Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Big picture k riadeniu v tíme

Tímový projekt 2018/2019

Názov tímu: MonAnt

Členovia tému: Matúš Barabás, Milan Cák, Dung Lam Tuan, Peter Lipták, David Tran Duc, Veronika Žatková Patrik Židuliak

Pedagogický vedúci: Ing. Ivan Srba, PhD.

Ak. rok: 2018/2019, zimný semester

Dátum poslednej zmeny: 6.12.2018

1. Úvod

Tento dokument dokumentuje priebeh práce, riadenia a stav výsledkov v rámci predmetu Tímový projekt počas zimného semestra akademického roka 2018/2019 tímu číslo 14, MonAnt. V rámci projektu tvoríme softvérové riešenie monitoringu antisociálneho správania na internete, predovšetkým šírenia falošných správ, trolovania a osobných útokov.

V druhej kapitole tohto dokumentu sú menovaní členovia tímu s popisom ich roly a zamerania v tíme, aj ich podielom na práci pri vývoji projektu a pri tvorbe tohto dokumentu. V tretej kapitole sú popísané aspekty organizácie tímu a manažmentu našej práce. V štvrtej kapitole sú zhrnuté doterajšie výsledky v jednotlivých šprintoch, ktoré zatiaľ prebehli.

2. Členovia tímu a podiel na práci v tíme

Tím Monant pozostáva zo 7 členov, ktorých hlavné zodpovednosti sú uvedené nasledovne:

Matúš Barabás - Backend programátor, venuje sa najmä programovaniu funkcionality centrálneho úložiska.

Milan Cák - V tíme zastáva primárne úlohu scrum mastera. V rámci vývoja projektu je zodpovedný najmä za front end aplikácie.

Dung Lam Tuan - Backend programátor, venuje sa najmä crawlovaniu.

Peter Lipták - Backend programátor, venuje sa najmä programovaniu API rozhraní a funkcionality centrálneho úložiska.

David Tran Duc - Frontend programátor, venuje sa najmä aplikácií manažmentu crawlerov a parserov

Veronika Žatková - Backend programátor, venuje sa crawlovaniu, parsovaniu a API nad centrálnym úložiskom.

Patrik Židuliak - Backend programátor, venuje sa najmä centrálnemu úložisku

2.1 Podiel práce na dokumentácii inžinierskeho diela

Nasledujúca tabuľka zachytáva percentuálne vyjadrenie podielu práce členov tímu na jednotlivých častiach dokumentácie inžinierskeho diela:

Člen	Ciele pre ZS	Celkový pohľad na systém	Analýza systému	Návrh systému	Implementácia	Testovanie
Matúš Barabás	10	0	25	90	5	30
Milan Cák	0	10	0	0	0	0
Dung Lam Tuan	0	10	20	0	80	0
Peter Lipták						
David Tran Duc			30	10		

Veronika Žatková	90	50	25	0	5	5
Patrik Židuliak						

Tabuľka 2.1: Podiel práce na dokumentovaní inžinierskeho diela

2.2 Podiel práce na dokumentácii riadenia

Tabuľka 2.2 zobrazuje podiel práce členov tímu na jednotlivých častiach dokumentácie riadenia.

Člen	Role členov	Manažment práce	Sumarizácie šprintov
Matúš Barabás	10	25	0
Milan Cák	20	0	60
Dung Lam Tuan	0	10	0
Peter Lipták		0	0
David Tran Duc	10	10	5
Veronika Žatková	20	5	35
Patrik Židuliak	10	50	0

Tabuľka 2.2: Podiel práce na dokumentovaní riadenia

3. Aplikácie manažmentov

Hlavným nástrojom pre organizovanie práce je metodológia Scrum, od ktorej sa odvíjajú viaceré postupy a pravidlá, aplikované na spoločných stretnutiach i počas práce na projekte. Súčasťou organizovania tímu sú dve pravidelné stretnutia na týždennej báze. Prvé trojhodinové stretnutie, spolu s product ownerom, prebieha každý utorok o 8:00, a jeho účelom je zhrnutie progresu za uplynulých sedem dní. Okrem toho sa na tomto stretnutí každý druhý týždeň vyhodnocujú výsledky ukončeného šprintu retrospektívou a taktiež sa plánuje nasledujúci šprint. Druhého trojhodinového stretnutia sa product owner nezúčastňuje. Toto stretnutie slúži predovšetkým na spoločnú prácu na projekte.

Manažment práce na projekte je bližšie popísaný súborom metodík, ktorých účel, formát, a proces tvorby je špecifikovaný v metodike pre tvorenie, udržiavanie a dostupnosť metodík.

3.1 Manažment úloh

Na manažment úloh v rámci projektu používame nástroj **Jira**. V hierarchii úloh rozlišujeme *epiky* (väčšie celky, ktoré predstavujú časť produktu), ktorým prislúchajú *user stories*, z ktorých každá predstavuje väčší celok z plánovanej práce. Tie obsahujú jednotlivé úlohy, *sub-tasky*. Úlohy, ktoré nepatria do žiadnej user story, označujeme ako *tasky*. Každá z úloh okrem toho obsahuje informáciu o komponente, ku ktorému sa viaže. User stories sú udržiavané v produktovom *backlogu*, ktorý je prioritne zoradený podľa ich dôležitosti.

