

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 19 Bratislava 4

Prepájanie dát o vývoji softvéru

Dokumentácia k riadeniu projektu

Členovia tímu č. 8:

- Bc. Peter Bobovský
- Bc. Michal Kráľ
- Bc. Peter Kučera
- Bc. Marek Mura
- Bc. Miriama Pomffyová
- Bc. Lukáš Račko
- Bc. Michal Slovík

Predmet: Tímový projekt II
Vedúci: Ing. Martin Konôpka
Akademický rok: 2016/2017

Obsah

1. Úvod	5
2. Členovia tímu	5
2.1 Rozdelenie manažérskych úloh	6
2.2 Rozdelenie vývojárskych úloh	7
3. Aplikácie manažmentov	8
3.1 Manažment dokumentácie	8
3.2 Manažment testovania	8
3.3 Manažment komunikácie	9
3.4 Manažment chýb	9
3.5 Manažment úloh	9
3.6 Manažment prehliadok kódu	10
3.7 Manažment verzií	10
4. Sumarizácia šprintov	11
4.1 Šprint 1 Anthriscus yellows	11
4.2 Šprint 2 Banna	11
4.3 Šprint 3 Chlamydia	11
4.4 Šprint 4 Diarrhoea	12
4.5 Šprint 5 Ebola	13
4.6 Šprint 6 Fiji	14
4.7 Šprint 7 Giardia	15
4.8 Šprint 8 Herpes	15
4.9 Šprint 9 Influenza	16
4.10 Šprint 10 Jumping Frenchmen of Maine	17
4.11 Šprint 11 Kernicterus	17
4.12 Šprint 12 Lupus	18
4.13 Šprint 13 Malaria	19
4.14 Šprint 14 Nausea	19
4.15 Šprint 15 Obesity	20

4.16 Šprint 16 Pertussis	20
4.17 Šprint 17 Quadrantanopia	21
4.18 Šprint 18 Rotavirus	21
4.19 Šprint 19 Scoliosis	22
4.20 Šprint 20 Tuberculosis	22
4.21 Globálna retrospektíva	22
5. Používané metodiky	24
5.1 Metodika dokumentácie	25
5.2 Metodika písania a komentovania zdrojového kódu	27
5.3 Metodika úloh	31
5.3.1 Vymedzenie pojmov – skratiek	31
5.3.2 Definovanie jednotlivých úrovní úloh	31
5.3.3 Súvisiace – nadväzujúce metodiky	32
5.3.4 Diagram nadväznosti procesov	33
5.3.5 Procesy	34
5.4 Metodika komunikácie	36
5.5 Metodika manažmentu chýb	37
5.6 Metodika manažmentu verzií	38
5.7 Metodika testovania	44
5.8 Metodika prehliadok kódu	47
6. Zápisnice	49
TÍMOVÝ PROJEKT TRACKS 1	49
TÍMOVÝ PROJEKT TRACKS 2	54
TÍMOVÝ PROJEKT TRACKS 3	58
TÍMOVÝ PROJEKT TRACKS 4	60
TÍMOVÝ PROJEKT TRACKS 5	62
TÍMOVÝ PROJEKT TRACKS 6	64
TÍMOVÝ PROJEKT TRACKS 7	66
TÍMOVÝ PROJEKT TRACKS 8	68
TÍMOVÝ PROJEKT TRACKS 9	70

TÍMOVÝ PROJEKT TRACKS 10	72
TÍMOVÝ PROJEKT TRACKS 11	74
TÍMOVÝ PROJEKT TRACKS 12	76
TÍMOVÝ PROJEKT TRACKS 13	77
TÍMOVÝ PROJEKT TRACKS 14	79
TÍMOVÝ PROJEKT TRACKS 15	81
TÍMOVÝ PROJEKT TRACKS 16	82
TÍMOVÝ PROJEKT TRACKS 17	84
TÍMOVÝ PROJEKT TRACKS 18	85
TÍMOVÝ PROJEKT TRACKS 19	87
TÍMOVÝ PROJEKT TRACKS 20	89
TÍMOVÝ PROJEKT TRACKS 21	90
TÍMOVÝ PROJEKT TRACKS 22	92
TÍMOVÝ PROJEKT TRACKS 23	93

1. Úvod

Táto dokumentácia opisuje prácu na tímovom projekte s názvom Prepájanie dát o vývoji softvéru. Ide o zadanie z predmetu Tímový projekt, na ktorom sa učíme, ako pracovať na projektoch v tíme, kde každý člen vypracováva svoje úlohy ako samostatne, tak aj za pomoci ostatných. Dokumentácia opisuje každého člena tímu a jeho príspevok práce na projekte, vďaka ktorým sa snažíme dosiahnuť využitie potenciálu ľudí pre splnenie zadania.

Vedúcim nášho tímu je Ing. Martin Konôpka a názov nášho tímu je Tím Blank. Na projekte pracujeme agilnou metodikou Scrum, pričom trvania našich šprintov sú dva týždne. Úlohy, ktoré majú členovia tímu, sú spísané v tejto dokumentácii riadenia a manažmentu. Taktiež sú v tomto dokumente zhrnuté metodiky, sumarizácia jednotlivých šprintov a zoznam zápisníc zo stretnutí.

2. Členovia tímu

Bc. Peter Bobovský

Venuje sa programovaniu v jazyku C#, grafickému programovaniu a popri tom písaniu anglických poviedok. Študuje inžiniersky odbor informačné systémy na FIIT STU

Bc. Michal Král'

Má skúsenosti s multithreadovými riešeniami, databázami SQL a NoSQL (MongoDB) a frameworkami JEE a GWT. Väčšinu svojich skúseností nadobudol v práci, kde sa venuje webovým aplikáciám (frontend aj backend). Svoj voľný čas trávi outdoorovým lezením.

Bc. Peter Kučera

Nadšenec pre mobilné technológie. Širšie skúsenosti má s vývojom na OS Android, OpenCV alebo programovaním v jazyku Python. Voľný čas rád trávi PC hrami, knihou alebo hrou na hudobný nástroj.

Bc. Marek Mura

Má skúsenosti so skriptovaním automatizovaných testov v jazyku VisualBasicScript, ako aj s tvorbou neurónových sietí v jazyku Python, ktorým sa venoval v rámci bakalárskej práce. Voľný čas trávi programovaním hier a štúdiom japonského jazyka.

Bc. Miriama Pomffyová

Je absolventkou bakalárskeho študijného programu Informatika na Fakulte informatiky a informačných technológií a v bakalárskej práci sa venovala spracovaniu dát (spotreba elektrickej energie) a predikcii. Riešila spracovávanie vstupov a výstupov, vyhodnocovaniu algoritmov na predikciu časových radov v jazyku R. Pracovala na aplikáciách v jazyku C# a Java s využitím technológií HTML, CSS a JavaScript.

Bc. Lukáš Račko

Venuje sa programovaniu prevažne v jazyku Java a vývoju mobilných aplikácií pre operačný systém Android. Má skúsenosti aj s tvorbou webových aplikácií s využitím PHP a Javascriptu. Vo voľnom čase sa rád odreaguje na bicykli, alebo pri stolnom tenise. Je študentom Softvérového inžinierstva na STU FIIT.

Bc. Michal Slovík

Sa venuje programovaniu mobilných aplikácií na platformu Android a svoje vedomosti si vo voľnom čase rozširuje s platformou Linux (Ubuntu) na RaspberryPi. Spomedzi novších technológií ho najviac zaujal Python, ktorý využíva pri svojich aktuálnych projektoch. V bakalárskej práci sa venoval zdieľaniu mobilných služieb na počítače, cez ktoré potom dokážu používatelia komunikovať pohodlnejšie.

2.1 Rozdelenie manažérskych úloh

V tejto kapitole je zobrazené pridelenie krátkodobých a dlhodobých úloh

Meno	Rola v tíme
Peter Bobovský	Manažér komunikácie.
Michal Král'	Manažér úloh.
Peter Kučera	Manažér prehliadok kódu
Marek Mura	Manažér chýb a kvality.
Miriama Pomffiová	Manažér dokumentácie a rozsahu projektu.
Lukáš Račko	Manažér verzií a zdieľaných prostriedkov.
Michal Slovík	Manažér testovania a webovej stránky

2.2 Rozdelenie vývojárskych úloh

Meno	Úlohy
Peter Bobovský	Dohliadanie na komunikáciu, udržiavanie Discordu.
Michal Kráľ	Podpora pre MSSQL databázu, udržovanie a riadenie. Webový frontend aj backend.
Peter Kučera	Vedúci vývoja klientskej aplikácie.
Marek Mura	Korektúra písomností, dohliadanie na ontológie.
Miriama Pomffyová	Dohliadanie na tvorbu zápisníc, dodržiavanie termínov.
Lukáš Račko	Podpora a udržiavanie Apache Jena Fuseki.
Michal Slovík	Správa webovej stránky a zdieľanie dokumentov. Zadávanie úloh do TFS.

3. Aplikácie manažmentov

V tejto kapitole sa nachádza opis manažmentov, pomocou ktorých prebieha uskutočňovanie projektu. Druhy manažmentov sú doplnené o používané metodiky v neskoršej kapitole.

3.1 Manažment dokumentácie

Dokumentácia je podstatnou časťou projektu pri vývoji a následnom používaní produktu. Preto, aby bola pre ostatných členov tímu a vedúceho tímu čo najprínosnejšia a naučiteľnejšia, každý člen tímu vypracováva pridelenú tému dokumentácie tak, aby bola stručná a vecná. Vďaka dokumentáciám má každý člen tímu predstavu o práci jednotlivých členov tímu a vie sa zorientovať v problematike.

Dokumentácie vypracovávame v slovenčine a zdrojové kódy a komentáre píšeme v anglickom jazyku.

Riadenie procesu dokumentovania:

- Na každom stretnutí sa určí 1 člen tímu, ktorý píše zápisnicu – píše, čo sa riešilo na stretnutí, kto sa zúčastnil a kto nie, kto mal aké nápady a pripomienky, prejdú sa zadané úlohy, či sú splnené / nesplnené, zápis rozhodnutí, ktoré boli prečo a ako urobené. Následne zápisnicu uloží na GoogleDisk tímu.
- Na každom stretnutí zapisujeme zadávané úlohy do TFS.
- Jednotlivé dokumentácie vypracovávajú členovia tímu po ich priradení.

Proces tvorby dokumentácie opisuje metodika dokumentácie. Súčasťou dokumentácie je aj komentovanie a písanie zdrojového kódu. Pravidlá písania komentárov a kódu opisuje metodika Písania a komentovania zdrojového kódu.

3.2 Manažment testovania

Testovanie nám umožňuje overiť správnosť vytváraného systému alebo jeho komponentov a verifikovať splnenie požiadaviek, ktoré boli špecifikované.

Funkčné testovanie rozdeľuje do niekoľkých častí. Prvým dvom sa venuje samotný programátor: **Unit Testy** a **Integračné testovanie**. Oba testy vytvára a spúšťa programátor. Unit testami sa overuje správnosť fungovania najmenších častí systému. Tieto testy sa vykonávajú vo vývojom prostredí a je možné ich automatizovať. Integračnými testami overuje programátor správnosť interakcie (spolupráce) jednotlivých modulov systému.

Ďalšia dva testy nevykonáva programátor. Ide o **systémové testovanie** a **regresné testovanie**. Pri systémovom testovaní sa testuje systém ako celok. Test sa vykonáva v prostredí, ktoré je podobné produkčnému prostrediu, pričom sa testuje splnenie funkčných a technických špecifikácií. Tieto testy vykonáva špecializovaný tím. Pri regresnom testovaní je cieľom otestovať, či odstránením chyby alebo zmeny v kóde nedôjde k vzniku novej chyby.

Posledný druh testu by mal vykonávať zákazník a ide o **akceptačné testovanie**, kde sa vykonávajú pripravené scenáre a testovacie prípady.

3.3 Manažment komunikácie

Komunikácia medzi jednotlivými členmi tímu navzájom a medzi členmi tímu a vedúcim projektu je kľúčovou aktivitou pre dosiahnutie správneho fungovania tímu. Vďaka dobrej komunikácii dokážeme predísť nedorozumeniam, jednotliví členovia tímu môžu rýchlejšie napredovať pri plnení svojich úloh, pretože ak sa zaseknú na nejakom probléme, po krátkom odkonzultovaní sa problém môže rýchlejšie odstrániť.

Komunikačné platformy v tíme fungujú dve, jedna formálna a jedna menej formálna. Formálnou komunikačnou platformou sú emaily prostredníctvom googlegroup, ku ktorej sú všetci prihlásení, a ktorá distribuuje emaily všetkým členom, ako aj umožňuje oficiálnu komunikáciu s productownerom, prípadne s inými osobami.

Neformálna komunikačná platforma je discordový server s kategorizáciou kanálov ako aj plnou históriou. Tento server podporuje pripojenie aj z mobilu, a tak sú členovia tímu zastihnuteľní všade.

3.4 Manažment chýb

Riadenie požiadaviek na zmenu a manažment chýb v softvéri:

- v prípade kontrolovania splnenia úlohy člena tímu, ten, kto kontroluje, spíše požiadavky na zmenu a chyby nájdené v kontrolovanej časti projektu – informuje o nich vykonávateľa úlohy, priloží na ďalšom stretnutí tomu, kto zapisuje – súčasť retrospektívy.
- v prípade stretnutia s vedúcim tímu (zákazníkom), ten, kto píše zápisnicu, spíše požiadavky na zmenu a chyby v rozpracovaných úlohách, poprípade v už uzatvorených.
- Vzniknuté požiadavky a chyby sa zapíšu do TFS:
 - Nové User Story a Tasks.
 - Neakceptovanie splnenia starých Tasks a presun do nasledujúceho šprintu.

3.5 Manažment úloh

Definovanie úloh je najzákladnejšia jednotka pri tvorbe softvéru. Predtým, ako začneme niečo programovať musíme vedieť, čo chceme naprogramovať.

Preto je potrebné na začiatku mať s product ownerom pár prvotných sedení, na ktorých prevedieme jeho ideu do epicov, ktoré nám poslúžia na neskoršiu analýzu projektu.

Prvotné sedenia prebiehajú dovtedy, pokiaľ nebude jasná základná myšlienka vyvíjaného softvéru.

Keď sa spolu s product ownerom definujú základné epicy, pokračuje sa robením jednotlivých epicov do tzv. feature.

Tak ako pri tvorbe epicov, tak aj pri tvorbe feature musí byť dodržaná ich granularita.

Po definovaní základnej funkcionality začínajú šprinty. Ako prebieha plánovanie úloh v rámci šprintu je opísané v metodike úloh.

3.6 Manažment prehládok kódu

Pri vývoji sa ucelené časti funkcionality vyvíjajú v zvlášť vetvách vývoja. Pri zlúčení vedľajších vetiev do hlavnej vetvy vývoja je potrebné nie len aby kód prešiel testami, ale aj skontrolovanie kódu iným členom tímu. Skontrolovaný kód je následne možné vrátiť autorovi na opravenie alebo v prípade, že člen tímu, ktorý prehliadku robí nemá voči danému kódu výhrady, môže ju automaticky zlúčiť s hlavnou vetvou vývoja.

Zmyslom takejto prehliadky kódu je odhalenie pachov v kóde, poukázanie na nedodržanie konvencií alebo prípadný návrh na vylepšenie kódu. Výsledkom procesu je kód, ktorý zodpovedá konvenciám a je často robustnejší a efektívnejší, keďže sa na jeho tvorbe a kontrolovaní podieľal okrem autora ešte aj člen tímu, ktorý prehliadku vykonával.

Na prehliadky kódu využívame funkcionality verziovacieho systému GIT - Pull requests (Bližšie popísané v metodike prehliadok kódu).

3.7 Manažment verzií

Jednotlivé funkcionality sa vyvíjajú v samostatných vetvách, do ktorých sa postupne pridávajú nové prvky, až kým nie sú splnené požiadavky na User Story. Čiastkové zmeny sa postupne commitujú, čím sa vytvárajú verzie, ku ktorým sa je možné kedykoľvek v prípade potreby vrátiť.

Po skončení práce na ucelenej funkcionalite vybraný člen alebo členovia skontrolujú správnosť a funkčnosť pridanej funkcionality. Keď sa nenájdu žiadne chyby, daná vetva sa zlúči s hlavnou vývojovou vetvou.

Pre správu verzií používame TFS Git. Bližšie informácie k používaniu Gitu sú uvedené v metodike manažmentu verzií.

4. Sumarizácia šprintov

4.1 Šprint 1 Anthriscus yellows

V prvom šprinte bolo naším cieľom rozdeliť si úlohy v rámci tímu. Každému členovi tímu bola pridelená určitá oblasť projektu, ktorej sa bude primárne venovať. Oboznámili sme sa so súčasným stavom a cieľmi projektu, ako aj s technológiami, ktoré budeme používať pri jeho realizácii.

Zároveň sme sa v tomto šprinte dohodli na finálnom názve nášho tímu a vytvorili sme spoločný plagát. Spísali sme tiež metodiku používania gitu a konvencie písania kódu. Zvolili sme Discord ako nástroj pre neformálnu tímovú komunikáciu a oboznámili sa so systémom TFS.

Zhodnotenie šprintu

- + Spísali sme metodiky používania gitu a code conventions
- + Dohodli sme sa na spôsobe komunikácie
- + Podarilo sa nám rozbehať projekt a pridať skúšobné zmeny
- + Všetci sme sa prihlásili do systému TFS

- Nestihli sme vytvoriť stránku tímu
- Úlohy sme neohodnotili story pointami, neboli vyplnené ani odpracované hodiny

4.2 Šprint 2 Banna

V druhom šprinte bolo našou úlohou vytvoriť webstránku tímu, na ktorej budú umiestnené informácie ohľadne členov tímu, cieľov projektu a stavu, v akom sa nachádza. Zároveň nastal čas oboznámiť sa s databázovými technológiami apache-jena-fuseki, do ktorej budú ukladané dáta vo formáte RDF trojíc.

