

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičová 3, 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 I
Vedúci: Ing. Martin Konôpka
Akademický rok: 2016/2017

Obsah

| | |
|---|----|
| 1. Úvod | 4 |
| 2. Členovia tímu | 4 |
| 2.1 Rozdelenie manažérskych úloh | 5 |
| 2.2 Rozdelenie vývojárskych úloh | 6 |
| 3. Aplikácie manažmentov | 7 |
| 3.1 Manažment dokumentácie | 7 |
| 3.2 Manažment testovania | 7 |
| 3.3 Manažment komunikácie | 7 |
| 3.4 Manažment chýb | 8 |
| 3.5 Manažment úloh | 8 |
| 3.6 Manažment prehliadok kódu | 8 |
| 3.7 Manažment verzí | 9 |
| 4. Sumarizácia šprintov | 10 |
| 4.1 Šprint 1 Anthriscusyellows | 10 |
| 4.2 Šprint 2 Banna | 10 |
| 4.3 Šprint 3 Chlamydia | 10 |
| 4.4 Šprint 4 Diarrhoea | 11 |
| 4.5 Šprint 5 Ebola | 12 |
| 4.6 Šprint 6 Fiji | 13 |
| 4.7 Globálna retrospektíva | 14 |
| 5. Používané metodiky | 15 |
| 5.1 Metodika dokumentácie | 15 |
| 5.2 Metodika písania a komentovania zdrojového kódu | 19 |
| 5.3 Metodika úloh | 22 |
| 5.4 Metodika komunikácie | 26 |
| 5.5 Metodika manažmentu verzí | 27 |
| 5.6 Metodika testovania | 31 |
| 5.7 Metodika prehliadok kódu | 32 |
| 6. Zápisnice | 33 |
| 6.1 TÍMOVÝ PROJEKT TRACKS 1 | 34 |
| 6.2 TÍMOVÝ PROJEKT TRACKS 2 | 39 |
| 6.3 TÍMOVÝ PROJEKT TRACKS 3 | 43 |
| 6.4 TÍMOVÝ PROJEKT TRACKS 4 | 45 |
| 6.5 TÍMOVÝ PROJEKT TRACKS 5 | 47 |

| | | |
|-----|--------------------------------|----|
| 6.6 | TÍMOVÝ PROJEKT TRACKS 6 | 49 |
| 6.7 | TÍMOVÝ PROJEKT TRACKS 7 | 51 |
| 7. | Big picture..... | 52 |
| 7.1 | Úvod | 53 |
| 7.2 | Globálne ciele projektu | 53 |
| 7.3 | Celkový pohľad na systém | 53 |
| 7.4 | MS SQL databáza..... | 55 |
| | Prílohy..... | 56 |

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 Kráľ

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 Pomffyová | Manažér dokumentácie a rozsahu projektu. |
| Lukáš Račko | Manažér verzií a zdieľaných prostriedkov. |
| Michal Slovík | Manažér testovania a webovej stránky a úloh. |

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 najužitočnejšia, 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 – zadáva úlohy do TFS, 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.
- Jednotlivé dokumentácie vypracovávajú členovia tímu po ich priradení.
- Uvádza sa autorstvo jednotlivých častí dokumentácie.

Proces tvorby dokumentácie opisuje metodika dokumentácie.

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 prehliadok 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 requesty (Bližšie popísané v metodike prehliadok kódu).

3.7 Manažment verzí

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 verzí používame TFS Git. Bližšie informácie k používaniu Gitu sú uvedené v metodike manažmentu verzí.

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 codeconventions
- + 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

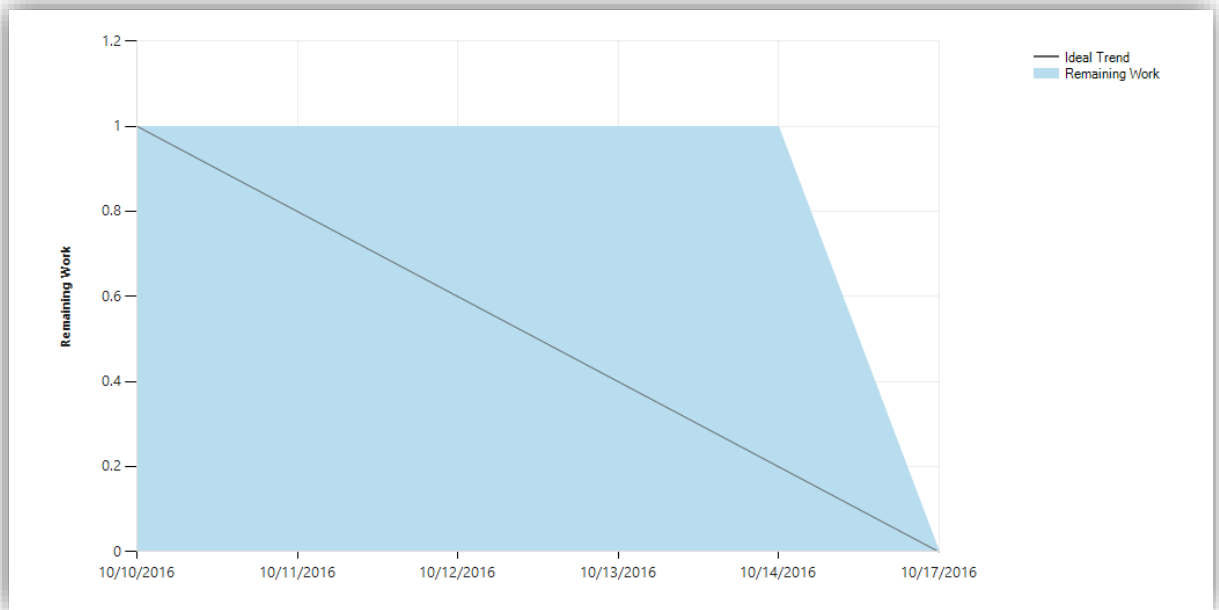
4.3 Šprint 3 Chlamydia

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

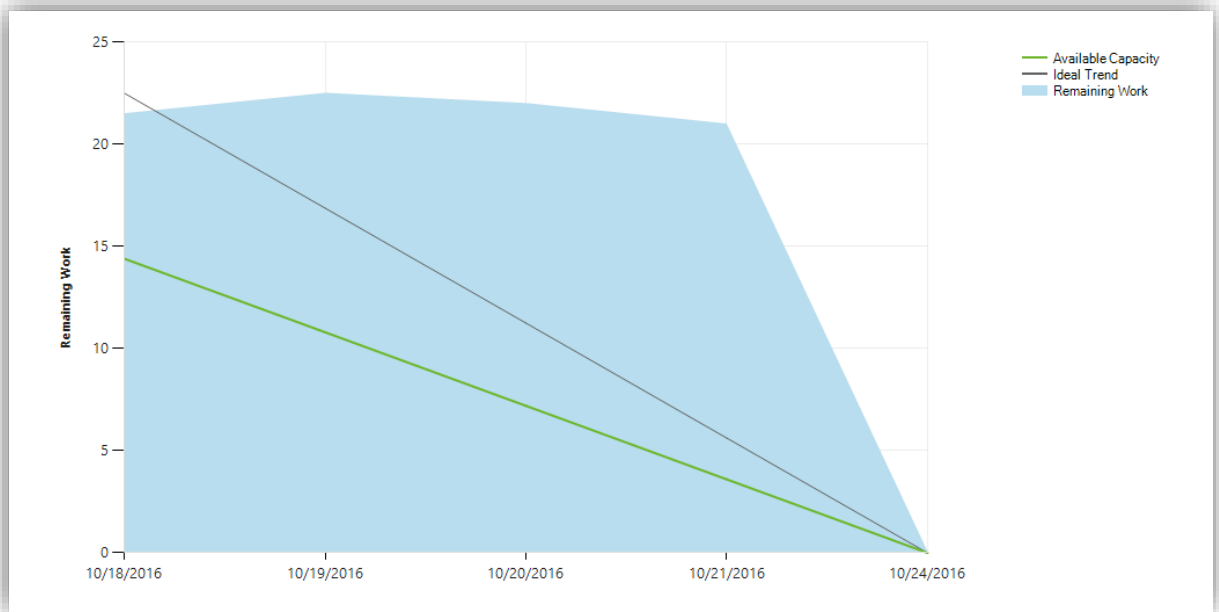
- + MSsqldatabáza
- skrytie klientskej aplikácie v tray
- analyzovanie existujúcich API na sťahovanie z jednotlivých repozitárov



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.

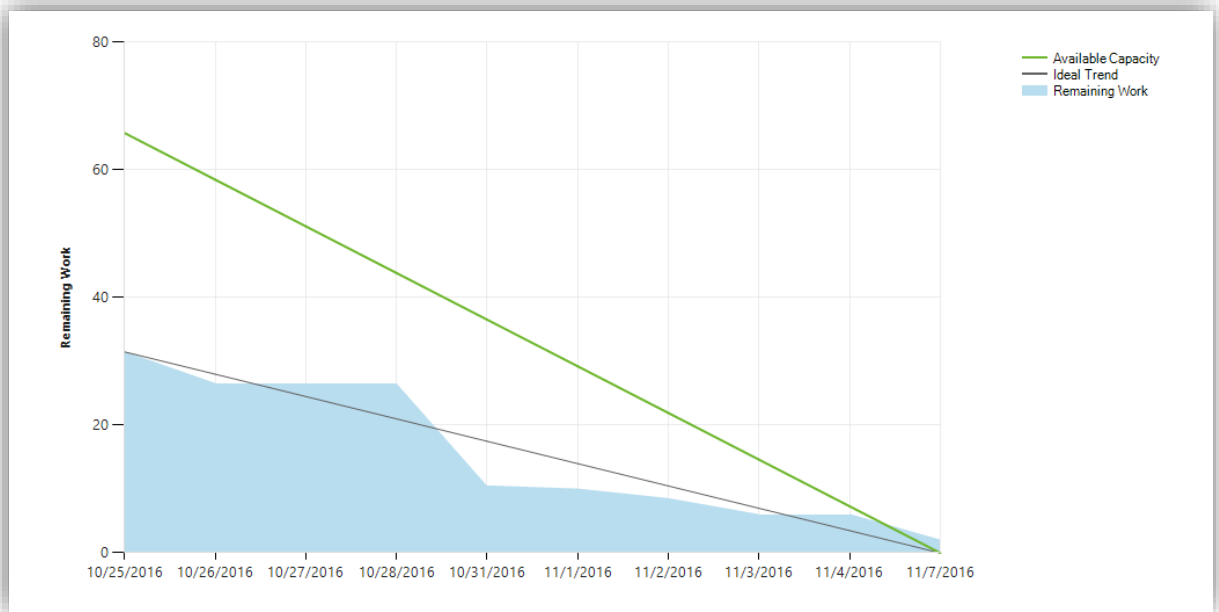
- + implementácia controllera.
- + DLL knižnica
- + API pre BugZillu
- prihláška na TP Cup nebola skontrolovaná všetkými členmi



4.5 Šprint 5 Ebola

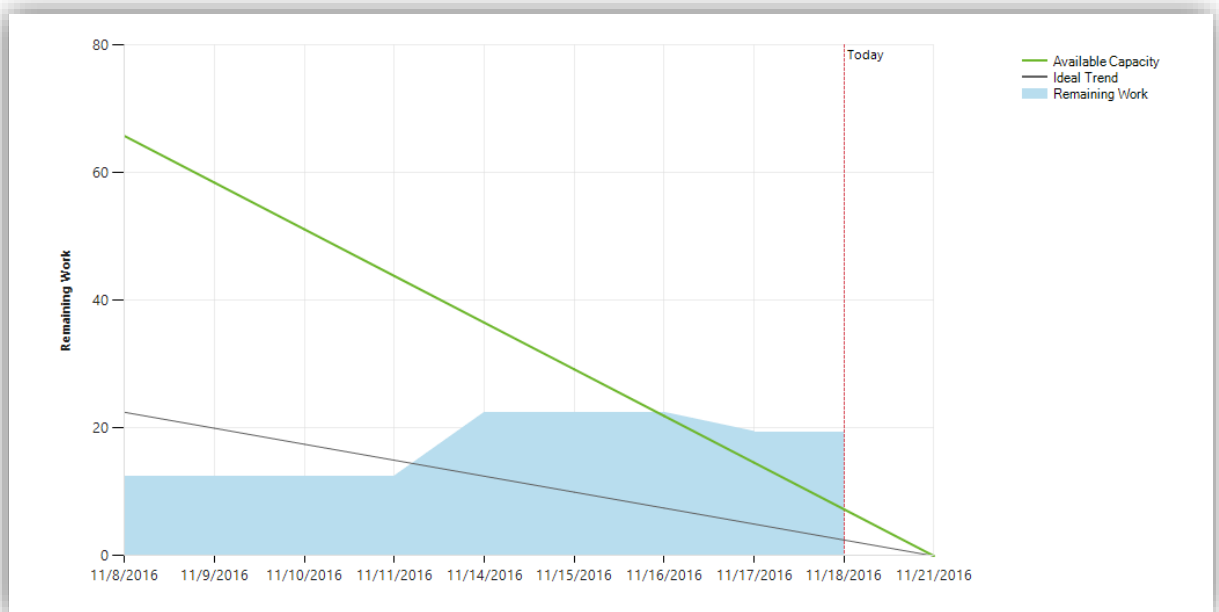
Prvý dvojtýždňový šprint. Rozhodovanie o implementácii modelu relačnej databázy. Zhodnutie sa na dvojtýždňovom šprinte a stanovenie prvého míľniku na 10. semestrový týždeň.