Úlohy sú spoločne plánované v dvojtýždenných šprintoch na spoločných stretnutiach. Pri plánovaní vyberáme z backlogu niekoľko user stories. Ich náročnosť spoločne procesom anonymného hlasovania (metódou scrum poker) ohodnotíme takzvanými story pointmi. Cieľom plánovania je navrhnúť prácu na najbližšie dva týždne tak, aby bola každá story zrealizovateľná v období nadchádzajúceho šprintu. Následne každú user story rozdelíme na niekoľko úloh (sub-taskov). V rámci šprintu nie je možnosť pridania ďalšej user story, je však možné kedykoľvek pridať nový sub-task.

Sub-tasky majú definovaný životný cyklus, ktorý je riadený členom tímu, ktorý na ňom pracuje použitím “kanban” tabule, ktorú ponúka Jira. Každý sub-task má počiatočný stav *To do* s nepriradeným členom tímu. Keď člen tímu začína prácu na danej úlohe, priradí si ju, a nastaví jej stav na *In progress*. Keď svoju prácu dokončí, označí ju ako *In review*. V tomto stave čaká na schválenie ostatnými členmi tímu (predovšetkým prehliadkou kódu). Ak úloha prejde procesom review, a jej výsledkom sú vopred dohodnuté výstupy, presúva sa do stavu *Done*. Úlohu je možné, ak ešte nebola ukončená, zamietnuť presunutím do stavu *Rejected*. Ak je progres v úlohe blokovaný, jej stav sa mení na *Blocked*. Ak sú všetky úlohy v rámci user story ukončené (sú v stave *Done* alebo *Rejected*), považuje sa jej nadradená user story za hotovú. Ak user story na nasledujúcom stretnutí product owner schváli, považuje sa za akceptovanú. Ak práca

na user story nie je v šprinte ukončená podľa plánu, presúva sa do nasledujúceho šprintu s opätovným procesom ohodnotenia scrum pokerom, s ohľadom na čiastočné dokončenie.

Manažment úloh je bližšie opísaný v metodike pre manažment úloh.

3.2 Manažment verziovania kódu

Na riadenie verzií všetkých zdrojových kódov používame **Git** a ako gitové úložisko používame **GitHub**. Na github momentálne máme všetky časti projektu rozdelené na nasledovné štyri repozitáre:

- **MonAnt-Assets**: zdrojový kód, ktorý nie je priamou súčasťou produktu (napríklad prezentačný web tímu)
- **MonAnt-Webapp**: Django aplikácia pre manažment crawlerov
- **MonAnt-Crawler-Parser**: crawlery a parsery
- **MonAnt-FlaskAPI**: centrálne úložisko projektu

V každom repozitári sú vždy dve hlavné vetvy, *master* a *develop*, v ktorých sú funkčné verzie projektu. Každý člen tímu, ktorý aktuálne pracuje na nejakej úlohe si vytvára novú vetvu s názvom v tvare *feature/nn-dd*, kde *nn* je identifikátor úlohy z nástroja Jira a *dd* je jej krátky popis.

Každá commit správa je písaná v angličtine, začína slovesom s veľkým prvým písmenom a končí sa číslom úlohy v Jire.

Autor vývojovej vetvy po dokončení práce vytvára v repozitári na GitHube *pull request* (PR), ktorý po prebehnutí prehliadky zdrojového kódu zlúči danú vetvu s vetvou *develop*. Pri tvorbe PR určuje aspoň dvoch ľudí (predovšetkým s dostatočnou znalosťou v danej oblasti technológie alebo produktu), ktorí túto prehliadku vykonajú. Proces schvaľovania prebieha komentovaním zdrojového kódu a schválením, prípadne zamietnutím zmien. V prípade, že je PR zamietnutý, autor prehliadky špecifikuje svoje výhrady. Autor vývojovej vetvy dané pripomienky zapracuje, a opätovne priradí PR na prehliadku členovi tímu, ktorý ho predtým zamietol. Po schválení PR môže autor vetvu zlúčiť s vetvou *develop*, musí sa však uistiť, že nenastanú žiadne konflikty.

Proces verziovania kódu je bližšie popísaný v metodike verziovania kódu.

3.3 Manažment komunikácie

V rámci tímu komunikujeme rôznymi spôsobmi. Osobná komunikácia prebieha predovšetkým na osobných stretnutiach všetkých členov tímu.

Mimo stretnutí komunikujeme predovšetkým používaním nástroja **Slack**. V tomto nástroji je diskusia rozdelená do kanálov podľa obsahu správ, primárne do nasledujúcich:

- **#general** - komunikácia všeobecných záležitostí tímového projektu, ktoré svojou podstatou nespadajú do žiadneho iného kanálu
- **#product-owner** - primárne vytvorený pre komunikáciu product-ownera s členmi tímu
- **#tpcup** - komunikácia týkajúca sa súťaže TP CUP, do ktorej sa náš tím zapojil
- **#stretnutia** - diskusia o formálnych a neformálnych stretnutiach tímu, časové upresnenia stretnutí, prípadné meškania či absencie jednotlivých členov tímu na stretnutia
- **#github** - organizačné informácie ohľadom verziovania zdrojového kódu, informovanie ostatným členov o vytvorení vetvy, pull requestu, upozornenia na commity prípadne merge a iných organizačných záležitostí týkajúcich sa githubu
- **#jira** - diskusia ohľadom organizačných záležitostiach týkajúcich sa user stories alebo ich subtaskov v jire (vytváranie, prerozdeľovanie úloh medzi členmi tímu, evidencia času, meškania splnení úlohy ...), neslúži na komunikáciu riešenia jednotlivých user stories
- **#branding** - komunikácia ohľadom názvov, tvorby loga a tímového plagátu
- **#dokumenty** - diskusia ohľadom akejkoľvek dokumentácie projektu (netýka sa metodík)
- **#metodiky** - akákoľvek komunikácia, ktorá sa týka metodík tímového projektu, upozornenie na zmenu obsahu jednotlivých metodík, rozdelenie zodpovedností za jednotlivé metodiky
- **#webstranka** - komunikácia ohľadom webovej stránky tímového projektu, jej obsah, úpravy, nasadzovanie nových verzií stránky a pod.
- **#random** - diskusia, ktorá môže ale primárne nemusí súvisieť s tímovým projektom, poskytuje výlučne informácie, ktoré nie sú pre tím dôležité a nie je povinnosťou čítať ich
- **#private** - súkromný kanál na internú komunikáciu medzi členmi tímu