- + Webstránka tímu bola úspešne vytvorená
- + Úspešné lokálne rozbehnutie apache-jena-fuseki

- Stránka bola zatiaľ vytvorená bez požadovaného obsahu

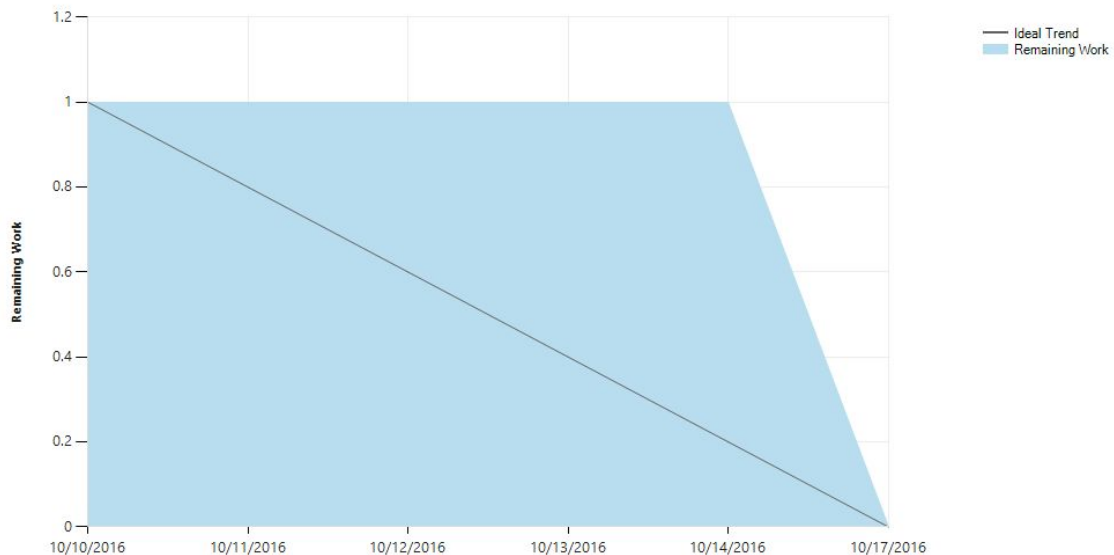
4.3 Šprint 3 Chlamydia

V tomto šprinte sme sa naplno venovali stanoveniu jednotlivých metodík (napríklad pre prácu s Gitom, kód, metodika pre uzatváranie taskov). Ďalšou úlohou bolo rozbehnutie aktuálneho projektu vo Visual Studiu.

Zároveň bola začatá práca s MSSQL databázou a jej komunikácia s balancerom. Bol vytvorený prototyp klienta, ktorý komunikuje s balancerom.

- + Funkčný prototyp klienta
- + Metodiky
- + MSsqldatabáza

- skrytie klientskej aplikácie v tray
- analyzovanie existujúcich API na sťahovanie z jednotlivých repozitárov

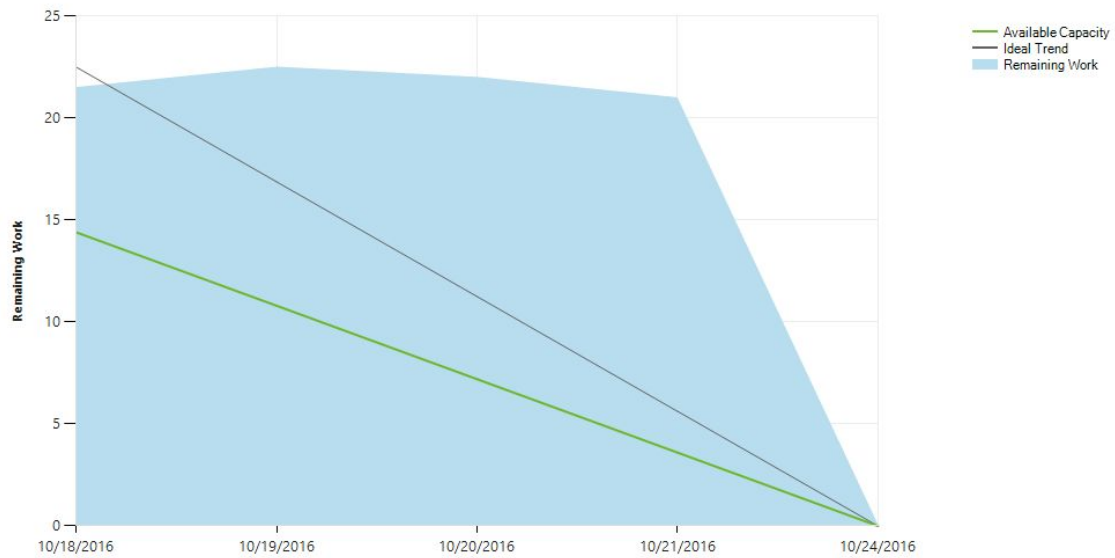


4.4 Šprint 4 Diarrhoea

V tomto šprinte sme si zvolili repozitáre na sťahovanie a spustili databázy pre ukladanie dát (apache-jena-fuseki). Vytvorenie DLL knižnice zodpovednej za sťahovanie repozitárov. Vytvorenie prihlášky na TP Cup.

- + implementácia controllera.
- + DLL knižnica
- + API pre BugZillu

- prihláška na TP Cup nebola skontrolovaná všetkými členmi

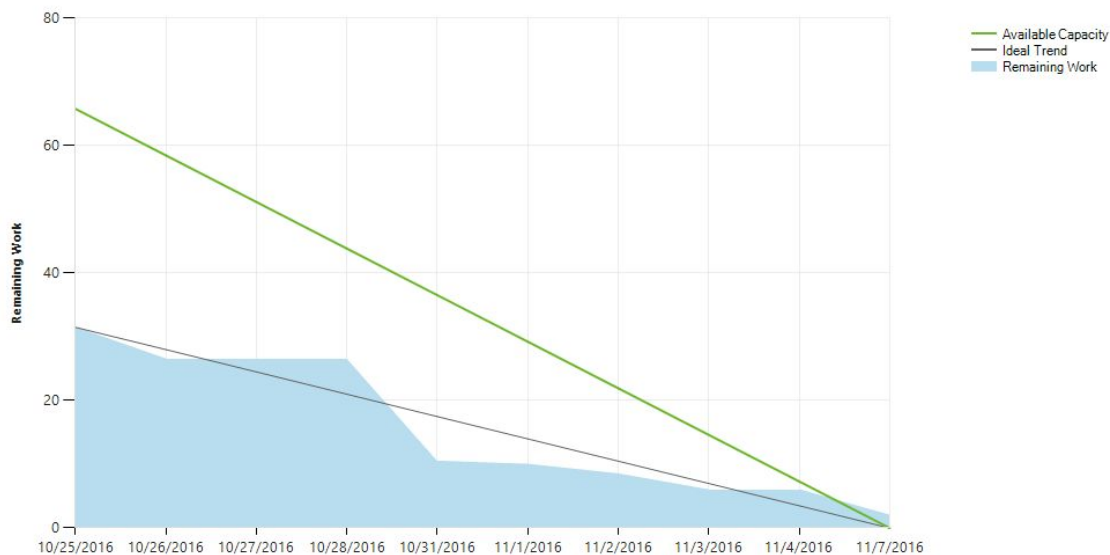


4.5 Šprint 5 Ebola

Prvý dvojtýždňový šprint. Rozhodovanie o implementácii modelu relačnej databázy. Zhodnutie sa na dvojtýždňovom šprinte a stanovenie prvého míľniku na 10. semestrový týždeň.

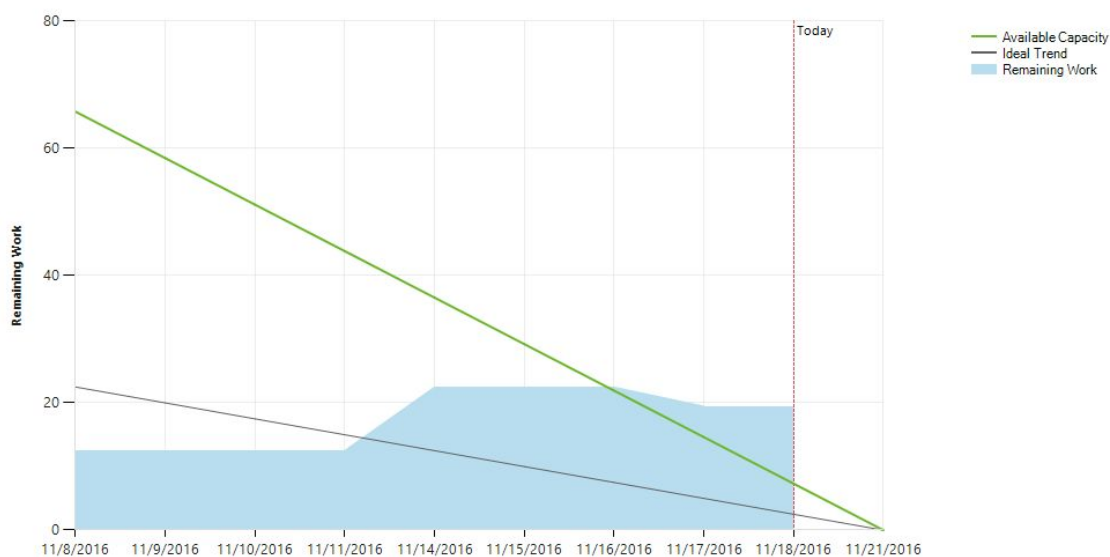
- + pridanie configItemu
- + vytvorenie controllera pre update už vytvoreného repozitára
- + Kontrola RDF podľa OWL ontológií
- + vytvorenie komunikácie s DLL
- + vytvorenie komunikácie medzi klientom a balancerom

- analýza stratégií pre sťahovanie jobov



4.6 Šprint 6 Fiji

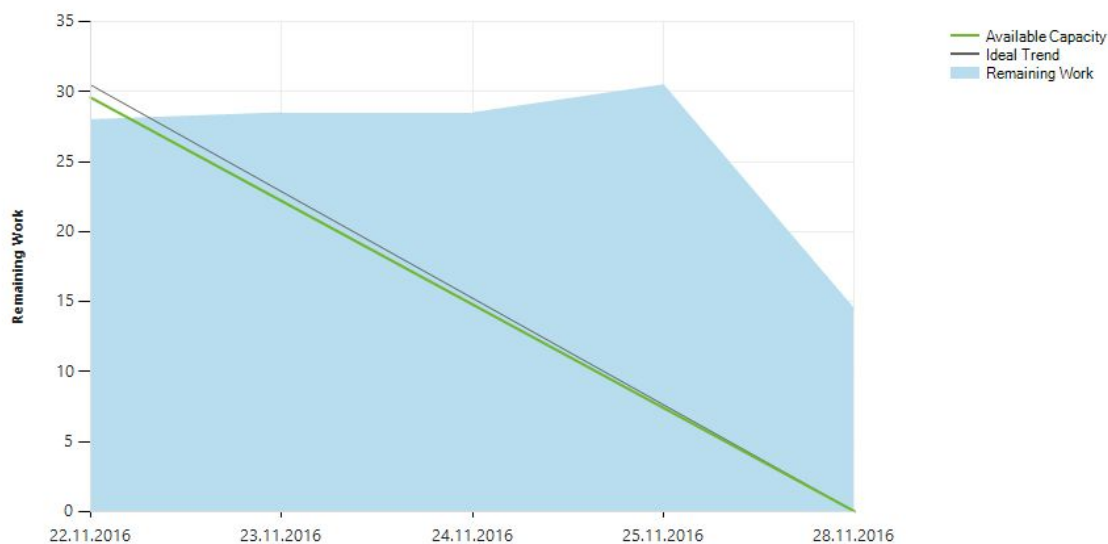
Aktuálny šprint (v čase odovzdania prvého kontrolného bodu). Je potrebné spísať ďalšie metodiky. Implementovať službu na sťahovanie stackOverFlow. Vytvorenie ontológie, podľa ktorej sa budú kontrolovať JobUri a TripleUri. Ďalej je potrebné vytvorenie kontroly dostupnosti klienta a jeho stav sťahovania. Zistiť aktuálny API pre Git Eclipse a Gerrit.



- + sťahovanie zo StackOverflow
- + oznamovanie dostupnosti klientov na server
- + úspešne odovzdaný prvý kontrolný bod
- komunikácia s Bugzillou
- písanie si hodín do tfs
- nesprávne uzatváranie taskov v TFS

4.7 Šprint 7 Giardia

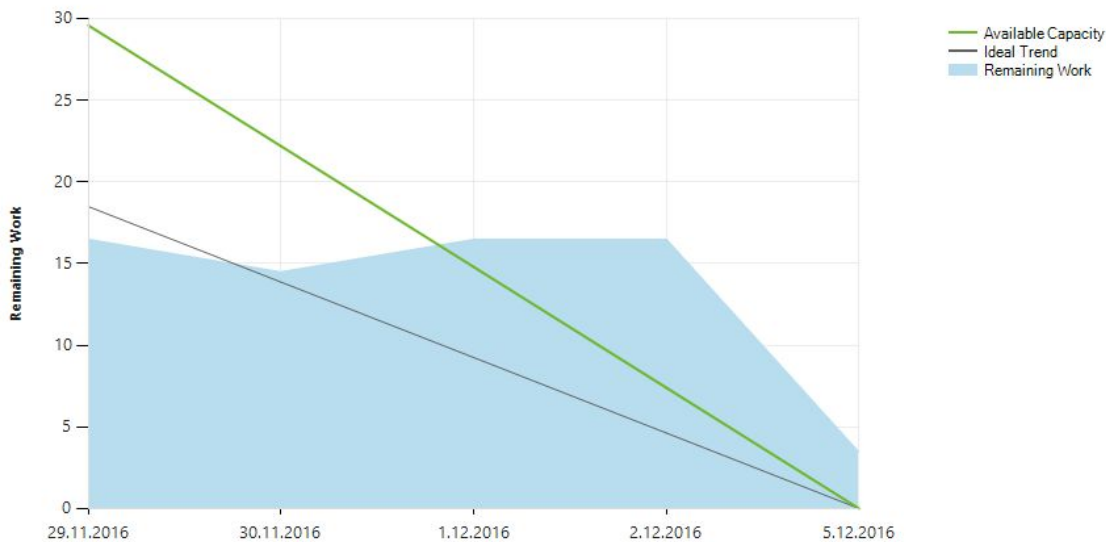
V tomto šprinte sme sa venovali primárne príprave na mentoring ohľadom biznisu, agilného vývoja a riadenia tímu, ku ktorému sme vypracovali prezentáciu a súhrn otázok týkajúcich sa agilného vývoja. Okrem toho sa podarilo v klientovi implementovať získavanie JobExecutions od servera a pridelovanie JobExecutions voľným klientom.



- + úspešná príprava a absolvovanie mentoringu
- + server prideluje JobExecutions klientovi
- + klient získava JobExecutions od servera

4.8 Šprint 8 Herpes

V tomto šprinte sme prvýkrát vytvorili funkčné dll pre komunikáciu s Bugzillou a Gitom pomocou HTTP requestov na REST api daných repozitárov. Zároveň sa podarilo vytvoriť z dát stiahnutých zo StackOverflow RDF trojice, vytvorenie použiteľnej ontológie ostáva na ďalší šprint.

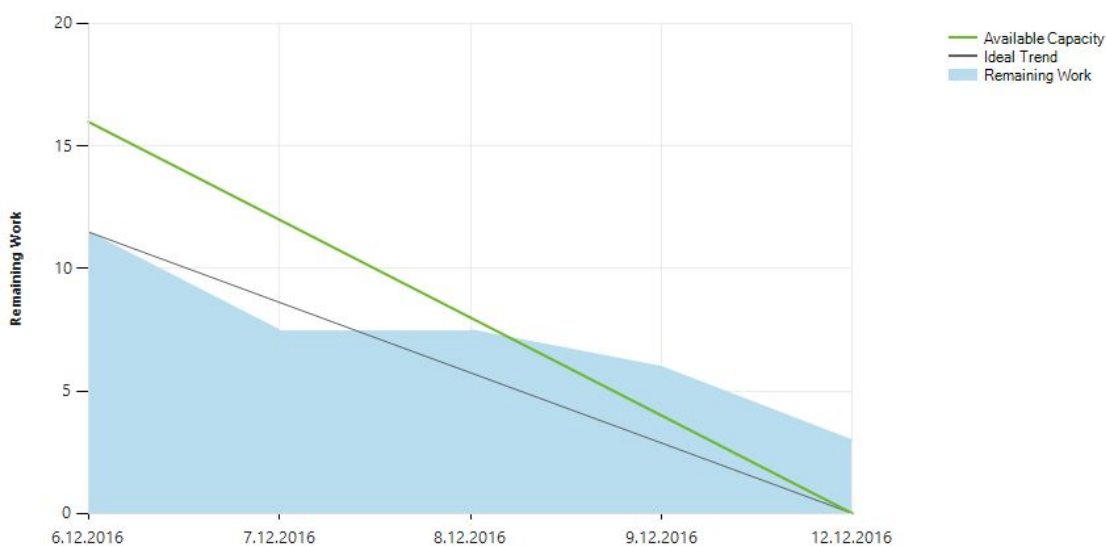


- + spojzdená základná komunikácia s viacerými repozitármi
- + ukladanie dát zo StackOverflow do RDF trojíc
- + klient dokáže stiahnuť Job Executions
- + klient dokáže zaznačiť do databázy, že úloha bola dokončená
- + klient zobrazuje vykonávané joby a mieru vyriešenosti

- ontológie sa nepodarilo dokončiť

4.9 Šprint 9 Influenza

Tento šprint je posledným šprintom v tomto semestri a zároveň pokrýva čas odovzdania druhého kontrolného bodu, ktorý celý tím kolaboratívne vypracoval. Hlavnou náplňou šprintu sú teda tasky týkajúce sa vypracovávaní dokumentácie. Okrem toho sme dokončili niekoľko taskov, vypracovali a úspešne odprezentovali prezentáciu manažmentov na predmet Manažment v IS/SI.