- + 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 TrippleUri. Ďalej je potrebné vytvorenie kontroly dostupnosti klienta a jeho stav sťahovania. Zistiť aktuálny API pre Git Eclipse a Gerrit.



4.7 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 spojzdníť 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.

5.1 Metodika dokumentácie

Táto metodika zobrazuje procesy pri vytváraní dokumentácii.

1. Určenie typu dokumentácie na základe požiadaviek

(Vystupujú tu: productowner, členovia tímu)

Inicializácia:

- Zhromaždenie požiadaviek na dokumentáciu.

Vykonávanie:

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

Ukončenie:

- Definovanie dokumentácie pre vypracovanie.

2. Pridelenie dokumentácie na vypracovanie

(Vystupujú tu: productowner, členovia tímu)

Inicializácia:

- Výber plánovanej dokumentácie.
- Zhodnotenie spôsobilosti účastníkov projektu.

Vykonávanie:

- Priradenie dokumentácie účastníkovi projektu.

Ukončenie:

- Pridelená dokumentácia na vypracovanie.

3. Spísanie metadát o dokumentácii

(Vystupujú tu: vybraný autor z členov tímu)

Inicializácia:

- Vytvorenie dokumentu určeným autorom.

Vykonávanie:

- Definovanie metadát o dokumentácii.
 - komu je určená,
 - súvisiace dokumentácie,
 - kto ju vytvoril,
 - posledná zmena,
 - koľká je to verzia,
 - referencie, ak sme vychádzali z nejakých zdrojov,

- skratky,
- pojmy, ktoré sú špecifické a doménové.
- Popis roly, ktorých sa dokumentácia týka a ktoré v nej vystupujú.

Ukončenie:

- Spísané metadáta dokumentácie.

4. Písanie jadra dokumentácie

(Vystupujú tu: autor z členov tímu)

Inicializácia:

- Definovanie cieľov dokumentácie.

Vykonávanie:

- Ak ide o:
 - metodiku:
 - Spísanie procesov a ich:
 - inicializáciu,
 - vykonávanie
 - a ukončenie.
 - 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 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 podaná na kontrolu.

5. Kontrola dokumentácie

(Vystupujú tu: kompetentné osoby z členov tímu)

Inicializácia:

- Predloženie dokumentácie na kontrolu poverených osobám.

Vykonávanie:

- Kontrola metadát dokumentácie.
- Kontrola jadra dokumentácie.

Ukončenie:

- 2 možnosti:
 - Schválenie správnosti dokumentácie.
 - Dokumentácia podaná na prepracovanie.

6. Uloženie dokumentácie

(Vystupujú tu: autor dokumentácie)

Inicializácia:

- Spísaná a odkontrolovaná dokumentácia.

Vykonávanie:

- Výber miesta uloženia dokumentácie.
- Uloženie dokumentácie.

Ukončenie:

- Uložená odkontrolovaná dokumentácia.

7. Sprístupnenie dokumentácie

(Vystupujú tu: autor dokumentácie, productowner, členovia tímu)

Inicializácia:

- Dokumentácia je uložená.
- Rozhodnutie sa o potrebe použitia dokumentácie.
- Výber dokumentácie.

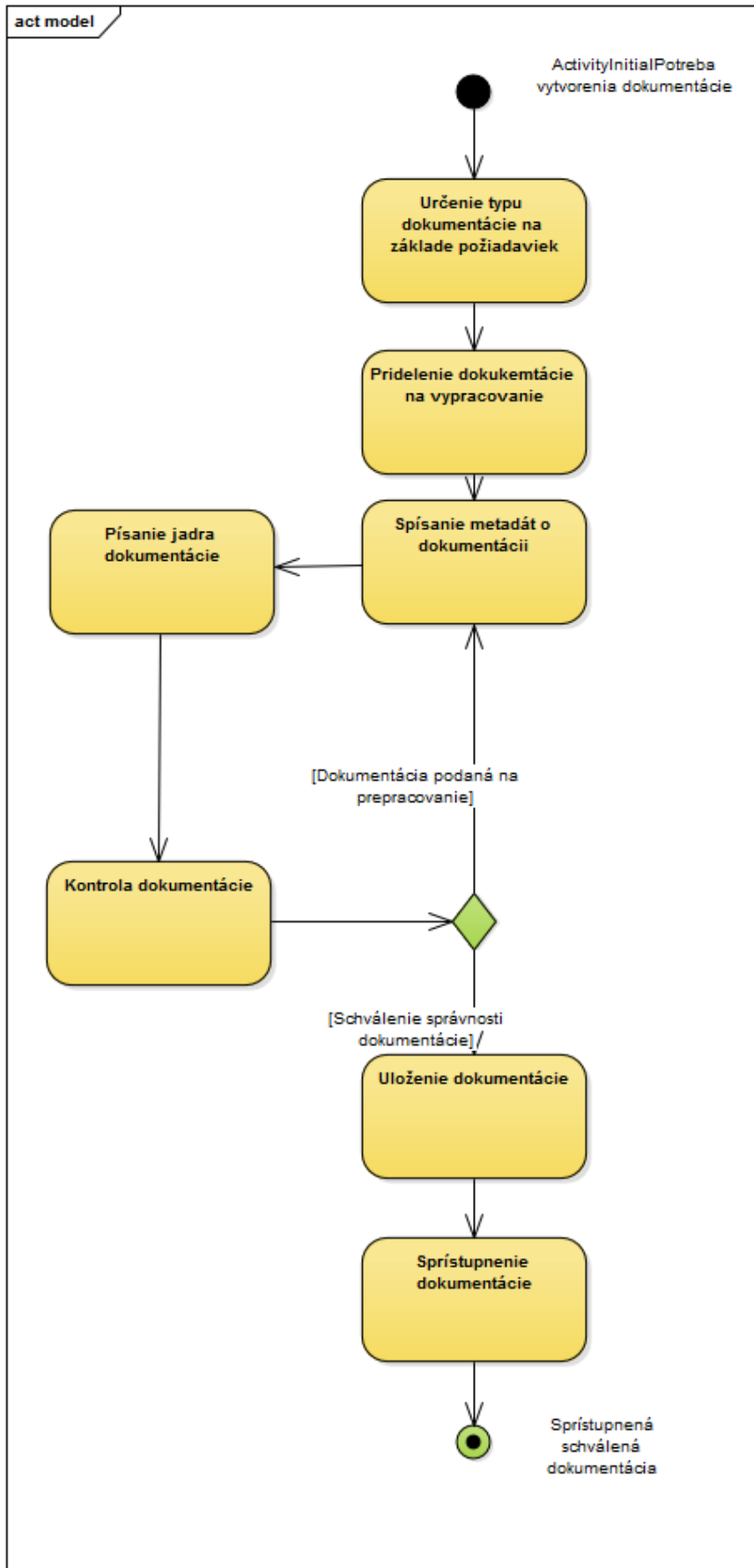
Vykonávanie:

- Sprístupnenie úložiska s dokumentáciou.
- Nájdenie najaktuálnejšej verzie dokumentácie.
- Prezeranie vybranej dokumentácie.

Ukončenie:

- Dokumentácia je sprístupnená.

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.html

1. Názvy premenných , kapitalizácia písmen

1.1 PascalCase

Príklad:

BackColor

Použitie:

Nazvy namespace, metód, suborov, publicproperties

1.2 camelCase

Príklad:

backColor

Použitie:

Nazvy private variables 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)

`HtmlHelper``htmlHelper`;

- 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

```
// Naming the following variable inputInt is misleading.
```

```
// It is a string.
```

```
var inputInt = Console.ReadLine();
```

```
Console.WriteLine(inputInt);
```

- Pred názvy Interfacov dať I

```
1. public interface IShape
```

```
2. {
```

```
3. }
```

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


```

static string GetValueFromArray(string[] array, int index)
{
    try
    {
        return array[index];
    }
    catch (System.IndexOutOfRangeException ex)
    {
        Console.WriteLine("Index is out of range: {0}", index);
        throw;
    }
}

```

• Premenné držať navrchu classy, statické vždy najvyššie.

```

1. public class Account
2. {
3.     public static string BankName;
4.     public static decimal Reserves;
5.     public Account()
6.     {
7.         // Constructor
8.         // ...
9.     }
10. }
11. }

```

3. Komentáre

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


```

// The following declaration creates a query. It does not run
// the query.

```

4. Unit testy

Po dokončení porobiť unit testy programátorom rovno aby to fungovalo. (následne kroky `sumerged` `develop` 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

Vymedzenie pojmov – skratiek

Branch – vetva

Task – úloha

User story – US

Story pointy – SP

Productowner – PO

Scrummaster – SM

Productbacklog – PB

Tím – T

Súvisiace – nadväzujúce metodiky

- Manažment verzií
- Manažment testovania
- Manažment verzií projektu
- Manažment konfigurácií systému
- Manažment ukončenia práce na projekte

Roly

- Productowner
 - 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.
- Scrummaster
 - Pomáha tímu spolupracovať.
 - Organizuje stretnutia.
 - Rieši problémy v tíme
 - Dohliada na správny formát a granularitu jednotlivých úloh vo všetkých úrovniach.
- 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.

Manažérske procesy

(V zátvorke sa vždy nachádza, kto sa na danom procese zúčastňuje)

- 1) Nastavenie systému na sledovanie projektu. (SM, T)
 - a) Vid' Manažment konfigurácií systému.
- 2) Vytvorenie epicov. (PO, SM, T)
 - a) Pretvorenie PO vízie do epicov a pridanie epicov do PB.
 - b) Určenie priority jednotlivých epicov.
 - c) Rozbitie jednotlivých epicov na menšie časti – feature.¹
 - i) Nové epicy môžu pribúdať aj počas vývoja produktu.²
- 3) Vybratie časti produktu, na ktorej sa ide pracovať. (SM, T)
 - a) Výber epicu/ epicov z PB s najväčšou prioritou.
 - b) Výber feature z daných epicov.
 - c) Rozbitie feature na US.
- 4) Definovanie US. (SM, T)
 - a) Rozbitie US na tasky.
 - i) Definovanie jednotlivých taskov.
 - ii) Ak je možné danú US nejako otestovať, tak testovanie je pridané ako samostatný task v rámci danej US.
 - b) Ohodnotenie US so SP.
 - i) V prípade, že má nejaká US príliš veľa SP (hodnota „príliš veľa“ sa môže v každom tíme líšiť, preto nie je presne špecifikovaná), tak sa daná US musí rozdeliť na viacej US.
 - c) Ohodnotenie taskov hodinami.
 - d) Pridelenie US a taskov jednotlivým členom tímu.
- 5) Práca na US. (T)
 - a) Vytvorenie novej branche.
 - i) Vid' manažment verzií.
 - b) Práca na US.
 - c) Kontrola US.
 - i) Vid' manažment testovania.
 - d) Vyhodnotenie US
 - i) Uzatvorenie US³/ prenesenie US do ďalšieho šprintu / odloženie US na neurčito.⁴
- 6) Vydanie novej verzie projektu. (T)
 - a) Vid' manažment verzií projektu.
- 7) Dodanie produktu. (PO, SM)
 - a) Vid' manažment ukončenia práce na projekte.

