentácie	37
10.5.1	Základné požiadavky	37
10.5.2	Základné pravidlá	37
10.5.3	Správa dokumentácie	38
11.	Metodika písania Doxygen dokumentácie	38
	Pravidlá	38
	Forma zápisu	39
	Class	39
	Funkcia	39
	Premenná	39

1 ÚVOD

Tento dokument obsahuje postupy a metódy, ktoré boli definované naším tímom za účelom riadenia tímového projektu. Projekt rieši segmentáciu 3D scény z rekonštruovaných 3D dát. Hlavnou myšlienkou je umožniť segmentáciu scény z dát v ľubovoľnom formáte pričom výstup by mal byť použiteľný softvérom *AutoCAD*¹.

Štafetu sme prebrali po minuloročnom tíme, ktorý viedol Bc. Lukáš Hudec. Vzhľadom nato, sme pokračovali v používaní niektorých metodík (metodika písania zdrojového kódu), avšak definovali sme si aj vlastné, ktoré sú špecifické pre náš tím.

Projekt je výskumného charakteru. Je teda potrebné umožniť testovanie rôznych segmentačných metód, ktoré po evaluácii môžu skončiť ako prototyp na zahodenie. Samotná práca s 3D dátami je výpočtovo, ale aj pamäťovo náročná. V súvislosti s tým, sme sa spočiatku začali uberať smerom, ktorý by urýchlil prototypovanie (ukladanie medzivýsledkov, ktoré neovplyvňujú výsledky testovania novej metódy). Ako ďalší z našich cieľov, je úplná segmentácia tzv. *outlayerov* (kľučka, svetlo atď.), ktoré vytvárajú nepresnosti pri segmentácii objektov, pri ktorých sa nachádzajú, prípadne sú ich súčasťou (stena, stôl atď.). V neposlednom rade, rozširujeme aj možnosti formátov vstupných dát. Ako ďalšie rekonštruujeme 3D dáta nasnímané pomocou zariadenia *Kinect*² z rôznych uhľov pohľadov.

Výsledkom našej práce je softvérový produkt, ktorý poskytuje rôzne metódy segmentácie 3D dát využiteľná hlavne pre architektov a vo výskumnej sfére. V rámci vývoja tohto produktu je samozrejme dostupné aj používateľské rozhranie, ktoré umožňuje používateľovi ľahko a efektívne, testovať nové segmentačné metódy.

1.1 PREHĽAD DOKUMENTU

V časti 2 sú opísané manažérske činnosti a zodpovednosti členov nášho tímu. V podkapitolách môžeme nájsť aj podiel práce členov tímu na tomto dokumente. V 3 kapitole sme uviedli aplikácie manažmentov, ktoré používame pri riadení nášho tímu. V kapitole 4 sme sumarizovali výsledky našich 3 šprintov. V rámci vývoja nášho softvéru používame určité metodiky, ktoré sú uvedené v časti 5. V ďalších častiach tohto dokumentu sa nachádzajú exporty z nástroja používaného na manažovanie (TFS), globálna retrospektíva a zápisnice zo stretnutí.

¹ Nástroj určený na dizajn <http://www.autodesk.com/products/autocad/overview>

² Hardware Senzor od Microsoftu <http://www.xbox.com/en-US/xbox-one/accessories/kinect>

2 ROLE ČLENOV TÍMU A PODIEL PRÁCE

2.1 MANAŽÉRSKE ČINNOSTI

TABUĽKA 1: ROZDELENIE ÚLOH V TÍME

Meno	Rola	Zodpovednosť
Jakub Ginter	vedúci tímu, biznis manažér	analýza a návrh požiadaviek, komunikácia s produkt ownerom
Miroslav Haščič	manažér kvality	vykonávanie testov, riešenie a správa chýb (bug-ov)
Viktor Košťan	manažér vývoja a architektúry	návrh a implementácia architektúry, správa hlavných segmentáčnych metód
Mário Hunka	integrátor, manažér propagácie	prepájanie jednotlivých modulov projektu, tvorba a udržiavanie web stránky
Richard Pintér	manažér dokumentácie	udržiavanie dokumentácie a záznamov zo stretnutí

2.2 PODIEL PRÁCE

TABUĽKA 2: PODIEL PRÁCE NA DOKUMENTÁCIÍ RIADENIA

Časť	Člen tímu
Úvod	Mário Hunka
Role členov tímu a podiel práce	Mário Hunka
Manažment komunikácie	Jakub Ginter
Manažment vývoja a integrácie	Jakub Ginter
Manažment plánovania	Mário Hunka
Manažment riadenia	Miroslav Haščič
Manažment dokumentovania	Richard Pintér
Sumarizácia šprintov	Richard Pintér
Metodika zadávania úloh do TFS	Viktor Košťan
Metodika prehliadok kódu	Mário Hunka
Metodika verziovania kódu	Jakub Ginter
Metodika testovania	Miroslav Haščič
Metodika Doxygen	Richard Pintér
Metodika dokumentácie	Richard Pintér
Globálna retrospektíva	Mário Hunka a Jakub Ginter

Export úloh z TFS	Jakub Ginter
Webové sídlo	Mário Hunka
Integrácia a finalizácia Kontrolný bod 1	Mário Hunka
Integrácia a finalizácia Kontrolný bod 2	Richard Pintér
Integrácia a finalizácia po LS	Jakub Ginter

TABUĽKA 3: PODIEL PRÁCE NA DOKUMENTÁCIÍ K INŽINIERSKEMU DIELU

Časť	Člen tímu
Úvod	Jakub Ginter
Globálne ciele pre ZS	Jakub Ginter
Globálne ciele pre LS	Jakub Ginter
Celkový pohľad na systém	Viktor Košťan
Moduly systému – úvod	Viktor Košťan
Segmentácia Outlayerov	Mário Hunka
Rekonštrukcia dát z Kinectu	Miroslav Haščič
Zlepšenie vizualizéra	Jakub Ginter
Ukladanie medzivýsledkov	Mário Hunka
Analýza metód na rozpoznávanie a segmentáciu stien	Viktor Košťan
Inštalčná príručka	Jakub Ginter a Miroslav Haščič
Technická dokumentácia	Jakub Ginter a Richard Pintér
Integrácia a finalizácia	Jakub Ginter

3 APLIKÁCIE MANAŽMENTOV

3.1 MANAŽMENT KOMUNIKÁCIE

Naša komunikácia v tíme by sa dala rozdeliť na dva hlavné celky. Prvým sú tímové stretnutia a druhým komunikácia v nástrojoch

3.1.1 TÍMOVÉ STRETNUTIA

Spoločne sme sa stretali podľa rozvrhu každý štvrtok o 09:00 v laboratóriu VGG 4.46. Na týchto stretnutiach sme spolu s vedúcou tímu zhodnocovali čo sa spravilo a čo sa bude robiť. Každý člen vysvetlil na čom pracoval a kde boli problémy, resp. aké by

mohli v budúcnosti nastať. Tieto stretnutia prebiehali podľa agilnej metódy SCRUM, ktorá je opísaná v manažmente plánovania. Tam je možné nájsť ako presne tieto stretnutia prebiehali.

Mimo týchto stretnutí sme sa stretávali aj v iných dňoch a riešili problémy, na ktoré sme narazili a bolo potrebné ich vyriešiť čo najskôr alebo jednoducho pracovali. Tieto stretnutia prebiehali zväčša vo VGG alebo v knižnici.

3.1.2 KOMUNIKAČNÉ NÁSTROJE

Na našu komunikáciu sme najčastejšie používali dva nástroje (Slack a Google Drive). V prvotných začiatkoch prác na projekte sme komunikovali prostredníctvom skupinového Facebook chatu. To sa však ukázalo ako neefektívne bolo potrebné nájsť vhodný nástroj.

Slack je aplikácia určená na komunikáciu v tímoch a zvýšenie ich produktivity. Je možné vytvoriť komunikačné kanály čo umožňuje rozdeliť komunikáciu do tematických celkov. V rámci kanálov je možné nastavovať notifikácie pre jednotlivých členov alebo skupinu. To má veľké využitie obzvlášť pri mobilnej aplikácii, ktorá je tiež k dispozícii. Používateľ tak môže nastaviť upozornenia iba na správy kde bol označený a vie, že je potrebné reagovať čo najskôr.

Pomocou tejto aplikácie je možné nahrávať súbory, zdieľať ukážky zdrojového kódu alebo upozorňovať pri commite vetvy.

Google Drive sme používali ako tímové úložisko, kde sme uchovávali dokumentáciu, záznamy zo stretnutí a rôzne iné dokumenty potrebné pre tím. Rovnako sme tu mali uložené datasety pre náš projekt.

Menej často sme používali Skype alebo Gmail, na ktorom sme mali založený tímový alias odkiaľ sa pošta preposielala každému členovi.

3.2 MANAŽMENT VÝVOJA A INTEGRÁCIE

V našom projekte je úloha manažéra vývoja a manažéra integrácie rozdelená medzi dvoch členov. Ich úlohy pozostávajú z manažmentu verzií, integrácie súčastí do jedného veľkého celku ale aj konfigurácia systému a iné. Manažér integrácie je priamo zodpovedný za repozitáre v TFS a udržuje vetvy projektu. Rieši konflikty a pridáva novú funkcionálnosť do hlavnej vetvy. Viac o manažovaní projektových vetiev a postup ako ich v projekte 3Drecon udržujeme sa nachádza v dokumente Metodika verziovania.

Aby sme dokázali udržiavať prehľadný kód a projekt používame TFS (Team Foundation Server) kde medzi sebou zdieľame vetvy projektu, manažujeme úlohy aj zapisujeme bugy. Tento nástroj je priamo integrovaný do Visual Studia, ktoré používame na vývoj a preto je pre nás TFS najlepšou možnou alternatívou. Nástroj ponúka jednoduchú synchronizáciu a aktualizovanie zmien, ktoré priamo prenáša medzi našim úložiskom aj

s prípadnými správami.

Každý z členov tímu je zodpovedný za časť systému, ktorú vyvíja teda prípadné nedostatky a chyby v týchto častiach musí primárne vyriešiť jej tvorca. Tvorca nie je zodpovedný iba za funkcionálnu stránku ale aj za výstup a úpravu, teda za celkové spracovanie (správne okomentovaný a zadokumentovaný kód). Následná integrácia do projektu je vykonaná integrátorom. Tvorca s ním túto skutočnosť odkomunikuje podľa pravidiel manažmentu komunikácie a poskytne integrátorovi potrebnú súčinnosť pri práci.

3.3 MANAŽMENT DOKUMENTÁCIE

Písanie dokumentácie v našom tíme prebieha priebežne. Dokumentácia je vytváraná spôsobom spoločného dokumentu, typu GOOGLE DOC, ktorý máme uložený na spoločnom úložisku GOOGLE DRIVE. Nakoľko dokumentácia sa skladá z viacerých častí, na GOOGLE DRIVE sú postupne pridávané dokumenty, ktoré si následne prejdú viacerí členovia tímu a po spoločnej konzultácii sa dohodnú, či je dokument vyhovujúci alebo je nutné ho prerobiť. V prípade, že je dokument vyhovujúci je následne zaintegrovaný do finálnej dokumentácie. Aby sme predišli rôznym kolíziám typu odlišné riadkovanie alebo font písma, dohodli sme sa dodržiavať pravidlá písania dokumentácie, ktoré sú spísané v dokumente menom Metodika písania dokumentácie.

V rámci manažmentu dokumentácie sme taktiež riešili dokumentovanie zdrojového kódu. Na tento účel sme sa rozhodli používať nástroj Doxygen. Pri pushovaní vetvy do devu musel byť kód správne zadokumentovaný podľa pravidiel spísaných v dokumente Metodika písania Doxygen dokumentácie. Manažér dokumentácie mal za úlohu vždy urobiť *Code Review* so zameraním na správny formát komentárov a prípadne upozorniť na nezrovnalosti.