V rámci Slacku využívame aj možnosť priameho posielania súkromných správ iným členom tímu, ak sú adresované iba im, a netýkajú sa iných členov tímu.

E-mailová komunikácia prebieha prostredníctvom služby Gmail výhradne cez mail tímového projektu fiit-tp14@googlegroups.com, aby sa informácia dostala ku všetkým členom tímu. Slúži na prijímanie, posielanie e-mailov všetkú oficiálnu komunikáciu tímu MonAnt, product ownera a verejnosti.

Komunikácia jednotlivých úloh projektu sa odohráva aj v nástroji na evidenciu úloh Jira. Každý člen tímu má možnosť pridať k úlohe komentár, ktorým vyjadrí nejaký problém, prípadne položí otázku. Je potrebné, aby otázka obsahovala označenie na všetkých členov tímu resp. jednotlivcov. Taktiež je potrebné, aby prípadné vytvorené dokumenty, linky a iné záležitosti boli pridané do komentárov po skončení práce na danej úlohe. Do komunikácie v Jire má možnosť pristupovať aj product-owner (vedúci tímu).

Podrobnejšie je komunikácia v tíme popísaná v metodike pre komunikáciu.

3.4 Manažment tvorby zdrojového kódu

Prevažná väčšina zdrojového kódu je písaná v programovacom jazyku **Python** (verzia 3.7). Ďalšie použité technológie sa líšia podľa komponentov:

- Hlavnou časťou **centrálneho úložiska** je server s RESTful API rozhraním, ktorá poskytuje klientom prístup k údajom uloženým v databázovom systéme *PostgreSQL*. Toto rozhranie je implementované vo frameworku *Flask*. Na programatický prístup k databáze a jej manažment sú použité knižnice *SQLAlchemy* (objektovo-relačný mapovač) a *Alembic* (manažment migrácií databázovej schémy).
- Webová aplikácia **manažmentu monitorov** je implementovaná použitím webového frameworku *Django*.
- Jednotlivé **crawlery** sú implementované použitím knižnice *Scrapy*. **Parsery** sú implementované prevažne použitím knižnice *BeautifulSoup*.

Zdrojový kód je členmi tímu implementovaný prevažne individuálne, prípadne v menších (<= 3 ľudia) skupinách. Pri písaní zdrojového kódu členovia tímu dbajú na konzistentný formát a štýl zdrojového kódu vychádzajúcim prevažne z metodiky PEP 8.

Pri tvorbe zdrojového kódu autori dbajú na dodržiavanie kvality udržiavaním dokumentácie, dodržiavaním konvencií, udržiavaním závislostí, ako aj pokrytím kódu automatizovanými testami. Na testovanie programu v Pythone sa používa testovací framework *PyTest*.

Pravidlá pre tvorbu zdrojového kódu sú bližšie popísané v metodike pre písanie zdrojového kódu.

3.5 Manažment kvality tvorby zdrojového kódu

Po napísaní zdrojového kódu, sa kód sprístupní ostatným členom tímu. Pre zdieľanie kódu sa používa git v prostredí GitHub. Za kontrolu kvality zdrojového kódu je zodpovedný iný člen tímu, ako ten čo písal zdrojový kód. Kontrolu vykonáva člen, ktorý má k doméne danému riešeniu najbližšie.

Pri kontrole zdrojového kódu sa pripomienky zapisujú vo forme komentárov do pull requestu danej funkcionality. V prípade nájdenia nedostatkov, člen tímu, ktorý písal zdrojový kód sa vyjadrí ku komentáru alebo opraví danú záležitosť a aktualizuje pull request. Tento cyklus sa opakuje, pokým kontrolóri nenájdu žiadne ďalšie problémy. V tom prípade je riešenie schválené a môže sa spojiť do vetvy pre vývoj kódu.

Pravidlá pre prehliadky zdrojového kódu sú bližšie popísané v metodike pre prehliadky zdrojového kódu.

3.6 Manažment testovania

Testy sa robia po každom pridaní funkcionality, po implementácii viacerých funkcionalít alebo pred odovzdaním softvérového produktu. Testovanie sa vykonáva na na troch úrovniach:

- Unit testovanie
- Integračné testovanie
- Akceptačné testovanie

3.6.1 Unit testovanie

Unit testy sa vytvárajú pri pridaní akejkoľvek funkcionality, pretože ich úspešné prevedenie značí správnosť implementácie. Všetky testy sa ukladajú do adresára **tests**. Sú na sebe nezávislé, to znamená, že sa dajú spustiť samostatne a v akomkoľvek poradí. Pre dosiahnutie efektivity je potrebné, aby jednotlivé testy trvali len veľmi krátky čas (niekoľko milisekúnd). Musia pokrývať celú implementovanú funkcionalitu, všetky valídne i nevalidné scenáre, výnimky a pod.