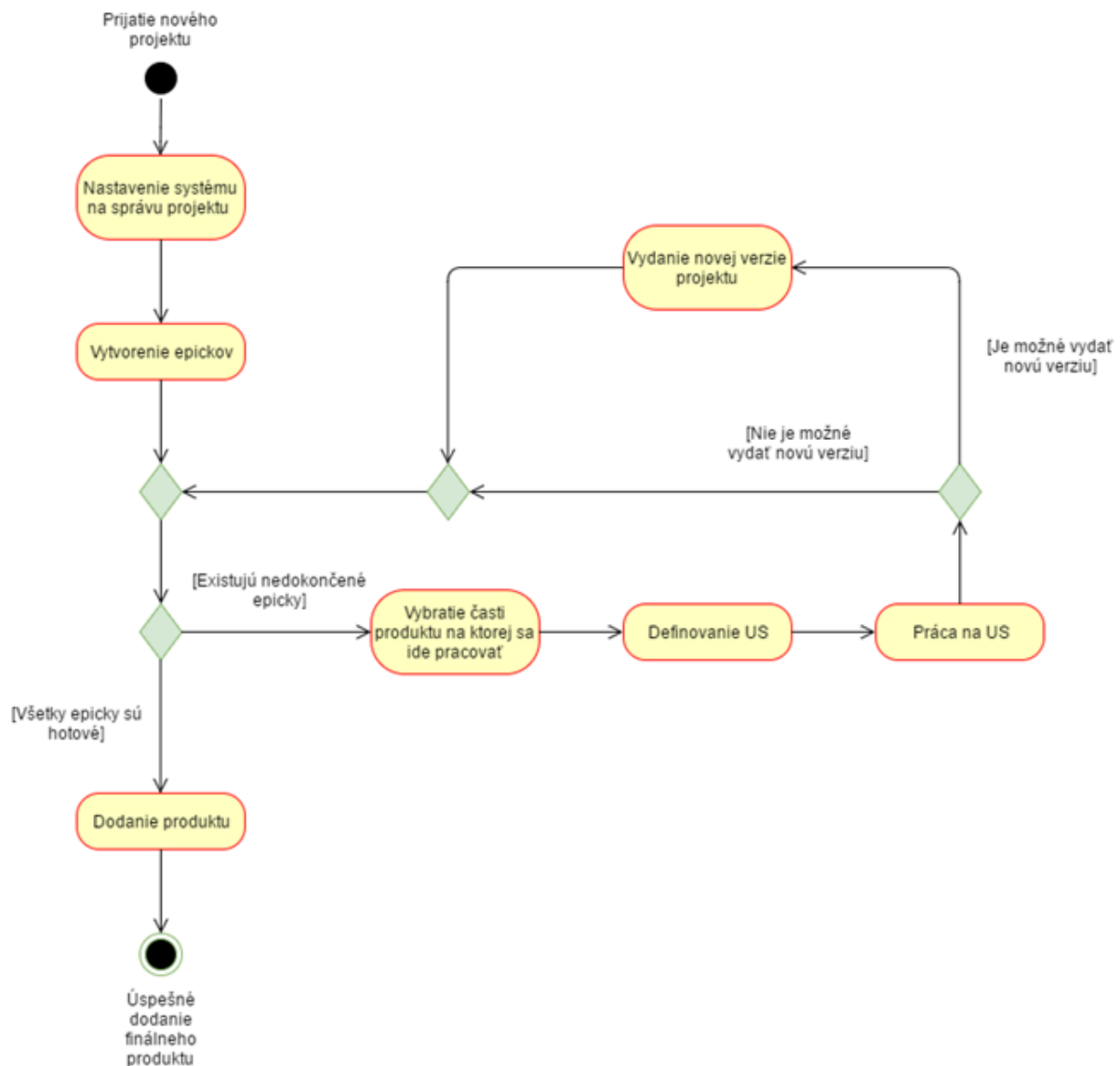
¹ Nový epic sa musí rozbiť na feature už vo fázi pridania. Rozbíjanie nemôže byť odložené na neurčito z dôvodu špecifikácie epicu.

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

³ US sa uzatvára až keď ju PO schváli.

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

Diagram nadväznosti manažérskych procesov



Obrázok 2 Diagram procesov pre zadávanie úloh

Definovanie možných stavov tasku

New – Nový task.

Active – Na tasku sa pracuje.

Closed – Task je hotový.

Definovanie možných stavov 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.

Definovanie možných stavov feature

New – Nový feature.

Active – Na feature sa pracuje.

Closed – Všetky US daného feature sú closed.

Definovanie možných stavov epicu

New – Nový epic.

Active – Na epicu sa pracuje.

Closed – Všetky feature daného epicu sú closed.

Definovanie mena epicu, feature, US, tasku

Meno musí byť samoopisné, často krát ním býva celá veta.

Napríklad:

epic - Šťahovanie dát z Git.

feature - Reprezentácia dát z Git v RDF trojiciach.

US - Určenie spôsobu sťahovania dát z Git.

task - Analyzovať navrhnuté ontológie pre Git.

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,
 - 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,
 - 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
 - Nezáväzná 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 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 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. Následne po spoločnom schválení aktuálneho stavu sa develop môže merge-núť s master vetvou, alebo v prípade nájdených problémov sa vytvorí nová vetva za účelom opravenia chýb.

Vývojový cyklus

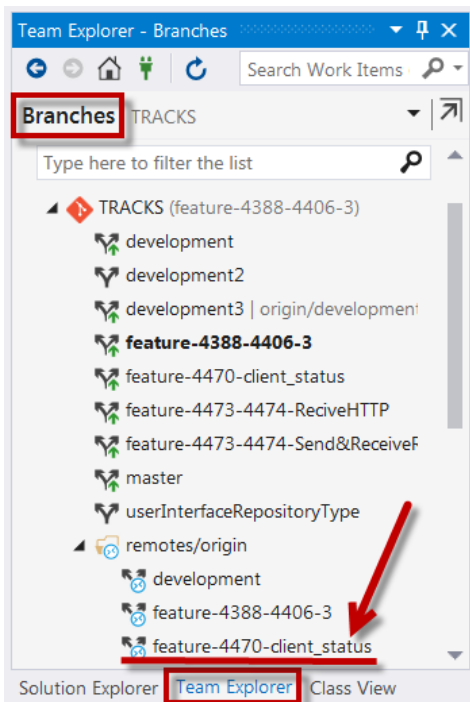
Pri začatí práce na novej funkčnosti postupujeme podľa nasledujúcich krokov:

Cez VisualStudio:

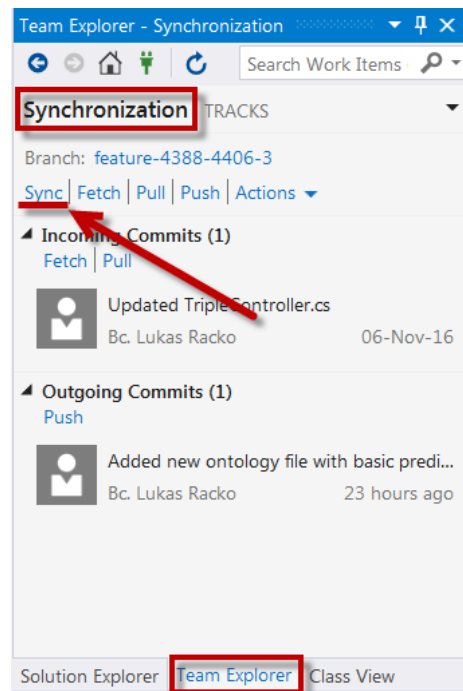
1. Dostal som zadanú novú úlohu
2. Zobrazím si zoznam vetiev v *Team explorer/branches*, zvolím si jednu, z ktorej budem vychádzať z Remotes/Origin a zvolením možnosti *Checkout/New localbranchfromsi* stiahnem vetvu ku sebe. Pomenujem ju podľa vyššie spomenutých konvencií a pri potvrdení odškrtnem možnosť *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 kódach 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 kódach 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:

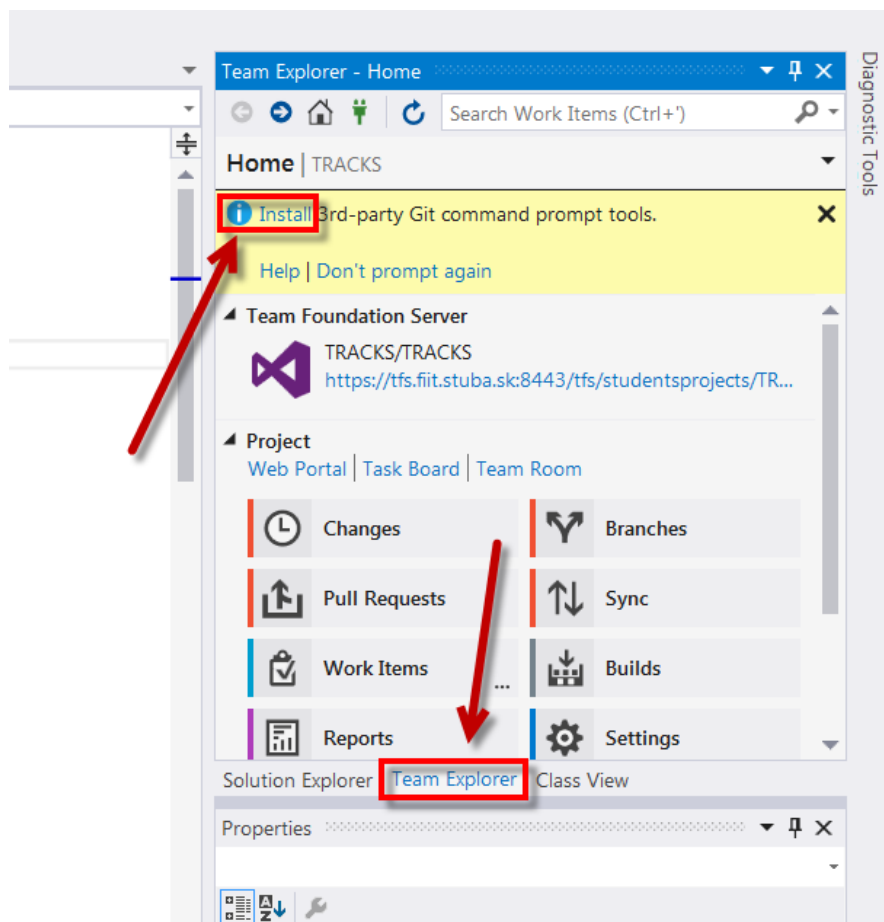
1. Dostal som zadanú už existujúcu úlohu
2. Zistím, či požadovaná vetva existuje lokálne `git branch`
3. Prepnem sa do požadovanej vetvy `git checkout nazov_vetvy`

- a. Pokiaľ vetva lokálne neexistuje, stiahnem si ju z gitu `git fetchoriginremote_nazov_vetvy: local_nazov_vetvy`
4. Pokiaľ som novú vetvu práve nestiahol z gitu, aktualizujem ju `git pulloriginnazov_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 pushoriginnazov_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 column options) alebo v detaile user story.