3.4 MANAŽMENT PLÁNOVANIA

Predpokladom úspešného plánovania je naplnený tzv. product backlog aspoň na 2 šprinty vopred. Product backlog sa vytváral na základe požiadaviek nášho zadávateľa, ktorý je v tomto prípade zároveň náš vedúci tímu. Vzhľadom na požiadavky, boli hierarchicky vytvorené časti softvéru (Epics, Features), pre ktoré sa vytvárali backlog itemy. Tie sa radia vždy podľa priority – dôležitosť s akou treba daný backlog item spraviť, aby sme naplnili požiadavky čo najefektívnejším spôsobom.

Proces plánovania prebieha na začiatku každého šprintu. Podľa priority sú ku šprintu priradené backlog itemy, na ktorých sa pracuje počas najbližšieho šprintu. Backlog item má určité kritéria akceptácie, ktoré musia byť dodržané, aby bola úloha splnená. Na naplnenie týchto kritérií je väčšinou potrebné viacero úloh (taskov).

Následne sa ohodnotia jednotlivé úlohy vzhľadom na prvú ohodnotenú úlohu, ktorá bola

vopred určená (base). Na každom stretnutí sa hrá tzv. *Scrum Poker*, pri ktorom sa odhaduje náročnosť úloh pomocou hlasovania, čo sprevádza diskusia o úlohách o ktorých náročnosti sa hlasuje. Touto aktivitou je zabezpečený čo najlepší odhad k jednotlivým úlohám. Vzhľadom na minulé šprinty prispôbujeme náročnosť ďalšieho šprintu, preto je veľmi dôležité zabezpečiť čo najlepší odhad.

Následne sú zadané úlohy do manažovacieho nástroja TFS, ktoré sú buď priradené jednotlivým členom tímu, alebo si ich členovia tímu vyberajú počas šprintu. Progres v jednotlivých úlohách sa taktiež sleduje v nástroji TFS.

3.5 MANAŽMENT RIADENIA

Riadenie v našom tíme bolo založené na agilnej metóde vývoja SCRUM. V časti manažment plánovania sa môžete dozvedieť ako sme pomocou tejto metódy plánovali postup v našom projekte a ako sme určovali úlohy. Tieto úlohy sme potom evidovali v nástroji TFS. V každom šprinte sme vytvorili definované backlog items a k nim patriace úlohy. Každý člen mal definované svoje úlohy a ich hodnotu (ako sa táto hodnota získava nájdete v časti manažment plánovania). Niektoré úlohy nemali pridelenú zodpovednú osobu nakoľko nebolo jasné kto to bude robiť a nebolo ani potrebné to definovať. Dohodnuté však bolo že sa to musí urobiť a ten kto mal najviac času takúto úlohu pripísal v TFS sebe a urobil ju.

Pre úlohy existujú tri základné stavy: To do, In progress, Done. Logicky je cieľom, každú z definovaných úloh dopracovať do štádia done. Ak by sa však niektoré úlohy v šprinte nestihli urobiť, presúvajú sa do ďalšieho šprintu kde majú najvyššiu prioritu.

Dohľad nad vykonávaním úloh a nad ich rozdeľovaním má vedúci tímu. Ten je zodpovedný za správne rozdelenie úloh, prípadne riešenie problémov v tejto oblasti.

4 SUMARIZÁCIA ŠPRINTOV

4.1 ŠPRINT 1

V prvom šprinte sme sa primárne snažili urobiť všetko podľa odporúčaní odborníkov. Narazili sme však na niekoľko problémov, ktoré sa odzrkadlili aj na jeho kvalite. No práve vďaka týmto chybám teraz vieme omnoho lepšie ako na šprintoch pracovať. Veľkým problémom bolo, že sme sa museli naučiť fungovať ako tím a dokázať tak zvýšiť efektivitu našej práce. Druhým problémom bolo, že pokračujeme v projekte po minuloročnom tíme a na začiatku sme mali minimálne skúsenosti a znalosti v oblasti, ktorej sa projekt venuje. Preto bolo potrebné dlhé analyzovanie, pretože bez neho nebolo

možné ďalej implementovať novú funkcionálnosť. Fakt, že celý projekt je výskumného charakteru túto situáciu sťažoval ešte o trochu viac. So SCRUM-om sme v tíme skúsenosti nemali a s jeho použitím na výskumnom projekte už vôbec nie.

Začali sme používať nové nástroje ako SLACK na komunikáciu a TFS na evidenciu úloh. Nejaký čas nám trvalo aj naučiť sa pracovať s týmito nástrojmi, hlavne s TFS.

V prvom šprinte sme definovali 4 backlog items.

1. Zrýchlenie výpočtov ukladaním medzivýsledkov
2. (bug) Slabá viditeľnosť procesov
3. Analýza celkového kódu
4. (bug) Zlyhanie programu pri načítaní konfiguračného súboru (.xml)

4.2 ŠPRINT 2

Pri retrospektíve prvého šprintu sme sa všetci členovia tímu jednohlasne zhodli, že našou najväčšou slabinou bolo TFS a jeho správne používanie, čo sme sa rozhodli do ďalšieho šprintu zmeniť. V druhom šprinte sme sa po dohode s vedúcou projektu a s členom minuloročného tímu Lukášom Hudecom rozhodli pustiť do reštruktulizácie celého programu. Vedeli sme, že tento krok nás spomalí v napredovaní rozširovania funkcionálnosti ale na druhej strane sme vedeli, že ak si to navrhne a prerobíme podľa seba, bude to mať pozitívny účinok do budúcnosti. Najprv sme vytvorili návrh class diagramu nového projektu, ktorý sme následne diskutovali s vedúcou cez Skype hovor. Na konci hovoru, po odsúhlasení štruktúry projektu sme si rozdelili úlohy a postupne sme na nich začali pracovať. Počas šprintu sme narazili na zopár problémov pri rozbiehaní projektu na notebookoch, ktoré sa nám podarilo našťastie vyriešiť. Na konci šprintu sme zistili, že sme v stanovenom čase nedokázali spraviť všetku prácu, ktorú sme si naplánovali a, že v tom budeme musieť pokračovať aj v treťom šprinte.

4.3 ŠPRINT 3

Pri retrospektíve druhého šprintu sme zistili, že naša práca s TFS je už na vyššej úrovni ako to bolo pri prvom šprinte no stále sme nerobili niektoré veci tak ako by sme mali respektíve ako by sme si predstavovali. Zistili sme, že pri taskoch nezadáваме správne hodnoty v story pointoch a, že to má výrazný vplyv na nežiadúci výzor nášho Burndown chart diagramu. Taktiež sme zhodnotili že by sme mali v TFS riešiť aj neprogramátorské úlohy a mohli by sme vylepšiť aj spôsob robenia nášho code review. Tieto problémy sme si na stretnutí vydiskutovali a navrhli sme vhodné riešenie na ich odstránenie. Pri vytváraní Backlog Items pre tretí šprint sme takmer všetok čas nášho šprintu investovali do dokončenia úloh z predchádzajúceho šprintu, no taktiež sme už načrtli víziu do budúcnosti a identifikovali sme si ciele ako vyriešiť outliers a taktiež možnosť pracovať na našom projekte pomocou Kinectu. Pri hraní Scrum Pokeru sme si určili vzorový task na základe, ktorého sme prideliť hodnoty ostatným úlohám.

4.4 ŠPRINT 4

Plánovanie. Plánovanie pri šprinte 4 prinieslo rozdelenie nášho tímu do viacerých smerov. Na rozdiel od predošlých šprintov, kde sme riešili refaktoring a novú architektúru softvéru, tieto tasky boli konkrétnejšie a ich úlohou bolo priniesť novú funkcionálnu. Venovali sme sa segmentácií outlayerov, ktorá je aplikovateľná pri segmentácií stien a môže priniesť lepšie výsledky celkovej segmentácie. Ďalej sme sa venovali problému zalomenia stien, ktorý vzniká pri segmentácií. Jeden člen z nášho tímu hľadal možnosti využitia MS Kinect v našom projekte - rekonštrukcia nasnímaných dát. V neposlednom rade sme sa venovali vylepšovaniu nášho vizualizéra, ktorý je potrebný pri testovaní a samozrejme aj pri prezentácií nášho projektu. Ako posledný task sme si zadali priebežnú dokumentáciu kódu. Určili sme člena, ktorý sa bude venovať DoxyGen-u a bude kontrolovať aké komentáre sa píše a či spĺňajú všetky dohodnuté pravidlá.

Priebeh. V šprinte išlo hlavne o analytické úlohy, ktoré mali priniesť vhodné návrhy riešení, ktoré budú otestované. Analýzy sme predstavili na stretnutí v strede šprintu, kde sme ich bližšie diskutovali s našou vedúcou.

Retrospektíva. Retrospektíva priniesla výraznú zmenu pre náš tím. Zistili sme 2 problémy, ktoré sú kritické. Je potrebné riadiť verziovanie a zmeniť dĺžky šprintov. V prípade analytických šprintov (ako bol z veľkej časti tento) sme sa rozhodli pre 1 týždňový šprint, ktorá nám dopomôže k tomu, ako naplánovať ďalší 2 týždňový šprint čo najpodrobnejšie. Taktiež je potrebný manažér verziovania, keďže sa stáva, že sme často závislí jeden od druhého a je potrebné, aby niekto kontroloval priebeh a dokončenie novej funkcionality (schválené pull requesty, nové branche a pod.).

4.5 ŠPRINT 5

V rámci skúškového obdobia a zimnej prestávky sme sa spolu s našou vedúcou rozhodli realizovať tzv. zimný šprint, ktorý bol primárne určený na to aby sa každý člen zorientoval vo svojej problematike nakoľko sme sa rozhodli ísť rôznymi cestami. Zvolil sme si zvlášť človeka, čo bude riešiť vizualizér nakoľko táto časť projektu nebola zrovna user friendly a taktiež sme sa rozhodli zapracovať 3D myš. Jeden člen tímu sa začal venovať práci s kinectom, takže musel naštudovať všetko potrebné ohľadom kinectu od knižníc po snímanie dát. V rámci toho šprintu sme riešili aj špeciálne problém zalamovania stien čo bola takisto rozsiahla úloha, ktorej sa venoval jeden člen tímu. Ďalší člen sa venoval problému lokalizácie outlayerov. Taktiež sme si zvolili člena, ktorý sa prioritne venoval dokumentácii typu doxygen jej písaniu a aj generovaniu. Na konci šprintu všetci prezentovali dosiahnuté výsledky a zistenia aby mali všetci členovia tímu dostatočný prehľad o celom projekte.

4.6 ŠPRINT 6

V tomto šprinte sme sa pustili do realizácie niektorých problémov, ktoré sme analyzovali počas predošlého šprintu. Hlavným bodom bolo zapojenie Kinectu. Rozhodli sme sa, že pre tento problém budeme mať samostatne oddelený solution a vlastným GUI. Taktiež sme pracovali na vyriešení zalamovania stien, serializácií objektov, konfigurácií projektu pre použitie myši a ďalších úlohách. V tomto šprinte bola taktiež vytvorená nová metodika pre písanie komentárov s využitím Doxygen. Identifikovali sme problém, že nemáme definovaný formát pre písanie komentárov a preto sme sa rozhodli vytvoriť metodiku.

4.7 ŠPRINT 7

V rámci 7meho šprintu sme sa snažili spraviť si nejaký prehľad toho na čom je potrebné zapracovať. Jeden člen tímu sa venoval použitiu harrisovych hranových bodov. Taktiež sme zistili, že by bolo vhodné aby sa ukladali medzivýsledky segmentačných metód, čo mal na starosti tiež samotný člen tímu. Pracovali sme na vizualizéri a jeho zdokonaľovaní a taktiež sme bojovali s funkcionalitou 3d myši. V rámci tohto šprintu prebehla aj implementácia lokalizácie outlierov. Rozhodli sme sa vytvoriť samostatný projekt pre kinect a zvoliť si vhodnú knižnicu, kde sme sa nakoniec rozhodli vybrať knižnicu KINFU. Nakoľko kinect, nebol kompatibilný s knižnicou pcl 1.7 musel jeden člen tímu vytvoriť návod na prechod na knižnicu PCL 1.8. Taktiež sme sa rozhodli, že nakoľko náš projekt je časovo výpočtovo náročný bolo by vhodné aby vyskúšať nejakú paralelizáciu. Preto si jeden člen tímu zobral na starosti analýzu kódu aby našiel časti vhodné na používanie multithreadingu. Bolo nutné si naštudovať knižnicu OpenMP, ktorá je kompatibilná s Visual Studiom. Taktiež bolo potrebné skontrolovať, či webová stránka nášho tímu obsahuje všetky potrebné informácie a následne spraviť update.