- + ontológie pre StackOverflow, odosielanie stavu dokončenosti daemonovi
 - + klient sa ozve serveru, keď skončí sťahovanie
 - + klient zobrazuje dokončenosť úloh percentuálnym pomerom
 - + template pre unit testy, prvé funkčné unit testy
- úspešné odprezentovanie manažmentov

4.10 Šprint 10 Jumping Frenchmen of Maine



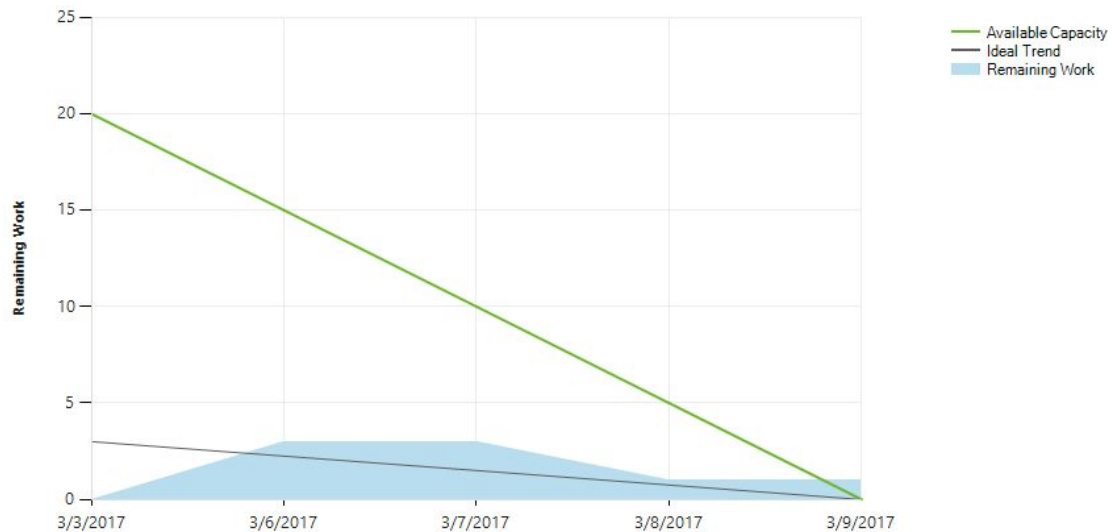
- + vytváranie RDF trojíc pre StackOverflow, BugZilla, Git
- + Spojaznenia servera online
- + spojaznenie CI

4.11 Šprint 11 Kernicterus



- + nasadenie Apache Jena Fuseki
- + spojznenie servera online
- + riešenie bugov, získavania klient ID od balancera
- + logger na unix serveri

4.12 Šprint 12 Lupus



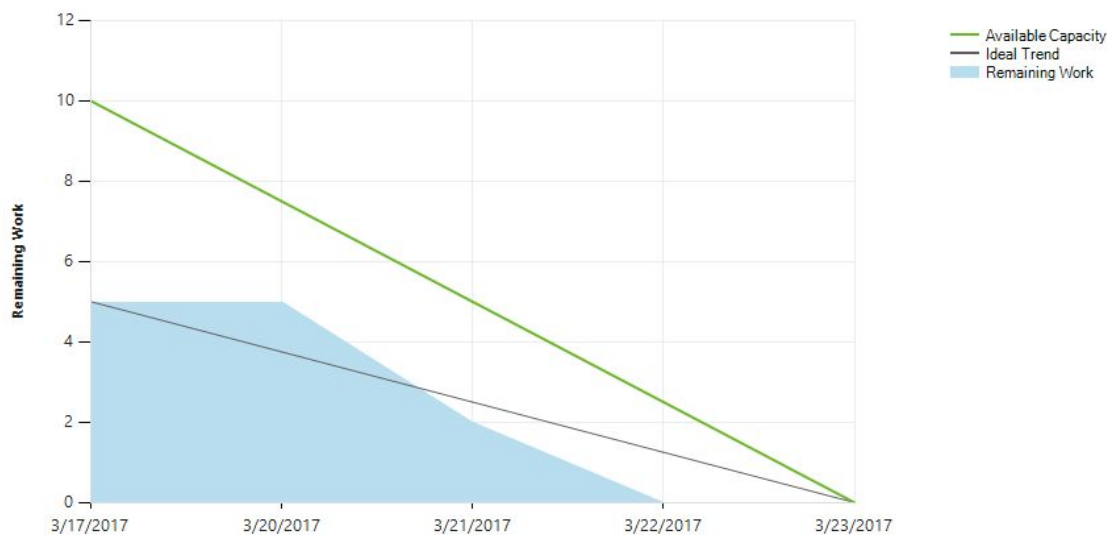
- + odskúšanie novej DB na školskom serveri
- + rozbaľovanie zip súborov z rôznych nástrojov - vytvorenie nástroja na zipovanie
- + riešenie bugov s CI a unix serverom

4.13 Šprint 13 Malaria



- + pridanie loadera po kliknutí na štart projektu, úprava progress bar
- + uprava gitignore

4.14 Šprint 14 Nausea



- + prerobenie dll na prácu s IGrافu
- + update článku na IIT.SRC
- + inštrukcie k vypíňaniu DLL
- + úprava grafického rozhrania klienta

4.15 Šprint 15 Obesity



- + stanovenie priorít pre sťahovanie jobExecutions
- + vyrobenie nugetu
- + oznámenie o úspešnom ukončení alebo naplnení limitu
- + upravenie ukladanie dát do Jeny

4.16 Šprint 16 Pertussis



- + Bugy s klientom (prideľovanie ID, findJob)
- + Zastavenie sťahovania
- + úprava JobExecutions na webe

4.17 Šprint 17 Quadrantanopia



- + Filter na JobExecutions
- + Odblokovanie dll súborov
- + Vytvorenie config file,
- + získavanie a spracovanie IGrafu

4.18 Šprint 18 Rotavirus



- + Vytvorenie plagátu na IITSRC
- + Vytvorenie prezentácie na robime.it
- + Sprístupnenie Jeny a FTP
- + Spísanie príručky klienta a pre server

4.19 Šprint 19 Scoliosis

- + Konferencia IITSRC
- + Spísanie finálnej dokumentácie
- + Vytváranie manuálov pre nasadzovanie servera
- + technická dokumentácia

4.20 Šprint 20 Tuberculosis

- + Otestovanie produktu iným tímom
- + testovanie cudzieho produktu
- + písanie záverečnej dokumentácie

4.21 Globálna retrospektíva

Prvé dva šprinty mali pre náš tím predovšetkým introdukčný charakter. Oboznámili sme sa s naším projektom a jeho hlavnými cieľmi. Všetci členovia tímu si rozdelili manažérske úlohy, vytvorili sme spoločný plagát a webovú stránku. Každý člen tímu sa úspešne prihlásil do systému TFS, ktorý používame na riadenie práce a formálnu komunikáciu. Úspešne sme rozbehali doposiaľ existujúcu časť projektu a postupne sa oboznámili s technológiami, ktoré používame pri práci.

Zároveň bolo pre nás dôležité osvojiť si metodiku scrum a riadiť sa jej pravidlami. Už od počiatku sme rozdeľovali úlohy na epicy, user stories a tasks. Nejasnosti ohľadom pridelovania story pointov jednotlivým taskom spôsobili, že sme tasky v prvých dvoch týždňoch výrazne nadhodnotili. Tím sa postupne oboznámil so spravovaním úloh v systéme TFS, vďaka čomu retrospektívy v nasledujúcich šprintoch prebiehali oveľa hladšie.

V priebehu úvodných dvoch šprintov sa nám podarilo lokálne rozbehať viacero komponentov projektu. Spustili sme existujúci projekt a spojzdnili správu verzií cez systém TFS Git.

V treťom a štvrtom šprinte sme sa venovali finalizácii dokumentácie metodík. Dokončili sme spisovanie chýbajúcich metodík a vytvorili uniformnú šablónu pre zápisnice zo stretnutí. Webová stránka tímu, ktorá bola po prvých dvoch šprintoch vytvorená bez obsahu, bola postupne obohatená o informácie o projekte, členoch tímu a aktuálnom stave nášho postupu.

Zároveň sme začali pracovať na vývoji komponentov projektu. Hlavnými cieľmi bolo spojzdníť mssql databázu a databázu apache-jena-fuseki pre ukladanie RDF-trojíc, keďže tieto technológie sú nevyhnutné pre komunikáciu medzi balancerom a klientom. Medzitým sme vytvorili jednoduchý prototyp klientskej aplikácie, ktorá beží ako daemon, obsahuje základné grafické používateľské rozhranie a po zatvorení je skrytá v system trayi. Zároveň

bola vytvorená dll knižnica pre sťahovanie dát z repozitárov.

Dôležitou udalosťou počas týchto dvoch šprintov bolo odovzdanie prihlášky na TP Cup. Spísali sme prihlášku v zdieľanom textovom dokumente. Všetci členovia tímu nestihli načas prihlášku skontrolovať, preto sme ju odovzdali s krátkym oneskorením. V prihláške tím písomne vyjadril záujem o coaching v súvislosti s riadením vývoja softvéru.

Piaty šprint bol prvým obdobím, kedy sme sa dohodli na dvojtýždňovom trvaní šprintu. V tomto šprinte sa nám podarilo spojazdniť komunikáciu medzi balancerom a klientskou aplikáciou a komunikáciu s dll knižnicami, ktoré obsahujú joby pre sťahovanie jednotlivých repozitárov. Hlavnou úlohou pre šiesty a zároveň súčasný šprint je vytvorenie jobov pre sťahovanie konkrétnych repozitárov, menovite Bugzilla a StackOverflow.

Naším hlavným cieľom v nasledujúcich šprintoch bude spojiť dokopy doteraz vytvorené komponenty projektu a vytvoriť tak funkčný prototyp, ktorý dokáže sťahovať dáta z konkrétnych repozitárov a odosielať ich z klientskej aplikácie na server. Tento cieľ chceme splniť do 10-teho semestrového týždňa, ktorý sme si stanovili ako prvý míľnik.

5. Používané metodiky

Táto kapitola obsahuje metodiky, ktorými sa náš tím pri práci na projekte riadi. Ako prvé si predstavíme roly, ktoré sú platné naprieč všetkými metodikami:

Scrum master

- Organizuje a motivuje tím, aby dokázal pracovať efektívne
- Dohliada na dodržiavanie termínov, dokončovanie úloh
- Snaží sa odstraňovať prekážky, ktoré by mohli spomaľovať členov tímu
- Moderuje diskusie a porady
- Dohliada na správny formát a granularitu jednotlivých úloh vo všetkých úrovniach.
- Komunikuje s projektom manažérom a product ownerom
- Vypĺňa medzery vo vývoji medzi členmi tímu (zaoberá sa vývojom produktu)

Člen tímu

- Vyberajú si úlohy, za ktoré sú zodpovedný
- Implementujú task a k nemu príslušné testy (unit, integračné)
- Pravidelné stretnutia so scrum mastrom, kde prezentujú pokrok, v retrospektíve hodnotia minulé pridelené úlohy a vyberajú si nové úlohy

Tester

- Testuje implementáciu, ktorú tím vytvorí,
- Implementuje testy na rôznych úrovniach
- Pracuje oddelene od členov tímu, pretože členovia tímu objektívne nevedia dostatočne otestovať to, čo implementovali

Product owner

- Ide o vlastníka produktu
- Definuje požiadavky a špecifikácie produktu
- Zodpovedný za biznis časť, že sa vytvára správny produkt
- Je tímu k dispozícii a pomáha rozhodovať o výbere epicov a feature, na ktorých sa bude pracovať.
- Môže akceptovať alebo zamietnuť produkt podľa toho či spĺňa alebo nespĺňa požiadavky

Tím

- Skladá sa z členov vhodných pre daný produkt.
- Pomáha PO identifikovať epicy a feature.
- Identifikuje US a tasky.
- Kolaboratívne sa rozhoduje, ktorý člen bude pracovať na ktorej úlohe.
- Navrhuje a implementuje technické riešenia jednotlivých úloh.
- Každý člen tímu je zodpovedný za časť, ktorú implementoval.

5.1 Metodika dokumentácie

Metodika slúži na opis procesov spojených s vytváraním dokumentácie, ktoré je potrebné dodržiavať pre zachovanie konzistencie dokumentácií. Zároveň metodika pomáha pri vytváraní nových dokumentácií. Táto metodika zobrazuje procesy pri vytváraní dokumentácií.

1. Určenie typu dokumentácie na základe požiadaviek

(Vystupujú tu: product owner, členovia tímu)

Vstupy:

- Nie sú.

Vykonávanie:

- Definovanie témy dokumentácie.
- Určenie typu dokumentácie
 - metodika,
 - zápisnica,
 - príručka,
 - dokumentácia produktu,
 - dokumentácia riadenia.

Ukončenie:

- Zhromaždené požiadavky na dokumentáciu a určený typ dokumentácie.

2. Pridelenie dokumentácie na vypracovanie

(Vystupujú tu: product owner, členovia tímu)

Vstupy:

- Zhromaždené požiadavky na dokumentáciu a určený typ dokumentácie.

Vykonávanie:

- Zhodnotenie spôsobilosti a schopností účastníkov projektu.
- Priradenie dokumentácie účastníkovi projektu.

Ukončenie:

- Pridelená dokumentácia na vypracovanie.
- Určený autor dokumentácie.

3. Spísanie jadra dokumentácie

(Vystupujú tu: vybraný autor z členov tímu)

Vstupy:

- Pridelená dokumentácia na vypracovanie.
- Určený autor dokumentácie.

Vykonávanie:

- Spísanie jadra dokumentácie podľa typu dokumentácie.
- Ak ide o:

- metodiku:
 - Spísanie procesov a ich:
 - vstupy,
 - vykonávanie,
 - výstupy.
- zápisnicu:
 - Spísanie:
 - času a miesta konania stretnutia.
 - zúčastnených a nezúčastnených členov.
 - preberaných bodov.
 - splnených a nesplnených úloh.
- dokumentáciu riadenia:
 - Spísanie:
 - manažmentov,
 - metodík,
 - sumarizácie šprintov,
 - a sumarizácie zápisníc.
- dokumentáciu produktu:
 - Spísanie:
 - analýzy,
 - návrhu,
 - implementácie
 - a overenia riešenia pre daný produkt.
- príručku:
 - Spísanie opisu produktu, ukážky rozhrania a návod pre používanie produktu.

Ukončenie:

- Spísaná dokumentácia autorom dokumentácie.

4. Uloženie a sprístupnenie dokumentácie

(Vystupujú tu: autor dokumentácie)

Vstupy:

- Spísaná dokumentácia.

Vykonávanie:

- Uloženie dokumentácie na Google Disk tímu.
- Sprístupnenie úložiska s dokumentáciou všetkým rolám.
- Sprístupnenie dokumentácie na stránke tímu.

Výstupy:

- Uložená a sprístupnená dokumentácia všetkým rolám.

Nasledujúci diagram popisuje následnosť vykonávania týchto procesov.



Obrázok 1 Diagram aktivít zobrazujúci procesy tvorby dokumentácie

5.2 Metodika písania a komentovania zdrojového kódu

Dobré porovnanie C# k Java na prečítanie

http://www.harding.edu/fmccown/java_csharp_comparison.htmlhttp://www.harding.edu/fmccown/java_csharp_comparison.htmlhttp://www.harding.edu/fmccown/java_csharp_comparison.htmlhttp://www.harding.edu/fmccown/java_csharp_comparison.htmlhttp://www.harding.edu/fmccown/java_csharp_comparison.html

1. Názvy premenných , kapitalizácia písmen

1.1 PascalCase

Príklad:

BackColor

Použitie:

Nazvy namespace, metód, suborov, publicproperties

1.2 camelCase

Príklad:
backColor

Použitie:
Nazvyprivatevariables a parametrov

- Jediná výnimka je pre premenné so skratkami ktore by mali byť veľkým a majú viac ako 2 veľké znaky porade, použiť PascalCase pravidlo (tieto znaky malým)
`HtmlHelperhtmlHelper;`

- Pri premenných typu Boolean je vhodné zvolit prefix "Can","Is" alebo "Has".

- Čitateľnosť názvov premenných je preferovaná pred rýchlosťou napísania
`CanScrollHorizontally>ScrollableX`

- V názvoch premenných sa pokúšať nepoužívať slovíčka z programovacieho jazyka

```
// NamingthefollowingvariableinputIntismisleading.  
// Itis a string.  
varinputInt = Console.ReadLine();  
Console.WriteLine(inputInt);
```

- Pred nazvyInterfacov dať I

1. `publicinterfaceIShape`
2. {
3. }

- Dĺžka riadka obsahujúca kód alebo komentár by nemala presahovať 80 znakov

2. Implementačné konvencie

- Pokúšať sa nechávať iba potrebné množstvo returnov z funkcií (ideálne 1)

- V podmienkach zátvorkovať celky aby bolo jasné čo ku čomu patrí

```
if ((val1 > val2) && (val1 > val3))  
{  
// Takeappropriateaction.  
}
```

- V prípade viacej ako dvoch podmienok v jednom ife, rozbiť na viac riadkov

```
if ((val1 > val2)  
&& (val1 > val3)  
&& (val2 > val3)  
{  
// Takeappropriateaction.  
}
```

- V prípade spájania stringov jednoducho, stačí používať +, v prípade loopovania a


```
1. publicStringNumber {get; set;}
2. publicDateTimeDateOpened {get; set;}
3. publicDateTimeDateClosed {get; set;}
4. publicdecimalBalance {get; set;}
5.
6. // Constructor
7. publicAccount()
8. {
9. // ...
10. }
11. }
```

3. Komentáre

- Komentý na vlastne riadky (nie na riadok za kod) pridať space hneď za // značkou. Tiež sa ich pokúšať držať čitateľné od riadkovaním. Prevažne sa snažiť mať pred samotnou metódou komentár o tom čo tato metóda robí. V angličtine.

```
// Thefollowingdeclarationcreates a query. Itdoesnot run
// thequery.
```

4. Unit testy

Po dokončení porobiť unit testy programátorom rovno aby to fungovalo. (následne kroky sumergedvelop do feature a resolve)

Dôležitý note:

Ak kód nespĺňa codeconventions, nemože sa testovať, opravovanie kódu sa tohto netýka, má to robiť rovno programátor.

