

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Dokumentácia k riadeniu projektu

Automatické testovanie v prostredí Internetu vecí

Vedúci projektu: doc. Ing. Tibor Krajčovič, PhD.

Product owner: Ing. Lukáš Ondriga

Členovia tímu:
Bc. Tomáš Bujna
Bc. Marián Ján Franko
Bc. Rastislav Kováč
Bc. Igor Labát
Bc. Miroslav Sabo
Bc. Filip Starý
Bc. Stanislav Širka

Akademický rok: 2018/2019

1 Úvod	4
2 Role členov tímu	5
2.1 Členovia tímu	5
2.1.1 Bc. Tomáš Bujna	5
2.1.2 Bc. Marián Ján Franko	5
2.1.3 Bc. Rastislav Kováč	6
2.1.4 Bc. Igor Labát	6
2.1.5 Bc. Miroslav Sabo	6
2.1.6 Bc. Filip Starý	7
2.1.7 Bc. Stanislav Širka	7
2.1 Podiel práce	8
3 Aplikácia manažmentov	9
3.1 Plánovanie	9
3.1.1 User story	9
3.1.3 Product backlog	10
3.2 Zdrojový kód	10
3.2.1 Verziovanie	10
3.2.2 Komentovanie	11
3.3 Komunikácia	11
3.3.1 Slack	11
3.3.2. Jira	12
3.3.3. Tímový email	12
3.4 Tímové stretnutia	12
3.4.1 Otvorenie šprintu	13
3.4.2 Retrospektíva	13
4 Sumarizácia šprintov	14
4.1 Šprint č.1	14
4.1.1 Úlohy	14
4.1.2 Retrospektíva	15
4.2 Šprint č.2	15
4.2.1 Úlohy	15
4.2.2 Retrospektíva	15
4.3 Šprint č.3	15
4.3.1 Úlohy	16
4.3.2 Retrospektíva	16
4.4 Šprint č.4	16
4.4.1 Úlohy	16
4.4.2 Retrospektíva	17
4.5 Šprint č.5	17
	2

4.5.1 Úlohy	17
4.5.2 Retrospektíva	17
5 Globálna retrospektíva	18
6 Záver	19
Príloha A - Metodiky	20
Metodika tvorby webovej stránky	20
Metodika manažmentu úloh	20
Metodika verziovania kódu	21

1 Úvod

Cieľom predmetu Tímový projekt je oboznámenie študentov s vytvorením nového systému v rámci menšieho tímu. Študenti pracujú podľa agilnej metódy vývoja tzv. Scrum. Scrum má presne definované pravidlá, ktoré je potrebné dodržiavať. Výhodou tejto agilnej metódy je plánovanie len nasledujúceho šprintu. Z toho vyplýva že keď sa niečo pokazí, členovia tímu zistia rýchlo problém a riešia hneď nápravu. Jeden šprint trval dva týždne. Na začiatku šprintu sa vytvorili príbehy, kde každý príbeh mal viacero úloh. Jednotlivé úlohy boli pridelené členom tímu, na ktorých pracovali počas trvania šprintu. Po ukončení šprintu sa vyhodnocovali výsledky a konala sa retrospektíva, pomocou ktorej sa tím snažil eliminovať nedostatky v najbližšom šprinte.

Pri vykonávaní úloh bolo potrebné, aby sa všetci členovia riadili podľa vytvorených pravidiel. Súhrn pravidiel sa nazýva metodika. Vytvorené boli viaceré metodiky týkajúce sa manažmentu konkrétnej oblasti.

Tím sa skladá zo siedmich členov. Každý člen mal počas každého šprintu pridelenú úlohu. Úlohy boli pridelené členom podľa preferencií a skúsenosti v danej oblasti.

Dokument je rozdelený do viacerých kategórií ako krátke predstavenie členov s ich záujmami v oblasti IT, manažment, jednotlivé šprinty. Súčasťou dokumentu sú aj prílohy.