4.8 ŠPRINT 8

Tento šprint sme využili na prechod na PCL 1.8. Taktiež sme pokračovali v práci na interaktore vizualizéra a snažili sme sa zapracovať multithreading do nášho projektu, čo v konečnom dôsledku nebolo úspešné a rozhodli sme sa od tohto problému upustiť nakoľko sme sa zhodli, že sa blíži súťaž TP cup, do ktorej sme sa prihlásili a bolo by viac vhodné sa v ďalšom šprinte začať venovať veciam okolo tohto projektu. V rámci tohto šprintu jeden člen tímu vypracoval prezentácie, toho ako funguje náš projekt. Taktiež sa nám podarilo integrovať 3d myš a dostať sa k prvotnej funkcionalite 3d myši.

4.9 ŠPRINT 9

V rámci toho šprintu, sme pokračovali v práci na Kinecte a prebehla migrácia Kinect app do GUI. Taktiež prebehla serializácia BirkysNormalSegmentation. Na koľko sme vedeli, že TPcup sa blíži, vznikla aj prvotná verzia plagátu a taktiež prvotná verzia videa, ktoré sme na stretnutí prediskutovali a dohodli sa na zapracovaní nových prvkov. Počas šprintu sme natáčali materiál do videa na TP cup. Počas trvania tohto šprintu sme sa niekoľko krát stretli mimo času v rozvrhu v laboratóriu a pracovali na príprave prezentácie na TP Cup a finalizácií projektu.

4.10 ŠPRINT 10

V rámci tohto šprintu sme sa venovali podobne ako v minulých šprintoch inovovaniu funkcionalít vizualizéru. Migrovali sme projekt na školský počítač a pripravovali plagát a video. Video sme po viacerých iteráciách rozhodli výrazne zmeniť a spraviť ho na novo. Plagát sme upravovali v denných iteráciách pomocou zdieľanej prezentácie. Pripravovali sme si scenár prezentovania a začali sme s finalizáciou dokumentácie.

5 GLOBÁLNA RETROSPEKTÍVA

5.1 GLOBÁLNA RETROSPEKTÍVA ZA ZS

V rámci celkovej retrospektívy sme do tohto bodu semestra narazili na viacero problémov, ktoré sme sa snažili riešiť v každom šprinte. Celkovo môžeme vytýčiť niekoľko skutočností, ktoré ovplyvnili vývoj nášho tímu najviac.

1. **Práca s TFS** – jednoznačne jeden z najväčších problémov. Keďže sme doteraz nemali možnosť pracovať v tíme pozostávajúcom z viac ako dvoch ľudí, je to pre nás nová skúsenosť. Nemali sme žiadne skúsenosti s používaním systémov na manažment úloh. Taktiež spadajú pod tento problém aj iné náležitosti, ako napr. tvorba backlogu či zadávanie estimate-u pri taskoch a vlastne celkové poznatky v oblasti manažmentu úloh.

V rámci 3. šprintu vidieť značný progres v tejto časti. Každý pracuje s tfs, interaguje s tabuľou, zadáva bugy a pod. Výsledok je vidieť na burn-down charte, ktorý naberá správny spád a taktiež aj na velocity, ktorá sa postupne

zvyšuje.

2. **Plánovanie** – v rámci plánovania šprintu sme učinovali viacero chýb. Nemali sme vytvorený backlog, z ktorého by sme len podsúvali podľa priority item-y do šprintov. Bolo to ovplyvnené hlavne tým, že pokračujeme po minuloročnom tíme a nevedeli sme presne kam sa chceme pohnúť a navyše je projekt výskumného charakteru, čo znamená, že backlog sa nám neustále mení vzhľadom na výsledky jednotlivých prototypov. Planning poker sme nehrali korektne a nezapisovali sme celý jeho opis rovno do description pri jednotlivých taskoch.

V rámci 3. šprintu sme v tejto časti spravili jednoznačne najväčší krok vpred. Aj keď je pravdou, že výskumný projekt nie je možné naplánovať na pol roka dopredu, lebo nikdy nie je isté čo-ako dopadne. Napriek tomu sme sa posnažili naplniť backlog minimálne na najbližšie 3 šprinty. Z týchto item-ov sme podľa priority vybrali tie s najvyššou a presunuli sme ich do šprintu. Je cítiť oveľa väčší prehľad v tom čo robíme.

Planning poker v 3. šprinte bol jednoznačne prínosnejší ako tie predtým. Hlasovalo sa o všetkých taskoch a diskusia sa písala rovno do ich opisu. Brainstorming zabezpečil lepší odhad a taktiež aj presný opis toho čo je treba v budúcnosti urobiť.

V neposlednom rade môžeme vyzdvihnúť našu komunikáciu. Od začiatku používame *Slack*, štruktúrujeme komunikáciu na jednotlivé témy a všetci sa aktívne zapájame. Taktiež využívame *Google Drive* na zdieľanie jednotlivých dokumentov. Okrem iného, sa stretávame osobne, mimo nášho oficiálneho stretnutia – to hodnotíme kladne a určite v tom chceme pokračovať aj naďalej.

5.2 GLOBÁLNA RETROSPEKTÍVA ZA LS

V letnom semestri sme sa rozhodne poučili z chýb a neidentifikovali sme výrazné problémy. Pokračovali sme vo využívaní všetkých nástrojov a pridali sme ešte písanie doxygen dokumentácie. Naša komunikácia bola stále na dobrej úrovni a ostali sme aj pri spoločných stretnutiach mimo vyhradeného času v rozvrhu.

Z dôvodu časovej a výkonnostnej náročnosti výpočtov sme začali vo veľkej miere využívať *TeamViewer*, ktorý nám priniesol možnosť pracovať na školskom počítači. Efektívne striedanie sa sme dosiahli hlavne vďaka komunikáciám.

Druhým zlepšením bolo výrazne lepšie verziovanie. Všetci sme začali efektívnejšie používať *tf*s a pracovať s vetvami v rámci projektu.

Všetky tieto veci sme robili aj v ZS, avšak až časom sme sa naučili robiť ich efektívne a podporiť tým produktivitu.

6 POUŽÍVANÉ METODIKY

TABUĽKA 4: POUŽÍVANÉ METODIKY

METODIKA ZADÁVANIA ÚLOH DO TFS	Metodika opisuje spôsob plánovania jednotlivých šprintov a napĺňania backlogu.
METODIKA PREHLIADOK KÓDU	Opisuje spôsob ako vykonávať prehliadky kódu a na čo sa pri nich zamerať.
METODIKA VERZIOVANIA KÓDU	Opisuje spôsob členenia projektu na jednotlivé verzie a ako správne používať Git.
METODIKA TESTOVANIA	Ako správne testovať náš systém.
METODIKA PÍSANIA DOKUMENTÁCIE	Tu sú spísané pravidlá, ktoré je potrebné pri písaní dokumentácie dodržať.
METODIKA PÍSANIA DOXYGEN DOKUMENTÁCIE	Dokument obsahuje postup, ako správne písať komentáre v zdrojovom kóde.

7 EXPORT ÚLOH

Project: 3Drecon Server: 3drecon.visualstudio.com\3drecon Query: 3Drecon Team - Sprint 1 - Backlog List type: Tree						
ID	Work Item Type	Title 1	Title 2	Status	Assigned To	Description
19	Product Backlog Item	Zrychlenie výpočtov ukladaním medzivýsledkov		Done	Mario Hunka	Prioritou tohto backlogu je ukladať medzivýsledky a obmedziť tak zbytočné čakanie na to isté pri každom spustení. Nakoľko program používa zložité výpočty bolo na bežnom počítači veľmi zdĺhavé testovať novú funkcionality. Do gui je potrebné pridať checkbox aby bolo možné exportovať medzivýsledky.
10	Task		Pridanie checkboxu pre export medzivýsledkov	Done	Mario Hunka	
7	Task		Ukladanie medzivýsledkov do formátu .pcd	Done	Mario Hunka	Vytvorenie funkcionality ukladania výsledkov výpočtov náročných funkcií programu s cieľom urýchlenia ladenia a vyvíjania programu.
31	Task		Načítavanie .pcd uložených medzivýsledkov.	Done	Mario Hunka	Uložené medzivýsledky je potrebné správne načítať do objektu ClosedSpace, kvôli ďalšej práci s nimi.
21	Bug	Slabá viditeľnosť		Done		
13	Task		Vytvorenie loading screenov	Done	Jakub Ginter	Používateľa je potrebné informovať o tom, že program práve pracuje na výpočtoch.
8	Product Backlog Item	Analýza celkového kódu		Done		Po prevzatí projektu po predošlom tíme je potrebné zanalizovať jeho štruktúru, jednotlivé moduly a triedy. Výsledky analýzy budú referované členmi tímu na ďalšom stretnutí.
11	Task		Vyhľadanie nástroja na zapisovanie chýb	Done	Mario Hunka	Nakoľko v každom projekte vznikajú nejaké chyby, je potrebné ich nejakým spôsobom evidovať. Treba analyzovať dostupné možnosti a vybrať tú najvhodnejšiu pre náš projekt.
14	Task		Analýza modulu 3DReconstruction/Methods	Done	Viktor Košťan	
22	Task		Analýza modulu 3DReconstruction/Objects	Done	Jakub Ginter	Analýza objektov používaných pri rekonštrukcii.
23	Task		Analýza modulu 3DReconstruction/Primitives	Done	Jakub Ginter	Analýza tried v Primitives, ktoré sú využívané pri rekonštrukcii.
26	Task		Analýza zvyšných tried v module DataHandling	Done	Mario Hunka	Analýza tried z DataHandlingu mimo konfigurácie.
28	Task		Analýza modulu gui/GUI	Done	Mario Hunka	Analýza používateľského okna.
29	Task		Analýza modulu gui/Widgets	Done	Mario Hunka	Analýza Widget tried, ktoré zabezpečujú výpočty v projekte GUI.
39	Task		Prefarbenie bodov	Done	Jakub Ginter	Prefarbit body na ine farby pre lepsie orientovanie v kóde
40	Task		odstátnenie zbytočných kopírovaní PointCloud	Done	hascicm.mh	po preskumaní som zistil ze kopirovania su opodstatnene, ale otvorila sa moznost navrhnut vlastny data parser do pcl:-PointXYZ
25	Task		Analýza modulu	Done	hascicm.mh	
9	Bug	Zlyhanie programu pri načítaní		Done	Viktor Košťan	
32	Task		Implementovanie bezpečného načítania konfiguračných	Done	Viktor Košťan	

Project: 3Drecon Server: 3drecon.visualstudio.com\3drecon Query: 3Drecon Team - Sprint 2 - Backlog List type: Tree						
ID	Work Item Type	Title 1	Title 2	Status	Assigned To	Description
75	Product Backlog Item	Refaktoring tried segmentácie			Richard Pinter	
76	Task		Identifikácia častí kódu týkajúcich sa segmentácie		Richard Pinter	
77	Task		Osamostatnenie kódu týkajúceho sa segmentácie		Richard Pinter	
74	Product Backlog Item	Vytvorenie abstraktnej triedy pre Segmentáciu			Richard Pinter	
81	Task		Vytvorenie triedy pre segmentáciu		Richard Pinter	
72	Product Backlog Item	rekonštrukcie pre nový model			hascicm.mh	
83	Task		prepracovanie triedy do nového projektu		hascicm.mh	
97	Task		objektov do nového projektu		hascicm.mh	
71	Product Backlog Item	refaktoring tried rekonštrukcie			hascicm.mh	
84	Task		refaktoring tried		hascicm.mh	
65	Product Backlog Item	Vytvorenie nového projektu Interior3DRecon			Viktor Košťan	
66	Task		Vytvorenie a nastavenie projektu Interior3DRecon		Viktor Košťan	projektov 3DReconstruction, DataHandling a Exporter.
51	Product Backlog Item	Refactoring načítania dát zo súborov			Viktor Košťan	
82	Task		Premiestnenie DataReader do Interior3DRecon		Viktor Košťan	
50	Product Backlog Item	Refactoring triedy ClosedSpace			Viktor Košťan	
67	Task		Vytvorenie základnej verzie triedy ClosedSpace		Viktor Košťan	