5.3 Metodika úloh

Na evidovanie úloh využívame TFS. Nástroj má plnú podporu pre SCRUM. Cieľom metodiky je stanovenie procesov, ktoré vedú k vytvoreniu požadovaného produktu. Metodikou sa tím riadi po celý čas práce na projekte.

V kapitole „Roly“ sú opísané jednotlivé roly v rámci scrum tímu, ktorých sa táto metodika týka a za čo primárne zodpovedajú.

5.3.1 Vymedzenie pojmov – skratiek

Branch – Vetva.

Task – Úloha.

Scrum – Metodika agilného vývoja produktu.

Šprint – Iterácia vývojového procesu.

Product backlog – Zoznam všetkých požiadaviek.

User story – US

Story pointy – SP

Product owner – PO

Scrum master – SM

Product backlog – PB

Tím – T

5.3.2 Definovanie jednotlivých úrovní úloh

Epic

Úloha, ktorá popisuje veľkú časť produktu, ktorá sa nedá spraviť za krátky čas. Skladá sa z viacerých menších častí – feature. Každý epic má pridelenú prioritu vykonávania.

Možné stavy epicu

New – Nový epic.

Active – Na epicu sa pracuje.

Closed – Všetky feature daného epicu sú closed.

Feature

Menšia úloha ako epic ale stále dosť veľká nato aby sa dala rozložiť na menšie časti US.

Popisuje možné funkcie epicu.

Možné stavy feature

New – Nový feature.

Active – Na feature sa pracuje.

Closed – Všetky US daného feature sú closed.

US

Skladá sa z menších častí – taskov. Má vždy prideleného jedného človeka, ktorý je

zodpovedný za jej dokončenie. Má pridelenú hodnotu spory point, ktorá definuje ako náročná je daná úloha.

Možné stavy US

New – Nová US.

Active – Na US sa pracuje.

Resolved – US je hotová a čaká na otestovanie.

Closed – US je korektná (spĺňa zadanie US a prešla testami) a PO ju schválil.

Task

Najmenšia úloha zo všetkých. Je presne definovaná nie len názvom ale aj podrobnou špecifikáciou. Počas práce na danej úlohe je člen tímu, ktorému bola úloha pridelená povinný zapisovať progress do TFS.

Možné stavy tasku

New – Nový task.

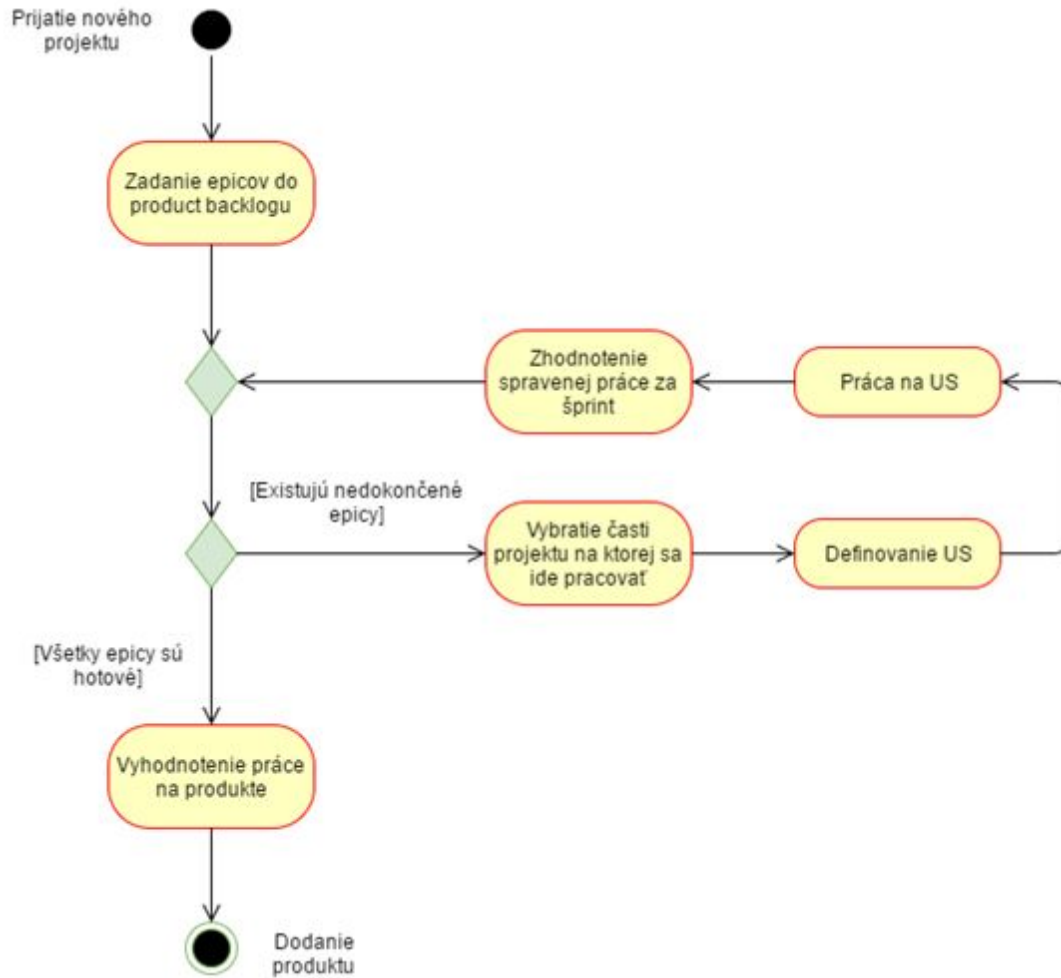
Active – Na tasku sa pracuje.

Closed – Task je hotový.

5.3.3 Súvisiace – nadväzujúce metodiky

- Metodika testovania – Určuje vytváranie testov pre otestovanie jednotlivých častí projektu.
- Metodika verzií – Definuje, kedy je možné vytvoriť novú verziu projektu, čo vytvoreniu novej verzii predchádza a ako sa daná verzia vytvára.
- Metodika zberu požiadaviek – Jeho úlohou je definovanie požiadaviek zo strany PO a vytvorenie základného modelu projektu.
- Metodika branchov

5.3.4 Diagram nadväznosti procesov



5.3.5 Procesy

1. Zadanie epicov do product backlogu.

Info: Na začiatku práce na projekte sa urobí jedno dlhšie stretnutie. Na tomto stretnutí sa určia základné epicity aj s ich featurami.

- a. Vytvorenie PB v TSF.
- b. Pretvorenie PO vízie do epicov a pridanie epicov do PB.
- c. Určenie priority jednotlivých epicov.
- d. Rozbitie jednotlivých epicov na menšie časti – feature.¹
 - i. Nové epicity môžu pribúdať aj počas vývoja produktu.²

2. Vybratie časti projektu, na ktorej sa ide pracovať.

Info: V tomto bude začína šprint. Na prvom stretnutí sa vyberajú časti, na ktorých sa bude počas šprintu pracovať.

- a. Vytvorenie šprintu v TFS.
 - i. Názov šprintu musí byť známy vírus alebo choroba. Názvy šprintov idú podľa abecedy.
- b. Výber epicu/ epicov z PB s najväčšou prioritou.
- c. Výber feature z daných epicov.
- d. Rozbitie feature na US.
 - i. Počas vytvárania US sa tím riadi vetou: „Ako POUŽÍVATEĽ chcem NIEČO aby NIEČO“.
 - ii. Názov US musí byť výstižný, presný a krátky.

3. Definovanie US.

Info: Po definovaných US pre šprint sa musia US rozbiť na menšie časti tasky. Akonáhle sú všetky US rozbité tak sa im priradia SP. Nakoniec sa pridelia jednotlivé tasky členom tímu.

- a. Rozbitie US na tasky.
 - i. Definovanie jednotlivých taskov.
 1. Meno Tasku musí byť výstižné ale nemusí byť krátke. V jeho opise sa potom nachádzajú dodatočné informácie.
 - ii. Ak je možné danú US otestovať, tak testovanie je pridané ako samostatný task v rámci danej US.
- b. Ohodnotenie US so SP.
 - i. SP sa priradia pomocou plannig poker.
 - ii. V prípade, že má US viac ako 13 SP, tak sa daná US musí rozdeliť na viacej US.
- c. Pridelenie US a taskov jednotlivým členom tímu.
 - i. Členom tímu sa priradia US podľa ich doterajších skúseností.
- d. Ohodnotenie taskov hodinami.
 - i. Definuje si každý člen tímu sám.

4. Práca na US.

¹ Nový epic sa musí rozbiť na feature už vo fázi pridania. Rozbíjanie nemôže byť odložené na neurčito z dôvodu špecifikácie epicu.

² Každému takto pridanému epicu sa musí určiť prioritou a musí sa rozbiť na menšie časti – feature.

Info: V tomto bode začína reálne implementovanie US pre daný šprint. Každá US musí mať vytvorenú jednu branchu, v ktorej bude implementovaná. Po dokončení práce na danej US sa daná vetva mergeje do vetvy develop.

- a. Vytvorenie novej branchy.
 - i. Vid' manažment branchov.
- b. Práca na US.
 - i. Poradie vypracovania jednotlivých taskov v US je na členovi tímu, ktorému bol daný task pridelený.
 - ii. Po vypracovaní tasku je člen tímu povinný zadať do TFS informácie o jeho vypracovaní.
- c. Kontrola US.
 - i. Vid' manažment testovania.
- d. Vyhodnotenie US.
 - i. Uzatvorenie US³ / prenesenie US do ďalšieho šprintu / odloženie US na neurčito⁴.

5. Zhodnotenie spravenej práce za šprint.

Info: Po uplynutí šprintu nasleduje stretnutie členov scrum tímu. Na stretnutí sa diskutuje o tom, čo sa za uplynulý šprint stihlo/ nestihlo urobiť a členovia tímu majú možnosť predniesť svoje pripomienky na zlepšenie práce na projekte. Počas stretnutia sa aj vyhodnotí, či je možné vydať novú verziu produktu.

- a. Vyhodnotenie čo všetko sa stihlo spraviť.
 - i. Vyhodnotenie burnchartu.
- b. Retrospektíva.
 - i. Retrospektíva úloh a práce na projekte.
- c. Ak je to možné tak vydanie novej verzie produktu.
 - i. Vid' manažment verzií.

6. Vyhodnotenie práce na produkte.

Info: Po uplynutí všetkých šprintov nasleduje celková retrospektíva projektu a potom začína údržba produktu.

- a. Celková retrospektíva.
- b. Údržba produktu.

³ US sa uzatvára až keď ju PO schváli.

⁴ V prípade, že sa US odkladá na neurčito tak treba zadať vážny dôvod tohto odkladu. Najčastejšie to býva z dôvodu zlej špecifikácie, nedostatku informácií od PO alebo čakania na inú kľúčovú US.

5.4 Metodika komunikácie

Komunikáciu v tíme delíme na formálnu a neformálnu, taktiež dodržiavame zásady komunikácie, ktoré táto dokumentácia opisuje.

Používania komunikačných nástrojov:

- Nástroje formálnej výmeny informácií tím používa:
 - Spoločný email tímu: skupina8fiitteam@googlegroups.com - komunikácia s vedúcim tímu, pokiaľ členovia nereagujú v nástroji na menej formálnu komunikáciu.
 - TFS – Team Foundation Server – zadávanie úloh, spravovanie zdieľaného kódu.
 - GoogleDisk a GoogleDoc – vytváranie a ukladanie zdieľaných dokumentov.
- Formy menej formálnej komunikácie:
 - o Discord
 - komunikácia medzi členmi tímu navzájom a vedúcim projektu,
 - Odkaz na pripojenie sa: <https://discord.gg/z5FsnWJ>
 - v tomto nástroji je komunikácia rozdelená podľa zameraní jednotlivých tém:
 - Súbory a linky – po vytvorení nového dokumentu sa tu pošle odkaz naň, poprípade nejaký komentár k nemu,
 - General – riešenie všeobecných problémov
 - Public - channel určený na komunikáciu s ľuďmi mimo nášho tímu.
 - Updates - oznamy, v ktorých členovia tímu zdieľajú informácie, ktoré sa týkajú celého tímu, aj keď nie možno okamžite
 - Development - kanál určený na riešenie problémov, ktoré vznikajú prístupom ku súborom, prípadne k spolupráci medzi členmi
 - Tutoriály - odkazy na externé návody a nástroje použiteľné v riešení projektu
 - Nezáväzné úlohy - zoznam dobrovoľných úloh nie okamžite potrebných, ale ktoré by mohli uľahčiť prácu ak by sa ich niekto ujal
 - Spam – najnižšia úroveň neformálnej komunikácie,

Pravidlá komunikácie:

- Najskôr sa snažia členovia tímu vyriešiť problém sami.
 - Pokiaľ si nevie niekto poradiť, kontaktuje člena tímu, s ktorým je spojený do dvojice na riešenie a testovanie.
 - Pokiaľ sa problém stále nevyrieši, kontaktujú ďalších členov tímu a vedúceho cez Discord.
 - Pokiaľ sa problém aj tak nevyrieši, preberie sa na oficiálnom stretnutí.
 - Pokiaľ člen tímu kladie otázku určitému človeku, označí ju v konverzácii @menoStudenta, alebo ak je správa smerovaná na všetkých označí sa @everyone.
- Medzi hlavný komunikačný prostriedok patrí tímové stretnutie, ktoré prebieha každý týždeň

formou diskusie. Na každom stretnutí sa preberie zoznam úloh, ktoré boli pridelené, preberie sa ich riešenie, rozhodne sa o tom, či sú splnené alebo nie. Na stretnutí sa prekonzultujú vzniknuté problémy a nedorozumenia a prebieha brainstorming ohľadom nápadov na riešenie problémov.

5.5 Metodika manažmentu chýb

Úvod

V tomto dokumente sa nachádza opis manažmentu chýb v projekte. Na správu chýb používame systém TFS. Jedna chyba je reprezentovaná ako task v TFS a jedna feature vetva v gite.

Postup

Životný cyklus jednej chyby sa skladá z nasledovných krokov:

1. Identifikácia chyby

Člen tímu odhalí v projekte chybu.

2. Vytvorenie nového tasku v TFS

Vytvorí sa nový task (ak je potrebné aj user story) a priradí sa členovi tímu na vyriešenie.

3. Riešenie chyby

Riešiteľ si v gite vytvorí novú feature vetvu, v ktorej bude opravovať chybu. Počas práce s gitom v rámci riešenia chyby sa riešiteľ riadi metodikou git workflow. Po úspešnom vyriešení vytvorí riešiteľ pull request a task týkajúci sa chyby v TFS označí ako *Resolved*.

4. Kontrola vyriešenia chyby

Po vyriešení chyby musí byť riešenie skontrolované iným členom tímu.

Riešiteľ dá zvyšku tímu na vedomie, že chyba je vyriešená a čaká na kontrolu. Po vzájomnej dohode jeden člen tímu vykoná kontrolu.

- Ak zistí, že chyba pretrváva, vráti task v TFS späť do stavu Active (návrat na krok 3)

- Ak zistí, že chyba bola odstránená, schváli pull request podľa metodiky git workflow a stav tasku v TFS zmení na Closed

Konvencia pomenovania

Keďže na správu chýb používame rovnaký systém ako na manažment úloh, v záujme prehľadnosti je potrebné chyby odlíšiť od bežných úloh.

- Task opisujúci chybu začína slovom "Bug".
- Vetva v gite je pomenovaná podľa konvencie zadefinovanej v metodike git workflow. Popis vetvy však musí obsahovať slovo "Bug" alebo inú indikáciu, že sa jedná o vetvu zaoberajúcu sa chybou, nie novou feature.

5.6 Metodika manažmentu verzí

Spôsob práce s gitom



V projekte sa budú udržiavať dve hlavné vetvy:

- **Master** - Hlavná vetva, ktorá odráža otestovaný funkčný kód
- **Develop** - Vetva, do ktorej sa priebežne “pushujú” zmeny z ďalších vetiev
- **Feature vetvy** - Vedľajšie vetvy, každá predstavuje samostatnú funkčnosť

Po skončení práce na konkrétnej feature sa merge daná vetva s developom prostredníctvom PullRequestu. Po dôkladnom otestovaní novej funkčnosti aspoň jedným ďalším členom tímu sa môže PullRequest schváliť a merge do developu sa vykoná. Po spoločnom schválení stavu projektu aj s product ownerom sa na konci väčšieho celku môže develop merge-núť s master

vetvou. V prípade nájdených problémov je nutné najprv vyriešiť identifikované problémy v samostatných vetvách.