2 Role členov tímu

Členovia tímu sa na spolupráci dohodli už pred začatím zimného semestra, v ktorom bola zahájená práca na tomto projekte. Väčšina z členov tímu sa vzájomne poznala už pred začatím práce, čo bolo veľkým prínosom, pretože bariéry v komunikácii medzi členmi tímu boli minimálne. V úvode semestra boli identifikované a definované roly, potrebné pri práci na projekte. Tieto roly boli po vzájomnej dohode pridelené medzi členov tímu. Pridelenie rolí bolo ovplyvnené najmä predošlými skúsenosťami v oblastiach, ktoré sa vybranej roly týkali. Názvy rolí sa v úvodných fázach práce upravili a ustálili. Na manažmente tímu sa podielali súčasne všetci členovia, pretože si uvedomovali vzájomnú zodpovednosť za celý projekt.

2.1 Členovia tímu

V tejto kapitole si predstavíme jednotlivých členov tímu. Identifikujeme ich role a taktiež popíšeme ako sa zapájali v tvorbe samotného softvéru pre tímový projekt.

2.1.1 Bc. Tomáš Bujna

Rola: Webový inžinier, REST API dizajnér a druhá úroveň Robot Framework testing

Absolvent bakalárskeho stupňa štúdia na FIIT STU v odbore Informatika. K jeho záujmom patrí vývoj a návrh informačných systémov, pričom sa rád zameriava na nové technológie a výzvy pre neho predstavujú neoddeliteľnú súčasť projektov. Aktuálne sa sústreďuje hlavne na oblasť bezpečnosti IS. Pri svojej doterajšej práci sa oboznámil s programovacími jazykmi C, C++, Java a Python.

V rámci tímu bola jeho úloha s počiatku vývoj a implementácia tímového webového sídla. Aktuálne ide už len o správu a rozširovanie webového sídla. Po úspešnom vytvorení webového sídla sa do jeho správy zverilo aj dizajnovanie REST API rozhrania, ktoré je nevyhnutne potrebné pre komunikáciu v nami navrhovanom IoT systéme. Taktiež vypomáha s ďalšími úlohami, ktoré sa ukazujú ako problémové napr. výpomoc pri vytváraní akceptačných testov vo vývojovom prostredí Robot Framework.

2.1.2 Bc. Marián Ján Franko

Rola: Tvorba testov pomocou Robot Framework

Marián Ján je absolventom bakalárskeho stupňa štúdia na FIIT STU v odbore Informatika. Jeho hlavným záujmom je vývoj mobilných aplikácií, ktoré bolo aj zadaním jeho bakalárskej práce. Vyzná sa v automatizačnom testovaní a to je aj jeho rola v tímovom projekte. Programovacie jazyky ovláda: Java, C, SQL a taktiež technológie ako Robot Framework, Selenium, REST API.

Jeho hlavná úloha je vývoj automatizačných testov nad webovou aplikáciou zariadenia ComoNeo. Taktiež pomáha pri vývoji REST API.

2.1.3 Bc. Rastislav Kováč

Rola: Requirements inžinier

Rastislav Kováč absolvoval bakalársky stupeň štúdia na FIIT STU v odbore Internetové technológie. Medzi jeho záujmy v oblasti IT patrí najmä vývoj webových a mobilných aplikácií.

Jeho úlohou v tíme je naštudovanie si problematiky danej úlohy a následne konzultuje zistené poznatky s hardvérovými a softvérovými inžiniermi. V zimnom semestri sa podieľal na výbere vhodného webového servera a na zabezpečení komunikácie RTU s ARM, ktoré sú súčasťou vývojovej dosky BeagleBone Black.

2.1.4 Bc. Igor Labát

Rola: Softvérový inžinier

Igor Labát je absolventom bakalárskeho štúdia na fakulte Informatiky a informačných technológií STU v odbore Internetové technológie. Medzi jeho oblasti záujmu patria sieťové technológie, primárne SDN. Ovláda programovacie jazyky Python, C a základy x86 jazyku symbolických inštrukcií.