The screenshot shows the 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 Board' with a table of work items. A modal window for creating a 'User Story' is visible at the top, with a yellow title field and an 'Add' button. The table below has columns for 'Order', 'ID', 'Work Item Type', and 'Title'. A red arrow points to the row with ID 4323, which is a 'Task' titled 'Task 1: vytvorit klienta v WPF'. Other tasks in the list include 'Task 2: Vytvorit server', 'Task 3: Vytvorit defaultne DLL', and 'Task 4: Rozbehnutie databazy apache-jena-fuseki'. Below these are several 'User Story' items with titles like 'Pouzivatel sa chce ozvat serveru' and 'Server zistuje kolko je aktivnych volnych uzlov v sieti'.

| Order | ID | Work Item Type | Title |
|-------|------|----------------|---|
| 1 | 4317 | User Story | Pouzivatel si chce nainstalovat a spustit klientsku aplikaciju |
| | 4323 | Task | Task 1: vytvorit klienta v WPF |
| | 4324 | Task | Task 2: Vytvorit server |
| | 4325 | Task | Task 3: Vytvorit defaultne DLL |
| | 4327 | Task | Task 4: Rozbehnutie databazy apache-jena-fuseki |
| 2 | 4318 | User Story | Pouzivatel sa chce ozvat serveru |
| 3 | 4319 | User Story | Server zistuje kolko je aktivnych volnych uzlov v sieti |
| 4 | 4320 | User Story | Pouzivatel sa ponukne ze chce pomoc, a chce od balancera link na... |
| 5 | 4321 | User Story | Server odosle kniznicu na konkretny repozitar |

Testovanie

Po dokončení ľubovoľnej funkcionality je potrebné, aby bola dôsledne otestovaná niekým druhým z tímu. Keď sa dokončia všetky tasky 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>

Detailná dokumentácia ku gitu: <https://git-scm.com/doc>
<https://git-scm.com/doc>

5.6 Metodika testovania

Vytváranie unit testov, integračných testov a testovacie scenáre. Pre každú ucelenú časť kódu najlepšie metódu, ak je metóda zložitá bolo by vhodné vytvoriť viac testov, je potrebné vytvoriť samostatný unit test (triedu a metódu ktorú bude využívať). V našom prípade je pre každú samostatnú .lib súbor vytváorená trieda do ktorej sa vkladajú metódy, ktoré sa testujú.

```
// unit test code
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace Tracks.Web.TripleStore.Lib.Test.Controllers
{
    [TestClass]
    public class JobControllerTest
    {
        [TestMethod]
        public void FinalizeExistingJobExecution()
        {
        }
    }
}
```

Názvy testovacích metód by mali obsahovať názov pôvodnej metódy a názov testu, ak testujem vkladanie (insertTest....) a podobne. Pre vytváranie unit testov platia rovnaké podmienky pri vytváraní kódu, teda správne označovať premenné a komentovať kód.

Pred testami je možné stanoviť podmienky a ohraničenia, ktoré sa zohľadňujú pri testovaní. Napríklad nastavenia maximálneho času vykonávanie metódy

```
[Timeout(2000)] // Milliseconds
```

alebo v prípade očakávanej výnimky

```
[ExpectedException(typeof(ArgumentException))]
```

Všetky unit testy sa spúšťajú pri buildovaní projektu. Je možné ich pustiť aj samostatne v zložke Test->Run->All Test. Je možné spustiť aj jeden test nad konkrétnou metódou.

Pre viac informácií pri vytváraní unit testov a používaní (<https://msdn.microsoft.com/en-us/library/hh694602.aspx>)

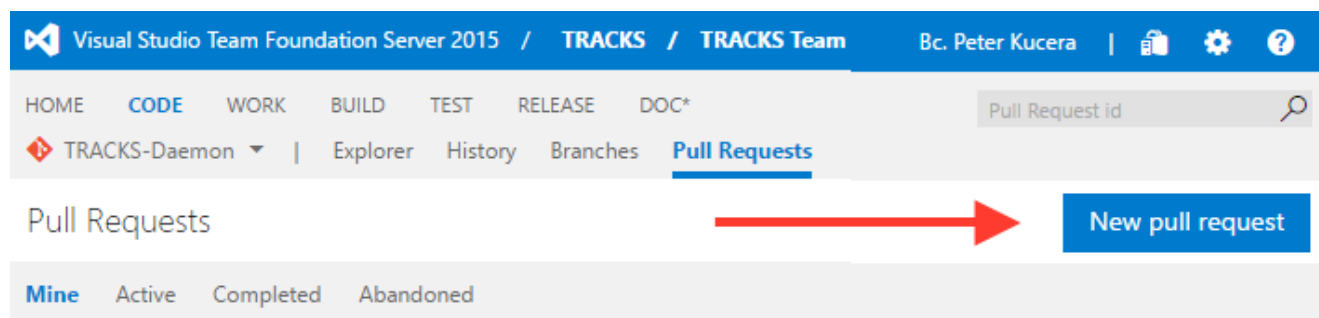
5.7 Metodika prehliadok kódu

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

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 displays a pull request review interface. The main area shows a diff for 'MainWindow.xaml.cs' between 'develop' and 'master'. The diff highlights changes in green, including the addition of logging code. On the right, the 'Active' section shows 'No merge conflicts' and 'No response from reviewers'. Below that, there are buttons for 'Complete pull request' and 'Abandon'. The 'Reviewers' section shows 'Bc. Peter Kucera' with a 'No response' status.

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

6.1 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) |
| Nepítomní | 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 | MiKr | 3.10.2016 |

| | | | | |
|----|-----------------------|--|-------------|-----------|
| | | projekte (html , JavaScript, bootstrap).Peter Lacko dodá informácie. | | |
| 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 masterbranch – 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 na zápisnice. Uložiť na GoogleDisk. | MiPo | 28.9.2016 |
| 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 navrhnúť 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 / nespĺnené, 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.
- Každú úlohu 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 sebelokálne a po release možnosť stiahnuť správny vyhládávaní online.
- Potrebný .Net Core – na stránke 2 veci vyskočia na stiahnutie – VisualStudio 2015 Update 3 a .Net Core 1.0.1.

6.2 TÍMOVÝ PROJEKT TRACKS 2

Zápisnica

| | |
|---------------------------------------|--|
| Dátum konania / Miesto konania | 3.10.2016, 3.21 |
| Zúčastnení | Ing. Martin Konôpka (MaKo) Miriam Pomffiová (MiPo) Michal Slovák (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Král (MiKr) Lukáš Račko (LuRa) |
| Nepřítomní | - |
| 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 | celý tím | 10.9.2016 |

| | | | | |
|-----|--|--|----------|-----------|
| | | s vedúcim – dohodne sa termín | | |
| 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 client-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 / funkcionálnosť 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, feature s, tasky do TFS | Prepísať údaje z písania na stretnutí do TFS | MiSl | 10.9.2016 |
| 20. | Spojzdenie klient server komunikácie | http client + nastavy, webapi | Nepridelené | N/A |
| 21. | Odosielanie dát z klienta | Odosielanie výsledkov, ktoré klient stiahne | Nepridelené | N/A |
| 22. | Vytvorenie prototypu aplikácie | Vlastný 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, WebApiClient

Knižnice pri pridávaní do projektu – NuGetpackages

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 managmentu

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

6.3 TÍMOVÝ PROJEKT TRACKS 3

Zápisnica

| | |
|---------------------------------------|--|
| Dátum konania / Miesto konania | 10.10.2016, 3.21 |
| Zúčastnení | Ing. Martin Konôpka (MaKo) Miriam Pomffýová (MiPo) Michal Slovík (MiSl) Peter Bobovský (PeBo) Peter Kučera (PeKu) Marek Mura (MaMu) Michal Král (MiKr) Lukáš Račko (LuRa) |
| Nepřítomní | - |
| 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 Král – 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.

6.4 TÍMOVÝ PROJEKT TRACKS 4

ZÁPISNICA

| | |
|---------------------------------------|---|
| DÁTUM KONANIA / MIESTO KONANIA | 17.10.2016, 3.21 |
| ZÚČASTNENÍ | ING. MARTIN KONÔPKA (MAKO) MIRIAMA POMFFYOVÁ (MIPO) MICHAL SLOVÍK (MISL) PETER BOBOVSKÝ (PEBO) PETER KUČERA (PEKU) MAREK MURA (MAMU) MICHAL 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 predošlému 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 voľne 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.

6.5 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 Král' (MiKr) Lukáš Račko (LuRa) |
| Nepřítomní | - |
| Zapísal | <i>Michal Král'</i> |
| Prílohy | - |

PROGRAM PORADY

1. Retrospektíva k predošlému šprintu
2. Diskusia
3. Vytvorenie nových úloh pre ďalší šprint
4. Závěrečné 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 buggzili 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 pridelovali 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 mílnika. Je načasovaný na 10 týždňov v semestri. Za prvý mílnik 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.

6.6 TÍMOVÝ PROJEKT TRACKS 6

ZÁPISNICA

| | |
|---------------------------------------|---|
| Dátum konania / Miesto konania | 7.11.2016 – FIIT 3.21 |
| Zúčastnení | Ing. Martin Konôpka Miriama Pomffýová Michal Slovík Peter Bobovský Peter Kučera Marek Mura Michal Král Lukáš Račko |
| Neprítomní | - |
| Zapísal | Michal Slovík |
| 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 Král). 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.

6.7 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 Král (MiKr) Lukáš Račko (LuRa) |
| Neprítomní | - |
| 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 viac.

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 spustiť uloženie dát najskôr.

Michal Král spojzadnil 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.

7 Big picture

7.1 Úvod

V tejto časti sa venujeme projektu ako cieľovému produktu našej práce. Popisujeme tu čo je hlavným cieľom projektu, návrh riešenie a následnú implementáciu.

Programátori, rovnako ako aj my, pri tvorbe softvéru pracujú s veľkým množstvom podporných systémov, ktoré im pomáhajú pri vývoji. Medzi najznámejšie patria vývojové prostredia (Eclipse, Visual Studio), 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). Ako sme už opísali v motivácii, 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 zo StackOverflow, na ktorej vývojár svoje riešenie založil.

Cieľom tohto projektu bude vytvoriť zberač dát z jednotlivých podporných systémov a následne ich spracovať tak, aby nadobúdali prínosnú hodnotu ako pre samotného programátora, tak aj pre manažéra projektu a ostatných členov tímu. Každý účastník vývoja bude mať dokonalý prehľad o tom, ku ktorému „Task“ patrí ktorý „Code Review“, „Activity“ a „Source Code“, poprípade, „Knowledge about Software“. Takýmto spôsobom sa naplní význam názvu projektu – TRACKS.

7.2 Globálne ciele projektu

Hlavným cieľom projektu je vytvorenie systému TRACKS pre zber a prepájanie dát, ktoré po sebe zanechali vývojári pri vývoji softvéru, t.j. „Tasks, Code Reviews, Activities, Source Code and Knowledge about Software“. Zber dát by mal byť vykonávaný pomocou botnet klientov na základe zadání od hlavného uzla. Nasleduje prevedenie dát do spoločnej reprezentácie, ich samotné prepájanie a sprístupnenie tretím stranám. Jedným z hlavných cieľov je navrhnúť infraštruktúru tak, aby bola ľahko nasaditeľná ako na fakulte, tak aj interne v softvérovej firme a bola využitá záverečnými projektmi na fakulte.

7.3 Celkový pohľad na systém

Kľúčovými cieľovými komponentmi systému TRACKS pre vytvorenie sémantickej vrstvy nad dátami o vývoji softvéru by mali byť:

Balancer – hlavný (master) uzol pre plánovanie zberu dát, dáva Daemonovi informácie o tom, čo sa má v akých intervaloch sťahovať a posilať od klienta.