Každý unit test musí mať názov <nazov_testu>_test.py. Názvy musia byť presné, výstižné, aby z nich bolo čo možno najviac zreteľne vidieť, o akú funkcionalitu sa opierajú.

Vykonávanie testov bude prebiehať prostredníctvom frameworku *pytest.*

3.6.2 Integračné testovanie

Pri integračných testoch sa testuje aplikácia ako celok a taktiež integrácia medzi viacerými časťami aplikácie. Integračné testy slúžia hlavne na identifikáciu defektov týkajúcich sa interakcie medzi jednotlivými komponentmi aplikácie. Vykonávajú sa po implementácii viacerých funkcionalít.

3.6.3 Akceptačné testovanie

Akceptačné testy tvoria formálny popis správania implementovaného softvérového produktu. Validujú splnenie požiadaviek na systém. Testujú sa rôzne scenáre. Akceptačné testovanie sa vykonáva pred odovzdaním softvérového produktu. Celý proces akceptačného testovania je bližšie opísaný v metodike pre testovanie.

3.7 Manažment dokumentovania

Dokumentácia sa uchováva v nástroji confluence. Dokumenty Dokumentácia k riadeniu v tíme a Dokumentácia k inžinierskemu dielu sú dokumenty nachádzajúce sa na google drive pre jednoduchšiu kolaboráciu. Ostatné dokumenty na google drive sú archivované, pre účely kompilácie v špecifickom časovom bode.

Všetky identifikované užitočné procesy, sa dokumentujú vo forme metodík. Technické návody sú tiež dokumentované. Musia obsahovať jednoznačné informácie potrebné pre sfunkčnenie alebo používanie riešenia.

Technické dokumentácie, v ktorých sú obsiahnuté informácie o REST API na centrálne úložisko, sú generované priamo z kódu.

Celý proces dokumentovania je bližšie opísaný v metodike pre písanie dokumentácie a metodike pre písanie metodík.

4. Sumarizácie šprintov

1. Šprint - Absint

V piatom týždni semestra sme ukončili náš prvý šprint. Nakoľko bol tento šprint špecifický tým, že obsahoval nielen technické úlohy, tak sme sa rozhodli dať si pred nasledujúcim šprintom týždeň prestávku, počas ktorej sme doplnili chýbajúce metodiky a pripravili si úlohy na ďalší šprint. Úlohy v uplynulom šprinte však boli dôležité, až nevyhnutné pre realizáciu ďalších implementačných úloh na projekte, nakoľko vďaka nim sme spolu schopní urobiť viaceré dôležité rozhodnutia ohľadom plánovania implementácie a ďalších krokov (napríklad výber druhu databázy, nástrojov pre crawlovanie webu či stanovenie si pravidiel ohľadom ďalšej práce na projekte).

V prvom šprinte sme sa venovali hlavne analýze dostupných zdrojov a crawlerov. Okrem toho sme spísali metodiky, na základe ktorých budeme pracovať počas celých dvoch semestrov. Pripravili sme si potrebné nástroje na manažment práce v tíme a web predmetu. Taktiež sa nám podarilo navrhnúť predbežnú štruktúru centrálneho úložiska, ktoré bude slúžiť ako hlavný zdroj dát pre celý vyvíjaný projekt. Celkovo náš prvý šprint obsahoval 6 stories s celkovou hodnotou 20 storypoints, z ktorých boli kompletne akceptované 3 (a spálených bolo 10 z 20 storypoints).

Po druhom týždni sme v rámci projektu mali implementované 3 druhy vzorových crawlerov, identifikované databázové entity, vytvorený vlastný diagram architektúry riešenia na základe predlohy, spojazdnené všetky potrebné nástroje a preskúmanú väčšinu zdrojov dát (článkov a príspevkov na sociálnych sieťach), ktoré budeme používať ďalej v projekte. Pozitívum šprintu bolo, že na úlohách sa pracovalo priebežne.

Start

- Lepšie manažovanie spoločnej práce
- Byť efektívnejší pri ukončení šprintu
- Priebežne aktualizovať Jiru
- Pridávať komentáre k taskom v Jire (Jira == náš svet, čo tam nie je, to neexistuje, aj linky na výstupy k taskom, kľudne tam dat aj komunikáciu, dokumentácia sa môže generovať z komentárov)
- Do Jiry písať s diakritikou

Stop

- Tu sme neidentifikovali žiadne body.