Rola ktorú spĺňa v tíme je softvérový inžinier. Jeho hlavnou úlohou v zimnom semestri bolo naštudovať na akom princípe funguje Real Time Unit na vývojovom kite BeagleBone Black a napísať program ktorý otestuje funkčnosť RTU. Následne vytváral program na ovládanie digitálnych výstupov pomocou RTU.

2.1.5 Bc. Miroslav Sabo

Rola: Hardvérový inžinier

Miroslav Sabo absolvoval bakalárske štúdium na fakulte informatiky a informačných technológií STU v odbore Internetové technológie. V oblasti informatiky sa venuje hlavne práci s IoT zariadeniami ako aj vývoju softvéru spojeného s integráciou systémov. Ovláda najmä programovacie jazyky Java, C++ a C.

Miroslav Sabo zastupuje v tíme rolu hardvérového inžiniera, ktorého hlavnou úlohou v zimnom semestri bola analýza existujúceho zariadenia a jeho následný refaktoring, keďže toto zariadenie obsahovalo chyby, ktoré bolo nutné opraviť. Taktiež je jeho úlohou poskytnúť

ostatným členom tímu informácie o hardvérovej časti projektu akými sú napríklad čísla portov využívané na jednotlivé úkony.

2.1.6 Bc. Filip Starý

Rola: Softvérový inžinier

Filip Starý je absolventom bakalárskeho štúdia na FIIT STU v odbore Internetové technológie. Medzi jeho oblasti záujmu patrí najmä vývoj softvéru a aplikácií. Taktiež má skúsenosti s vývojom sieťových aplikácií, ktoré nadobudol pri tvorbe bakalárskej práce. Z programovacích jazykov najlepšie ovláda C#, Java a z jazykov nižšej úrovne C.

Filipova rola v tíme nesie názov softvérový inžinier a stará sa najmä o architektúru systému. V zimnom semestri Filip vytváral program, ktorý funguje na vývojovej doske BeagleBone Black a má za úlohu v reálnom čase pomocou RTU(Real Time Unit), ktorá je súčasťou tejto dosky, ovládať digitálne výstupy tejto dosky. Taktiež pracoval na komunikácii medzi RTU a procesorom ARM, na ktorom beží operačný systém Linux.

2.1.7 Bc. Stanislav Širka

Rola: Scrum Master

Stanislav Širka absolvoval bakalárske štúdium na fakulte informatiky a informačných technológií STU v odbore Informatika. Jeho hlavným záujmom sú modelovanie, architektúry informačných systémov a agilný prístup vývoja softvéru. Z programovacích jazykoch ovláda C, C++, Java, Python a rád si zamodeluje v UML, ArchiMate a BPMN.

Stanislav Širka v tíme zastupuje úlohu Scrum Master, ktorú sa snaží maximálne dodržať. Najčastejšie sa pozerá do nástroja Jira za pomoci ktorého spolu s Product Ownerom a tímom vytvárajú backlog úloh. V zimnom semestri aplikoval Scrum v tíme a rozbiehal nástroj Jiru. Bol súčasťou úloh pre RESTI API a BeagleBone Black, kde usmerňoval tím v správnom smere. Okrem toho sa snaží uchopiť projekt ako jeden celok a porozumieť jednotlivým častiam systému a ich závislostiam.

2.1 Podiel práce

Meno a priezvisko	Dokument Riadenie [%]	Dokument Inžinierske dielo [%]
Bc. Tomáš Bujna	15	15
Bc. Marián Ján Franko	15	15
Bc. Rastislav Kováč	10	20
Bc. Igor Labát	15	10
Bc. Miroslav Sabo	15	15
Bc. Filip Starý	10	20
Bc. Stanislav Širka	20	5

3 Aplikácia manažmentov

Kvalitne navrhnutý manažment je neodmysliteľnou súčasťou každého dobre fungujúceho tímu. Manažment v našom tíme bol dohodnutý na prvých stretnutiach spolu s vedúcim tímu a product ownerom (zamestnanec z firmy Kistler s.r.o.). Na tvorbe manažmentu sa podieľali taktiež všetci členovia tímu. Celý tím sa následne týmto manažmentom riadi pri tvorbe nášho tímového projektu.