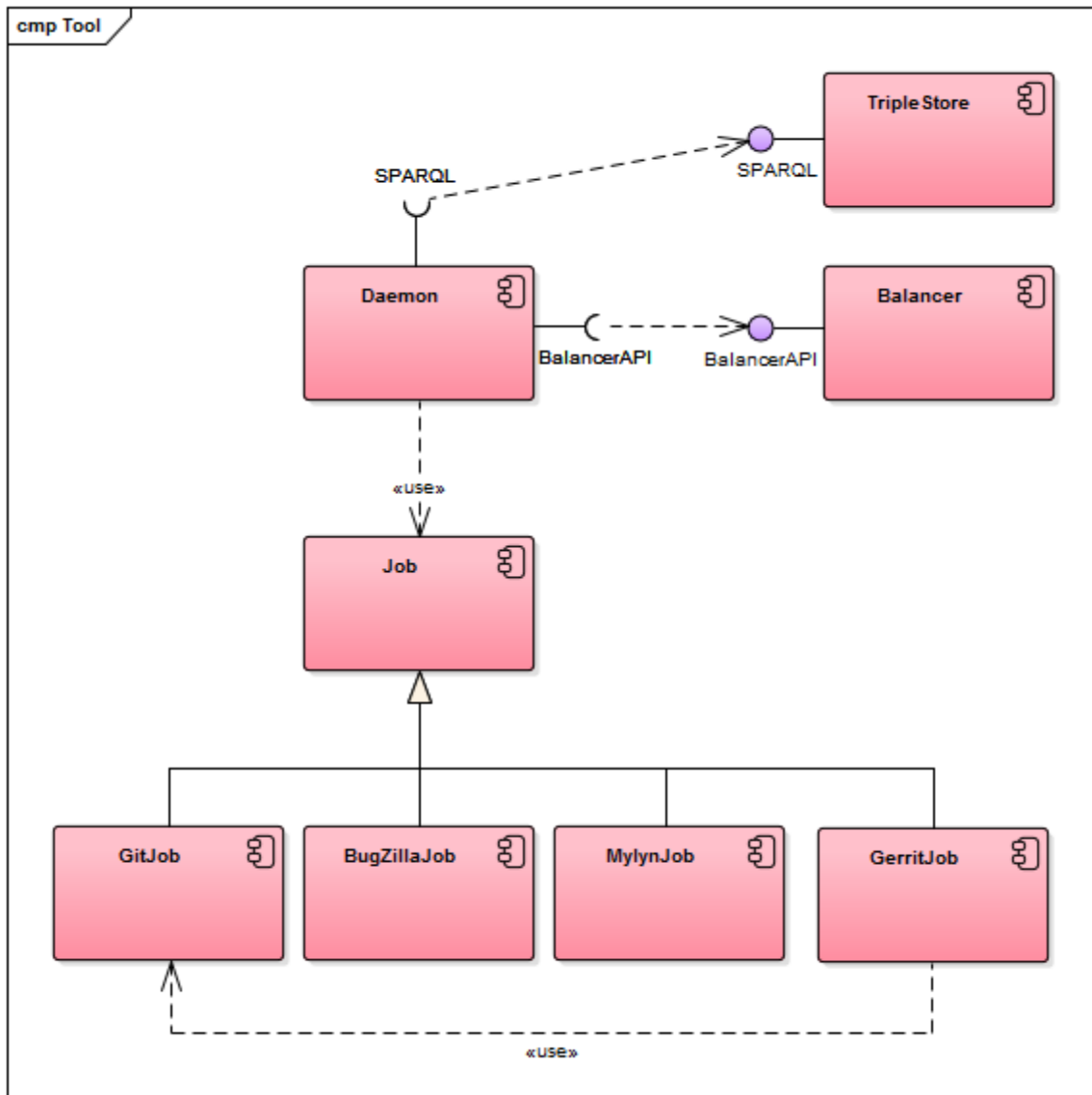
Klient API (Daemon) – distribuovaní botnet klienti pre zber dát. Po prijatí príkazu z Balancera spúšťa úlohu (job). Job bude obsahovať parametre pre konfiguračný súbor. Takto univerzálnemu klientovi nastavíme, čo chceme sťahovať. Následne prijíma a spracováva dáta, a ukladá ich na server Jena.

Webový server - poskytuje používateľovi-adminovi pridávať a upravovať jednotlivé projekty a dáta k nim.

MSSQL DB - databáza na uchovávanie dát o projektoch.

Apache Jena Fuseki – umožní pripojenie na databázu so zozbieranými dátami z repozitárov a odvodenými metadátami (TripleStore).

Nasledujúci diagram komponentov zobrazuje ich poprepájanie.



Obrázok 3 Diagram komponentov

Ukladanie dát je v trojiciach – RDF (Resource Description Framework). Využijeme tu dopytový jazyk SPARQL Query Language.

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.

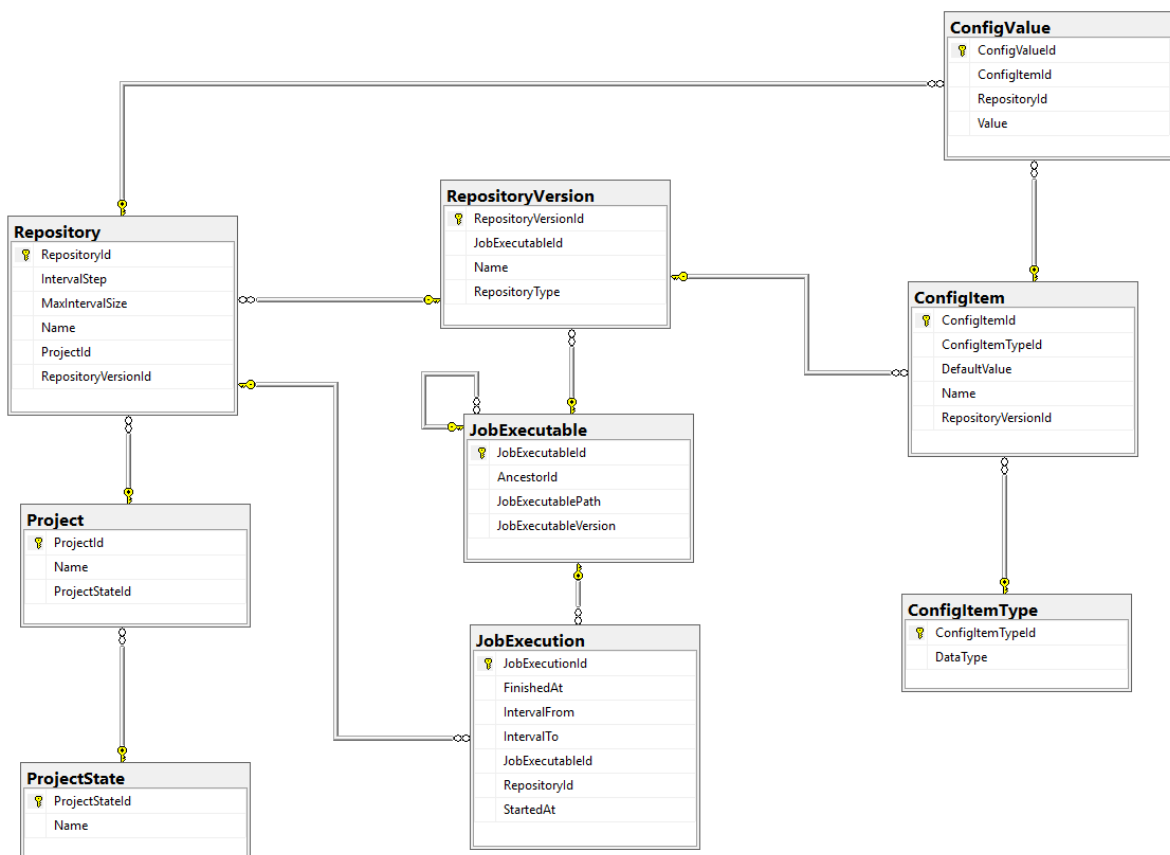
Každý systém má svoje dáta a samostatné prihlásenie používateľa pomocou prihlasovacích údajov.

Implementovať budeme projekt v prostredí Visual Studio v jazyku C#, pri čom využijeme .NETCore, MVC 6, Entity framework 7, Razor language, .NETRDF, SPARQL Query Language a RDF.

7.4 MS SQL databáza

V databáze sa ukladajú informácie o daemonoch, jednotlivých projektoch, ich repozitároch a joboch. K databáze prístupuje len webový server.

Schéma



Obrázok 4 Dátový model

Tabuľky

- ProjectState – Obsahuje možné stavy aké môže projekt nadobudnúť.
- Project – Obsahuje jednotlivé projekty vytvorené používateľom.
- Repository – Obsahuje konkrétne repozitáre určené na stiahnutie k danému projektu.
- JobExecution – Obsahuje rozparované joby pomocou stratégie. Práve tieto joby sťahujú jednotliví daemóni.
- JobExecutable – Obsahuje informácie o DLL, ktoré slúžia na ťahovanie jednotlivých typov repozitárov
- RepositoryVersion – Obsahuje verzia repozitárov.

- ConfigItemType – Obsahuje dátové typy.
- ConfigItem – Obsahuje configItemmi, ktoré je možné nastavovať pre daný repositoryVersion.
- ConfigValue – Obsahuje konkrétne hodnoty jednotlivých configItemov.

Prílohy

Project: TRACKS Server: tfs.fiit.stuba.sk\StudentsProjects Query: Blank Team - Anthriscus yellows - Backlog List type: Tree

| ID | Title 1 | Title 2 | Assigned To | Description | Story Points |
|------|------------------------|---|-----------------------|---|--------------|
| 4399 | Zápisnica | | Bc. Miriama Pomffyova | Zápisnica obsahuje zadané úlohy a ich priradenie členom tímu. Tiež zhŕňa získane informácie o projekte, hrubý náčrt fungovania cieľového produktu, rozpísanie komponentov a zoznam činností, ktoré musia členovia tímu vykonávať. | |
| 4400 | | Spísať zápisnicu | Bc. Miriama Pomffyova | Spísať zápisnicu zo stretnutia a vo formáte PDFpriložiť na stránku, dávať relatívne odkazy na zápisnice. Uložiť na GoogleDisk. | |
| 4401 | Úložisko dát | | | | |
| 4402 | | Vytvoriť úložisko | Bc. Michal Slovik | Vytvoriť úložisko na GoogleDisku pre dokumentácia zápisnice. | |
| 4392 | Rozbehanie projektu | | | Projekt má každý z tímu rozbehnutý. | |
| 4396 | | Zbuildnúť projekt, editovanie a uloženie zmien. | Bc. Michal Slovik | Zbuildnúť projekt, editovanie a uloženie,spísanie k tomu dokumentácie. Prepojenie s TFS (Team FoundationServer). Naklonovať repozitár z gitu do našich kónt. Spísať k tomumanuál. | |
| 4376 | Plagát | | | | 2 |
| 4379 | | Navrhnuť plagát | Bc. Michal Kral | Plagát bol navrhnutý a odsúhlasený celým tímom. Obsahuje názov tímu, mená, číslo tímu, názov projektua tému vystihujúcu projekt. | |
| 4412 | | Vytlačiť plagát | Bc. Michal Slovik | | |
| 4385 | TFS | | | | |
| 4387 | | Prihlásiť sa na TFS | | Všetci členovia tímu sa prihlásili na TFS a majú sa oboznámiť s týmto prostredím. | |
| 4397 | Konvencie písania kódu | | | Konvencie pre písanie kódu boli spísané a zavesené na úložisko. | |

| | | | | | |
|------|---------------|---------------------------------|--------------------|---|--|
| 4398 | | Spísať code convention | Bc. Peter Bobovsky | Spísať code convention, podľa možností priložiť tabuľku porovnania kódu Java do C#. Napr. používanie – namespace/ using, alebo opremennej var – nemusím definovať explicitne, vezme si ho sám pri vytvorení inštancie. Uvádzať aj príklady. | |
| 4389 | Metodika Gitu | | | | |
| 4391 | | Spísať metodiku používania gitu | Bc. Lukas Racko | Spísanie podrobnej metodiky gitu, akým spôsobom sa vyvíjajú nové funkcionality, vytváranie a pomenovanie nových vetiev, zlučovanie vetiev do hlavnej vetvy a ďalšie konvencie týkajúce sa verziovania | |

Project: TRACKS Server: tfs.fiit.stuba.sk\StudentsProjects Query: Blank Team - Banna List type: Tree

| ID | Title 1 | Title 2 | Assigned To | Story Points | Description |
|------|--|--|-----------------------|--------------|--|
| 4320 | Vytvorit požiadavku na balancera a link na repozitar, z ktoreho ma stahovat data | | | | |
| 4373 | Vytvorit stránku tímu | | Bc. Michal Slovik | 3 | Pripojenie sa cez MobaXterm na virtualny server, spustit instalaciu ubuntu, rozbehat web server |
| 4374 | | Vytvorit navrh stranky | Bc. Michal Kral | | Stránka bola vytvorená podľa požiadaviek tímu. |
| 4375 | | Dodať fotky | Bc. Miriama Pomffyova | | Odfotiť všetkých členov tímu, použiť filter v PRISMA a poslať Mišovi K. Odfotila som členov tímu pomocou nástroja PRISMA s použitím rovnakého efektu a poslala som ich Mišovi K. |
| 4413 | | Uploadnúť stránku na FIIT server | Bc. Michal Slovik | | Spojzdnit stranku na serveri. Stránka spojzdnená team08-16.studenti.fiit.stuba.sk |
| 4471 | Oboznámenie sa s Jenou | | Bc. Lukas Racko | | |
| 4327 | | Rozbehať lokálne databázu apache-jena-fuseki | Bc. Lukas Racko | | Lokálne rozbehnutie apache-jena-fuseki databazy do ktorych by klient vedel vkladat vysledky v RDF trojiciach. https://jena.apache.org/download/#jena-fuseki |