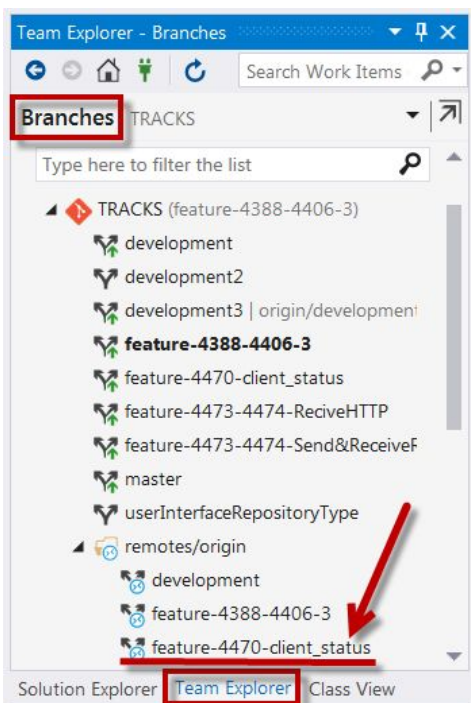
Vývojový cyklus

Pri začatí práce na novej funkčnosti postupujeme podľa nasledujúcich krokov:

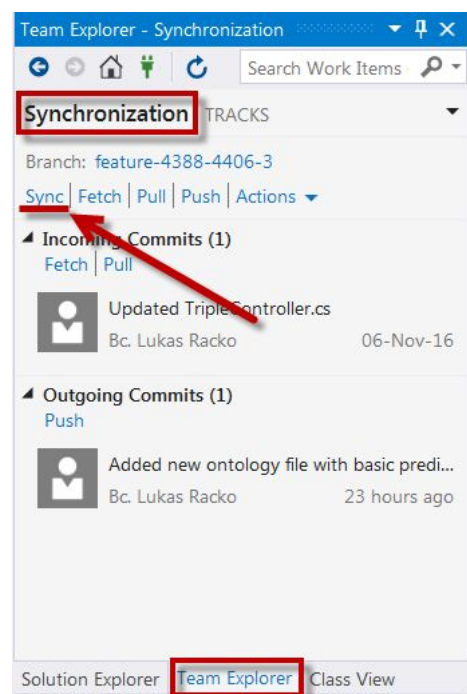
Cez VisualStudio:

1. Dostal som zadanú novú úlohu
2. Zobrazím si zoznam vetiev v *Team explorer/branches*, zvolím si jednu, z ktorej budem vychádzať z Remotes/Origin a zvolením možnosti *Checkout/New localbranchfrom* si stiahnem vetvu ku sebe. Pomenujem ju podľa vyššie spomenutých konvencií a pri potvrdení odškrtnem možnosť *Trackremotebranch*. Týmto spôsobom sa pri následnom synchronizovaní vytvorí kópia v Remotes/Origin s rovnakým názvom, aký sme zvolili a všetky zmeny sa budú ukladať do nej.
3. Vykonám ľubovoľné zmeny v zdrojových kódoch a iných súboroch
4. Commitnem zmeny v záložke *Changes* aj so stručným popisom zmien
5. Zmeny synchronizujem aj do Remote vetvy prostredníctvom záložky *Sync*

Obrázky:



Krok 2



Krok 5

Cez konzolu:

1. Dostal som zadanú novú úlohu
2. Prepnem sa do vetvy, z ktorej budem vychádzať `git checkout nazov_vetvy`
3. Aktualizujem túto vetvu (*master*, *develop*), pre prípad, že niekto medzičasom vykonal nejaké zmeny `git pull origin nazov_vetvy`
4. Vytvorím novú vetvu `git checkout -b nova_vetva`
5. Vykonám ľubovoľné zmeny v zdrojových kódoch a iných súboroch
6. Stagenem vykonané zmeny `git add -A`
7. Commitnem zmeny `git commit -m "popis_commitu"`
8. Zmeny 'pushnem' do gitu `git push origin nazov_vetvy`

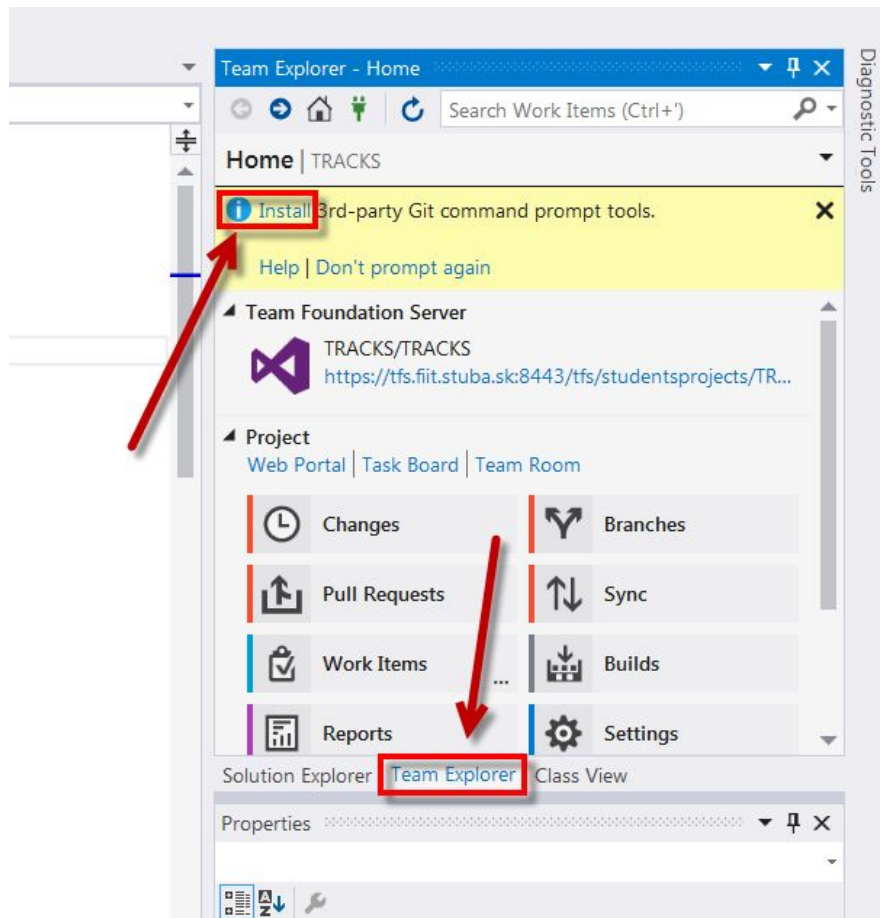
Pri pokračovaní práce na vetve, ktorá už v gite existuje postupujeme nasledovne:

1. Dostal som zadanú už existujúcu úlohu
2. Zistím, či požadovaná vetva existuje lokálne `git branch`
3. Prepnem sa do požadovanej vetvy `git checkout nazov_vetvy`
 - a. Pokiaľ vetva lokálne neexistuje, stiahnem si ju z gitu `git fetch origin remote_nazov_vetvy: local_nazov_vetvy`
4. Pokiaľ som novú vetvu práve nestiahol z gitu, aktualizujem ju `git pull origin nazov_vetvy`
5. Vykonám ľubovoľné zmeny v zdrojových kódoch a iných súboroch
6. Stagenem vykonané zmeny `git add -A`
7. Commitnem zmeny `git commit -m "popis_commitu"`
8. Zmeny 'pushnem' do gitu `git push origin nazov_vetvy`

Inštalácia gitu

- **Cez visual studio**

Pokiaľ git ešte nie je nainštalovaný, je možné ho stiahnuť cez visualstudio(VS). Po otvorení VS otvoríme záložku 'Team Explorer' a v hore sa nám zobrazí možnosť na stiahnutie gitu. Všetko potvrdíme a po stiahnutí nainštalujeme stiahnutý inštalčný súbor gitu.



- **Z internetu**

Najnovšiu verziu gitu môžeme stiahnuť priamo zo stránky gitu na <https://git-scm.com/downloads>

Pomenovanie vetiev

- Každá nová vetva sa vytvára za účelom vytvorenia/opravenia nejakej funkcionality a jej názov by mal jednoznačne hovoriť o tom, čoho sa týka.
- Pomenovanie: **feature-'číslo_user_story'-'stručný_popis'**
 - napr. feature-4317-instalacia_aplikacie
 - ID taskov a user stories sa dajú nájsť v backlogu (treba správne nastaviť columnoptions) alebo v detaile user story.

The screenshot shows a Jira Backlog interface. On the left, there is a sidebar with 'Backlogs' and 'Queries' tabs. Under 'Backlogs', there are sections for 'Epics', 'Features', and 'Stories'. The 'Current' section is expanded to show 'Iteration 1'. The main area displays a 'Backlog' view with a table of work items. A red arrow points to a task with ID 4323. A modal window for creating a 'User Story' is open, showing a 'Title' field and an 'Add' button.

Order	ID	Work Item Type	Title
1	4317	User Story	Používateľ si chce nainštalovať a spustiť klientsku aplikáciu
	4323	Task	Task 1: vytvoriť klienta v WPF
	4324	Task	Task 2: Vytvoriť server
	4325	Task	Task 3: Vytvoriť defaultné DLL
	4327	Task	Task 4: Rozbehnutie databázy apache-jena-fuseki
2	4318	User Story	Používateľ sa chce ozvať serveru
3	4319	User Story	Server zisťuje koľko je aktívnych voľných uzlov v sieti
4	4320	User Story	Používateľ sa ponúka, že chce pomôcť, a chce od balancera link na...
5	4321	User Story	Server odosle knižnicu na konkrétny repozitár

Testovanie

Po dokončení ľubovoľnej funkcionality je potrebné, aby bola dôsledne otestovaná niekým druhým z tímu. Keď sa dokončia všetky úlohy pre danú User Story, presunie sa do stavu 'Resolved', kde ktokoľvek z tímu otestuje správnosť danej funkcionality, prípadne skontroluje, či sú splnené **Acceptancecriteria** (ak sú zadané). Následne, ak je všetko v poriadku, user story sa dá do stavu closed.

Odporúčaná literatúra

Prehľad kľúčových pojmov a metodika gitu -

<http://labss2.fiit.stuba.sk/TeamProject/2010/team03is-si/data/metodiky/manazment-verzii.pdf>

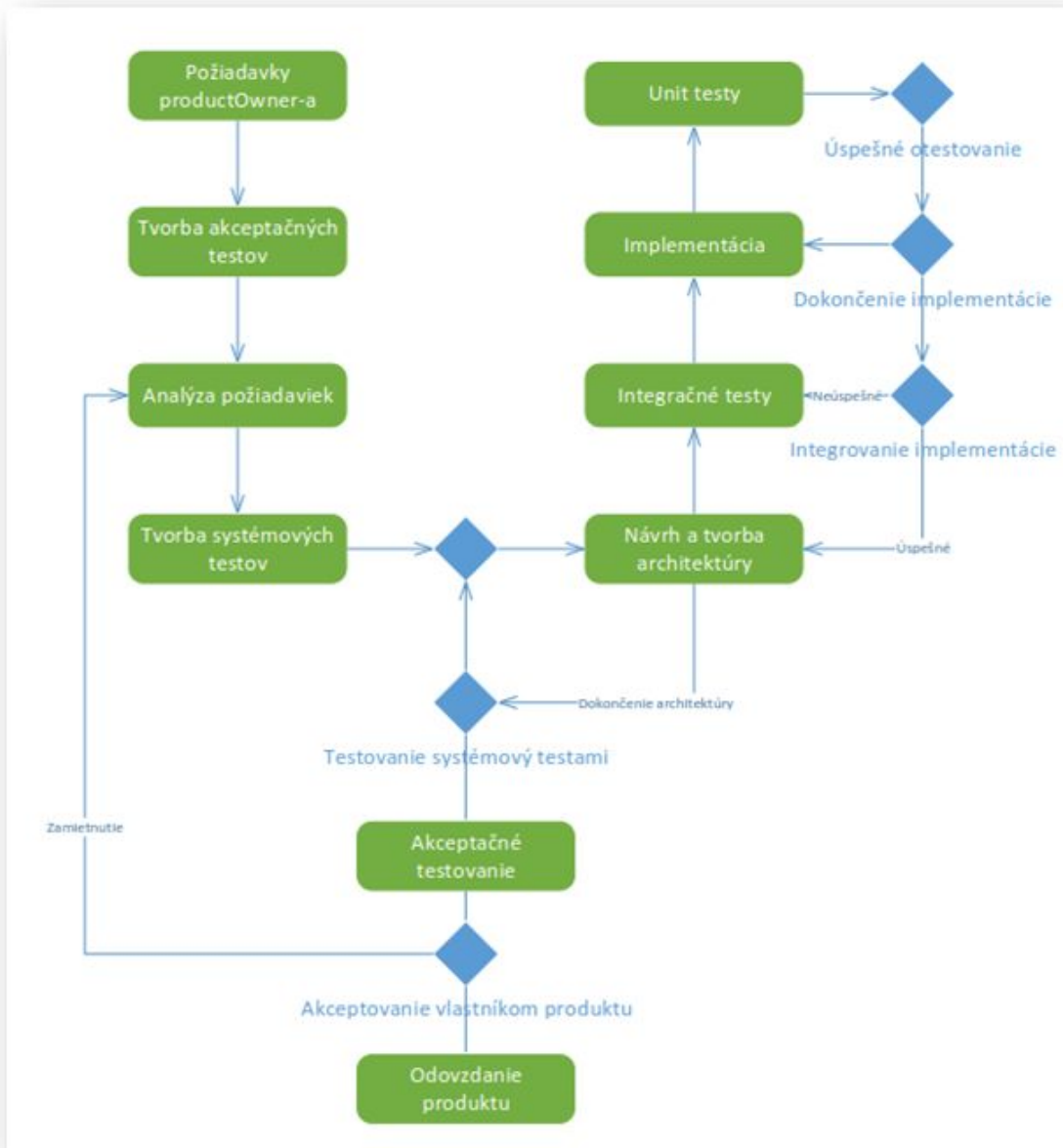
<http://labss2.fiit.stuba.sk/TeamProject/2010/team03is-si/data/metodiky/manazment-verzii.pdf>

<http://labss2.fiit.stuba.sk/TeamProject/2010/team03is-si/data/metodiky/manazment-verzii.pdf>

Detailná dokumentácia ku gitu:

<https://git-scm.com/doc>

5.7 Metodika testovania



1. Požiadavky productOwner-a

Roly: Product owner, scrum master, členovia tímu, tester)

Inicializácia: Zhromaždenie požiadaviek od vlastníka produktu

Vykonávanie: Definovanie požiadaviek a získanie špecifikácií od vlastníka produktu

Výstup: Definované požiadavky vlastníka produktu

2. Tvorba akceptačných testov

Roly: Tester, Product owner, Scrum master

Inicializácia: Definované požiadavky od vlastníka produktu

Vykonávanie:

- Definovanie kedy bude produkt akceptovaný vlastníkom,
- Vytvorenie akceptačných testov podľa požiadaviek
- Vytvorenie úspešného a neúspešného výsledku testu

Výstup: Akceptačné testovanie

3. Analýza požiadaviek

Roly: Tester, Scrum master, Členovia tímu

Inicializácia: Vytvorenie akceptačného testovania alebo vrátenie sa po neakceptovaní systému product-ownerom

Vykonávanie:

- Analyzovanie požiadaviek vlastníka
- Rozdelenie požiadaviek do epikov, feature a user story
- Rozdelenie user story na tasky

Výstup: Rozdelené analyzované požiadavky na systém

4. Tvorba systémoch testov

Roly: Tester, Scrum master

Inicializácia: Požiadavky na systém

Vykonávanie:

- Definovanie akceptovaných a neakceptovaných stavov
- Vytvorenie systémových testov na základe požiadaviek

Výstup: Systémové testy

5. Návrh a tvorba architektúry

Roly: Tester, Scrum master, členovia tímu

Inicializácia: Vytvorené systémové testy, analyzované požiadavky

Vykonávanie:

- Vytvorenie návrhu architektúry podľa požiadaviek
- Tvorba architektúry podľa požiadaviek na systém
- Rozloženie architektúry na komponenty

Výstup: Vytvorená architektúra systému

6. Integračné testy

Roly: Scrum master, členovia tímu

Inicializácia: Návrh architektúry systému alebo dokončenie implementácie

Vykonávanie:

- Vytvorenie integračných testov členmi tímu podľa návrhu architektúry ,
- Špecifikovanie požiadaviek na splnenie integračných testov,
- Spustenie integračných testov

Výstup: Vytvorené integračné testy

7. Implementácia

Roly: Členovia tímu, scrum master

Inicializácia: Vytvorenie integračných testov, návrh architektúry a špecifikované

požiadavky,

Vykonávanie:

- Implementovanie user story podľa požiadaviek product ownera,
- Vytváranie a implementovanie taskov v user story

Výstup: Implementovaný user story a tasky k nemu

8. Unit testy

Roly: Členovia tímu, Scrum master

Inicializácia: Implementovaný user story a tasky k nemu

Vykonávanie:

- Implementovanie unit testov
- Zabezpečenie automatizovaného spúšťania týchto testov

Výstup: Automatický spúšťané unit testy

9. Akceptačné testovanie

Roly: Tester, Product owner

Inicializácia: Implementovanie architektúry a úspešné systémové testy

Vykonávanie:

- Testovanie podľa zadaných akceptačných testov
- Akceptovanie alebo vrátenie product ownerom

Výstup: Výsledok akceptačného testovania

10. Odovzdanie produktu

Roly: Product owner, Scrum master

Inicializácia: Úspešné akceptačné testovanie

Vykonávanie:

- Nasadenie produktu,
- Odovzdanie produktu vlastníkovi
- Vytvorenie Podpora produktu

Výstup: Nasadený produkt

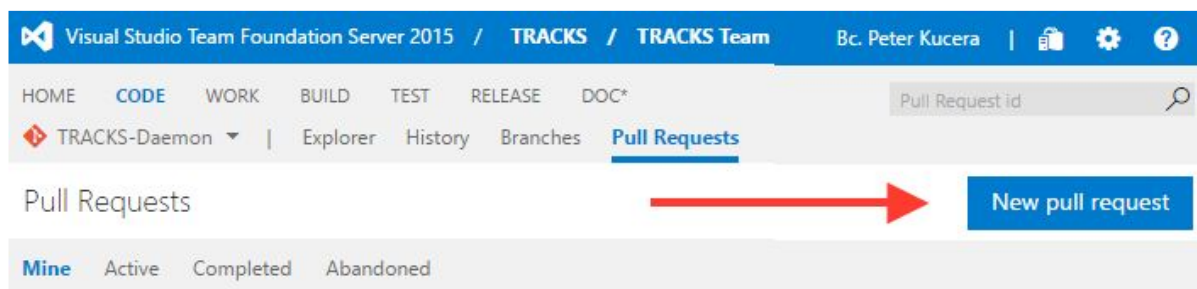
5.8 Metodika prehliadok kódu

Prehliadky kódu sú vykonávané pomocou funkcionality verziovacieho systému GIT - Pull requestov. Osoba vykonávajúca prehliadku kódu kontroluje, či je kód napísaný na základe konvencií opísaných v kapitole "Metodika písania a komentovania zdrojového kódu". Kód sa taktiež kontroluje na základe princípov DRY (Don't Repeat Yourself) a KISS(Keep It Simple, Stupid) tak aby boli jednotlivé časti kódu robustné a aby na nich mohli byť ľahko uskutočniteľné zmeny.