3.1 Plánovanie

Každý šprint začína plánovaním úloh. Dobre rozplánovanie úloh do daného šprintu si vyžadovalo získanie prehľadu o tom, ako tím pracuje a koľko úloh zvládne odovzdať za daný šprint. Ako nástroj na správu úloh sme si zvolili webový nástroj Jira, ktorý nám poskytla priamo firma Kistler. Vytvorili pre nás nový projekt ktorý sme si sami manažovali. Pre prístup k tomuto nástroju je potrebné pripojenie cez VPN na firmu Kistler. Nástroj Jira nám poskytuje skvelý prehľad o plánovaných úlohách, jednotlivé úlohy sa dajú ohodnotiť a taktiež sa dajú sledovať jednotlivé šprinty.

3.1.1 User story

User story je základná časť (jednotka) v plánovaní projektu. Je to konkrétna úloha, ktorá môže byť funkcionálneho charakteru (implementácia databázy) ale taktiež nie funkcionálneho charakteru (spísanie dokumentácie). Jednotlivé user story, ktoré tím spolu s product ownerom vytvoril, sú následne ohodnotené číslom, ktoré udáva, ako je úloha náročná. Toto číslo nepredstavuje jednotku reálneho času. Veľmi dobre to slúži na to, aby sme vedeli porovnať náročnosť danej user story voči ostatným user story.

Na ohodnotenie každej user story náš tím využíva techniku planning poker. Je to technika, pri ktorej všetci členovia tímu držia číselne ohodnotené karty (1, 2, 3, 5, 8 atď). Každý člen tímu si následne zvolí jednu kartu, ktoré predstavuje náročnosť úlohy podľa jeho úvahy. Ak sa čísla všetkých členov tímu nezhodujú, začína diskusia, pri ktorej každý povie prečo dal také číslo. Vyjadri sa čo očakáva a čo môže predstavovať problém/výhodu pri vypracovaní danej user story. Následne sa hlasuje znova, až pokým sa nezhodne celý tím na jednom konkrétnom čísle.

3.1.2 Task

Jednotlivé user story predstavujú väčšiu, celkovú funkcionálnosť (implementácia databázy) vytváraného projektu a preto je potrebné ich rozdeliť na menšie časti - tasky (zvoliť technológiu, implementovať zvolenú technológiu, navrhnuť tabuľky atď). Samotné tasky sú taktiež ohodnotené číslom, ktoré predstavuje počet hodín na ich dokončenie.

3.1.3 Product backlog

Product backlog slúži na uchovávanie jednotlivých user story. Pri otvorení šprintu, ktoré sa koná každý druhý týždeň sa z backlogu vyberajú user story, ktoré sa v danom šprinte vykonávajú a na konci šprintu odovzdávajú. Jednotlivé user story sa do šprintu vyberajú za pomoci product ownera a všetkých členov tímu.

Počet user story, ktoré sa vyberajú závisí od velocity daného tímu. Velocity predstavuje konkrétne číslo, koľko tím dokáže odovzdať user story bodov za jeden šprint. Toto číslo sa vypočíta ako súčet user story bodov, ktoré tím splnil a odovzdal ako dokončené na konci šprintu. Velocity v našom tíme sme si vedeli určiť po prvých dvoch šprintoch, v ktorých sme mali vybrané určité user story a na základe ich splnenia sme zistili, koľko úloh dokážeme zvládnuť.