Project: TRACKS **Server:** tfs.fiit.stuba.sk\StudentsProjects **Query:** Blank Team - Chlamydia **List type:** Tree

| ID | Title 1 | Title 2 | Assigned To | Description | Story Points |
|------|---------------------------|---|-----------------|--|--------------|
| 4421 | | Spísanie zápisnice | Bc. Lukas Racko | Spísanie zápisnice pre ďalší šprint | |
| 4523 | Uloženie DLL na webe | | Bc. Michal Kral | Ako používateľ chcem uploadnúť nové DLL na web aby som mohol sťahovať repozitáre. | 13 |
| 4386 | | Vytvoriť webové úložisko pre DLL knižnice | Bc. Michal Kral | Webové úložisko sa bude nachádzať v priečinku projektu. | |
| 4526 | | Analyzovať ako funguje file upload v ASP.NET Core MVC | Bc. Michal Kral | File-upload funguje identicky ako pri iných frameworkoch. | |
| 4525 | | Implementovať controller pre nahranie DLL cez upload | Bc. Michal Kral | Kontroler sa nachádza v balíku tracks.web. Nahraté DLL sa automaticky triedia do jednotlivých skupín podľa typov repozitárov. | |
| 4524 | | Implementovať rozhranie pre vkladanie DLL | Bc. Michal Kral | V menu webového rozhrania pribudli nové možnosti pre vkladanie a úpravu nahratých DLL. | |
| 4393 | Rozbehať MSSQL pre server | | Bc. Michal Kral | Zistiť ako funguje MSSQL server. Čo všetko treba spraviť aby sme ho mohli ladne zakomponovať do projektu. Samotné rozbehnutie a spustenie MSSQL DB v projekte. | 13 |
| 4395 | | Nainštalovať MSSQL | Bc. Michal Kral | Nainštalovanie prebehlo bez problémov. Inštalovala sa verzia SQLEXPRESS 2016. | |
| 4426 | | Otestovať vytvorenie databázy na serveri | Bc. Michal Kral | Databáza sa vytvorí pri prvom prístupe - otestovať asi cez existujúce API. | |
| 4427 | | Spísať manuál na rozbehnutie DB pre iných členov | Bc. Michal Kral | Pre úplné rozbehnutie DB na jednotlivých strojoch je nutné postupovať podľa manuálu ktorý sa nachádza na cloude tímu. | |

| | | | | | |
|------|---|---|------------------|--|----|
| 4429 | | Nevytvorenie databázy v MSSQL Express | Bc. Michal Kral | <p>Bolo potrebné:</p> <ul style="list-style-type: none"> -správne upraviť connection string. -správne nainštalovať DB -nakonfigurovať SQL server -zmeniť typ overenia používateľa pri vstupe na SQL server | |
| 4515 | | Analyzovať entity framework | Bc. Michal Kral | <p>entity framework 7, ktorý je aktuálne použitý v projekte má dostačujúcu funkcionality aká projekt vyžaduje. Ale dokumentácie EF 7 niesú úplné.</p> <p>Funguje obdobne ako v EntityManager v JEE.</p> | |
| 4388 | Ukladanie grafov RDF trojíc do TripleStore | | Bc. Lukas Racko | | 13 |
| 4472 | | Analyzovať existujúcu implementáciu použitia TripleStore v projekte | Bc. Lukas Racko | <p>Analyzoval som existujúcu implementáciu použitia TripleStore v projekte, Otestoval som vkladanie trojíc a grafu do RDF databázy Jena Overil som funkčnosť existujúcej funkcionality k Jobom a Repozitárom v TripleStore</p> | |
| 4380 | Repozitár pre Daemon klienta | | Bc. Peter Kucera | | 2 |
| 4381 | | Vytvorenie Git repozitára pre klientsku aplikáciu | | Repozitár vytvorený, 2 branche pridané - master a develop | |
| 4403 | | Pridať základný projekt pre klientsku aplikáciu | | Prvá verzia klientskej aplikácie pridaná do GITu | |
| 4317 | Základná verzia Daemon klientskej aplikácie | | Bc. Peter Kucera | Používateľ si chce nainštalovať a spustiť klientsku aplikáciu | 5 |

| | | | | | |
|------|--|---|-----------------------|--|---|
| 4323 | | Založiť projekt pre klienta vo WPF | Bc. Peter Kucera | Založil som nový WPF projekt vo Visual Studiu , s ukázkami komunikácie UI a kódu | |
| 4506 | Skrytie aplikácie v Trayi a zobrazenie Popupu v pravom dolnom rohu | | Bc. Peter Kucera | Ako používateľ chcem, aby sa aplikácia skrývala v trayim, hneď aj po spustení a po rozkliknutí jej ikony sa zobrazil popup (namiesto samostatného okna) | 8 |
| 4509 | | Umožniť zatvorenie Daemona cez kontextové menu nad tray ikonou. | | Podľa dokumentácie som vytvoril základnú implementáciu ikony v notifikačnom paneli a impementoval som skrytie aplikácie v notifikacnom paneli | |
| 4377 | Konvencie pre uzatvaranie úloh | | Bc. Miriama Pomffyova | Sú spísané nie len konvencie pre uzatváranie úloh ale aj prvotná verzia metodika práce na projekte. | 3 |
| 4378 | | Spísať konvencie pre uzatváranie taskov | Bc. Miriama Pomffyova | Sú spísané konvencie pre uzatváranie taskov. Nastavovanie stavov taskov a user stories, postup pri vytváraní branch, code review a oznacenie tasku za splnený. | |
| 4408 | | Spísanie konvencií pre mergovanie nových časti | Bc. Miriama Pomffyova | Na základe metodiky práce s gitom som spísala spôsob mergovania nových častí kódu. | |
| 4318 | Používateľ sa chce ozvať serveru | | Bc. Peter Kucera | Ako používateľ chcem aby moja aplikácia komunikovala so serverom cez REST API | 5 |
| 4423 | | Umožniť komunikáciu deamona cez REST API | Bc. Peter Kucera | Do kódu klienta som pridal funkčný príklad komunikácie so serverom cez REST API | |
| 4373 | Vytvorit stránku tímu | | Bc. Michal Slovik | Doladenie textov a dokumentov na stránke, ich uploadovanie na git a prepojenie s nim | 3 |
| 4416 | | Dodať texty | Bc. Miriama Pomffyova | Dodať všetky texty. | |

| | | | | | |
|------|--|---------------------------|-------------------|---|--|
| 4422 | | Vytvoriť repozitar na web | Bc. Michal Slovik | Vytvorenie repozitaru pre web. A nahratie aktualneho webu na git. | |
| 4424 | | Updatnut zmeny | Bc. Michal Slovik | Zmenit farby, texty, fotky | |

Project: TRACKS Server: tfs.fiit.stuba.sk\StudentsProjects Query: Blank Team - Diarrhoea List type: Tree

| ID | Title 1 | Title 2 | State | Assigned To | Story Points | Description |
|------|--|--|----------|-----------------------|--------------|---|
| 4721 | Opravenie onDelete funkcie pri vsetkych tabulkach v DB | | Closed | Bc. Michal Kral | 13 | V databaze nefunguje onDelete Restrict/Update na spravnych tabulkach. |
| 4722 | | Nastavit onDelete operacie pri jednotlivich tabulkach v DB | Closed | Bc. Michal Kral | | Zmenené pri väčšine tabulkách. |
| 4635 | Template na dokumentáciu | | Resolved | Bc. Michal Slovik | | Vytvorenie templatu na vseobecne dokumentacie |
| 4636 | | Vytvorit template na technicke dokumentacie | Closed | Bc. Michal Slovik | | Nemysli sa zapisnica, ale vseobecne dokumentacie. |
| 4409 | TP Cup | | Resolved | | 3 | V dokumente sa nachadzaju zle vysvetlene veci. |
| 4411 | | Spísať prihlášku na TP Cup | Closed | Bc. Miriama Pomffyova | | Prihláška bola spísaná a predložená ostatným členom tímu na odkontrolovanie a schválenie. |
| 4561 | Zadanie podporovaných repozitárov | | Resolved | Bc. Michal Kral | 13 | Ako používateľ chcem vytvoriť podporovaný typ repozitáru aby som následne mohol vytvárať projekty z týchto typov. |

| | | | | | | |
|------|--|---|----------|-----------------|----|---|
| 4326 | | Navrhnuť rozhranie pre zadanie typov repozitárov a určenie JobExecutable pre ne | Closed | Bc. Michal Kral | | V menu webového rozhrania pribudla možnosť na pridávaní/ upravovanie typov repozitárov. |
| 4562 | | Implementovať controller | Closed | Bc. Michal Kral | | Kontroler sa nachádza v balíku tracks.web.controller |
| 4510 | Zadanie projektu na sťahovanie | | Resolved | Bc. Michal Kral | 13 | Ako používateľ chcem zadať do systému, že chcem stiahnuť Eclipse Git, Eclipse Bugzillu atd. aby som ich mohol neskôr analyzovať. |
| 4511 | | Implementovať controller pre zadanie repozitárov | Closed | Bc. Michal Kral | | Kontroller sa nachádza v balíku tracks.web.controller. |
| 4512 | | Implementovať view pre zadanie repozitárov | Closed | Bc. Michal Kral | | V menu webového rozhrania pribudla možnosť na pridávanie/ upravovanie repozitárov. |
| 4513 | | Implementovať rozhranie pre aktuálny zoznam repozitárov | Closed | Bc. Michal Kral | | V menu webového rozhrania pribudla možnosť na pridávanie/ upravovanie repozitárov. |
| 4388 | Ukladanie grafov RDF trojíc do TripleStore | | Resolved | Bc. Lukas Racko | 13 | |
| 4475 | | Navrhnuť API pre zápis RDF trojíc do TripleStore | Closed | Bc. Lukas Racko | | Navrhol som API pre zápis RDF trojíc do databázy Jena. Cez funkciu insertGraph budú klienti posilať na server grafy obsahujúce trojice, ktoré sa následne uložia do Jeny. |

| | | | | | | |
|------|------------------------------|--|----------|--------------------|---|--|
| 4406 | | Implementácia webovej služby pre zápis do TripleStore | Closed | Bc. Lukas Racko | | <p>Vytvoril som novú triedu TripleData na reprezentáciu trojíc z RDF databázy.</p> <ul style="list-style-type: none"> - Implementoval som funkciu insertGraph pre zápis RDF trojíc do databázy Jena. Klienti môžu poselať na server grafy obsahujúce trojice, ktoré sa následne uložia do Jeny. - Vytvoril som pomocné funkcie pre prácu s TripleURI, podobne ako to majú aj JobUri a RepositoryUri. |
| 4476 | | Zdokumentovať API pre zápis do TripleStore | Closed | | | Zdokumentoval som základné použitie implementovanej funkcie insertGraph na ukladanie trojíc do grafu. |
| 4384 | Správa DLL pre JobExecutable | | Resolved | Bc. Peter Bobovsky | 5 | |
| 4325 | | Vytvoriť základnú implementáciu DLL knižnice pre JobExecutable | Closed | Bc. Peter Bobovsky | | Vytvorit defaultnu kniznicu DLL z ktorej sa budú vyrábať jednotlivé DLL na sťahovanie. |
| 4482 | | Definovať štruktúru knižnice pre JobExecutable | Closed | Bc. Peter Bobovsky | | Štruktúra knižnice bola zadaná v dokumente a popísaná aká a kde funkcionality sa očakávajú. Sú možné neskoršie zmeny podľa potreby |
| 4527 | | Implementovať fake JobExecution pre budúce testovanie | Closed | Bc. Peter Bobovsky | | Fake implementácia Executable, ktorú môžeme použiť pre testovanie klienta aj servera. Je implementáciou defaultnej DLL knižnice |