Continue

- Priebežná práca

Práca na projekte

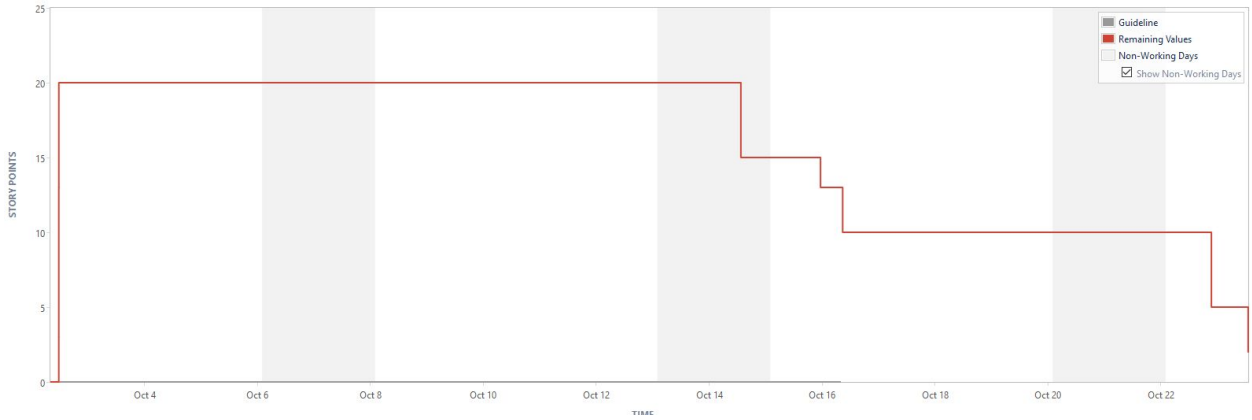
	Počet priradených taskov	Počet dokončených taskov	Zdôvodnenie nedokončených taskov	Počet hodín (hod)	Percentuálny podiel na šprinte (%)
Matúš Barabás	3	3		8	18
Milan Cák	6	5	Duplikát	8	14
David Tran Duc	3	3		6	17
Peter Lipták	3	1	Retrospektíva v review, Export úloh z Jiry v review	5	8
Dung Lam Tuan	2	2		3,5	10
Veronika Žatková	5	5		13	17
Patrik Židuliak	6	4	Metodika v review, Confluence bloknutý hardvérom	9	16

Stories

	Počet story points	Akceptácia story	Dôvod neakceptácie story
[MON-1] Analýza zdrojov	5	áno	-
[MON-2] Experimentálny crawler	4	áno	-

[MON-3] Schéma centrálneho úložiska	2	nie	Schéma nebola dostatočne spracovaná
[MON-4] Architektúra riešenia	1	áno	-
[MON-5] Setup nástrojov, webu	8	áno	-
[MON-6] Metodiky	2	nie	Niektoré metodiky na konci šprintu chýbali

Burn down chart



2. Šprint - Borovička

V rámci šprintu číslo 2 sme si vytýčili prvé implementačné úlohy. Cieľom viacerých z nich bolo vytvorenie “proof of concept” - prototypu na zahodenie, ktorý potvrdzuje naše predpoklady vyplývajúce z analýzy a implementovateľnosť návrhov.

Prvým cieľom bolo vytvoriť kostry hlavných modulov systému - centrálneho úložiska (databázy), API a crawlerov. Súčasťou bolo tiež setupovanie nástrojov a vytváranie metodík. Vytvorené boli taktiež prvé základné endpointy pre prácu s dátovými entitami.

Z predchádzajúceho šprintu neboli akceptované 3 stories (ktoré sa však v aktuálnom šprinte podarilo úspešne ukončiť). Okrem nich do šprintu pribudlo 8 nových stories. Velocity tímu pre aktuálny šprint dosiahla hodnotu 26 storypoints (resp. 36 storypoints v súčte so stories z predchádzajúceho šprintu). Počet úspešne ukončených a akceptovaných stories bolo 7 a počet spálených storypoints bol 21.

Po ukončení šprintu sme mali k dispozícii navrhnutú schému databázy a implementovanú kostru centrálneho úložiska spolu s API a základnými endpointami. Implementovaný bol taktiež prvotný crawler a scraper pre novinové články. V rámci šprintu taktiež vznikali nové metodiky.

Pozitívom šprintu zostala priebežná práca, námetom na zlepšenie do budúcnosti bola lepšia organizácia pull requestov. Pre ďalšiu prácu bol zadefinovaný pojem *sledovacia kampaň*, ktorá predstavuje projekt pre monitorovanie vybraného portálu a pozostáva z monitorov určených pre sledovanie a získavanie dát.

Start

- Aktualizovať obsah na webe
- Presunúť veci z Google Docs na Confluence
- Priraďovať na code review maximálne dvoch ľudí:
    - jeden expert
    - jeden, čo sa chce učiť danú technológiu
- Product owner nahodí product backlog do Jiry
- Lepšie dokumentovať
- Scrum master aktívnejšie monitorovať tím

Stop

- Vytváranie pull requestov na poslednú chvíľu
- Nie neskôr ako v piatok

Continue

- Logovanie času v Jire (aj reviews)
- Aktualizovanie Jiry
- Priebežná práca

Práca na projekte

	Počet priradených taskov	Počet dokončených taskov	Počet hodín (hod)	Percentuálny podiel na šprinte (%)
Matúš Barabás	3	3	5,5	10%
Milan Cák	5	5	17,25	20%
David Tran Duc	6	6	8	13%
Peter Lipták	4	4	10,5	10%