Keď sú user story pridané do šprintu, nasleduje pridelenie členov tímu. User story sa pridávajú členom tímu na základe toho, aká je jeho rola v tíme a aké majú skúsenosti s technológiami vystupujúcimi v danej user story. Ku každej user story sa taktiež pridáva review-er, ktorý má na starosti pozeráť a kontrolovať zmeny vykonané na user story. Aby sa user story dala akceptovať ako hotová musí mať taktiež napísané akceptačné kritériá (definition of done), ktoré určujú, kedy je úloha splnená a za akých podmienok.

3.2 Zdrojový kód

Pre splnenie a dokončenie nášho tímového projektu musíme taktiež implementovať určité softvérové časti. Keďže tieto časti implementuje a programuje viacej ľudí, je nesmierne dôležité dohodnúť sa na spoločnom úložisku zdrojového kódu a taktiež určitých pravidlách formátovania a komentovania vytvoreného kódu.

3.2.1 Verziovanie

Ako úložisko a nástroj na verziovanie sme si zvolili Git. Je to v dnešnej dobe najlepší nástroj na verziovanie, keďže sa veľmi jednoducho ovláda a je intuitívny. Na ovládanie nášho repozitára na Gite využívame nástroj GitLab, ktorý nám poskytla firma Kistler. Na prístup k tomuto nástroju je potrebné pripojenie pomocou VPN priamo na firmu Kistler. Nástroj GitLab uľahčuje ovládanie Git-u a taktiež poskytuje skvelý prehľad o repozitári. Umožňuje vytvárať pull request novo vytvorených častí, kde jednoducho vieme pridať review-era. Ten nám následne povolí pridať (merge) novo implementované časti, alebo nám vie okomentovať, čo je nutné zmeniť.

Dohodli sme sa na určitých pravidlách, ako verziovať kód a taktiež ako vytvárať nové branch. Na toto sa odkazuje metodika verziovania zdrojového kódu.

3.2.2 Komentovanie

Zdrojový kód softvérových častí vytvára v našom tíme viacej ľudí a preto sme si dohodli pravidlá pre komentovanie a formátovanie. Aby zdrojový kód nebol vytvorený rôznymi štýlmi, keďže každý člen môže implementovať rôzne, je potrebné sa týmito pravidlami riadiť. Vďaka týmto pravidlám je náš zdrojový kód jednotný a ucelený.

3.3 Komunikácia

Jedným z kľúčových faktorov pri práci na tímovom projekte je komunikácia. Komunikácia má veľký dopad na samotný úspech celého projektu. Pri zanedbaní komunikácie vzniká veľké množstvo problémov, ktoré môžu viesť k neúspechu. V prípade tohto projektu bolo potrebné zabezpečiť komunikáciu medzi jednotlivými členmi tímu, ako aj komunikáciu medzi tímom a product ownerom. Z tohto dôvodu boli pri práci využité viaceré komunikačné kanály. Tieto komunikačné kanály sú popísané nižšie.

3.3.1 Slack

Pri komunikácii tímu bol primárne využitý komunikačný nástroj Slack. Pre tento nástroj sa tím rozhodol po vzájomnej konzultácii. Tento nástroj umožňuje komunikáciu v rámci viacerých vlákien. Vďaka tomu je možné komunikáciu rozdeliť do konkrétnych celkov podľa obsahu komunikácie. Všetci členovia tímu mali prístup ku všetkým vláknam a podľa toho na čom práve pracovali komunikovali vo vlákne, určenom pre danú tému. Taktiež je vďaka tomuto nástroju možné aj zabezpečiť komunikáciu medzi dvomi členmi tímu. Takejto komunikácii sa však tím vyhýbal, pretože sa snažil udržiavať všetkých členov tímu v obraze a teda využívať pre komunikáciu verejné vlákna. V rámci nástroja Slack tím vytvoril tieto vlákna:

- **General** - v tomto vlákne prebiehala najmä komunikácia týkajúca sa organizácie tímu. Ďalej tu boli preberané záležitosti, ktoré sa týkali každého člena tímu.
- **Random** - v tomto vlákne boli rozoberané témy, ktoré priamo nezapadali do ostatných vlákien.
- **RTU** - v tomto vlákne prebiehala komunikácia týkajúca sa RTU(Real Time Unit). Keďže práca na RTU bola problematická, bolo potrebné riešiť množstvo problémov, ktoré postupne vznikali. Diskusia ako aj riešenie týchto problémov prebiehali v tomto vlákne.
- **Robot Framework** - toto vlákno bolo určené pre komunikáciu týkajúcu sa vytvárania testov pomocou rozhrania Robot Framework.
- **Webpage** - úlohou tohto vlákna bolo zabezpečenie komunikácie týkajúcej sa webovej stránky. Súčasťou tohto vlákna boli pripomienky na dizajn stránky, ako aj pripomienky na obsah a funkcionality stránky.

3.3.2. Jira

Na udržiavanie prehľadu úloh na tímovom projekte bol využitý nástroj Jira. Jira je nástroj na manažovanie a správu úloh v tíme, ktorý je vhodný pri agilnom vývoji Scrum. V tomto nástroji bol definovaný projekt, na ktorom sa pracovalo iteratívno-inkrementálne v kratších časových úsekoch zvaných Šprinty. Šprinty pri tomto projekte trvali dva týždne. V rámci projektu boli definované user stories, ktoré tvorili väčšie celky. Súčasťou user story boli jednotlivé menšie úlohy (task), ktoré bolo možné vypracovať samostatne. Každý user story je možné priradiť zodpovednú osobu.

Pre tento nástroj sa tím rozhodol po vzájomnej dohode. Výber nástroja ovplyvnil najmä fakt, že tento nástroj patrí medzi najpoužívanejšie a najznámejšie nástroje pri agilnom vývoji Scrum. Ďalším faktorom, ktorý ovplyvnil výber tohto nástroja bol fakt, že tento nástroj používajú vo firme Kistler, s ktorou tím spolupracuje na tomto projekte. Pri práci s nástrojom Jira nie je potrebná žiadna inštalácia a je možné s týmto nástrojom pracovať pomocou webového prehliadača. Nástroj poskytuje možnosť ohlasovať vykonané zmeny členom tímu prostredníctvom emailu.

S týmto nástrojom pracoval každý člen tímu, no v najväčšej miere s ním pracoval scrum master, ktorý poskytoval taktiež rady ostatným členom.

3.3.3. Tímový email

Na komunikáciu z vonkajšej strany s tímom slúži emailové konto Gmail. Emailová adresa tímu je fiit.tp.tim15@gmail.com.

3.4 Tímové stretnutia

Pre tímové stretnutia sme dostali k dispozícii miestnosť na fakulte, ktorá je dostupná pre všetkých členov tímu v priebehu celého týždňa. Tímové stretnutia spolu s product ownerom sa konajú každý pondelok v čase 14:00-20:00, pričom sa všetci členovia tímu zúčastňovali na všetkých stretnutiach. Stretnutie je rozdelené do dvoch základných častí:

1. Práca v prítomnosti product ownera. (14:00-17:30)
2. Samostatná práca tímu. (17:30-20:00)

Štandardne ale vyčlenený čas nestačí, a preto je tím nútený komunikovať priebežne počas celého týždňa a to aj vrátane dodatočnej komunikácie s product ownerom. Počas priebehu tímových stretnutí sa tím spolu s product ownerom dohodol na rozdelení jednotlivých úloh na "tasky", pričom jednotlivé úlohy boli na celý priebeh šprintu pridelené zodpovedným osobám.