| | | | | | | |
|------|---|---|----------|------------------|---|--|
| 4382 | Určenie spôsobu sťahovania dát z BugZilly | | Resolved | Bc. Marek Mura | 5 | |
| 4496 | | Analyzovať existujúce API pre sťahovanie z repozitárov Bugzilla Eclipse | Closed | Bc. Marek Mura | | Pre sťahovanie dát z Bugzilly existuje Bugzilla REST API. Jeho použitie v projekte by bolo obtiažne. Vhodnejšie by bolo nájsť knižnicu použiteľnú v .NET. |
| 4516 | | Analyzovať knižnicu pre sťahovanie BugZilla | Closed | Bc. Marek Mura | | Existuje nejaká knižnica pre sťahovanie BugZilla pre .NET? Alebo to budeme crawlovať alebo parsovať? - Pre Bugzillu existuje knižnica Bugzilla.NET, ktorá umožňuje sťahovanie dát o bugoch priamo z Bugzilly. |
| 4620 | | Výber dát Bugzilly na stiahnutie | Closed | Bc. Marek Mura | | Po preskúmaní možností knižnice na sťahovanie dát z Bugzilly boli vybrané dáta, ktoré budú o každom bugu získavané. Spísané v dokumentácii. |
| 4506 | Skrytie aplikácie v Trayi a zobrazenie Popup v pravom dolnom rohu | | Resolved | Bc. Peter Kucera | 8 | Ako používateľ chcem, aby sa aplikácia skrývala v trayim, hneď aj po spustení a po rozkliknutí jej ikony sa zobrazil popup (namiesto samostatného okna) |

| | | | | | | |
|------|-------------------|----------------------------------|----------|-----------------------|---|--|
| 4507 | | Pridať ikonu pre tray do WPF App | Closed | Bc. Peter Kucera | | Použil som Hardcodet NotifyIcon knižnicu na zobrazenie okna aplikácie |
| 4508 | | Navrhnuť rozhranie pre popup | Active | Bc. Michal Slovik | | Navrhnut loading jobov, infrastrukturu ako zobrazit pouzivatelovi priebeh stahovania. Restart jobu, nazov appky. Zobrazovanie zoznamu sťahovaných jobov. |
| 4631 | Logovanie Daemona | | Resolved | Bc. Miriama Pomffyova | 5 | Ako admin projektu chcem aby klienti logovali svoju cinnost do lokalnych suborov. |
| 4633 | | Pridat kniznicu pre logovanie | Closed | | | Asi log4net alebo nieco ine |

| Project: TRACKS Server: tfs.fiit.stuba.sk\StudentsProjects Query: Blank Team - Ebola List type: Tree | | | | | | |
|--|--|--|----------|-----------------|--------------|---|
| ID | Title 1 | Title 2 | State | Assigned To | Story Points | Description |
| 4409 | TP Cup | | Resolved | | 3 | V dokumente sa nachádzajú zle vysvetlene veci. |
| 4628 | | Skontrolovať prihlášku | Closed | | | Do 25.10.2016_ 23:59 |
| 4561 | Zadanie podporovaných repozitárov | | Resolved | Bc. Michal Kral | 13 | Ako používateľ chcem vytvoriť podporovaný typ repozitáru aby som následne mohol vytvárať projekty z týchto typov. |
| 4737 | | Pridať možnosť pridávania configitemov k typu repozitáru | Closed | Bc. Michal Kral | | DB bola prerobená tak aby pridávanie config itemov k jednotlivým typom repozitárov bolo normalizované. Každý typ repozitáru teraz obsahuje config itemy. |
| 4510 | Zadanie projektu na sťahovanie | | Resolved | Bc. Michal Kral | 13 | Ako používateľ chcem zadať do systému, že chcem stiahnuť Eclipse Git, Eclipse Bugzillu atd. aby som ich mohol neskôr analyzovať. |
| 4738 | | Použiť configValue table v súvislosti s vytváraním nového repozitáru | Closed | Bc. Michal Kral | | DB od teraz využíva configValue table pre ukladanie konkrétnych hodnôt config itemov. |
| 4723 | Uprava vytvoreného repozitáru | | Resolved | Bc. Michal Kral | 13 | Ako používateľ chcem upraviť už vytvorený repozitár v prípade, že som sa pomyšľil. |
| 4724 | | Vytvoriť view pre update uz vytvoreného repozitáru | Closed | Bc. Michal Kral | | Po kliknutí na change btn sa otvorí screen na úpravu repozitárov daného projektu. Autogenerované formuláre: meno projektu (nedá sa zmeniť), repoType, meno, heslo, url, intervalStep, maxIntervalSize, update btn, delete btn. View je vytvorené podľa požiadaviek US. Formuláre sú autogenerované. |
| 4725 | | Vytvoriť controller pre update uz vytvoreného repozitáru | Closed | Bc. Michal Kral | | Controller sa nachádza v balíku tracks.web.controller |
| 4388 | Ukladanie grafov RDF trojíc do TripleStore | | Resolved | Bc. Lukas Racko | 13 | |
| 4629 | | Kontrola vstupných trojíc podľa OWL ontologii | Closed | Bc. Lukas Racko | | Implementoval som validáciu vstupných trojíc podľa poskytnutých ontológií. - Prichádzajúci graf od klientov sa kontroluje po trojiciach, a odfiltrujú sa z neho všetky predikáty, ktoré nie sú obsiahnuté v dostupných OWL súboroch(ontológiach). - Ontológie sa načítavajú dynamicky zo zadaného priečinka v rámci projektu, je možné pridať nové ontológie bez nutnosti meniť kód, stačí nakopírovať nové OWL súbory do daného priečinka. |
| 4470 | Oznamovanie dostupnosti klientov na server | | Active | Bc. Lukas Racko | 3 | |
| 4624 | | Navrhnuť stránku s aktuálnym stavom klientov | Closed | Bc. Lukas Racko | | Navrhol som stránku s prehľadom klientov z SQL databázy. Zobrazuje sa tabuľka so všetkými základnými informáciami o klientoch - ID, posledná IP, dátum poslednej aktivity, meno a stav. |
| 4528 | | Rozšíriť DB model o klientov | Closed | | | Vytvorená nová databázová tabuľka slúžiaca na uchovanie informácií o klientoch. |
| 4324 | | Implementovať službu pre ohlásenie klienta na server | Closed | Bc. Lukas Racko | | Implementovaná služba updateClient na vytvorenie záznamu nového klienta, alebo aktualizovanie starého. Pomocou tejto funkcie sa môže klient ohlasovať serveru, vďaka čomu server vie, že danému klientovi môže priradiť úlohu. |

| | | | | | | |
|------|---|---|----------|--------------------|---|---|
| 4477 | | Navrhnuť API pre oznamovanie stavu klientov na server | Closed | Bc. Lukas Racko | | Navrhnutý spôsob ohlasovania stavu klientov na server prostredníctvom funkcie updateClient(), pomocou ktorej server získa základné informácie o klientovi - aktuálnu IP adresu, meno a aj čas poslednej aktivity. |
| 4478 | | Zdokumentovať API pre oznamovanie stavu klientov na server | New | | | - |
| 4384 | Správa DLL pre JobExecutable | | Resolved | Bc. Peter Bobovsky | 5 | |
| 4731 | | Upraviť DLL na interface | Closed | Bc. Peter Bobovsky | | Zo základnej implementácie vyrobiť vzorový a referencovateľný interface |
| 4739 | | Rozšíriť Fake job executable aby sa dalo testovať | Closed | Bc. Peter Bobovsky | | Potreba aby sa fake job executable dalo spúšťať a vracalo veci. Toto Fake executable bude slúžiť na prvé testovanie a low level implementáciu. Fake Job executable bolo vyrobené vo vlastnom repozitári pre DLL. |
| 4744 | Zistenie aktuálneho stavu executable | | Active | Bc. Peter Bobovsky | 8 | |
| 4745 | | Vytvorenie komunikácie s DLL | Closed | | | Medziprocesorova komunikácia cez eventy. Zabezpečiť spustanie samotných DLL z klienta. Vznikla potreba pre špeciálny subrepozitár pre interface a eventové triedy. |
| 4382 | Určenie spôsobu sťahovania dát z BugZilly | | Resolved | Bc. Marek Mura | 5 | |
| 4734 | | Spísať do dokumentácie | Closed | Bc. Marek Mura | | Možnosti a funkcie knižnice Bugzilla.NET boli spísané do dokumentácie. Dokumentácia obsahuje stručný popis knižnice, návod na základnú obsluhu a sťahovanie dát o bugoch a popis dôležitých dátových typov. |
| 4319 | Vytvorenie jobov pre stiahnutie BugZilly | | Active | Bc. Marek Mura | 8 | Ako výskumník chcem, aby balancer rozdelil sťahovanie repozitára do viacerých JobExecution použitím konkrétnej stratégie pre daný repozitár. |
| 4514 | | Analyzovať vytváranie stratégií pre sťahovanie Jobov | Active | Bc. Marek Mura | | |
| 4506 | Skrytie aplikácie v Trayi a zobrazenie Popup v pravom dolnom rohu | | Resolved | Bc. Peter Kucera | 8 | Ako používateľ chcem, aby sa aplikácia skrývala v trayim, hneď aj po spustení a po rozkliknutí jej ikony sa zobrazil popup (namiesto samostatného okna) |
| 4621 | | Implementovať popup ako WPF okno | Closed | Bc. Peter Kucera | | Zrušil som implementáciu okna klienta cez WinForms a nahradil som ju WPF oknom (XAML). |
| 4318 | Používateľ sa chce ozvať serveru | Komunikácia so serverom | Active | Bc. Peter Kucera | 5 | Výmena dát medzi klientom a serverom cez REST API |
| 4622 | | Implementovať zadanie mena používateľa v nastaveniach aplikácie | Closed | | | Implementoval som zadanie mena používateľa v aplikácii, ktoré sa bude ďalej odosielať na server |
| 4487 | Zobrazenie vykonávania jobov Daemonom | | Resolved | Bc. Peter Kucera | 5 | Ako používateľ chcem mať prehľad o práve sťahovaných údajoch v podobe ich zoznamu |
| 4488 | | Vytvoriť zoznam JobExecution v rozhraní Daemona | Closed | | | Navrhol a naimplementoval som zobrazenie jednotlivých JobExecutions v aplikácii pomocou listu itemov ListView |
| 4473 | Odosielanie RDF trojíc z klienta do TripleStore | | Resolved | Bc. Michal Slovik | 8 | |

| | | | | | | |
|------|--|---|----------|-----------------------|---|---|
| 4474 | | Implementovať klienta služby pre odoslanie grafov RDF trojíc do TripleStore | Closed | Bc. Michal Slovik | | <p>Klientska trieda umožňuje odosielať dáta do TripleStore, implementuje nejaké rozhranie, napr.: <code>ITripleStoreServiceClient</code>, s metódami pre odoslanie grafu RDF trojíc, napr.: <code>SendTriplesGraphAsync(triples : IEnumerable<Triple>) : Task<bool></code></p> <p>Inštancia tejto triedy sa potom vkladá (injectuje) do DLL pre sťahovanie repozitárov.</p> |
| 4529 | | Implementovať fake klienta pre testovanie | Closed | Bc. Michal Slovik | | Fake klient nebude dopytovať server, bude priamo vracaať výsledky. Túto fake implementáciu môžeme použiť pri testovaní Daemona. |
| 4491 | Dokončenie sťahovania job | | Closed | Bc. Michal Slovik | 5 | Ako používateľ chcem aby ma aplikácia pri pokuse ju zavrieť vyzvala, či to naozaj chcem, ak sťahovanie jobov je blízko konca aby som zbytočne neprerušil dokončenie jobu, ktorý je už na konci. |
| 4492 | | Implementovať kontrolu pri ukončení aplikácie na bežiacie joby | Active | Bc. Michal Slovik | | |
| 4742 | | Upozornenie na nedokončené sťahovanie repozitára | Active | Bc. Michal Slovik | | |
| 4743 | | Otestovať upozornenia sťahovania | Active | Bc. Michal Slovik | | Skusiť vypnúť počas sťahovania. |
| 4735 | Aktualizácia stránky | | Resolved | Bc. Michal Slovik | 2 | Aktualizácia stránky a jej obsahu |
| 4736 | | Aktualizovať stránku | Closed | Bc. Michal Slovik | | |
| 4631 | Logovanie Daemona | | Resolved | Bc. Miriama Pomffyova | 5 | Ako admin projektu chcem aby klienti logovali svoju činnosť do lokálnych súborov. |
| 4634 | | Pridať základné logy | Closed | Bc. Miriama Pomffyova | | <p>Kedy sa spustila aplikácia</p> <p>Kedy sa zavrela aplikácia</p> <p>Kedy sa skryla</p> <p>Volania na server</p> <p>Spustanie jobov</p> |
| 4746 | | Rozdelenie logov podľa tried | Closed | Bc. Miriama Pomffyova | | Loggovanie sme skonzultovali. Pre doterajší stav kódu je postačujúce. V loggu je uvedený z ktorej triedy aplikácia zaloggovala. |
| 4747 | | Ukladanie logov | Closed | Bc. Miriama Pomffyova | | <p>Logy sa ukladajú do priečinka</p> <p>"myPathToProject\Source\Repos\TRACKS-Daemon\TracksDaemon\TracksDaemon\bin\Debug\Log\Mylog_2013_04_27.log".</p> <p>Logy sú ukladané vo formáte XML, umožňuje to ďalšie spracovanie logu.</p> |
| 4495 | Určenie spôsobu sťahovania dát z Git | | Active | Bc. Miriama Pomffyova | 5 | |
| 4383 | | Analýza existujúcich API pre sťahovanie z repozitárov Git Eclipse | Active | Bc. Miriama Pomffyova | | |
| 4394 | | Analýza navrhnutých ontológií pre Git a Gerrit | Active | Bc. Miriama Pomffyova | | Karol navrhol ontológiu pre Git a Gerrit v OWL, pridať aj e-mailom, link aj na discord. |
| 4521 | | Analýza knižnice pre sťahovanie Git | Active | Bc. Miriama Pomffyova | | Môžeme použiť NGit alebo aj niečo iné, ak pre .NET sú. |
| 4638 | Vytvorenie konfiguračného súboru pre nastavenie DB | | Resolved | Bc. Miriama Pomffyova | 5 | Návod na vytvorenie konfiguračného súboru user.json je uverejnený na google disku tímu a súbor gitignore je upravený. |
| 4639 | | Čítanie konfiguračného súboru pri spustení | Active | Bc. Miriama Pomffyova | | <p>Vytvorenie konfiguračného súboru ktorý bude v tvare:</p> <p>name = value.</p> <p>Hodnota ktorá sa v nom bude nachádzať je DB connection string.</p> <p>Cieľom tohto konfiguračného súboru aby sa neprepisoval DB connection string.</p> <p>Cieľom tohto konfiguračného súboru sa nebude nikdy commitovať.</p> |