45	Product Backlog Item	Vytvorenie projektu pre GUI		Mario Hunka	rozhranie. Preto je potrebné vytvoriť nový projekt pre GUI a nastaviť všetko tak, aby aplikácia bežala bez problémov.
57	Task		Integrovanie QT do projektu	Mario Hunka	iný framework -QT. Z čoho vyplýva, že je nutné nastaviť cesty ku zdrojovým súborom QT. Taktiež nainštalovať addon do MCVS 2013, pomocou, ktorého je možné prepojiť QT Creator a MCVS.
58	Task		projektu s ostatnými projektami	Mario Hunka	Nastavenie dependencies pre GUI (iné moduly, ktoré GUI používa)
59	Task		Vytvorenie základného používateľského rozhrania	Mario Hunka	Vytvorenie triedy pre gui a základného template.
60	Task		Vytvorenie nového GUI pomocou QT	Mario Hunka	Design nového GUI.
61	Product Backlog Item	Prepojenie GUI s projektom		Mario Hunka	Je potrebné zabezpečiť prepojenie GUI s logickou časťou projektu.
62	Task		EventHandler pre rekonštrukciu	Mario Hunka	Handler pre elementy, ktoré súvisia s rekonštrukciou scény.
63	Task		EventHandler pre vizualizáciu	Mario Hunka	Handler pre elementy, ktoré súvisia s vizualizáciou.
64	Task		EventHandler pre načítanie súboru	Mario Hunka	Handler pre elementy, ktoré súvisia s načítaním/exportovaním súboru.
80	Task	Vytvorenie novej triedy pre vizualizer		Jakub Ginter	Vytvorenie projektu pre refaktoring starého kódu Vizualizácie priestoru
89	Task	Odstránenie nepotrebného kódu		Jakub Ginter	Zmazanie starého kódu, ktorý sa nepoužíva

Project: 3Drecon Server: 3drecon.visualstudio.com\3drecon Query: 3Drecon Team - Sprint 3 - Backlog List type: Tree							
Work Item ID	Work Item Type	Title 1	Title 2	Assigned To	Description	Remaining Work	Activity
146	Product Backlog Item	Dokumentácia k inžinierskemu dielu					
156	Task		Štruktúra dokumentu	Jakub Ginter	Vytvorenie nového dokumentu na google drive. Návrh štruktúry - kapitoly, podkapitoly.		
157	Task		Kapitola úvod	Jakub Ginter			
158	Task		Kapitola ciele	Jakub Ginter			
145	Product Backlog Item	Dokumentácia riadenia					
147	Task		Štruktúra dokumentu	Mario Hunka	Vytvorenie nového dokumentu na google drive. Návrh štruktúry - kapitoly, podkapitoly.		Documentation
148	Task		Finalizácia dokumentu	Mario Hunka	Po dokončení dokumentácie je potrebné naformátovať celú dokumentáciu do jedného štýlu.	1	Documentation
149	Task		Kapitola úvod	Mario Hunka	Napísanie kapitoly úvod.		
150	Task		Integrácia dokumentácie	Mario Hunka	Vzhľadom na to, že dokumentáciu robia všetci členovia tímu. Je potrebné dať všetko do jedného súhrného dokumentu.		Documentation
151	Task		Sumarizácia šprintu 1	Jakub Ginter			
152	Task		Export evidencie úloh	Jakub Ginter			
153	Task		Manažment komunikácie	Jakub Ginter			
154	Task		Manažment vývoja a integrácie	Jakub Ginter			
155	Task		Manažment riadenia	hasicm.mh			
159	Task		Globalna retrospektiva	Mario Hunka			
160	Task		Manazment planovania	Mario Hunka			
142	Bug	Zlá konverzia stringu pri načítavaní nového		Viktor Košťan		3	Testing

143	Task		Oprava konverzie stringu pri načítavání (TEST) Oprava konverzie stringu pri načítavání	Viktor Košťan				Development
144	Task						1	Development
140	Product Backlog Item	Webová prezentácia tímu		Mario Hunka				
141	Task		Update priebežnej dokumentácie na webe	Mario Hunka				Documentation
137	Product Backlog Item	Tvorba dokumentu na logovanie odrobených hodín		Viktor Košťan	na taskoch, aby mal produkt owner prehľad o odrobenej práci, keďže tfs nedovoľuje zdarma prístup do projektu pre viac ako 5 ľudí.			
138	Task		Vytvorenie excel sheetu na logovanie práce	Viktor Košťan	Je potrebné vytvoriť excel sheet na logovanie práce a vložiť do zdieľaného priečinku na drive.			Documentation
75	Product Backlog Item	Refaktoring tried segmentácie		Richard Pinter				
78	Task		Odstránenie nepotrebného kódu	Richard Pinter			5	
55	Product Backlog Item	Vizualizácia spracovaných dát		Jakub Ginter	Všetky dáta, ktoré načítavame a spracovávame je potrebné vizualizovať.			
91	Task		Vizualizácia Point cloud vo formáte XYZ	Jakub Ginter	Vizualizácia Point cloud vo formáte XYZ pre naskenovaný priestor		5	
93	Task		Vizualizácia Point cloud s farebným odlíšením	Jakub Ginter	Vizualizácia Point cloud s farebným odlíšením pre naskenovaný priestor		5	
52	Product Backlog Item	Refactoring ukladania (exportu) dát do		Viktor Košťan				
94	Task		Premiestnenie triedy Exporter do	Viktor Košťan				
95	Task		Refactoring triedy Exporter v	Viktor Košťan			3	
50	Product Backlog Item	Refactoring triedy ClosedSpace		Viktor Košťan				
88	Task		Vytvorenie segmentačného	Viktor Košťan			4	
106	Product Backlog Item	Finalizácia refaktoringu			Posledná časť refaktoringu, v ktorej sa vykoná merge všetkých branches refaktoringu do jednej výslednej.			
107	Task		Vytvorenie metódy calculate pre triedu (MERGE) Vytvorenie metódy calculate pre (TEST) Vytvorenie metódy calculate pre (REVIEW) Vytvorenie metódy calculate pre	Richard Pinter			5	Development
111	Task						1	Development
110	Task						2	Testing
112	Task						2	Development
108	Task		Vytvorenie metódy calculate pre triedu (MERGE) Vytvorenie metódy calculate pre (TEST) Vytvorenie metódy calculate pre (REVIEW) Vytvorenie metódy calculate pre	hasicm.mh			8	Development
113	Task						1	
114	Task						2	
115	Task						2	Development
109	Task		Integrácia segmentačných funkcií (MERGE) Integrácia segmentačných funkcií (TEST) Integrácia segmentačných funkcií (REVIEW) Integrácia segmentačných funkcií	Viktor Košťan			3	Development
116	Task						1	Development
117	Task						2	Testing
118	Task						1	Development
119	Task		Handler pre GUI (MERGE) Handler pre GUI (TEST) Handler pre GUI (REVIEW) Handler pre GUI	Mario Hunka	Prepojenie logickej časti v GUI. Pridanie handlerov na výpočet segmentácie konkrétnou metódou a vizualizácie pomocou nového vizualizéru.		3	Development
120	Task						1	Development
121	Task						2	Testing
122	Task						1	Development
123	Task		Vizualizácia segmentovaných dát	Jakub Ginter	Na vizualizér je potrebné odoslať dáta aj s informáciou, čo ideme vizualizovať. Je potrebné dorobiť možnosť vizualizácie už rekonštrovaných dát.		8	Development
124	Task		Interaktor vizualizéru	Jakub Ginter	Interakcia so starým vizualizérom nebola intuitívna. Treba zmeniť spôsob vizualizácie a urobiť interakciu v okne ľahšie použiteľnou.		8	Development
125	Task		(MERGE) Vizualizér				1	Development

126	Task		(TEST) Vizualizér			5	Testing
127	Task		(REVIEW) Vizualizér			2	Development
131	Product Backlog Item	Tvorba metodik					
132	Task		Manažment code review	Mario Hunka	Metodika na to ako robiť prehliadky kódu.		Documentation
133	Task		Manažment verzíí	Jakub Ginter	Metodika verziovania projektu		Documentation
134	Task		Manažment testovania	hascicm.mh			Documentation
135	Task		Manažment úloh	Viktor Košťan			Documentation
136	Task		Manažment dokumentácie	Richard Pinter		8	Documentation
100	Product Backlog Item	Migrácia existujúcej implementácie segmentácie outlayerov					
128	Task		Migrácia existujúcej implementácie	Mario Hunka	Presun logiky segmentácie outlayerov do nového projektu.	13	Development
129	Task		(TEST) Migrácia existujúcej			5	Testing
130	Task		(REVIEW) Migrácia existujúcej			2	Development

Project: 3drecon Server: 3drecon.visualstudio.com\3drecon Query: 3drecon Team - Sprint 4 - Backlog List type: Tree

ID	Work Item Type	Title 1	Title 2	Assigned To	Description	Remaining Work	Activity
209	Bug	Viacnásobný výpočet rovnakých výsledkov segmentačných metód		Viktor Košťan			Development
210	Task		Vytvorenie triedy zabezpečujúcej lazy evaluation	Viktor Košťan			Development
101	Product Backlog Item	Analýza rôznych metód lokalizácie outlayerov		Mario Hunka			
194	Task		Verifikácia analýzy	Mario Hunka			
195	Task		Analýza metódy ABOD	Mario Hunka			
196	Task		Analýza existujúcich metód	Mario Hunka			
197	Task		Analýza metódy založenej na hustote datasetu	Mario Hunka			
198	Task		Analýza určovania outliers pomocou neuronových sietí	Mario Hunka			
12	Bug	Zalamovanie stien.		Viktor Košťan			
20	Task		Identifikovať príčinu zalamovania stien	Viktor Košťan	Hypotéza je, že pridávanie hraničných bodov stien (polygónov) pomocou konkávnej obálky spôsobuje to, že sa medzi ne pridávajú body, ktoré neležia na rovine steny.		
206	Task		Opraviť identifikovanú príčinu zalamovania stien	Viktor Košťan	Identifikovaná príčina stien - nie je implementované prelietnutie bodov konkávnej obálky segmentu na rovinu segmentu pri upresnení ohraničenia segmentu.		Development
168	Product Backlog Item	Návrh prototypu pre lokalizáciu outlayerov		Mario Hunka			
169	Task		Vyhodnotenie analýzy	Mario Hunka			
214	Task		Návrh riešenie pomocou zobrazenia do 2D	Mario Hunka			
170	Product Backlog Item	Implementácia prototypu lokalizácie outlayerov		Mario Hunka			
171	Task		Vizuálne overenie	Mario Hunka		13	
172	Product Backlog Item	Zosnímanie testovacej vzorky Kinectom		hascicm.mh			
208	Task		snímanie dát pomocou Kinect SDK	hascicm.mh			
175	Task		Verifikácia nasnímaných dát	hascicm.mh			

173	Product Backlog Item	DataHandling pre Kinect dáta	načítanie a rekonštruovanie vytvoreného datasetu	hascim.mh		
213	Task		Verifikácia načítania súborov z Kinect	hascim.mh		
	Read-only	Analýza metód rekonštrukcie 3D dát nasnímaných z viacerých uhľov pohľadu		hascim.mh		
174	Product Backlog Item		Vyhodnotenie analýzy	hascim.mh		13
177	Task		analýza existujúcich metód	hascim.mh		
199	Task		analýza odborného článku	hascim.mh	http://www.tuit.ut.ee/sites/default/files/tuit/atprog-courses-bakalaureuset55-lot1.05.029-lembit-valgma-text-20160520.pdf	
202	Task					
178	Product Backlog Item	Analýza metód pre rozpoznávanie a segmentáciu stien		Viktor Košťan	Analýza metód rozpoznávania a segmentácie stien z point cloud dát (všetne te), ktorá sa používa v tomto projekte).	
179	Task		Vyhodnotenie analýzy	Viktor Košťan		21
205	Task		Analýza aktuálnej implementácie rozpoznávania a segmentácie stien.	Viktor Košťan		
207	Task		Rešeršovanie metód rozpoznávania a segmentácie stien	Viktor Košťan		
182	Product Backlog Item	Generovanie dokumentácie pomocou Doxygen		Richard Pinter		
183	Task		Verifikácia výstupu	Richard Pinter		
184	Product Backlog Item	Kontrola a oprava kódu podľa code convetions		Richard Pinter		
185	Task		Finalizacia opráv	Richard Pinter		13
187	Product Backlog Item	Zmena interaktoru vizualizéra		Jakub Ginter		
191	Task		Overenie funkčnosti vizualizéra	Jakub Ginter		8
188	Product Backlog Item	Vizualizácia segmentovaných polygónov		Jakub Ginter		
192	Task		Overenie vizualizácie segmentovaných dát	Jakub Ginter		3
200	Task		Oprava vizualizácie originálnych dát	Jakub Ginter	Oprava chyby pri vizualizovaní originálnych dát. Okno sa nedá zatvoriť ani nijak ovládať.	
201	Task		Implementácia vizualizácie segmentovaných dát	Jakub Ginter		
189	Product Backlog Item	Integrácia 3d myši		Jakub Ginter		
193	Task		Verifikácia interakcie s 3D myšou	Jakub Ginter		
203	Task		Konfigurácia projektu s novým SDK pre 3d myš	Jakub Ginter		
204	Task		Analýza 3D myši	Jakub Ginter	Analýza dostupných funkcií a knižnic obsiahnutých v SDK pre 3d myš. Hľadanie dostupných funkcií pre náš projekt.	

ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
187	Product Backlog Item	Zmena interaktoru vizualizéra		Done	Jakub Ginter	
191	Task		Overenie funkčnosti vizualizéra	Done	Jakub Ginter	
219	Task		Možnosť meniť obsah vizualizéra	Done	Jakub Ginter	V niektorých prípadoch môže byť užitočné pozrieť si obsah výstupu všetkých aplikovaných metód, prípadne originálne dáta. Preto sme sa rozhodli umožniť používateľovi aby pomocou konkrétnych kláves mohol zobrazovať aj iný obsah ako ten, ktorý zvolil v hlavnom okne aplikácie.
222	Product Backlog Item	Integrácia knižnic pre kinect		Done	hascim.mh	
223	Task		otestovanie knižnice	Done	hascim.mh	
12	Bug	Zalamovanie stien.		Done	Viktor Košťan	
206	Task		Opraviť identifikovanú príčinu zalamovania stien	Done	Viktor Košťan	Identifikovaná príčina stien - nie je implementované premietnutie bodov konkávnej obálky segmentu na rovinu segmentu pri upresnení ohraničenia segmentu.
189	Product Backlog Item	Integrácia 3d myši		Done	Jakub Ginter	
204	Task		Analýza 3D myši	Done	Jakub Ginter	Analýza dostupných funkcií a knižnic obsiahnutých v SDK pre 3d myš. Hľadanie dostupných funkcií pre náš projekt.
170	Product Backlog Item	Implementácia prototypu lokalizácie outlayerov		Committed	Mario Hunka	
224	Task		Najdenie a odlíšenie outlayerov	Done	Mario Hunka	
225	Task		Evaluácia výsledkov	Done	Mario Hunka	
182	Product Backlog Item	Generovanie dokumentácie pomocou Doxygen		Done	Richard Pinter	
183	Task		Verifikácia výstupu	Done	Richard Pinter	
184	Product Backlog Item	Kontrola a oprava kódu podľa code convetions		Done	Richard Pinter	
185	Task		Finalizacia opráv	Done	Richard Pinter	
188	Product Backlog Item	Vizualizácia segmentovaných polygónov		Done	Jakub Ginter	
192	Task		Overenie vizualizácie segmentovaných dát	Done	Jakub Ginter	
201	Task		Implementácia vizualizácie segmentovaných dát	Done	Jakub Ginter	

ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
249	Product Backlog Item	Ukladanie medzivýsledkov segmentačných metód		Done	Mario Hunka	Ukladanie prevažne pcl::PointCloud objektov, ktoré produkujú segmentačné metódy na urýchlenie výpočtov.
250	Task		Generalizácia nastavovania konfigurácie segmentačných metód	Done	Viktor Košťan	
233	Product Backlog Item	otestovanie Kinect v2 v projekte		Done	hascim.mh	
243	Task		vytvorenie projektu a vizuálny test	Done	hascim.mh	
12	Bug	Zalamovanie stien.		Done	Viktor Košťan	
216	Task		Testovanie opravenia chyby	Done	Viktor Košťan	
189	Product Backlog Item	Integrácia 3d myši		Done	Jakub Ginter	
203	Task		Konfigurácia projektu s novým SDK pre 3d myš	Done	Jakub Ginter	
237	Product Backlog Item	vytvorenie projektu a solution pre kinect		Done	hascim.mh	
238	Task		vytvorenie solution	Done	hascim.mh	
240	Task		naviazanie dependencies	Done	hascim.mh	
239	Task		vytvorenie GUI	Done	hascim.mh	

ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
253	Product Backlog Item	Analýza a dokumentácia		Committed	Richard Pinter	
257	Task		Analýza časti kodu vhodných pre multithreading	In Progress	Richard Pinter	
259	Task		Metodika pre Doxygen	Done	Richard Pinter	
260	Task		Analýza využitia GPU v	To Do	Richard Pinter	
272	Product Backlog Item	Paralelizácia výpočtov		New	Richard Pinter	
273	Task		Tutorial OpenMP	To Do	Richard Pinter	Vyskúšanie fungovania OpenMP na základných príkladoch a zistenie či je pre nás vhodné použiť ho.
264	Product Backlog Item	Aktualizácia webu		Done	Mario Hunka	
267	Task		Update dokumentácie	Done	Mario Hunka	
232	Product Backlog Item	Otestovanie použitia Harrisových hranových bodov.		Done	Viktor Košťan	Skôr než sa použijú body hrán na vylepšenie hľadania okrajových bodov objektov je treba vyskúšať segmentáciu týchto bodov z použitého datasetu.
234	Task		Akceptačný test použitia Harrisových hranových bodov.	Done	Viktor Košťan	
231	Product Backlog Item	Identifikácia nedostatkov hľadania okrajových bodov stien.		Done	Viktor Košťan	Na to aby sa vedelo ktoré časti procesu segmentácie rohových bodov objektov treba zmeniť/vylepšiť, ich treba najskôr identifikovať.
235	Task		Zdokumentovanie identifikovaných nedostatkov.	Done	Viktor Košťan	
274	Task		Review big picture.	Done	Viktor Košťan	
275	Task		Review metódy computePlanarCoefficients	Done	Viktor Košťan	
249	Product Backlog Item	Ukladanie medzivýsledkov segmentačných metód		Done	Mario Hunka	Ukladanie prevažne pcl::PointCloud objektov, ktoré produkujú segmentačné metódy na urýchlenie výpočtov.
251	Task		Opätovný výpočet použitím uložených dát	Done	Mario Hunka	
252	Task		Vytvorenie systému ukladania medzivýsledkov	Done	Mario Hunka	
268	Task		Identifikácia vhodných medzivýsledkov na uloženie	Done	Mario Hunka	
270	Task		Pridanie funkcionality na automaticky vypocet vsetkych metod	Done	Mario Hunka	
187	Product Backlog Item	Zmena interaktoru vizualizéra		Done	Jakub Ginter	
256	Task		Rotovanie objektu a kamery podľa tlačidiel	Done	Jakub Ginter	Interakcia pohybu vo vizualizačnom okne. Zmena spôsobu pohybovania sa v priestore a rotovania objektu. Rovnako sa zmení aj pohybovanie kamerou.
269	Task		Pohyb v okne vizualizera	Done	Jakub Ginter	Zmena pohybových vektorov pri interakcii s oknom. Aktuálne skáču o príliš veľké hodnoty. Je potrebné zmeniť logiku.
189	Product Backlog Item	Integrácia 3d myši		Done	Jakub Ginter	
246	Task		Implementácia konektivity pre 3D myš	Done	Jakub Ginter	Implementácia konektivity 3d myši. Zatiaľ nie je možné získavať eventy pri použití pohybov s 3d myšou.
170	Product Backlog Item	Implementácia prototypu lokalizácie outlayerov		Committed	Mario Hunka	
171	Task		Vizuálne overenie	Done	Mario Hunka	
271	Task		Test na novom datasete	In Progress	Mario Hunka	
237	Product Backlog Item	vytvorenie projektu a solution pre kinect		Done	hascim.mh	
244	Task		naviazanie kniznic pre releases verziu	Done	hascim.mh	
254	Product Backlog Item	Výber knižnice pre Kinect		Done	hascim.mh	
266	Task		overenie knižnice KINFU	Done	hascim.mh	
228	Product Backlog Item	vytvorenie návodu pre prechod na pcl 1.8		Done	hascim.mh	
241	Task		vytvorenie návodu	Done	hascim.mh	
242	Task		prechod stroja v učebni 4.48	Done	hascim.mh	

Proje						
ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
253	Product Backlog Item	Analýza a dokumentácia		Committed	Richard Pinter	
265	Task		Nahodenie zapiskov o stretnuti na net	Done	Richard Pinter	
278	Product Backlog Item	Update webu		Done	Mario Hunka	
294	Task		Aktualizácia zapiskov zo stretnuti	Done	Mario Hunka	
289	Product Backlog Item	Prechod na nové PCL (Miro)		Done	hascicm.mh	
292	Task		overenie funkčnosti	Done	hascicm.mh	
287	Product Backlog Item	Prechod na nové PCL (Kubo)		Done	Jakub Ginter	
187	Product Backlog Item	Zmena interaktoru vizualizéra		Done	Jakub Ginter	
245	Task		Overenie správania okna	Done	Mario Hunka	
255	Task		Prekladanie dvoch cloudov	Done	Jakub Ginter	Pre kontrolu potrebujeme prekladať vybrané pointcloudy aby sme dokázali skontrolovať výstup a porovnať si výsledky.
293	Task		Zmena pozície focal pointu	Done	Jakub Ginter	
300	Task		Zmena pozície koordinátov osí	Done	Jakub Ginter	
301	Task		Napísať dokument ku interakcií	Done	Jakub Ginter	
276	Product Backlog Item	Serializácia objektov		Done	Mario Hunka	Serializovanie objektov pre ukladanie medzivýsledkov.
295	Task		Moznosti serializácie v PCL	Done	Mario Hunka	
296	Task		Otestovanie serializácie	Done	Mario Hunka	
297	Task		Integrácia serializovateľnosti do ukladania medzivýsledkov	Done	Mario Hunka	
298	Task		Overenie funkčnosti	Done	Mario Hunka	
299	Task		Moznosti serializácie v C++	Done	Mario Hunka	
279	Product Backlog Item	Lokalizácia stredu datasetu		Done	Jakub Ginter	Nájsť stred datasetu, aby sa vo vizualizéry dal dynamicky nastaviť focal point.
280	Product Backlog Item	Prechod na novú knižnicu pre 3d senzory		Done	hascicm.mh	Napr. KinFuRemake
305	Task		test connectora	Done	hascicm.mh	
306	Task		build knižnic a dependencies	Done	hascicm.mh	
307	Task		zobrazovanie dat z kinect	Done	hascicm.mh	
282	Product Backlog Item	Dokončenie analýzy APS a PPE		Done	Viktor Košťan	Dokončenie analýzy AnalyticalPlanesSegmentation a PlanePointsExamination.
236	Task		Review metódy reconstructWallPlanes.	Done	Viktor Košťan	
304	Task		Zdokumentovanie analýzy	Done	Viktor Košťan	
189	Product Backlog Item	Integrácia 3d myši		Done	Jakub Ginter	
248	Task		Overenie funkčnosti 3d Myši	Done	Jakub Ginter	
291	Task		Vytvorenie pohybov v okne	Done	Jakub Ginter	Keď už získavame inputový formát dát z 3d Myši, je potrebné vytvoriť pohybové funkcie, podľa ktorých sa budeme pohybovať v okne. Knižnica žiadne neponúka, preto je potrebné vytvoriť to všetko od nuly.