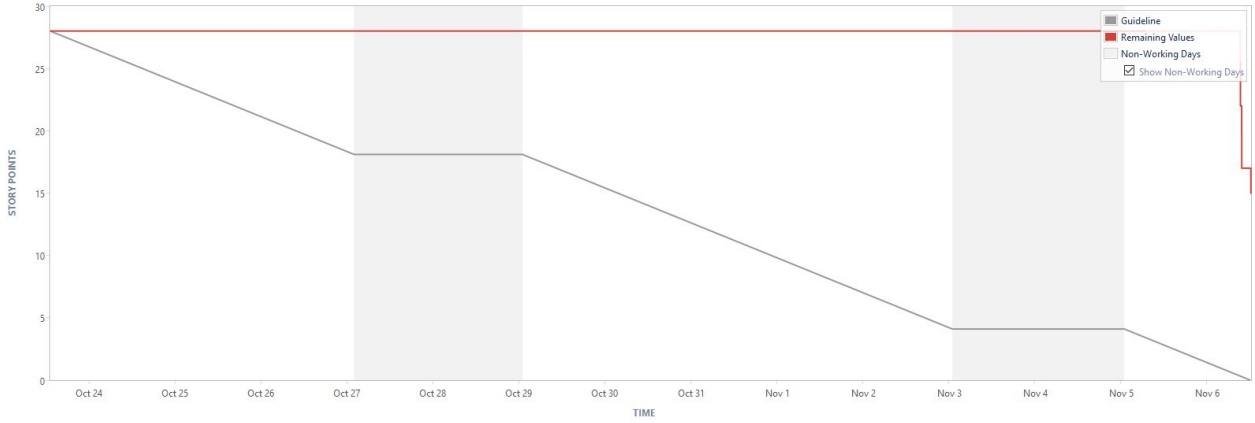
Dung Lam Tuan	2	2	9,5	15%
Veronika Žatková	2	2	8	12%
Patrik Židuliak	3	3	16	20%

Stories

	Počet story points	Akceptácia story	Dôvod neakceptácie story
Schéma centrálneho úložiska	5	áno	-
Setup nástrojov, webu	2	áno	-
Metodiky	3	áno	-
Kostra úložiska	3	áno	-
Endpointy pre uloženie URL	3	áno	-
Endpoint pre uloženie článku	2	áno	-
Dummy crawler	3	áno	-
Kostra BE a FE	3	nie	nedokončenie úloh
Formulár pre pridanie URL	2	nie	nedokončenie úloh

Endpointy pre uloženie a získanie HTML	5	nie	nedokončenie úloh
Dummy parser	5	nie	nedokončenie úloh

Burn down chart



3. Šprint - Cachaca

Tretí šprint nadväzoval na prototypy a kostry systému vytvorené v predchádzajúcich iteráciách. Cieľom bolo implementovať dostatočne veľkú časť funkcionality systému, pokrývajúcu správu centrálneho úložiska a jeho API, crawlovanie a parsovanie novinových portálov (konkrétne badatel.net). Na konci šprintu tak bol implementovaný crawler a parser, ktoré sú schopné získavať dáta z článkov z portálu badatel.net, endpointy pre API pre pridávanie HTML súborov, URL crawlovaných článkov, pridanie článku, prácu s monitormi v rámci sledovacej kampane a ich data providermi. Rovnako je implementované administrátorské prostredie, umožňujúce zadávať URL pre crawlovanie.

Z predchádzajúceho šprintu boli prenesené 4 neakceptované stories, ktoré sa v aktuálnom šprinte podarilo dokončiť a boli akceptované. Okrem nich bolo pridaných 6 nových stories a celková velocity tímu bola odhadovaná na 38 storypoints. Po ukončení šprintu sa podarilo spáliť 28 storypoints.

V rámci retrospektívy bola okrem implementačných detailov diskutovaná aj spoločná práca a komunikácia v tíme, na základe čoho boli dohodnuté ďalšie kroky pre zlepšenie spolupráce, ako napríklad včasné komunikovanie potreby pomoci alebo nestíhania.

Start

- refactoring testov
- refactoring API

- refactoring DB, doplnenie indexov,
- v rámci code review dôkladnejšie kontrolovať migrácie
- integrovanie storiek PRED stretnutím!
- zaviesť CI (Travis)
- zdieľať kolekcie requestov z Postman (pridať do repo)
- priebežne dokumentovať do confluence a link pridávať do user stories v Jira
- včas komunikovať nestíhanie
- nová metodika pre testovanie
- nová metodika pre code review
- rozšíriť metodiku pre verziovanie
- pair programming na spoločných stretnutiach
- ak mám problém, tak kričím čo najskôr a kontaktujem:
    - experta na danú oblasť
    - scrum mastera
    - product ownera

Stop

- Neboli identifikované žiadne body

Continue

- aktívnejší scrum master
- poctivejšie logovať čas v Jira

Práca na projekte

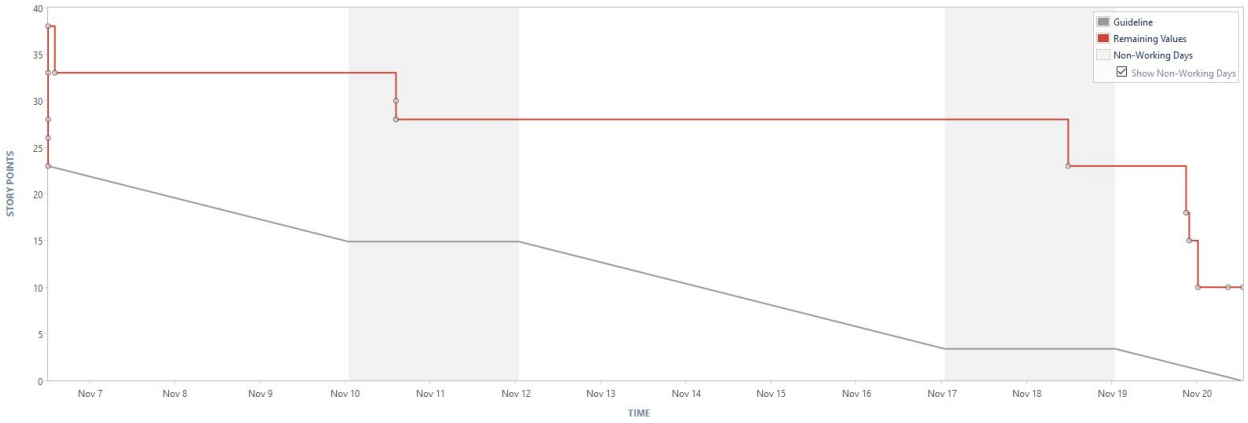
	Počet priradených taskov	Počet dokončených taskov	Počet hodín (hod)	Percentuálny podiel na šprinte (%)
Matúš Barabás	5	5	15	20
Milan Cák	3	3	5	10
David Tran Duc	2	2	22	20
Peter Lipták	2	2	6	10
Dung Lam Tuan	5	5	12	15