Vzorové príklady sú demonštrované na Microsoft TFS.

Postup :

1. Vytvorenie Pull requestu



2. Vytvorenie pull requestu, výber vetiev odkiaľ a kam sa má požadovaný kód zlúčiť (merge) a priradenie osobe, ktorá má prehliadku kódu vykonať .

Review changes in develop ▾ relative to master ▾ ↔

Merge develop to master

DESCRIPTION

- Added first basic logs
- Merge branch 'develop' into feature-4631-4634-AddBasicLog

REVIEWERS

Bc. Peter Kucera Search users and groups

RELATED WORK ITEMS

Add work items

[New pull request](#) [fewer options](#)

3. Obrazovka pre kontrolu kódu. Viditeľné sú súbory, ktoré boli upravované, a zmeny v nich vykonané (zelená farba). Vpravo sú viditeľné možnosti potvrdenia alebo zamietnutia pull requestu.

The screenshot shows the Visual Studio Code interface for a pull request. The main window displays the code for 'MainWindow.xaml.cs' with several lines highlighted in green, indicating changes. The right sidebar shows the pull request details, including the status 'Active', 'No merge conflicts', and 'No response from reviewers'. The 'Complete pull request' button is visible.

4. Tok prehliadky kódu sa následne vetví na

a) Vrátenie kódu na opravu

b) Schválenie pull requestu a zlúčenie kódu s vetvou vývoja

6. Zápisnice

TÍMOVÝ PROJEKT TRACKS 1

ZÁPISNICA

Dátum konania / Miesto konania	26.9.2016, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Ing. Karol Rástočný, Phd. (KaRa) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr)
Nepřítomní	Lukáš Račko (LuRa)
Zapísal	Miriama Pomffyová
Prílohy	-

Program porady

1. Zoznámenie sa tímu s vedúcim a konzultantom.
2. Oboznámenie sa s témou projektu. Pridelenie úloh.

Nové úlohy

P. č.	Názov úlohy	Popis úlohy	Zodpovedný	Termín splnenia
1.	Vytvorenie plagátu	Tím sa po stretnutí s vedúcim stretne a navrhne plagát – bude obsahovať názov tímu, mená, číslo tímu, názov projektu a tému vystihujúcu projekt.	celý tím	27.9.2016
2.	Teambuilding	Členovia tímu sa neformálne stretnú s vedúcim – dohodne sa termín – štvrtok (niekde v meste) alebo piatok (Stupava – Dni zelá).	celý tím	30.9.2016

3.	Stránka tímu	Vytvoriť webové sídlo projektu statické – informácie o tíme a o aktualizáciach práce na projekte (html , JavaScript, bootstrap).Peter Lacko dodá informácie.	MiKr	3.10.2016
4.	Prihlásiť sa na TFS	Všetci členovia tímu sa 1.-krát prihlásia do systému tfs.fiit.stuba.sk pod prihlasovacími údajmi do AIS. Heslo sa dá zmeniť. Je tu integrovaný kalendár, git, work – kto čo robí, koľko toho spravil; build – možné automatické pretestovanie, nasadzovanie na server; code - user story – kto čo programoval. Oboznámiť sa s prostredím a dať info vedúcemu, keď sa každý prihlási.	celý tím	3.10.2016
5.	Metodika gitu	Spísať metodiku používania gitu. Napr. vznikne úloha – vytváram branch – nazvem podľa úlohy – pracujem – mergnem s masterbrnach – otestujem či sa dá buildnúť, ak nie, vrátim do pôvodného stavu.	nepridelené	3.10.2016
6.	Rozbehanie projektu	Zbuildnúť projekt, editovanie a uloženie, spísanie k tomu dokumentácie. Prepojenie s TFS (Team Foundation Server). Naklonovať repozitár z gitu do našich kont. Spísať k tomu manuál.	MaMu, MiSI	3.10.2016
7.	Spísať Codeconvention	Spísať codeconvention, podľa možnosti priložiť tabuľku porovnania kódu Java do C#. Napr. používanie – namespace/ using, alebo o premennej var – nemusím definovať explicitne, vezme si ho sám pri vytvorení inštancie. Uvádzať aj príklady.	PeKu, PeBo	3.10.2016
8.	Spísať zápisnicu	Spísať zápisnicu zo stretnutia a vo formáte PDF priložiť na stránku, dávať relatívne odkazy	MiPo	28.9.2016

		na zápisnice. Uložiť na GoogleDisk.		
9.	Vytvoriť úložisko	Vytvoriť úložisko na GoogleDisku pre dokumentácie a zápisnice.	MiSI	27.9.2016
10.	EA	Získať licencie na EA (Enterprise Architect). Poslať model v EA a dokumentáciu k RDF.	MaKo	3.10.2016

Nové informácie

Cieľ projektu

Vytvorenie systému TRACKS pre zber a prepájanie dát, ktoré po sebe zanechali vývojári pri vývoji softvéru, t.j. „Tasks, CodeReviews, Activities, SourceCode, and Knowledgeabout Software“. Zber dát by mal byť vykonávaný pomocou botnet klientov na základe zadaní od hlavného uzla. Nasleduje prevedenie dát do spoločnej reprezentácie, ich samotné prepájanie a sprístupnenie tretím stranám. Jedným z hlavných cieľov je navrhnuť infraštruktúru tak, aby bola ľahko nasaditeľná ako na fakulte, tak aj interne v softvérovej firme, a bola využitá záverečnými projektami na fakulte.

System – Prepájanie dát o vývoji softvéru

- V procese vývoja softvérových projektov používame rôzne podporné nástroje, v ktorých zanechávame stopy o svojej činnosti:
 - vývojové prostredia (Eclipse, VisualStudio),
 - systémy pre správu zdrojových kódov (Git),
 - úloh a nahlasovania chýb (Bugzilla, Jira, Redmine),
 - prehliadok zdrojového kódu (Gerrit),
 - alebo aj diskusné fóra a dokumentácie (StackOverflow, MSDN).
- Problém nastáva, keď sa snažíme nájsť súvislosti medzi nimi. Príkladom je prepojenie otázky v StackOverflow od vývojára pracujúceho na úlohe z Bugzilla a jeho commity v Git. Pri následnej prehliadke zmien v Gerrit má kontrolór pred sebou iba výsledný návrh zmeny v Git a nie odpoveď z SO, na ktorej vývojár svoje riešenie založil. Všetky uvedené dáta o vývoji softvéru sú v prípade opensource projektov voľne dostupné cez API alebo web crawling. Každý zdroj dát je však samostatné riešenie, a tak sú dáta slabo alebo vôbec prepojené, napr. iba v komentároch (ID úlohy v commitmessage). Orientácia v množstve informácií je tak zbytočne náročná, znepríjemňuje analýzu dát výskumníkom, uplatnenie ich výsledkov v praxi, a zároveň obmedzuje vznik nových podporných nástrojov pre vývoj softvéru. A keďže nedokážeme upraviť všetky systémy, spravíme si vlastný TRACKS, ktorý nám a naším študentom umožní to pekne pospájať.
- Kľúčovými komponentmi systému TRACKS pre vytvorenie sémantickej vrstvy nad dátami o vývoji softvéru by mali byť:
 - hlavný (master) uzol pre plánovanie zberu dát,
 - distribuované botnet klienti pre zber dát, špecifické pre každý zdroj dát (napr. Git, Bugzilla, Gerrit),

- repozitáre zozbieraných dát a odvodených metadát,
- rozhranie pre vystavenie dát a rozšíriteľnosť systému o analýzu dát Príkladmi softvérových projektov, ktoré môžeme hneď zbierať a prepájať sú Eclipse, Android, Qt, OpenStack, i mnohé ďalšie na <http://gerrithub.io>, ku ktorým máme dostupné dáta rôzneho druhu.
- Každý systém má svoje dáta a samostatné prihlásenie používateľa – login (Git, tasky –úlohy, chyby) – chceme prepojiť súvis commitov.

Činnosti

- Master vetva – otestovaný zdrojový kód, vedľajšie vetvy – fixovanie chýb + vývoj ->merge.
- Unit testy majú mať čo najväčšie pokrytie.
- Prehľad kódov – codereview – kontrola commitov, opráv chýb, či boli správne opravené – Gerrit – nastavenie na Git.
- Sledovanie programátora počas práce – nájsť, čo mu dlho trvá, s čím má problémy, nájsť dobrých a efektívnych riešiteľov pre vybraný problém.
- Zadávať tasky, robiť reviews, vykonávať activities, zdieľať a dokumentovať knowledges, ...
- Zápisnica obsahuje – čo sa riešilo na stretnutí, kto sa zúčastnil a kto nie, kto mal aké nápady a pripomienky, prejdú sa zadané úlohy, či sú splnené / nesplnené, zápis rozhodnutí, ktoré boli prečo ako urobené. Ukladať na GoogleDisk.
- Viest' agilný vývoj – konzultácia so zákazníkom = vedúcim projektu.
- Za každý šprint je zodpovedný 1 člen tímu.
- Každéj úlohe definovať, kedy sa dá označiť za splnenú a dopredu odhadnúť ako dlho bude trvať jej vypracovanie – dobré pre vyhodnotenie BurnDownChart.
- Dokumentácia – ukladať na GoogleDisk – odkazy z webového sídla projektu na PDF:
 - Z priebehu šprintov – aké úlohy boli zadané a ako dopadli – nahodiť do TFS.
 - O produkte BigPicture, odkazy na dokumenty o implementácii.
 - Zápisy zo stretnutí.
- Zozbierané dáta sú vo formáte RDF:
 - ResourceDescriptionFramework.
 - 3 zložky – objekt, predikát, subjekt.
- funkcia podobná ako XML – ExtensibleMarkupLanguage.
- aby dáta vedeli čítať a premieňať na znalosti nie len ľudia, ale aj stroje.
- objekt je URI, o čom hovoríme, predikát je čo o objekte hovoríme a subjekt je znalosť.
- chceme dosiahnuť Linkdata – pospájať všetko so všetkým – súvislosti – vznikne graf. Napr. máme výpis – tento commit – tento autor – tento názov – tento čas, ... ->uložiť všetky dáta do trojíc (o, p, s).

Čo potrebujeme

- Desktop API:
 - Spracovanie dát – formuláre, a vizualizácia,
 - Sťahuje dáta,
 - Posiela dáta,
 - Beží u klienta, riešiť obmedzenie trafiku na sieti, výkonu – koľko jobov berie –

možnosť „vypnúť“ alebo „pozastaviť“.

- Server – webový – prijíma dáta a spracováva.
- Balancer – dáva info Daemonovi, čo sa má kedy sťahovať, posielat', desktopy si pýtajú úlohy.
- Daemon po prijatí príkazu z Balancera spúšťa job – pošle klientovi správu – zo zariadenia –klientského PC stiahne všetky požadované dáta. Job je JSONovská informácia s parametrami pre nastavenie configfile – univerzálnemu klientovi nastavím, čo chcem.

Už máme:

- Základný portál,
- Balancer – RepositoryTypeController – vráti typ repozitára podľa ID.
- TripleStore – ukladá dáta – vie naplánovať joby – JobExecutoryController – commit určujúci intervaly, od kedy do kedy chceme sťahovať.
- JobExecutionID dá potrebné informácie o tom, čo sa má vykonať.
- JobExecutable – odkiaľ pokiaľ sa má vykonať.
- model DB a model projektu (jeho komponenty: TripleStore, Balancer, Daemon, Job, Git Job, BuggzillaJob, MylynJob, GerridJob, StackOverflow) – dodané v EA.
- V controlleroch sú controllery pre APIčka.
- Vytvorené libky na testy – rozchodené testy – podobné xUnit.
- Rozbehaný connect na databázu RDF.

Máme spraviť:

- Rozbehať MS SQL server– databáza pre trojice v RDF, definovanie Jobov, intervalov, balancera – Express verzia zadarmo.
- Nainštalovať VisualStudio – verzia Enterprise zadarmo.
- Pre connect na databázu potrebné nainštalovať lokálne pre vývoj Apache Jena Fuseki – downloadnúť 1 zip – rozbaľiť, dá sa dať na Tomcat ako war súbor, ale lokálne stačí cez .bat súbor, beží na localhoste.
- Naštudovať SPARQL QueryLanguage – po trojiciach: ?o ?p ?s (dokumentácia na W3C).
- Rozhodnúť stratégiu testovania – či prí commitoch, či prí celkoch, alebo ináč.
- Pri .NET RDF je chybný repozitár, neurobiliešterelease s opravou, preto je potrebné naklonovať správny k sebe lokálne a po release možnosť stiahnuť správny vyhládávaní online.
- Potrebný .Net Core – na stránke 2 veci vyskočia na stiahnutie – VisualStudio2015 Update 3 a .Net Core 1.0.1.

TÍMOVÝ PROJEKT TRACKS 2

Zápisnica

Dátum konania / Miesto konania	3.10.2016, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	Peter Kučera
Prílohy	-

Program porady

1. Retrospektíva - prehľad o stave plnenia úloh
2. Pridelenie úloh na ďalší týždeň
3. Identifikácia základnej architektúry projektu
4. Pridelenie úloh na ďalší týždeň

Nové úlohy - tím

P. č.	Názov úlohy	Popis úlohy	Zodpovedný	Termín splnenia
1.	Doplniť reálne údaje do webu	Doplniť na web reálne texty o tíme + fotky členov, pridať číslo tímu, kontrast nápisu a pozadia, veľkým TRACKS, téma webu musí byť konzistentná s plagátom (BOTNET .NET)	MIKR	10.9.2016
2.	Teambuilding	Členovia tímu sa neformálne stretnú s vedúcim – dohodne sa termín	celý tím	10.9.2016

3.	Disk Google	Potrebné uploadnúť všetky súbory súvisiace s tímovým projektom na cloud Všetci potrebujú prístup na Disk Google	celý tím	10.9.2016
4.	Potreby nástroj na chat	Discord	celý tím	10.9.2016
5.	Konzolové príkazy pre GIT	Potrebné sa naučiť s GITom robiť cez konzolu (aspoň príkazy clone pushpullmerge a commit)	celý tím	10.9.2016
6.	Licencie na EA	Potrebné získať licencie na EA	celý tím	10.9.2016
7.	Fotky členov tímu	Zhotoviť fotky členov tímu a uploadnúť na web	MiPo	10.9.2016
8.	Rozbehanie projektu	Všetci si musia rozbehať projekt podľa tutoriálu na Disku Google	celý tím	10.9.2016
9.	TP Cup	Áno / Nie ? .. Treba sa dohodnúť	celý tím	10.9.2016
10.	StackOverlow datamining	Je možný cez nejaké verejné API ? Treba zistiť	celý tím	10.9.2016
11.	Nastudovanie klient – server komunikácie	Studovať Rest-API, komunikácia klient-server,		

Nové úlohy - jednotlivci

P. č.	Názov úlohy	Popis úlohy	Zodpovedný	Termín splnenia
12.	Definovať API	Formálny popis API, cez ktoré bude komunikovať klient so serverom	Nepridelené	N/A
13.	Základná funkcionálna klienta	Za použitia knižnice WPF vytvoriť desktopového klienta a spraviť HelloWorld so serverom.	PeKu	10.9.2016
14.	Základná kostra / funkcionálna servera	Rozbehať „helloworld“ medzi klientom a serverom vo webovej časti.	MiKr	10.9.2016
15.	Návrh štruktúry DLL	Navrhnuť DLL ktoré bude klient sťahovať	PeBo	10.9.2016

16.	Výber repozitárov	Výber repozitárov, ktoré je potrebné sťahovať	Nepridelené	N/A
17.	RDF trojice	Lokálne rozbehať apache-jena-fuseki RDF trojice	LuRa	10.9.2016
18.	Uploadnúť user stories, features, tasky do TFS	Prepísať údaje z TFS na stretnutí	MiSI	10.9.2016
20.	Spojzdnenie klient server komunikácie	http client + nadstavby, webapi	Nepridelené	N/A
21.	Odosielanie dát z klienta	Odosielanie výsledkov, ktoré klient stiahne	Nepridelené	N/A
22.	Vytvorenie prototypu aplikácie	Vlastny prototyp	Nepridelené	N/A

Nové informácie

Navrhnuté nástroje / knižnice

N-git knižnica pre .NET

Gerrit – codereviews

WPF – desktopový klient

Logovanie crashov – CrashReporter.NET

Volanie API – HttpClient, WebClient

Knižnice pri pridávaní do projektu – NuGet packages

Pripomienky k projektu

Ideálna práca na projekte po dvojiciach

Bude potrebné rozumne vyriešiť update klienta

Update len pri spustení (nie počas sťahovania)

Features

1. Vytvoriť klienta a prepojenie so serverom

- a. Možnosť upravovania nastavení
 - b. Možnosť nastavenia priority (filter)
 - c. Update (dokončenie úloh, a preinštalovanie)
 - d. Log o chybách, nedokončených ulohách, odoslať balancerovi pri každom spustení
 - e. Progress bar
 - f. Možnosť reportu
 - g. Zistenie stavu siete (kvalitu, rýchlosť), obmedzenie alebo nesťahovanie
 - h. Obmedzenie výkonu, pozastaviť sťahovanie, mimo aktívnych hodín
2. Získať job z balancera
 3. Web rozhranie pre používateľa na vytvorenie úloh pre balancera
 4. Reprezentácia dát (RDF)
 5. Balancer klientovi (joby)
 6. Udržiavanie dll
 7. Vytvoriť balancer
1. Kontrola ktorých dát je/nie je nutná
 2. Vytvorenie dll pre každý repozitár
 3. Vytvoriť databázu (tripleStore) jena
 4. Správa logov v balancerovi (crashreporter)
 5. Vidieť na serveri aktívnych klientov
 6. Správa user managementu

USER STORIES

1. Používateľ sa chce ozvať serveru
2. Server zistí koľko je voľných uzlov v sieti
3. Používateľ chce od balancera "link", ktorý ma sťahovať a server mu odošle dll súbor pre odoslanie notifikácie a spustím ju
4. Admin ma možnosť dodávať najnovšie knižnice, ktoré sú dostupné pre klientov

TÍMOVÝ PROJEKT TRACKS 3

Zápisnica

Dátum konania / Miesto konania	10.10.2016, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	Lukáš Račko
Prílohy	-

PROGRAM PORADY

1. Retrospektíva k predošlému šprintu
2. Vytvorenie nových úloh pre ďalší šprint
3. Vytvorené Epicy, featury, stories v TFS
4. Ohodnotenie user stories v TFS pomocou SCRUM kartičiek

DETAILY

V rámci retrospektívy sme sa pozreli na aktuálny stav riešenia vytvorených úloh, pozreli sme sa na novú verziu webovej stránky. Následne sme si stanovili nové user stories, features a epics a všetky prvky sme navzájom poprepájali v TFS.