Proj						
ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
332	Product Backlog Item	Otestovanie výstupu pre AutoCAD		Committed	Mario Hunka	
334	Task		Otestovanie kocky	Done	Mario Hunka	
315	Product Backlog Item	Vizualizácia 3D myšou		Done	Jakub Ginter	Je potrebné upraviť spôsob vizualizácie pomocou 3D myši. Vytvorený pohyb nie je dostatočne intuitívny.
316	Task		Upraviť userability	Done	Jakub Ginter	Posun v okne nie je dostatočne intuitívny. Je potrebné zmeniť hodnoty pri posunoch.
318	Task		Test na viacero datasetoch	Done	Jakub Ginter	Je potrebné otestovať fungovanie na viacero datasetoch.
320	Task		Sprehľadníť kód 3D Myši	Done	Jakub Ginter	Kód je zatiaľ neupravený. Je potrebné odstrániť nepotrebné kusy kódu a okomentovať ho.
322	Task		Odstrániť problém s renderovaním pri posune	Done	Jakub Ginter	Pri posúvaní kamery sa objekt stráca.
323	Task		Odstrániť problém s focalpointom	Done	Jakub Ginter	Pri pohybe sa preráta focalpoint a potom sa nedá vrátiť k objektu.
324	Task		Overiť funkčnosť 3D myši	Done	Jakub Ginter	Overiť bezproblémové fungovanie 3D myši.
327	Product Backlog Item	Zmena výstupu Kinect		Done	hascicm.mh	
321	Task		zmena formátu výstupu	Done	hascicm.mh	
310	Product Backlog Item	Migrácia kinect app do GUI		Done	hascicm.mh	
314	Task		overenie funkčnosti	Done	hascicm.mh	
309	Product Backlog Item	Serializácia BirkysNormalSegmentation		Done	Mario Hunka	- pri birkys cloudRGB a segmented_cloud - to iste 2x ?
313	Task		Test funkčnosti serializácie na väčšom datasete	Done	Mario Hunka	Analytical -planeCoefficients - error
283	Product Backlog Item	Úprava klasterov v PPE po odstránení outliers		Done	Viktor Košťan	Úprava klasterov v PlanePointsExamination po odstránení outliers
328	Task		Validácia funkčnosti	Done	Viktor Košťan	

Project: 3Drecon						
ID	Work Item Type	Title 1	Title 2	State	Assigned To	Description
352	Product Backlog Item	Vizualizér 3D Myš		Done	Jakub Ginter	
353	Task		Pridanie doxygen v novom kóde	Done	Jakub Ginter	
354	Task		Oprava načítavania binárneho cloudu	Done	Jakub Ginter	
355	Task		Inštalácia na školský počítač	Done	Jakub Ginter	
356	Task		Overenie	Done	Jakub Ginter	
357	Task		Odčlenenie metód pre 3D myš	Done	Jakub Ginter	
358	Task		Odstranenie chyby s vectorom pri birkis	Done	Jakub Ginter	
368	Task		Návod na inštaláciu 3D myši	Done	Jakub Ginter	
345	Product Backlog Item	Kinect dependencies pre labak, návod		Done	hascicm.mh	
348	Task		návod	Done	hascicm.mh	
349	Task		inštalácia	Done	hascicm.mh	
343	Product Backlog Item	zmena štruktúry zdrojového kódu		Done	hascicm.mh	
347	Task		zmena štruktúry	Done	hascicm.mh	
342	Product Backlog Item	MultiThreading pre kinect GUI		Done	hascicm.mh	
346	Task		overenie funkčnosti	Done	hascicm.mh	
341	Product Backlog Item	Parametrizácia Kinfu		Done	hascicm.mh	
344	Task		vizuálne overenie nových parametrov	Done	hascicm.mh	
339	Product Backlog Item	Implementácia Try a Catch v kóde		Committed	Mario Hunka	
340	Task		Verifikacia algoritmu	To Do	Mario Hunka	
361	Task		AnalyticalPlanesSegmentation	In Progress	Viktor Košťan	
362	Task		BirkysNormalSegmentation	Done	Mario Hunka	
363	Task		DataReader	Done	Viktor Košťan	
364	Task		InputDataParser	Done	Viktor Košťan	
365	Task		Exporter	Done	Mario Hunka	
366	Task		Objects	Done	Viktor Košťan	
367	Task		ClosedSpace	Done	Viktor Košťan	
369	Task		Serializer	Done	Mario Hunka	
370	Task		VisualizerWindow	Done	Jakub Ginter	
371	Task		Custominteractor	Done	Jakub Ginter	
336	Bug	Violation at writing location AnalyticalPlane		Done	Mario Hunka	
337	Task		Identifikacia bugu	Done	Mario Hunka	
338	Task		Uspesny cely algoritmus v debugu	Done	Mario Hunka	
332	Product Backlog Item	Otestovanie vystupu pre AutoCAD		Committed	Mario Hunka	
335	Task		Export apartmentu	In Progress	Mario Hunka	
319	Product Backlog Item	Odstránenie outlierov po fitovaní rovin		Committed	Viktor Košťan	
325	Task		Akceptačný test	To Do	Viktor Košťan	
326	Product Backlog Item	Logger pre Kinect		Done	hascicm.mh	
317	Task		overenie loggea	Done	hascicm.mh	
329	Task		vytvorenie Logger triedy	Done	hascicm.mh	
308	Product Backlog Item	Prezentácia na TPCUP		Done	Richard Pinter	
330	Task		Plagát	Done	Richard Pinter	
331	Task		Videoprezentacia	Done	Richard Pinter	
350	Task		Skontrolovať doxygen komentáre	Done	Richard Pinter	
351	Task		Dokumentácia	Done	Richard Pinter	

8 PREBERACIE PROTOKOLY

Tímový projekt 2016/2017

Tím č. 11 – 3DRecon

Predmet odovzdávania:

Dokumentácia riadenia

Projektová dokumentácia

Elektronické médium - zdrojové súbory, elektronická forma dokumentácie

Vedúci tímového projektu: Ing. Vanda Benešová, PhD.

Podpisom potvrdzuje prevzatie vyššie uvedených častí dokumentácie

V Bratislave

Dátum

Podpis

9 MOTIVAČNÝ DOKUMENT

Dobrý deň,

dovoľte mi predstaviť náš tím pre tímový projekt. Skladá sa zo 6 absolventov bakalárskeho štúdia na FIIT, pričom sme všetci študovali v odbore informatika. Menovite sa v tíme nachádza Jakub Ginter, Miroslav Haščič, Mário Hunka, Viktor Košťan, Richard Pintér a Matej Kollár. Nakoľko neštudujeme všetci rovnaký odbor sú v našom rozvrhu zastúpené takmer všetky odborné predmety inžinierskeho štúdia, ktoré nám poskytnú znalosti pre prácu na širokej škále projektov. Tím sme však neskladali náhodne. Pracovali sme spolu na viacerých menších projektoch v rámci štúdia, nie sme len kolegovia, ale aj kamaráti. Využívali sme rôzne technológie, primárne však prevládajú v našom tíme skúsenosti s vývojom v jazyku Java. S týmto jazykom majú široké skúsenosti všetci členovia nášho tímu. V rámci databázových technológií sú v našom tíme zastúpené PostgreSQL a MySQL ale tak isto aj MariaDB či Hibernate. Náš tím dokáže efektívne pracovať na vývoji ako webových aplikácií tak aj mobilných. Pokročilé skúsenosti máme s vývojom v Pythone(Django), C/C++ a už spomínaná Java. Pri tvorbe webovej aplikácie vieme využiť naše skúsenosti v HTML, CSS, JavaScript(AngularJS). V rámci školských povinností sme si vyskúšali prácu s použitím technológií v oblasti počítačových sietí, databáz, paralelného programovania alebo grafiky a videnia. V rámci bakalárskych prác sme pracovali na témach - vývoj aplikácie pre mobilné zariadenia s platformou Android, analýza správania používateľa, vizualizáciu dát v obohatenej realite a iné.

Počas mimoškolských aktivít sme nabrali skúsenosti v oblasti vývoja i testovania softvéru, tvorby webových stránok či v analýze dát.

Doposiaľ dosiahnuté vedomosti dokážeme efektívne využívať a rozdeliť si prácu, aby každý člen pracoval na tom čo ho zaujíma. Našou najsilnejšou stránkou je tímový duch a chuť pracovať na projekte spoločne a tím vytvoriť čo najzaujímavejší produkt s potenciálom pre reálne využitie praxi.

Motivácia EduSim

Náš tím sa prioritne zaujíma o tému Tvorba vzdelávacích simulácií, nakoľko pokladáme segment vzdelávania za veľmi neefektívny, obzvlášť čo sa týka využívania informačných technológií. Ako aj z vlastnej skúsenosti vieme, predstava fyzikálnych alebo chemických javov, ako napríklad gravitačná sila, elektromagnetizmus alebo fotosyntéza, môže byť pre žiaka veľmi náročná. Práve preto by sme radi pracovali na aplikácií, ktorá by žiakom zjednodušila učenie sa. Sami si vieme predstaviť ako by takáto aplikácia zefektívnila výučbu na školách, ktoré sme navštevovali.

Máme skúsenosti s tvorbou webových aplikácií, avšak prácu vo väčších tímoch sme si zatiaľ nemali možnosť vyskúšať, resp. sme pracovali iba v menších tímoch v rámci školských projektov. Práve preto chceme primárne pracovať na tejto téme, pretože zaujala všetkých členov nášho tímu. Skúsenosťami v tíme sa dokážeme navzájom dopĺňať, dokážeme vytvoriť funkčný frontend aj backend.

V rámci štúdia vieme podporiť vývoj znalosťami z premetov ako Pokročilé databázové technológie, Architektúry informačných systémov aj Spracovanie obrazu, grafika a multimédia či predmetom bakalárskeho štúdia PPGSO. Myslíme si že naše znalosti pre vypracovanie tohto projektu sú dostatočné a rovnako aj náš záujem o školstvo a proces edukácie samotnej. Budeme radi ak nám bude umožnené podieľať sa na tomto projekte.

Motivácia eMotion

Ako sekundárnu v rámci priority sme vybrali tému Manažment zdravotného stavu pacienta prostredníctvom monitoringu emócií. Táto téma nás zaujala z dôvodu, že vytvorením takéhoto systému je možné reálne pomáhať ľuďom v oblasti zdravia prostredníctvom informačných technológií. Tento segment pokladáme za veľmi zaujímavý a pri spolupráci so spoločnosťou, ktorá sa v ňom aj pohybuje by bolo možné vytvoriť veľmi zaujímavý systém.

Znalosti, ktoré by sme pri práci na tomto projekte vedeli využiť bude možné prepojiť v rámci štúdia s predmetmi ako Architektúra softvérových systémov, Objektovo-orientovaná analýza a návrh softvéru, Vizualizácia dát, Pokročilé databázové technológie, Vyhľadávanie informácií aj Objavovanie znalostí. Všetci členovia tímu majú pokročilé skúsenosti s programovaním v jazyku Java a tiež máme záujem získať vedomosti v oblasti programovania v jazyku Swift, ktorý ma veľký potenciál využitia. Naše skúsenosti a vedomosti pokladáme za dobrý predpoklad pre prácu na projekte eMotion a budeme radi ak nám to bude umožnené.

10 METODIKY

10.1 METODIKA ZADÁVANIA ÚLOH DO TFS

V rámci tohto projektu (3dRecon) sa úlohy manažujú v 3 fázach:

1. *Tvorba Backlogu*
2. *Plánovanie šprintu*
3. *Práca na taskoch počas šprintu*

Manažment úloh sa v tomto prípade týka **Backlog Items**, **Bugs** a **Tasks**. Epics a Features nie sú opísané, keďže sa s nimi manipuluje iba zriedka.

Najdôležitejšia časť metodiky pre členov tímu je *Práca na taskoch počas šprintu*.