Veronika Žatková	4	4	9	10
Patrik Židuliak	1	1	15	15

Stories

	Počet story points	Akceptácia story	Dôvod neakceptácie story
Kostra BE a FE	3	áno	-
Formulár pre pridanie URL	2	áno	-
Endpointy pre uloženie a získanie HTML	5	áno	-
Dummy parser	5	áno	-
Endpointy pre CRUD monitorov	5	áno	-
Badatel.net crawler článkov	5	áno	-
Badatel.net parser článkov	3	áno	-
Manažment	0	nie	nedokončenie úloh
CRUD monitorov	8	nie	nedokončenie úloh
Rozšírenie endpointov pre ukladanie dát	2	nie	nedokončenie úloh

Burn down chart



4. Šprint - Demänovka

Prácu vo štvrtom šprinte nám značne skomplikovala náročnosť zadaní, ktoré bolo potrebné odovzdať na iné predmety. Preto sa do nasledujúceho šprintu prenáša 6 stories z 8. Aj napriek tomu sa nám však podarilo spáliť 21 z 35 story points.

Podarilo sa rozšíriť endpointy pre ukladanie dát a refactoring databázy. Technický dlh, ktorý sa nazbieral počas doterajšej práce, bol väčší než sme pôvodne predpokladali.

Continue

- priebežne dokumentovať do confluence a link pridávať do user stories v Jira!
- písať komentáre k taskom v Jira
- nová metodika pre testovanie
- nová metodika pre code review
- rozšíriť metodiku pre verziovanie
- poctivejšie logovať čas
- aktívnejší scrum master
- lepšie vyhodnocovať riziká (deadlines na projekte / dealines na zadaniach)

Práca na projekte

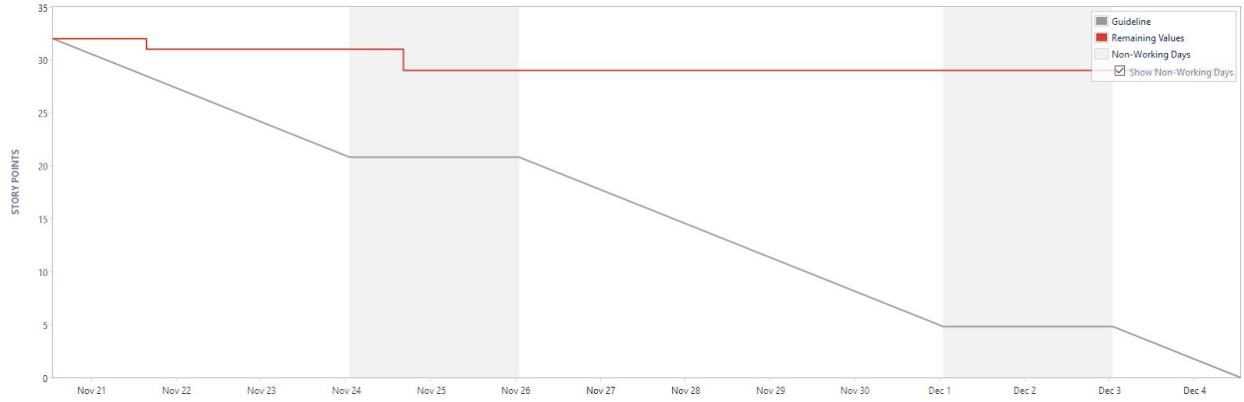
	Počet priradených taskov	Počet dokončených taskov	Počet hodín (hod)	Percentuálny podiel na šprinte (%)
Matúš Barabás	5	4	16	20
Milan Cák	4	2	8	10

David Tran Duc	2	1	15	20
Peter Lipták	1	1	6	10
Dung Lam Tuan	1	1	7	10
Veronika Žatková	1	1	9	10
Patrik Židuliak	8	5	14	20

Stories

	Počet story points	Akceptácia story	Dôvod neakceptácie story
Rozšírenie endpointov pre ukladanie dát	2	áno	-
Refactoring DB	1	áno	-
Manažment	0	nie	nedokončenie úloh
CRUD monitorov	8	nie	nedokončenie úloh
RSS parser článkov	5	nie	nedokončenie úloh
Refactoring API	5	nie	nedokončenie úloh
Refactoring testov	8	nie	nedokončenie úloh

Burn down chart



5. Šprint - Etanol

Stop

- Neskoré písanie retrospektív
- Zlé odhadovanie časových možností

Continue

- Pair programming

Práca na projekte

	Počet priradených taskov	**Počet dokončených taskov**	**Počet hodín (hod)**	**Percentuálny podiel na šprinte (%)**
Matúš Barabás	**3**	**3**	**14**	**15**
Milan Cák	**6**	**6**	**15**	**15**
David Tran Duc	**2**	**2**	**13**	**15**
Peter Lipták	**2**	**2**	**10**	**10**
Dung Lam Tuan	**3**	**3**	**12**	**15**
Veronika Žatková	**3**	**3**	**11**	**15**