| | | | | | | |
|------|--|--|-----|-----------------------|--|---|
| 4640 | | Upravenie .gitignore pre ignorovanie súboru s DB connection string | New | Bc. Miriama Pomffyova | | Konfiguračný súbor s connection string by nemal byť v Gite, preto by ho aj Git mal ignorovať pri sledovaní zmien. Súbor je špecifický pre každého člena. |
| 4641 | | Dokumentovanie postupu rozbehania projektu | New | Bc. Miriama Pomffyova | | Keďže config súbor s DB connection string je ignorovaný Gitom, tak pri Git Clone celého projektu bude chýbať. Preto je potrebné zachovať niekde túto znalosť - čo má vlastne vývojár spraviť, aby mu to fungovalo - v dokumentácii. |

| Project: TRACKS Server: tfs.fiit.stuba.sk\StudentsProjects Query: Blank Team - Fiji List type: Tree | | | | | | |
|---|---|---|----------|--------------------|--------------|--|
| ID | Title 1 | Title 2 | State | Assigned To | Story Points | Description |
| 4879 | Zmenenie aktuálneho managovania typov repozitárov | | Resolved | Bc. Michal Kral | 5 | Ako používateľ chcem naprv vytvoriť typ repozitáru a až potom mu priradiť DLL - nie naopak ako to funguje teraz. |
| 4880 | | Priradovanie typov repozitarov | Closed | Bc. Michal Kral | | Ako používateľ chcem repozitar a priradiť mu typ (GitHub, Gerrit...) a jobExecutable. |
| 4881 | | Zmenenie aktuálnej štruktúry DB | Closed | Bc. Michal Kral | | Podľa požiadaviek user story. |
| 4882 | Zadávanie aktuálneho stavu konkrétneho projektu | | Resolved | Bc. Michal Kral | 2 | Ako používateľ chcem zadať stav projektu aby som ich vedel managovať. Stavy: new, active, closed |
| 4883 | | Managovanie stavu projektu | Closed | Bc. Michal Kral | | Pridaná nová tabuľka do DB, ktorá obsahuje stavy projektu. Používateľ má teraz možnosť managovať stavy projektu. |
| 4885 | Sťahovanie z StackOverFlow | | Active | Bc. Peter Bobovsky | 5 | |
| 4886 | | Implementovať službu na sťahovanie stackOverFlow | New | Bc. Peter Bobovsky | | Pracuje sa na nej. |
| 4893 | | Analyzovať existujúce API na sťahovanie stackoverflow | Closed | Bc. Peter Bobovsky | | Stack Overflow poskytuje API na sťahovanie a používanie údajov z ich databázy. Toto API nám absolutne vyhovuje a otvára mnohé možnosti použitia v našom projekte. Môžeme používať verziu bez loginov alebo aj s, podľa potreby a dostávať presne iba údaje ktoré potrebujeme |

| | | | | | | |
|------|--|--|--------|------------------|---|--|
| 4887 | Vytvorenie ontologie na špeciálne prípady | | New | Bc. Lukas Racko | 3 | |
| 4888 | | Vytvorenie ontologiu v owl subore s JobUri a TripleUri | New | Bc. Lukas Racko | | Vytvorená základná ontológia s JobUri a TripleUri aby tieto hodnoty prešli kontrolou podľa OWL ontológií. |
| 4470 | Oznamovanie dostupnosti klientov na server | | Active | Bc. Lukas Racko | 3 | |
| 4863 | | Zdokumentovať API pre oznamovanie stavu klientov na server | New | | | Spísať to do samostatného dokumentu a uploadnúť na google drive |
| 4875 | | Implementovať kontrolu dostupnosti klienta | Closed | Bc. Lukas Racko | | Pridaná funkcia na automatické určenie dostupnosti klienta na základe jeho poslednej aktivity. |
| 4894 | | Zistenie IP klienta | New | | | |
| 4319 | Vytvorenie jobov pre stiahnutie BugZilly | | Active | Bc. Marek Mura | 8 | Ako vyskumník chcem, aby balancer rozdelil stahovanie repozitara do viacerých JobExecution použitím konkrétnej stratégie pre daný repozitar. |
| 4864 | | Analyzovať vytváranie stratégií pre sťahovanie Jobov | New | Bc. Marek Mura | | |
| 4865 | | Implementovať Job | New | Bc. Marek Mura | | |
| 4866 | | Implementovať sťahovanie | New | Bc. Marek Mura | | |
| 4318 | Používateľ sa chce ozvať serveru | | Active | Bc. Peter Kucera | 5 | Ako používateľ sa chcem po zapnutí klienta ohlásiť na server |
| 4623 | | Implementovať ohlásenie klienta na server | New | Bc. Peter Kucera | | Vykoná sa pri štarte klienta, ale aj pri zmene mena používateľa v nastaveniach. |

| | | | | | | |
|------|---|--|--------|-------------------|---|---|
| 4859 | Zobrazenie vykonávania jobov Daemonom (copy from Ebola) | | New | Bc. Peter Kucera | 8 | Ako používateľ chcem aby mi boli priradené joby z balancera, stiahnuť zo serveru executable a vidieť postup každého sťahovania v progress bare |
| 4490 | | Pridať progress bar pre každý job do ich zoznamu | New | Bc. Peter Kucera | | Pridaný progress bar do každého itemu v liste vykonávaných jobov |
| 4494 | | Implementovať klienta pre získanie priradených jobov z Balancera | New | Bc. Peter Kucera | | Trieda, ktorá umožňuje získať zoznam jobov z Balancera, pomocou HttpClient. Stiahnuť, nacať, uložiť, prípadne update |
| 4740 | | Implementovať na servery metodu na poskytnutie executable | New | Bc. Peter Kucera | | Odoslanie executable cez REST API |
| 4889 | Vytvorenie dokumentácie pre Prvy kontrolny bod | | New | Bc. Michal Slovik | | |
| 4890 | | Zistenie čo všetko je potrebné spísať | Closed | | | |
| 4891 | | Spísanie dokumentácie | Active | | | |
| 4892 | | Overenie stavu dokumentácie | New | | | |
| 4491 | Dokončenie sťahovania job | | Closed | Bc. Michal Slovik | 5 | Ako používateľ chcem aby ma aplikácia pri pokuse ju zavrieť vyzvala, či to naozaj chcem, ak sťahovanie jobov je blízko konca aby som zbytočne neprerušil dokončenie jobu, ktorý je už na konci. |
| 4860 | | Implementovať kontrolu pri ukončení aplikácie na bežiaci joby | Closed | Bc. Michal Slovik | | |
| 4867 | | Upozornenie na nedokončené sťahovanie repozitára | Closed | Bc. Michal Slovik | | |
| 4868 | | Otestovať upozornenia sťahovania | Closed | Bc. Michal Slovik | | Skúsiť vypnúť počas sťahovania. |

| | | | | | | |
|------|--|--|----------|-----------------------|---|---|
| 4937 | Kontrolný bod 1 | | Active | Bc. Miriama Pomffyova | | |
| 4938 | | Manažment testovania | New | Bc. Michal Slovik | | Doplniť podkapitolu o manažmente testovania a metodiku testovania. |
| 4939 | | Manažment chýb | New | Bc. Marek Mura | | |
| 4940 | | Manažment dokumentácie | New | Bc. Miriama Pomffyova | | Doplnenie podkapitoly manazmentu a metodiky komunikácie a dokumentácie. |
| 4941 | | Manažment verzíí | New | Bc. Lukas Racko | | |
| 4942 | | Manažment komunikácie a iterácií | New | Bc. Peter Bobovsky | | |
| 4943 | | Manažment prehliadok zdrojového kódu | New | Bc. Peter Kucera | | Doplnenie podkapitoly týkajúcej sa metodiky prehliadok kódu |
| 4944 | | Manažment úloh | New | Bc. Michal Kral | | Manžment je hotový a pripravený na implementáciu do finálneho dokumentu. |
| 4945 | | Finalizácia dokumentácie pre kontrolný bod | Active | Bc. Miriama Pomffyova | | Odkontrolovanie všetkých časti dokumentácie. |
| 4946 | | Exportovanie úloh z TFS | New | Bc. Michal Slovik | | |
| 4495 | Určenie spôsobu sťahovania dát z Git | | Active | Bc. Miriama Pomffyova | 5 | |
| 4874 | | Analyzovať existujúce API pre sťahovanie z repozitárov Git Eclipse | New | Bc. Miriama Pomffyova | | |
| 4873 | | Analyzovať navrhnuté ontológie pre Git a Gerrit | New | Bc. Miriama Pomffyova | | Karol navrhol ontológie pre Git a Gerrit v OWL, prídu aj e-mailom, link aj na discorde. |
| 4872 | | Analyzovať knižnicu pre sťahovanie Git | New | Bc. Miriama Pomffyova | | Môžeme použiť NGit alebo aj niečo iné, ak pre .NET sú. |
| 4895 | | Spísať do dokumentácie | New | Bc. Miriama Pomffyova | | |
| 4638 | Vytvorenie konfiguračného suboru pre nastavenie DB | | Resolved | Bc. Miriama Pomffyova | 5 | Návod na vytvorenie konfiguračného súboru user.json je uverejnený na google disku tímu a súbor gitignore je upravený. |

| | | | | | | |
|------|--|--|--------|-----------------------|--|--|
| 4869 | | Čítanie konfiguračného súboru pri spustení | Closed | Bc. Miriama Pomffyova | | Vytvorenie suboru user.json, v ktorom je zadaný connectionString. Každý prepíše názov svojej databázy a v súbore .gitignore je zapísaný, aby sa pri necommitoval. |
| 4870 | | Upravenie .gitignore pre ignorovanie súboru s DB connection string | Closed | Bc. Miriama Pomffyova | | Do súboru .gitignore bolo potrebné dopísať riadok user.json. Táto informácia je zahrnutá v dokumentácii. Bolo potrebné spustenie príkazu pre git pre vymazanie evidencie o tomto súbore. Dopísaný riadok user.json v súbore .gitignore.txt. Pri commitovaní sa vynecháva. |
| 4871 | | Dokumentovanie postupu rozbehania projektu | Closed | Bc. Miriama Pomffyova | | Spísala som dokumentáciu, v ktorej opisujem, čo je potrebné nastaviť, konfiguračný súbor fungoval správne. V dokumentácii je opísaný postup, ktorý musí používateľ dodržať. Dokumentácia sa nachádza na google disku tímu. Opisuje súbor, ktorý je potrebné vytvoriť a nastavenia pre správne fungovanie konfigurácie. |