SLOVNÍK

BLI - Backlog Item

PO - Product Owner

10.1.1 TVORBA BACKLOGU

Tvorba Backlogu je činnosť, ktorá prebieha stále. Vytvoriť nový Backlog Item alebo Bug je možné hocikedy a nemusí sa čakať s vytvorením na konzultáciu s ostatnými členmi tímu alebo s Product Ownerom, aj keď práve PO dáva najviac podnetov na BLI vzhľadom na tému.

Životný cyklus BLI

1. *Vytvorenie BLI*
2. *Schválenie BLI Product Ownerom*
3. *Odhadnutie Effortu BLI*
4. *Práca na BLI*
 - 4.1. *Priradenie do sprintu*
 - 4.2. *Odloženie (pozastavenie) BLI*
5. *Dokončenie BLI*

(body 4 a 5 sú opísané v kapitolách *Plánovanie šprintu* a *Práca na taskoch počas šprintu*)

10.1.1.1 VYTVORENIE BACKLOG ITEMU

Aj keď je možné vytvoriť BLI hocikedy je dôležité aby sa zadali všetky nižšie popísané atribúty BLI-u aby sa neskôr nebrzdili procesy zadávaním týchto atribútov. Tiež v čase

ich tvorby ich vieme opísať rýchlo, lebo sa nemusíme rozpamätávať.

Kroky tvorby Backlog Itemu:

- 1. Vhodne nazvať BLI - Title** - Názov Backlog Itemu by mal byť dostatočne popisný aby vystihoval prácu ktorú predstavuje, pričom by nemal obsahovať názvy tried, funkcií atď. pokiaľ nie sú nevyhnutné pre popis aby sme neobmedzovali tasky vo voľbe implementácie. (*miesto názvu funkcie alebo triedy skôr použiť opis funkcionality*). Zlý názov “Vytvorenie triedy Exporter” - Dobrý názov “Vytvorenie exportu segmentačných dát”
- 2. Pridať “Description” - Description** - Popis by mal byť natoľko podrobný, aby umožnil členom tímu odhadnúť *effort* a určiť tasky na jeho implementáciu.
- 3. Určiť “Acceptance Criteria” - Acceptance Criteria** - Kritérium akceptácie reprezentuje “*Definition of done*”. Slúži na verifikáciu dokončenosti BLI. Mala by byť definovaná minimálne 1 akceptačná podmienka.
- 4. Určiť “Value area”** - Ak sa BLI týka zmeny štruktúry programu a nepridáva funkcionality tak je *Architecture*, inak *Business*.
- 5. Určiť “Priority”** - Pri určení priority je dobré vychádzať z priorit ostatných BLIs a určiť ju vzhľadom na ne.
- 6. Priradiť ku “feature” (pridať vzťah patent)**
- 7. Priradiť predchodcov (pridať vzťah successor)** - Je dôležité priradiť predchodcu BLI, aby sa pri plánovaní šprintu vedelo, či bude možné BLI implementovať (či nebude musieť BLI čakať na predchodcu).
- 8. Usporiadať BLI podľa priority**

V rámci Backlog treba premiestniť nový BLI tak, aby boli Backlog Items v zozname usporiadané podľa priority.

10.1.1.2 SCHVÁLENIE BACKLOG ITEMU PRODUCT OWNEROM

Po schválení BLI Product Ownerom (väčšinou na stretnutiach) je potrebné zmeniť jeho stav z *New* na *Approved*. Ak sa BLI vytvára na podnet Product Ownera, tak po jeho vytvorení tiež treba zmeniť stav na *Approved*.

10.1.1.3 ODHADNUTIE EFFORTU BACKLOG ITEMU

Odhadnutie effortu BLI by sa malo vykonať na tímových stretnutiach pomocou *planning pokeru*, pričom sa môže upraviť aj jeho popis ak si ho počas *planning pokeru* členovia tímu lepšie upresnia.

10.1.2 PLÁNOVANIE ŠPRINTU

Pred začatím plánovania šprintu by mali mať všetky BLIs **určený effort**, aby sa vedelo podľa *velocity* z predošlého šprintu odhadnúť aké BLIs vybrať aby sa stihli spraviť do konca šprintu.

Kroky plánovania šprintu:

1. **Určiť Backlog Itemy pre nasledujúci šprint.** - Backlog itemy pre nasledujúci šprint sa určujú na základe *velocity* z predošlého šprintu a závislosti Backlog Itemov (niektoré môžu byť vykonané až po druhých)
2. **Určiť a vytvoriť tasky pre Backlog Itemy**
 - 2.1. *Title* - Názov tasku by mal konkrétne popisovať úlohu, ktorá sa ma vykonať. V prípade *Development* tasku by sa mali používať aj názvy funkcií a tried.
 - 2.2. *Description* - Popis tasku **netreba** pokiaľ je názov dostatočne opisný.
 - 2.3. *Activity* treba určiť podľa typu tasku
3. **Odhadnúť Remaining Work tasku**
4. **Určiť vzťahy medzi taskmi**
 - 4.1. *Predecessor/Successor* - potrebné pre všetky typy taskov, ak sa dá
 - 4.2. *Related* - vhodné pre tasky ako REVIEW a pod. ktoré môžu bežať súbežne a ovplyvňujú sa (súvisia) istým spôsobom
 - 4.3. *Tests/Tested By* - tasky typu *Testing* (Activity) majú vzťah *Tests* s taskami, ktoré testujú.

10.1.3 PRÁCA NA TASKOCH POČAS ŠPRINTU

Počas šprintu sa vykonávajú nasledujúce aktivity:

1. **Práca na tasku** - Pri začatí práci na tasku sa člen tímu priradí do daného tasku pokiaľ tak už neurobil a presunie task (zmení mu stav) do *In progress*.
2. **Dokončenie práce na tasku** - Ak sa dokončí práca na tasku je nutné ho presunúť do *Done*. Tiež sa skontroluje Backlog Item tasku, či má dokončené všetky tasky. Ak áno a sú **splnené Acceptance Criteria**, tak sa nastaví stav Backlog Itemu na *Done*.
3. **Obnovenie práce na tasku** - V prípade ak sa ukáže, že task nie je dokončený, napr. ak neprejde cez testy, zmení sa jeho stav späť na *In progress*.
4. **Objavenie chyby** - Ak sa objaví chyba vytvorí sa nový *Bug* do Backlogu a podľa

závažnosti a potreby sa môže pridať do aktuálneho šprintu. Pri jeho vytváraní sa nastaví atribúty *Severity*, *Effort*, *Remaining Work* a *Activity*. (atribúty *Effort* a *Remaining Work* sa nastaví na rovnakú hodnotu)

- 5. Pridávanie potrebných taskov** - Počas šprintu sa často stáva, že sa zistí, že treba vytvoriť nový task, lebo pri plánovaní sa s ním nerátalo, alebo sa objavil bug a treba vykonať tasky na jeho opravu. V takých prípadoch sa môže vytvoriť task a pridať do BLI alebo Bugu pre daný šprint. Nevadí, že Burdown Chart stúpne, pretože podľa toho ako často a o koľko stúpne sa vie vyhodnotiť, ako dobre sa plánuje, prípadne ako často nastávajú chyby.

10.2 METODIKA PREHLIADOK KÓDU

Tento dokument pojednáva o spôsobe vykonávania prehliadok naším tímom. Každý *pull-request* musí prejsť schválením, ktorému predchádza prehliadka kódu.

10.2.1 ZÁKLADNÉ USTANOVENIA

1. Schváleniu predchádza prehliadka kódu, ktorú vykonáva vopred určený člen tímu. V nevyhnutných prípadoch je možné, aby prehliadku spravil niekto iný, avšak je potrebné dostatočne vopred oznámiť, že prehliadku daný človek nebude môcť vykonať, pričom treba uviesť relevantný dôvod.
2. Nikto by nemal robiť prehliadku pre kód dlhší ako **400** riadkov kódu. V prípade, že je kódu viac, je potrebné, aby bola prehliadka vykonaná viac ako jedným členom.
3. Prehliadka by mala trvať maximálne **1 hodinu**.

10.2.2 PRIEBEH PREHLIADKY

Pri prehliadke je potrebné pozrieť sa na kód z viacerých pohľadov, za účelom odhaliť čo najviac chýb a tak zabezpečiť vyššiu kvalitu kódu.

ARCHITEKTÚRA

- **Single responsibility principle** - každá trieda je zodpovedná za **práve** jednu konkrétnu vec. Tento princíp je možné uplatniť aj pri metódach. Ak metóda robí 2 veci, nie je to správne.
- **Open/Closed principle** - každá trieda musí byť orvorená pre rozširovanie (dá sa rozširovať, je možné pridávať novú funkcionality...), ale uzavretá pre modifikáciu (existujúcu implementáciu, je možné použiť inými triedami/modulmi)
- **Duplikácia kódu** - ak si všimneme duplikovaný kód, treba zvážiť či nie je možné vytvoriť metódu, ktorú je potom možné využiť na viacerých miestach.
- **Error handling** - používané metódy je potrebné ošetriť proti zlyhaniu (napr. načítanie *.pcd* súboru - *FileNotExistException* - je ošetrená?). Inými slovami, vždy musí byť alternatívny scenár, ktorý sa vykoná v prípade neúspechu primárneho priebehu.
- **Efektívnosť** - treba sa zamyslieť či daný algoritmus nie je možné urýchliť. Napr. iterovanie celého listu v prípade, že to nie je potrebné (hash tabuľka, dictionary apod.)
- **Potenciálne chyby** - táto časť je jednoznačne najťažšia, ale je potrebné sa

zamyslieť aj nad ňou. Každý kód by mal prejsť samozrejme testovaním, avšak, je možné, že sa nachádza v kóde nejaká logická chyba ? Ak si ju všimneme je výborné nato poukázať.

ŠTÝL

Vzhľadom na skutočnosť, že pokračujeme v minuloročnom projekte, je vhodné aby sme dodržiavali code conventions, ktoré boli zavedené minulý rok. Táto metodika sa nachádza v minuloročnej dokumentácii a je možné si ju prezrieť. Pri prehliadke postupujeme preto podľa tejto metodiky.

V minuloročnej metodike sú konkrétne návody ako písať kód. Uvediem preto zopár konvencií, ktorých je všeobecne správne sa držať a je vhodné nedostatky týchto konvencií hľadať pri prehliadkach kódu.

- **Mená metód a premenných** - mená musia byť jednoznačné. Ak napadne pri prehliadke niekomu lepšie pomenovanie, nebojte sa napísať svoj návrh.
- **Dĺžka metódy** – 50 riadkov
- **Dĺžka triedy** – 200-300 riadkov
- **Komentovaný kód** - komentovaný kód nemá čo hľadať pri žiadosti o merge novú funkcionality. Ak chcete dať návrh, že niečo treba spraviť, niečo inak atď. Napíšte to do komentára a diskutujte to s ostatnými.
- **Komentáre** - sú dostatočne okomentované všetky metódy, premenné atď? Ak nie, treba nato upozorniť. Je veľmi dobré komentovať aj parametre, ktoré do metódy vstupujú, nie len samotnú funkcionality.

10.2.3 TESTING

- **Testy** - pre každú funkcionality, musí byť dokončené testovanie. Ak nie je, treba počkať dokedy bude.
- **DFD** - spĺňa kód *Definition of Done* (zabezpečuje funkcionality, ktorú má?)

10.2.4 AKO ROBIŤ PREHLIADKU

1. Pýtajte sa
2. Diskutujte (označte tagom v *tf*s daného človeka)
3. Argumentujte
4. Dajte návrhy nato, ako danú vec zlepšiť
5. Nezabudnite pochváliť, ak je to zaslúžené :)

Na záver ešte jedno zlaté pravidlo.

Skúste vždy pred pull requestom spraviť code review sám sebe.

10.3 METODIKA VERZIOVANIA KÓDU

Na manažment verzí využívame TFS od spoločnosti Microsoft, ktorý je priamo integrovaný do Visual Studia.

SLOVNÍK

- **Branch** – vetva s projektovým kódom, prípadne jeho časťou.
- **Commit** – odoslanie zmien do úložiska.
- **Merge** – Zlúčenie vetiev kódu, vyriešenie konfliktov.
- **Konflikt** – vznikne v prípade že rovnaká vetva bola na rovnakom mieste upravená dvoma spôsobmi, resp. v dvoch vetvách.
- **Pull-Request** - teda získanie zmien zo vzdialeného repozitára a ich zamergovanie do hlavnej vetvy.