Po retrospektívne sme si opäť so zákazníkom objasnili architektúru systému a väzby medzi jednotlivými komponentami v projekte.

Rozvrhli sme nové úlohy týkajúce sa klienta, servera, databázy RDF a ďalších dokumentácií. Pri jednotlivých úlohách prebehlo hlasovanie o ich náročnosti, kde každý člen tímu vyjadril svoj názor na danú úlohu.

Jednotlivé úlohy sme rozdelili medzi členov tímu s tým, že zatriedenie sa ešte mohlo neskôr zmeniť:

- Michal Kráľ – rozbehanie MS SQL na manažovanie úloh
- Lukáš Račko – rozbehanie jeny, webová služba
- Peter Kučera – klient, gui, popup
- Peter Bobovský – DLL, pomoc s klientom
- Michal Slovík – upratanie úloh v TFS, práca na klientovi
- Miriama Pomffyová – Repozitáre
- Marek Mura – Repozitáre
- Michal Slovík – Repozitáre

Dňa 11.10. sa uskutočnilo druhé stretnutie k tomuto šprintu, na ktorom sa dokončilo ohodnocovanie a zatriedenie úloh v TFS a priradili sme zodpovedných ľudí k jednotlivým user stories. Nestihli sme vytvoriť všetky potrebné úlohy k jednotlivým user stories, čo sme nechali na priebežné doplnenie počas tohto a aj ďalších šprintov priebežne, ako budú nasledovať.

TÍMOVÝ PROJEKT TRACKS 4

ZÁPISNICA

DÁTUM KONANIA / MIESTO KONANIA	17.10.2016, 3.21
ZÚČASTNENÍ	ING. MARTIN KONŮPKA (MAKO) MIRIAMA POMFFYOVÁ (MIPO) MICHAL SLOVÍK (MISL) PETER BOBOVSKÝ (PEBO) PETER KUČERA (PEKU) MAREK MURA (MAMU) MICHAL KRÁL' (MIKR) LUKÁŠ RAČKO (LURA)
NEPRÍTOMNÍ	-
ZAPÍŠAL	MAREK MURA
PRÍLOHY	-

Program porady

1. Retrospektíva k predošlému šprintu
2. Určenie taskov pre nový šprint a presunutie nedokončených taskov do aktuálneho šprintu

Počas retrospektívy k predpošlému šprintu sme zhodnotili, ktoré tasky boli splnené a v akej miere. Po uplynulom šprinte sa podarilo nasledovné:

- podarilo sa rozbehnutie mssql databázy a ukladanie jobov do databázy
- bola vytvorená základná štruktúra pre dll. Súbory
- klientská aplikácia beží ako daemon
 - treba dohodnúť, ako daemon bude komunikovať s jobom
- webstránka projektu je na stránke fakulty aj vo vlastnom repozitári na tfs, možno vo ľne upravovať

- reprezentácia dát stiahnutých z gitu pomocou trojíc

Úlohy, ktoré zatiaľ neboli dokončené sa presúvajú do aktuálneho šprintu.

Zároveň sme si so zákazníkom znovu ujasnili štruktúru projektu a spôsob, akým bude zabezpečená komunikácia medzi jednotlivými komponentmi softvéru. Na identifikáciu klientskeho zariadenia poslúži meno používateľa zadané v nastaveniach daemona, jeho id a ip adresa, z ktorej komunikuje. Zatiaľ nevyriešeným problémom ostáva určenie spôsobu, akým bude daemon komunikovať s konkrétnym jobom. User story "klient sa chce ozvať serveru" klesol na priorite a jeho dokončenie sa odkladá.

TÍMOVÝ PROJEKT TRACKS 5

ZÁPISNICA

Dátum konania / Miesto konania	24.10.2016, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	<i>Michal Kráľ</i>
Prílohy	-

PROGRAM PORADY

1. Retrospektíva k predošlému šprintu
2. Diskusia
3. Vytvorenie nových úloh pre ďalší šprint
4. Záverečné ustanovenia

DETAILY

V retrospektíve sme zhodnotili priebeh posledného šprintu.

Po retrospektíve prebiehala diskusia počas ktorej sme rozoberali nasledovné:

- Lepšie navrhnutie modelu relačnej databázy MSSQL. Rozhodli sme o ponechaní pôvodnej tabuľky configValues z dôvodu normalizácie dát v databáze.
- Zhodnotenie jennatriplestore databázy. Priradili sme konkrétny element k premennej jobUri.
- Úprava štruktúry jobExecutabledll na interface. Dohodli sme sa čo bude obsahovať dictionary a na zmenení návratových hodnôt z metód na boolean.
- Rozhodli o sťahovaní dát z Buggzily podľa bugov a nie chronologicky podľa klientov. Z dôvodu, že možno nestiahneme z projektu všetko. Takto budú vždy kompletne aspoň jednotlivé bugy a nie len časť toho čo komitoval používateľ.
- Nájdené riešenia pre popup okno v daemonovy.

Po diskusii sme dopĺňali a upravovali jednotlivé user stories v backlogu. Po úprave sme im prideľovali story pointy.

Úlohy, ktoré sa nedokončili sa presunuli do ďalšej iterácie.

Presunuli sme US z backlogu do novej iterácie a vytvorili sme k nim tasky.

Zákazník navrhol vytvorenie prvého míľnika. Je načasovaný na 10 týždňov v semestri. Za prvý míľnik sa pokladá základné všetkých komponentov v projekte a ich vzájomné prepojenie a komunikácia. Tak aby bola zabezpečená základná funkcionálna v projekte.

Na záver sme sa dohodli na tom, že najbližší šprint bude z dôvodu zložitejších úloh a sviatku až dvojtýždňový. Počas neho sa uskutoční jedno stretnutie aby sme sa informovali o prograse v jednotlivých US. Toto stretnutie sme dohodli na 3.11.2016 o 10:00 v Jobsovom štúdiu.

TÍMOVÝ PROJEKT TRACKS 6

ZÁPISNICA

Dátum konania / Miesto konania	7.11.2016 – FIIT 3.21
Zúčastnení	Ing. Martin Konôpka Miriam Pomffiová Michal Slovík Peter Bobovský Peter Kučera Marek Mura Michal Kráľ Lukáš Račko
Nepítomní	-
Zapísal	<i>Michal Slovík</i>
Prílohy	-

PROGRAM PORADY

Zhodnotenie posledného šprintu.

Nové úlohy na nový šprint (dvojtýždňový)

Úspešne kontrolovanie RDF trojíc a ich vkladanie do jeny (Lukáš Račko). Evidovanie fungujúcich a pripojených klientov, ešte treba dokončiť automatické odhlásenie po určitom čase, kedy je používateľ neaktívny (napr.: po 1 hodine).

Pracovanie s DB úspešné a funkčné (Mišo Kráľ). Možnosť pridávania, upravovania a mazania aj s databázou funkčné.

DLL knižnice, s ktorými dokáže pracovať klient, vie si ich otvoriť a pracovať s metódami (Peťo Bobovský) je funkčné, ešte je potrebné vytvoriť funkčný submodul.

BugZilla dokumentácia je vytvorená a prehľadná (Marek Mura). Je potrebné ešte vytvoriť

Skrytie PoPup ako WPF okno sa tiež podarilo (Peťo Kučera).

Dokončenie a pripojenie do hlavnej vetvy odosielanie rdf trojíc medzi klientom a webom (Michal Slovík).

Logovanie klienta, vytvorenie konfiguračného súboru pre nastavenie DB.

Tvorenie nových úloh pre ďalší šprint

Vytvorenie nových úloh pre ďalší šprint.

Dohodnutie priebežného stretnutia počas šprintu.

Rozdelenie jednotlivých manažmentov na predmet MSI/MIS.

TÍMOVÝ PROJEKT TRACKS 7

ZÁPISNICA

Dátum konania / Miesto konania	14.11.2016 – FIIT 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Peter Bobovský
Prílohy	-

Program porady

Zhodnotenie prvého týždňa šprintu

Retrospektíva a zhodnotenie

Lukáš spravil ontológie pre triplestore a joburi a manažovanie klientov, ešte neurobil pullrequest. Vyskytla sa otázka či trackovať aktuálne aktívnych userov na základe alebo v rámci bežiackej aplikácie na serveri.

Peter Bobovsky riešil submodule aby sa updatoval podľa konkrétneho branchu v gite v zdieľanom repozitári, namiesto mastera. Analyzoval Stackapi ale vznikajú otázky ohľadom toho či chceme sťahovať stack a koľko odtiaľ. Taktiež na stratégii vyšlo napojenie, vzhľadom na sťahovanie jednoduchých dát a potom po zistení napojenia by sa stiahlo viacej.

Peter Kučera meškal 52 minút na stretnutie. Za týždeň narazil na problém mergovania vetiev v gite.

Michal Slovík narazil na problém zatvárania klientskej aplikácie a uloženia dát. Konkrétne ak by nastala situácia že používateľ aplikáciu chce zavrieť, volanie na zatvorenie sa nedá pozdržať a pustiť uloženie dát najskôr.

Michal Kráľ spojzdnil zadávanie projektov pre používateľa aj keď to prakticky ešte nefunguje. Verziovanie jobexecutable prirobil aby tam bola tabuľka pre jednotlivé repozitáre. Globálne repository type je ešte stále staticky zadané, v ďalších user stories by sa to malo zmeniť na dynamickejšie, zadávané userom. Chceme dať používateľovi možnosť zadávať aj tieto typy a ak

áno necháme ho vybrať stratégiu podľa ktorej sa to bude sťahovať? Vedel by používateľ čo stratégia znamená?

Mirka vytvorila dokumentáciu ku kontrolnému bodu a zadaniu taskov na doplnenie tejto dokumentácie. Tiež vyrobila konfiguračný súbor ku používaniu databázy.

TÍMOVÝ PROJEKT TRACKS 8

Zápisnica

Dátum konania / Miesto konania	21.11.2016,
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	Miriama Pomffyová
Prílohy	-

PROGRAM PORADY

1. Retrospektíva k predošlému šprintu
2. Vytvorenie nových úloh pre ďalší šprint
3. Vytvorené Epicy, featury, stories v TFS
4. Ohodnotenie user stories v TFS pomocou SCRUM kartičiek

DETAILY

Peter Kučera spravil okienko pre zobrazenie id vráteného zo servera. Ukladá sa do properties, dá sa zadať aj meno človeka. Na servery spravil http get request na strane servera. Nepodarilo sa mu urobiť update sťahovania. Keď chce serializovať DLL, dá sa rozbiť do stringu?

Pozrieť: Ako sa sťahujú súbory v klientovi a potom ako sa pošlú na server.

Marek Mura mal problém s vraj funkčným DLLkom na oficiálnej stránke- link je nefunkčný. Pomocou inštancie servera sa má pripojiť na Bugzillu. Nedarí sa to – nedostáva response na základe url.

Pozrieť: má to fungovať tak, že sa prihlási na API, od ktorej si dáta sťahuje. Pozrieť novú API, ktorá

bude funkčná, nebude mať staré xml.

Michal Kráľ konzultoval kód stratégií. Delenie podľa commitov a chronologicky? Podľa toho, čo vieme ťahať.

Keď vytvárame nový repository version, user je nútený vybrať buď už vytvorené repository, alebo vyrobiť nový. Vyberieme aký typ repository chceme sťahovať. Môžeme ich menovať, ale nie zmazať keď už sú priradené.

Rozšíril databázu o ProjectState, vytvárajú sa hodnoty New, In progress.

Lukáš Račko pull requestol sa kód je v develope – vlastná ontológia pre špeciálne prípady.

Oznamovanie dostupnosti klientov na server - vytvorené funkcie pre oznamovanie. Spísal dokumentáciu pre prípady: keď sa prihlasuje klient prvýkrát, získava ID a musí si ho uložiť, pokiaľ druhý a ďalšíkrát prihlasuje sa pomocou zapamätaného ID.

Peter Bobovský vie sťahovať zo StackOverflow, vie nastavovať filtre – podľa zadaných parametrov. TryGetValue podľa názvov key získa hodnoty. Musíme zistiť názvy key. Názvy tagov zistiť podľa repository.

Miriama Pomffyová pracovala na dokumentácii Kontrolný bod, prvotné a finálne formátovanie.

Prechádzala API na sťahovanie dát z GitHub. Branch vytvárať nie z TRACKS-Daemon ale z TRACKS-Job.

Dorobila nastavenie pre každého používateľa user.json. Zapísané v gitignore, aby sa pri committe neuploadoval.

Michal Slovák exportoval úlohy z TFS a upravoval dokumentáciu na Prvý kontrolný bod. Dokončenie sťahovania job – pri zatváraní aplikácie sa kontroluje stav, aby sa nezavrela skôr ako dokončí pridelenú úlohu, sťahovanie.

Stratégie:

Na klientovi, ktorá bude hovoriť ktorý repository type používaš podľa toho budeš sťahovať.

Stratégia, ktorý execution dať klientovi.

Ako sťahovať repository type.

Úloha pre tím:

Písať si hodiny do TFS.

Správne uzatvárať user story a riešiť ich presúvanie.

TÍMOVÝ PROJEKT TRACKS 9

Zápisnica

Dátum konania / Miesto konania	28.11.2016
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Lukáš Račko (LuRa)
Neprítomní	Michal Kráľ (MiKr)
Zapísal	Michal Slovík
Prílohy	-

PROGRAM PORADY

retrospektíva šprintu,

presunutie tasku do ďalšieho šprintu

RETROSPEKTÍVA

Mirke bol pridelený Bug Task, pretože nejde spustiť unit testy v dôsledku user.json.

Peter Bobovský potrebuje vytvoriť nové ontológie potrebné pre sťahovanie StackOverflow, zvyšné tasky prechádzajú do ďalšieho šprintu.

Michal Slovík dokončenie sťahovanie a zipovanie .DLL a úspešne extrahovanie. Potrebné prerobiť na nie napevno dané cesty. A otestovať plus unit testy.

Michal Kráľ vytvoril stratégiu ktorá vytvára interval v určenom rozsahu (stratégia na základe času). Hotové a code revision vykonal Lukáš Račko.

Lukáš Račko pridal pridelenie JobExecutions vo ľným klientom, ktorý sa ohlasujú na server. Dorobiť JobExecutions údaje. Pridal do tabuľky JobExecutions cudzí kľúč na klienta.

Marek Mura vytvorenie jobov pre stiahnutie BugZilly. Podporu majú pre mozzilu, pre ostatné repozitáre to nemá podporu.

Peter Kučera vytvoril získavanie JobExecutions od servera. Spustenie JobExecutions ešte nebolo vykonané.

Miriama Pomffyova určenie spôsobu sťahovania dát z Git-u. Cez nástroj Fiddler ukazané.

PLÁN NA ĎALŠÍ ŠPRINT

Aby sa JobExecutions posielali zo servera do klienta. Dokončiť veci z predošlého šprintu.

Vytvoriť otázky na AGILNE a BIZNIS na stretnutie.

TÍMOVÝ PROJEKT TRACKS 10

Zápisnica

Dátum konania / Miesto konania	5.12.2016
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffiová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	<i>Peter Bobovský</i>
Prílohy	-

Program porady

Zhodnotenie posledného šprintu.

Retrospektíva a zhodnotenie

Peter Bobovský spravil generovanie trojíc a ich ukladanie do grafu. Ontológiu rozpracoval ale narazil na problémy s jej formálnym zápisom. Kontakt so serverom ešte nedokončil, ale už rozpracoval reportovanie statusu downloadovania klientovi. Riešil všeobecné problémy so submodulami.

Marek Mura analyzoval bugzillove downloadovanie a vyrobil prvú funkčnú verziu v DLL.

Mirka sa venovala dopytovanie cez http requestom v githube. Narazila na problém s asynchrónnym volaním. Taktiež sa venovala príprave otázok a prezentácie na mentoring.

Lukáš dokončil možnosť pre klienta stiahnuť si job executions, všetko už je zamergované v develope. Tiež riešil zaznačenie úlohy do databázy že klient ukončil úlohu.

Mišo Kráľ pomohol opraviť bug Mirke a inak riešil všeobecnú organizačnú synchronizáciu.

Michal Slovik riešil stiahnutie zip súboru z balancera vytvorenie dynamických ciest a kontrola s databázou, mimo iné aj unit test. V práci mu prekážal bug s visualkom, ktorý spôsobil nemožnosť spustiť server. Bug včasne vyriešil a do TFSka zapísal postup riešenia bugu.

Peto Kucera managoval UX na klientovi, zakomponoval kontakt so serverom, ako napríklad zobrazovanie job executions v zozname na klientovi aj proces sťahovania.

Prišli sme na problém so submodulami a ich synchronizáciou.

Tvorenie nových úloh pre ďalší šprint

Vytvorenie nových úloh pre ďalší šprint.

Dohodli sme sa na termíne na bonusové stretnutie týkajúce sa dokumentácií.

TÍMOVÝ PROJEKT TRACKS 11

Zápisnica

Dátum konania / Miesto konania	12. 12.2016 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Marek Mura
Prílohy	-

PROGRAM PORADY

Retrospektíva šprintu

RETROSPEKTÍVA

Google disk a dropbox repozitáre boli upratané, dohodli sme sa na konvencii pomenovávaní súborov.