Patrik Židuliak	**2**	**2**	**16**	**15**

Stories

	Počet story points	**Akceptácia story**	**Dôvod neakceptácie story**
RSS parser článkov	**5**	**áno**	**-**
Badatel.net crawler článkov	**3**	**áno**	**-**
CRUD monitorov	**8**	**áno**	**-**
Refactoring API	**5**	**áno**	**-**
Refactoring testov	**8**	**áno**	**-**
Manažment	**0**	**nie**	**Niektoré metodiky na konci šprintu chýbali**

6. Globálne retrospektívy

6.1 Retrospektíva za zimný semester

Počas zimného semestra sme sa toho veľa naučili. Práca v takomto tíme bola pre nás veľmi prínosná a ukázala nám, ako môže fungovať vývoj softvéru v agilnom tíme.

Na začiatku sme dostali veľkú výzvu v podobe softvéru, ktorý má byť základom pre viacerých diplomantov, bakalárov a tiež tímom zo zahraničia. Analýze a návrhovým rozhodnutiam sme preto venovali podstatnú časť celého prvého šprintu. Rozhodli sme vývoj začať *centrálnym úložiskom*, pretože to bude základnou časťou celého softvéru. Centrálne úložisko sme obalili do API rozhrania, ktoré zabezpečuje správnu prácu s databázou a tiež ponúka jednoduchý prístup pre všetky ostatné komponenty, ktoré úložisko používajú.

Vďaka tomu, že sa náš tím skladá z členov, ktorí majú skúsenosti s rôznymi technológiami a oblasťami informatiky sme dokázali paralelne s centrálnym úložiskom vyvíjať aj komponenty na crawlovanie a parsovanie obsahu. Podarilo sa nám implementovať crawler a parser pre rôzne webové stránky a rovnako aj RSS zdroje.

Prácu v jednotlivých šprintoch sme si vždy rozdelili tak, aby centrálne úložisko obsahovalo potrebné API rozhrania pre vyvíjanú funkcionalitu ostatných komponentov. Takto sa nám podarilo v neskorších šprintoch všetky komponenty spolu vždy integrovať a overiť tak funkčnosť celého systému.

Poslednou vyvíjanou časťou v tomto semestri bol nástroj na správu a zobrazenie výsledkov monitorovania. Tento nástroj sme vyvíjali už od druhého šprintu a rozdelili sme ho na viacero častí.

Start

- nový scrum master
- aktívnejší scrum master
- rýchlejšie robiť retrospektívy po stretnutí
- lepšie informovať product ownera
- úprava product backlogu a ohodnocovanie stories robiť na stretku počas sprintu

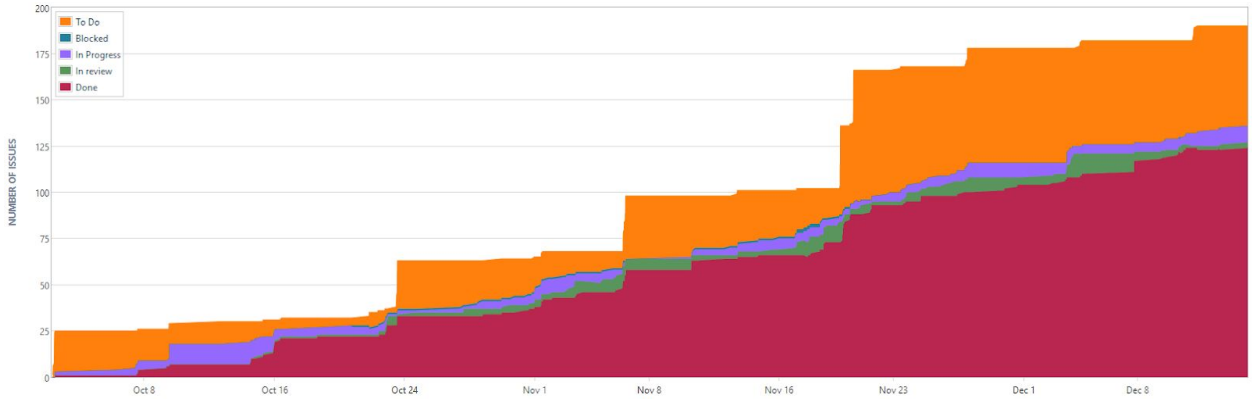
Stop

- spoločné stretnutia naplánovať na iný deň než hneď po TP

Continue

- ešte intenzívnejší pair programming na spoločných stretnutiach
- využívať stretnutia na osobnú prehliadku kódu

- pravidelné trackovanie odpracovaného času



Tento diagram zobrazuje počet úloh a ich stav. Je tu vidieť, že naša schopnosť dodávania výsledkov sa počas posledných 2 šprintov v porovnaní s prvými 3 šprintami rapídne znížila. Príčinou tohto poklesu bolo náročnejšie obdobie v škole počas ktorého sme museli odovzdať viacero ďalších zadaní. Pri plánovaní šprintov sa nám nepodarilo dostatočne dobre odhadnúť časovú náročnosť ďalších školských zadaní.

7. Webové sídlo projektu

Webové sídlo projektu sa nachádza na adrese: http://team14-18.studenti.fiit.stuba.sk/.