10.3.1 BRANCHES

Projekt je vetvený do 2 hlavných vetiev (*branches*).

- **master** – vetva obsahujúca iba skontrolovanú a overenú funkcionálnosť. Je to hlavná projektová vetva. O túto vetvu sa stará integrátor projektu a iba on môže pripojiť do tejto vetvy novú funkcionálnosť. Z tejto vetvy sa aplikácia nasadzuje.
- **dev** – je to developerská vetva určená pre verziovanie kódu. Do takejto vetvy je možné poslať iba funkčný kód po teste programátorom. Do tejto vetvy môže pridať funkcionálnosť každý developer, musí však dodržať pravidlo, že všetka funkcionálnosť bude bez problémov fungovať aj s ostatnou v tejto vetve. Ostatní developeri teda nebudú mať problém použiť ju pri práci na svojej úlohe.
- **Pomocné vetvy** – každý developer môže vytvárať svoje pomocné vetvy a pracovať v nich. Každá vetva sa však vytvára za účelom práce na konkrétnom tasku alebo backlog iteme. Avšak ak chce túto vetvu pridať do nadradenej, musí dodržať určité pravidlá spísané nižšie v dokumente. V prípade, že celý tím bude pracovať na konkrétnom okruhu funkcionality, môže sa vytvoriť dočasná developerská vetva, z ktorej si všetci urobia svoju vetvu a pri dokončení sa tieto vetvy spoja. V jednej vetve pracuje jeden človek. Výnimkou však môžu byť určité kedy bude v jednej vetve pracovať viac ľudí ale je potrebný ďalší manažment.

V prípade, že sa developer rozhodne pridať novú funkcionálnosť do master vetvy musí mať kód patrične zdokumentovaný podľa pravidiel dokumentovania. Kód musí byť okomentovaný a otestovaný. Ak je všetko splnené informuje o tom integrátora. Ten projekt skontroluje a ak bude všetko v poriadku spojí vetvy, ak

nie, ohlási nájdené chyby developerovi a ten ich musí opraviť.

Medzi vetvami je možné prepínať sa iba v prípade, že všetky zmeny boli odoslané alebo ak majú vetvy rovnakú verziu. Ak nemajú rovnakú verziu nebude možné prepnúť sa medzi vetvami.

10.3.2 PULL-REQUEST

Pull-Request vetvy podlieha nasledujúcim pravidlám:

1. Funkcionalita vetvy je otestovaná a nenašli sa žiadne chyby.
2. Vetva je bez problémov skompilovateľná.
3. Nadradené sú vždy pravidlá vetvy, do ktorej sa pod-vetva pridáva.
4. Ďalšie pravidlá sú definované v dokumente manažment prehliadok, ktorým musí prejsť každá nová funkcionálnosť.

Každý Pull-Request podlieha procesu schvaľovania tzv. *code review*.

Nikto si nesmie svoj Pull-Request schváliť sám!

Požiadavka o kontrolu, teda nový Pull-Request sa objaví v komunikačnom nástroji. Jeden z developerov (nie autor) si vezme úlohu z boardu ohľadne schvaľovania tejto vetvy a skontroluje či je všetko splnené. Ak áno povolí Pull-Request, ak nie oznámi autorovi pridaním komentáru, prípadne aj v komunikačnom nástroji, aké našiel chyby a nedostatky a ten je povinný ich napraviť.

Ak pri Pull-Requeste nastanú kolízie je potrebné informovať o tom integrátora, ktorý ich vyrieši a urobí merge vetiev.

10.3.3 COMMIT

Commit je vlastne nahranie zmien do repozitára, čím vzniká nová revízia v histórii. Všetky zmeny urobené v počítači konkrétneho developera sa zapisujú do úložiska. Každý commit obsahuje správu ktorá pozostáva zo zoznamu zmien vo funkcionalite spolu s ich krátkym opisom, novú funkcionalitu s opisom a prípadné nezrovnalosti alebo možné budúce chyby a problémy. Zo správy musí byť jasné čo je nové a prečo to bolo vytvorené.

10.3.4 SÚHRN PRAVIDIEL ODOSIELANÉHO KÓDU

Kód, ktorý chce developer pridať do nadradenej vetvy podlieha nasledujúcim pravidlám.

Odosieľa sa výlučne:

- fungujúci kód
- otestovaný kód
- okomentovaný kód
- stabilný kód

10.4 METODIKA TESTOVANIA

Tento dokument určuje spôsob akým je nutné testovať funkčnosť pred tým ako sa zmena v projekte schváli a integruje sa do nadradenej branch. Keďže pracujeme na výskumnom projekte, pre ktorý nie je možné efektívne vyvinúť automatické testovanie, je nutné aby testovanie prebiehalo na čo najľahšie testovateľných častiach.

POJMY

- **Unit testovanie** – unit testovanie, alebo testovanie komponentov je testovanie, ktoré sa zameriava na špecifickú časť kódu, resp. konkrétnu funkčnosť
- **Integračné testovanie** – cieľom integračného testovania je overenie správnej interakcie jednotlivých modulov systému.
- **White box testovanie** – Testovania softvéru zamerané na vnútornú logiku a štruktúru systému.

10.4.1 ZÁKLADNÉ PRAVIDLÁ

- Testovanie je nutné vykonať pre každý pull request
- Každý programátor je zodpovedný za testovanie svojej pridanej funkčnosti
- Je nutné vykonať unit test na pridanú funkčnosť
- Po unit testoch programátor vykoná integračné testovanie
- Po úspešnom otestovaní môže pokračovať vo vytváraní pull request

Programátor nesmie vytvoriť pull request bez otestovania svojej práce.

10.4.2 PRIEBEH TESTOVANIA

Testovanie prebieha v podobe unit testov, ktoré vykonávajú samotní programátori. Pre každý task z backlog si zodpovedný programátor vytvorí branch. V rámci vytvorenej branch pracuje na tasku a po dokončení programovania a okomentovaní kódu otestuje svoju prácu. Testovanie prevedie hlavne na svoju časť kódu, ale taktiež otestuje celú integritu kódu po vytvorenej zmene.

10.5 METODIKA PÍSANIA DOKUMENTÁCIE

Tento dokument slúži ako návod na písanie dokumentácie.

10.5.1 ZÁKLADNÉ POŽIADAVKY

Dokumentácia je vytváraná v nástroji MS WORD (Microsoft Word) od spoločnosti Microsoft, tento nástroj je zahrnutý v balíku MS Office, ktorý musí byť licencovaný. Dokumentácia musí byť písaná v slovenskom jazyku a v prípade anglických výrazov je nutné aby boli v dokumente preložené. Dokumentácia musí byť písaná pomocou diakritiky. V prípade, že dokonale neovládate slovenský pravopis alebo ho ovládať nechcete využívajte pomocný nástroj na kontrolu pravopisu, ktorý je integrovaný v MS WORD.

10.5.2 ZÁKLADNÉ PRAVIDLÁ

- **Font písma** – Times New Roman
- **Veľkosť písma** – 11
- **Farba písma** - čierna
- **Riadkovanie** – 1,5
- **Hlavný nadpis** – na titulný nadpis dokumentu používajte nadpis typu Názov
- **Podnadpis** – na podnadpis dokumentu používajte nadpis typu Nadpis 1
- **Zoznam** – v prípade zvolenia zoznamu využívajte odrážky
- **Obrázky** – Obrázok musí mať vždy priradený popis a musí byť aj správne očíslovaný
- **Diagramy** – Diagram musí mať vždy priradený popis a musí byť správne očíslovaný
- **Tabuľka** – Tabuľka musí mať vždy priradený popis a musí byť správne očíslovaná
- **Popis** – Popis k obrázkom, diagramom a tabuľkám musí byť v tvare objekt. Číslo – opis
- **Umiestnenie popisu** – popis musí byť umiestnený vždy pod objektom, v prípade, že sa nezmestí je nutné objekt s obrázkom vložiť na ďalšiu stranu

- Kapitola – každá nová kapitola musí začínať na novej strane, je nutné aby pred ňou bol zlom strany
- V prípade odkazovania sa na nejaký dokument je potrebné aby odkaz obsahoval hyperlinku

10.5.3 SPRÁVA DOKUMENTÁCIE

Každý dokument musí byť nahraný na spoločné úložisko Google Drive, aby bola prístupná aj ostatným členom tímu, Na tomto úložisku sa nachádza finálny dokument, do ktorého bude váš dokument v prípade, že je vyhovujúci zaintegrován. Dokument na Drive nazývajte spôsobom čosomspravil_ktotospravil. V prípade nejasností kontaktuje manažéra dokumentácie, ktorý s vami bude problémy riešiť.

11.METODIKA PÍSANIA DOXYGEN DOKUMENTÁCIE

Tento dokument slúži ako návod na písanie dokumentácie Doxygen.

PRAVIDLÁ

Na začiatok súboru sa pridáva dokumentácia k danému súboru vrátane názvu súboru, pričom:

- @file sa používa na deklaráciu názvu súboru. Tento príkaz je dôležitý, pretože ak sa v špecifických prípadoch vynechá, celý súbor bude z dokumentácie vynechaný.
- @brief sa používa na stručný popis daného súboru.
- @author špecifikuje meno autora daného súboru. Ak má program viac autorov, každého autora je potrebné zdokumentovať zvlášť prostredníctvom príkazu @author.
- @date špecifikuje dátum vytvorenia súboru.
- @ref vytvára referenciu (odkaz) na sekciu/podsekciiu alebo stránku.
- @see začína odstavec, v rámci ktorého môže byť špecifikovaná jedna alebo viac krížových referencií na súbory, funkcie, premenné, URL adresy, prípadne triedy a metódy (v objektovom programovaní).

Každá funkcia by mala mať pred jej definíciou uvedenú vlastnú dokumentáciu, pričom:

- @brief je stručný popis, ktorý končí odstavcom, a teda popis uvedený za týmto odstavcom je považovaný za detailný popis.
- @param popisuje konkrétny parameter. Ak má funkcia viac parametrov, každý

parameter je potrebné zdokumentovať zvlášť prostredníctvom príkazu @param.

- @return popisuje návratovú hodnotu danej funkcie.
- @author a @date je možné použiť aj v rámci dokumentácie funkcie.

FORMA ZÁPISU

CLASS

Takéto komentáre sa píše iba do hlavičkových súborov „*.h“.

Tento komentár by mal obsahovať stručný opis danej classy a taktiež meno autora.

Tento komentár by mal byť písaný v takomto tvare.

```
/*! \class BirkysMethodReconstruction BirkysMethodReconstruction.h "BirkysMethodReconstruction.h"
 * \brief This is a derived class from class Reconstruction.
 * Contains methods and variables for 3D reconstruction using normal estimation, normal histogram,
 * point labeling/coloring, region growing.
 *
 * \author Robert Birkus
 */
```

Obrázok 1 Názorná ukážka hlavného komentára pre classu

FUNKCIA

Komentár funkcia – tento komentár by mal obsahovať stručný opis toho, čo daná funkcia robí a taktiež jej vstupné a výstupné parametre a mal by byť písaný v takomto tvare.

```
/*!
 * \brief function for normal estimation of a point cloud
 * @param [in] cloudXYZ - XYZ point cloud pcl::PointCloud<pcl::PointXYZ>::Ptr
 * @param [in] numberOfCores - number of cores for the parallel computation
 * @param [in] numberOfNeighbours - number of neighbours considered during normal estimation computation
 * @param [out] normals of the input point cloud pcl::PointCloud<pcl::Normal>::Ptr
 */
```

Obrázok 2 Názorná ukážka komentára pre funkciu

PREMENNÁ

Komentár premennej – tento komentár by mal obsahovať stručný opis toho, na čo daná premenná slúži a mal by byť písaný v takomto tvare

```
std::vector<int> pointIdxRadiusSearch; /*!< Neighbours indeces of the searched point*/
```

Obrázok 3 Názorná ukážka komentára premennej