Michal Kráľ vytvoril template pre unit testy a niekoľko unit testov, upratol google disk.

Lukáš Račko zobrazuje v klientovi dokončenosť úloh percentuálnym pomerom dokončených a všetkých úloh.

Marek Mura spravil vytváranie RDF trojice zo sťahovaných dát z bugzilly, upratal dropbox.

Peter Kučera implementoval do klienta funkcionalitu, aby za ozval serveru keď skončí sťahovanie.

Peter Bobovský vytvoril ontológi pre StackOverflow a implementoval odosielanie stavu dokončenosti daemonovi.

Mirka Pomffyová spravila ukladanie stiahnutých JSON objektov do .net tried.

Michal Slovík pracoval na prezentácii tímu a dokumentácii.

Zároveň sa celý tím venoval dokončovaniu dokumentácie a inžinierskeho diela pre odovzdanie kontrolného bodu. V priebehu stretnutia sme prediskutovali, kto a do akej miery dokončil svoju časť a spojili sme ich dohromady v jednom dokumente.

Bol predložený návrh na použitie Source 3 na formátovanie kódu. Pokúsime sa o inštaláciu v ďalšom semestri.

PLÁN DO BUDÚCNOSTI

Najbližšie stretnutie je naplánované na 12.1.2017, keďže v tom čase väčšina členov tímu bude na FIITke kvôli skúške. Dovtedy každý člen tímu spraví úlohy, ktoré mu ostali.

TÍMOVÝ PROJEKT TRACKS 12

Zápisnica

Dátum konania / Miesto konania	16.2.2017, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	<i>Miriama Pomffyová</i>
Prílohy	-

PROGRAM PORADY

Zhodnotenie súčasného stavu projektu.

Pridelenie vytvorených Epicov, featury, stories v TFS.

Ohodnotenie user stories v TFS pomocou SCRUM kartičiek.

ETROSPEKTÍVA

Dohodli sme sa na týždenných šprintoch.

Peter Bobovský bude SCRUM master.

Vytvorili sme dokument na GoogleDisku pre článok na IITSRC. Prediskutovali sme, čo je potrebné ešte zahrnúť do rozpracovaného článku. Odovzdanie je dnes do polnoci.

Pridelili sme si user stories. Ktoré neboli ukončené zo zimného semestra sme preniesli do tohto šprintu. Vytvorili sme nové US a tasky - zapísané v TFS.

PLÁN DO BUDÚCNOSTI

Najbližšie stretnutie je naplánované na 23.2.2017. Sú pridelené US na vypracovanie.

TÍMOVÝ PROJEKT TRACKS 13

Zápisnica

Dátum konania / Miesto konania	23.02.2017
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriana Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Michal Kráľ
Prílohy	Marek Mura meškala kvôli autoškole. Peter Bobovský musel odísť skôr kvôli diplomovej práci.

PROGRAM PORADY

Zhodnotenie súčasného stavu projektu.

Pridelenie vytvorených stories v TFS.

Ohodnotenie user stories v TFS pomocou SCRUM kartičiek.

RETROSPEKTÍVA

Ontológia pre stackOverflow je hotová.

Manuál pre vytváranie ontológie je hotový.

CI pre buildovanie deamona je funkčné.

CI pre buildovanie servera zatiaľ nejde.

FileZilla pre FTP pripojenie zatiaľ stagnuje.

Unix server je pripravený na publishnutú verziu z TFS. Dokáže projekt aj publishnúť a spojzdníť online.

Odstránenie submodulov a nahodenie namiesto toho nugovet, už skoro funguje. Je to na správnej ceste.

Preposielanie zip súborov medzi klientom a serverom funguje. Bude treba trochu zmeniť logiku posielania týchto súborov. Momentálne posieľa vždy najnovší.

Ontológia pre bugzillu je hotová ale treba prerobiť pár vzťahov.

Článok na IITSCR bol korektne odovzdaný v riadnom čase.

DETAILY

Rozhodli sme sa poskytnúť deamona žiačke na bakalárskom stupni kvôli bakalárskemu projektu.

PLÁN DO BUDÚCNOSTI

Ostáva už len mesiac na dokončenie projektu, preto sme si zobrali viacej taskov a budeme v ďalších šprintoch viacej pracovať.

TÍMOVÝ PROJEKT TRACKS 14

Zápisnica

Dátum konania / Miesto konania	02.03.2017
Zúčastnení	Ing. Martin Konôpka (MaKo) Hostiteľ: Ing. Karol Rastočný, Phd. Miriam Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Michal Slovák
Prílohy	

PROGRAM PORADY

Zhodnotenie súčasného stavu projektu.

Pridelenie vytvorených stories v TFS.

Ohodnotenie user stories v TFS pomocou SCRUM kartičiek.

RETROSPEKTÍVA

Server bol založený online na adrese www.team08-16.studenti.fiit.stuba.sk:8008

Demo prijímanie deamona DLL a executable.

Nuget bol vytvorený a úspešne importovaný. Problém je pri uploadovaní na vzdialený server (DevACT).

FTP beží na serveri a bolo úspešne spojené s TFS.

Ontologie pre BugZillu a Git.

Vyrabanie trojic pre ontologiu Git-u.

DETAILY

Sťahovanie Git-u rozdelené podľa denných commitov.

Preklopenie submodulov na nugety

Tracks - Shared

Track - Job-Shared

PLÁN DO BUDÚCNOSTI

Rozbehať normálny a štandardný logger na serveri. Rozbehnutie CI naprieč TFS a správnym nakonfigurovaným ciest a skriptov na serveri.

Vytvorenie demá pre klienta aj s grafickým rozhraním.

Vytvorenie dema pre sťahovanie bugZilly.

A ďalšie tasky sú v TFS.

TÍMOVÝ PROJEKT TRACKS 15

Zápisnica

Dátum konania / Miesto konania	9.3.2017, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Marek Mura
Prílohy	-

PROGRAM PORADY

Zhodnotenie uplynulého šprintu:

- upravený repozitár pre NuGety, vytvorený manuál
- integrované CI s generovaním NuGetov
- plynulé sťahovanie userov so stacku, odosielanie progresu klientovi
- dáta z Gitu sťahované po intervaloch pre konkrétnych userov podľa konfigurácie a odosielené na server
- dáta o bugoch z Bugzilly sťahované po častiach a odosielené na server
- opravených niekoľko bugov
- zmena databázy na školský server

Riešenie problémov

V rámci stretnutia boli vyriešené bugy a problémy, ktoré si vyžadovali osobnú účasť a komunikáciu viacerých členov tímu.

Na stretnutí sa účastnil aj Karol Rástočný. Po konzultácii s ním sme zistili, že spôsob načítavania ontológie a overovanie RDF grafov, ktorý v súčasnosti používame, je nekorektný a potrebujeme ho prerobiť.

PLÁN DO BUDÚCNOSTI

Koniec nasledujúceho šprintu bol stanovený na 16.3. Boli naplánované tasky a US na tento šprint.

TÍMOVÝ PROJEKT TRACKS 16

Zápisnica

Dátum konania / Miesto konania	16.03.2017
Zúčastnení	Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	Martin Konôpka
Zapísal	<i>Peter Kučera</i>
Prílohy	

PROGRAM PORADY

Zhodnotenie súčasného stavu projektu.

Pridelenie vytvorených stories v TFS.

Ohodnotenie user stories v TFS pomocou SCRUM kartičiek.

RETROSPEKTÍVA

Bol naimplementovaný kód pre IGraph. Padla pripomienka ze je to ale drasticky inak s povodnou implemntaciou RDF grafov.

Bol pridany loading bar na server zobrazujuci priebeh stahovania. Taktiez boli najdene a opravene a zdokumentovane bugy v serverovej casti aplikacie.

Pri stahovani udajov zo StackOverflow sme v klientovi narazili na problem pri balancovani requestov.

Bolo plne spojzdnene CI.

Bol vytvoreny instalacny balicek aplikacie z ktorej je mozne TRACKS klienta nainstalovat.

Boli opravene chyby v klientovi z minuleho tyzdna.

PLÁN DO BUDÚCNOSTI

Dokončit tasky z minuleho tyzdna.

Zimplementovat IGraph implementáciu RDF grafov.

Vyriesit najdene buggy v serverovej casti aplikacie.

Vyriesit problem s podpisovanim setup balika na klientovi.

TÍMOVÝ PROJEKT TRACKS 17

Zápisnica

Dátum konania / Miesto konania	23.03.17
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Peter Bobovský
Prílohy	-

PROGRAM PORADY

Peťo Bobovský prerobil generovanie trojíc priamo do IGraphu podľa funkcionality igraphu, ale narazil na problémy s Karolovym novým interfacom

Lukáš pridal filter na jobExecutions, ale mal tam ešte chybu ktorú nestihol vyriešiť. Mišovi Kráľovi sa jeho riešenie až tak nepáčilo takže je možnosť to trosku upraviť.

IIT.SRC článok sme upravili podľa pripomienok a uploadli do systému easychair, ale zabudli sme to poslať Kompanovi, takže to sa dodatočne robilo. Text sme tiež nemali v šablóne takže dodatočne sa to dorába.

Peťo Kučera urobil redesign klienta ako aj updatovanie progress baru podľa toho čo DLL robí.

Michal Slovík upratal stránku, zmanažoval funkciu klienta s online serverom a ich komunikáciu, poopravoval všetky errorry na ktoré pritom narazil.

Michal Kral pridal inštrukcie ku jobom na serveri aby používateľ vedel ako nastaviť config itemy.

Rozhodli sme sa že názov projektu sa bude posúvať až DLL a to ho bude uploadovať s dátami aby sa do Jeny uploadovala na vlastné repozitáre

Ďalej sme sa dohodli na zmenách na serveri, aby sa dali upravovať existujúce a bežiacie projekty a zobrazovanie progressu v repozitároch.

Mirka doplnila informácie o sťahovaní z bugzilly, nastavila sťahovanie repozitárov cez organizácie. Testovala limit sťahovania

TÍMOVÝ PROJEKT TRACKS 18

Zápisnica

Dátum konania / Miesto konania	30.3.2017, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Lukáš Račko
Prílohy	-

PROGRAM PORADY

Zhodnotenie uplynulého šprintu:

- Vyriešenie dependencies, opravene bugy v klientovi
- Ošetrenie limitu pri sťahovaní z repozitárov
- Finalizácia jobExecutable pre sťahovanie dát z BugZilly
- Oznámenie úspešného alebo neúspešného sťahovania JobExecution-ov
- Rozdelenie nedokončeného JobExecution na dva, nedokončenú časť je možné prideliť inému klientovi
- Otestovanie Clienta s UNIX serverom
- Úprava ukladania dát do jeny cez PostRDFGraph funkciu
- Pridaná kontrola v databáze, či projekt, pre ktorý sa sťahujú dáta ešte existuje
- Zobrazovanie progresu v ManageRepositories

Riešenie problémov

V rámci stretnutia boli vyriešené bugy a problémy, ktoré si vyžadovali osobnú účasť a komunikáciu viacerých členov tímu.

PLÁN DO BUDÚCNOSTI

Na stretnutí sme si stanovili nové úlohy, ktoré majú za cieľ spojazdniť celý proces od vytvorenia projektov, repozitárov a ďalších dát na serveri, generovanie JobExecution, prideľovanie úloh klientov, stiahnutí dát, spracovanie dát do grafu trojíc, poslanie na server a uloženie do Jeny

Koniec nasledujúceho šprintu bol stanovený na 6.4. Boli naplánované tasky a US na tento šprint.

TÍMOVÝ PROJEKT TRACKS 19

Zápisnica

Dátum konania / Miesto konania	6.4.2017, 3.21
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Marek Mura
Prílohy	-

PROGRAM PORADY

Zhrnutie uplynulého šprintu:

- Mirka dokončila prerobenie Git executable na interface a korektne odosiela progres klientovi.
- Marek refaktoroval Bugzilla executable a narazil na problém s rozdeľovaním sťahovania na intervaly, ktorého riešenie bolo prediskutované na stretnutí.
- Peter Bobovský implementoval prerušenie práce dll pri prekročení limitu alebo vyžiadaní zastavenia od klienta.
- Peter Kučera implementoval možnosť zastavenia job executable z klienta a zároveň vyriešil niekoľko bugov a drobností v klientovi.
- Michal Kráľ riešil logovanie crash reportov, ktoré sa teraz správne ukladajú do databázy. Zároveň riešil prioritizáciu úloh zo strany servera.
- Lukáš Račko dorábala filter kde bol už dlhšiu dobu bug, teraz funguje pre akúkoľvek kombináciu zadaných parametrov. Zároveň pridal chýbajúce parametre a možnosť nastavenia limitu zobrazených executions
- Michal Slovík riešil odosielanie informácií loggeru od klienta balancerovi.

PLÁN DO BUDÚCNOSTI

Na stretnutí sme stanovili úlohy na nasledujúci šprint. Cieľom úloh je vytvorenie konfiguračných súborov pre všetky dll, spravovanie limitov sťahovania, práca na dokumentácii projektu a príprava tímu na IIT.SRC konferenciu vytvorením plagátu, spozajdnením dema a premyslením prezentácie.

Koniec nasledujúceho šprintu bol stanovený na 13.4. Boli naplánované tasky a US na tento šprint.

TÍMOVÝ PROJEKT TRACKS 20

Zápisnica

Dátum konania / Miesto konania	13.4.2017
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	<i>Miriama Pomffyová</i>
Prílohy	-

PROGRAM PORADY

Čo sa stihlo alebo nestihlo urobiť za posledný šprint,

Prechádzať jednotlivé úlohy na tfs.

ZHODNOTENIE UPLYNULÉHO ŠPRINTU

Prišiel Karol Rástočný s požiadavkou na vytvorenie používateľskej príručky pre vytvorenie jobov. Michal Slovák ukázal video s vytvorením jobu klientovi a sťahovaním u klienta a poslaním dát na server.

Ukážka od Peťa Kučeru Download Client, stiahol sa setup.exe.

PLÁN DO BUDÚCNOSTI

Na stretnutí sme si stanovili nové úlohy, ktoré majú za cieľ ...

Koniec nasledujúceho šprintu bol stanovený na 20.4. Boli naplánované tasky a US na tento šprint.

TÍMOVÝ PROJEKT TRACKS 21

Zápisnica

Dátum konania / Miesto konania	20.4.2017
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Nepřítomní	-
Zapísal	Michal Kráľ
Prílohy	-

PROGRAM PORADY

Diskusia o jednotlivých úlohách a aktuálnom stave TRACKS.

Čo sa stihlo alebo nestihlo urobiť za posledný šprint.

Vytváranie nových úloh v TFS.

DETAILY

Problém s FTP na unixe je zažehnaný.

Expirovali heslá na pripojenie do windows serveru. Martin ich obnovil.

Devacts aktuálne stagnuje, čakáme na jeho spojzdenie.

Ipraph funguje! Aj jeho preposielanie medzi klientom a balancerom!

Máme hotový plagát na TRACKS, po pár úpravách s ním všetci členovia súhlasia. Tričká stále stagnujú ale sme odhodlaní ich zabezpečiť.

V časti retrospektíva sme zhodnotili aktuálny stav.

Projekty, ktoré dokážeme sťahovať - Mozilla, Bugzilla.

ZISTENIA

V Jene musí byť všetko zadané pomocou URI.

PLÁN DO BUDÚCNOSTI

Otestovať OneClick.

Spísať veľa tutoriálov.

Stretnúť sa s Karolovou bakalárkou a vysvetliť jej ako to celé funguje.

Začať písať dokumentáciu.

Zabezpečenie technického vybavenia pre IITSRC.

Koniec nasledujúceho šprintu bol stanovený na 27.4. Boli naplánované tasky a US na tento šprint.

TÍMOVÝ PROJEKT TRACKS 22

Zápisnica

Dátum konania / Miesto konania	28.4.2017
Zúčastnení	Ing. Martin Konôpka (MaKo) Miriam Pomffyová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	-
Zapísal	Peter Bobovský
Prílohy	-

RETROSPEKTÍVA

V klientovi bolo upravené spúšťanie DLL na nový interface, Bolo upravené UI klienta aby bolo príjemnejšie na pohľad.

Logy z klienta sa už posielajú na server keď sa klient spustí.

Prebehli všeobecné zlepšenia o informovanosti ako prebieha progres jednotlivých projektov, pridávania repozitárov do projektu a ich mazanie.

Všetky DLL prešli na nový štandard komunikácie s klientom. Všetky DLL teraz aj budú posielať info o repozitári naspäť na server s dátami, takže sa môžu dáta ukladať do jeny do vlastných projektov.

TÍMOVÝ PROJEKT TRACKS 23

Zápisnica

Dátum konania / Miesto konania	4.5.2017
Zúčastnení	Miriama Pomffyová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Kráľ (MiKr) Lukáš Račko (LuRa)
Neprítomní	Martin Konôpka
Zapísal	<i>Miriama Pomffyová</i>
Prílohy	

PROGRAM PORADY

Zhodnotenie súčasného stavu projektu.

Pridelenie vytvorených stories v TFS.

RETROSPEKTÍVA

Pridali sme komentáre k metódam a premenným, ktoré chýbali pre vygenerovanie technickej dokumentácie.

Michal Kráľ našiel spôsob pre generovanie technickej dokumentácie z kódu.

Začali sme s finalizáciou dokumentácie. Doplnili sme návody.

Peter Kučera kontaktoval slečnu, ktorá má súvisiacu bakalársku prácu. Dodali jej návody.

Odprezentovali sme projekt na TP CUPE. Bol zhotovený scénar prezentovania, spravili sme nálepky a tričká a plagát, Michal Slovík ho dal na vytlačenie.

PLÁN DO BUDÚCNOSTI

Spísať finálnu dokumentáciu. Pospájať vytvorené priebežne súčasti dokumentácie.

Vygenerovať export úloh z TFS, doplniť komentáre k úlohám, doexportovať technickú dokumentáciu, zápisnice.

Vytvoriť testovacie scenáre a podať na otestovanie tretej strane.

Otestovať produkt druhého tímu.

a.