3.4.1 Otvorenie šprintu

Otvorenie šprintu sa konalo vždy každé dva týždne. Na začiatku šprintu sa ako prvé zdefinovali user stories za pomoci product ownera. Pre každú user story product owner stanovil akceptačné kritériá, podľa ktorých je jasne dané, aké kritériá musí story spĺňať, aby sa mohlo prehlásiť, že je story splnená. Tiež sa vytvorili akceptačné kritériá tímu, ktoré musia členovia tímu spraviť aby bola user story prezentovaná product ownerovi. Ako je vyššie spomenuté user story sa potom rozčlenila na menšie úlohy, ktoré boli pridelené jednotlivým členom tímu.

3.4.2 Retrospektíva

Medzi najdôležitejšie agilné praktiky patrí retrospektíva. Cieľom je vypočuť si názory všetkých členov na danú vec, nie každý má rovnaký pohľad na vec. Členovia tímu hodnotia prácu počas šprinty, vymenujú klady a zápory a navrhnu prípadne zmeny. Vďaka tomu vieme, ktoré veci treba zlepšiť v rámci tímu. Retrospektíva sa vykonáva po každom šprinte, aby sa problémy odstránili prípadne zlepšenia aplikovali včas.

V tíme sme si vytvorili 3 stĺpce do ktorých sme vkladali poznámky. Stĺpec dobre je určený na veci, ktoré tím počas šprintu spravil dobre, v čom by mali pokračovať. V stĺpci zle sú veci, ktoré treba zlepšiť v nasledujúcom šprinte. Posledný stĺpec sa nazýva myšlienky. Sem môže každý člen pridať nejaký návrh prípadne poznámku na zlepšenie.

4 Sumarizácia šprintov

Samotný vývoj výsledného produktu, podlieha agilnému vývoju, pričom príbeh je rozdelený na takzvané šprinty. Šprint predstavuje časový úsek konkrétne 2 týždne, pričom je každý šprint vieme rozdeliť na 3 základné časti:

1. Inicializácia(Start) - výber jednotlivých úloh z backlogu.
2. Priebeh - práca na jednotlivých úlohách
3. Uzatvorenie šprintu a retrospektíva - zhodnotenie výsledkov jednotlivých úloh, overovanie výsledkov zo strany product ownera a výsledná bilancia.

4.1 Šprint č.1

Predstavuje úvodný šprint, pri ktorom bolo nevyhnutné identifikovanie zoznamu úloh potrebných pre korektné splnenie požiadaviek zadania, ktoré sme obdržali od product ownera. Na začiatku prvého šprintu boli obodované identifikované základné časti systému, ktoré bude potrebné implementovať. Následne boli jednotlivé úlohy dekomponované na menšie časti, aby ich bolo možné vyriešiť v priebehu jedného šprintu. Ďalej sme uskutočnili výber úloh do prvého šprintu. Úlohy boli vyberané na základe nadväznosti a priorit, ktoré je možné a nutné realizovať v priebehu prvého šprintu a to s prihliadnutím na to, že ide o prvý šprint, t.j. úlohy s menšou náročnosťou.

4.1.1 Úlohy

V prvom šprinte boli zvolené úlohy, aby sa tím zoznámil so technológiami, ktoré sa budú používať v projekte. Boli tam úlohy na:

- Vytvorenie Webovej stránky tímu - webová stránka tímu vytvorená (úloha splnená)
- Vytvorenie metodík - vytvorená iba časť metodík (úloha rozpracovaná)
- Nastavenie nástrojov a procesov - nástroje a procesy v tíme pridané (úloha splnená)
- Zvolenie technológie pre Web Server na BeagleBone Black - technológia zvolená (úloha splnená)
- Načítanie programu do RTU na BeagleBone Black - v tomto šprinte sa zistilo že táto úloha je omnoho náročnejšia a skrývajú sa za tým ďalšie náročné úlohy (úloha rozpracovaná)
- Komunikácia medzi CPU a RTU - kvôli úlohe “Načítanie programu do RTU na BeagleBone Black” sa táto úloha nezačala (úloha zostala v backlogu)
- Analyzovanie dosky - analýza dosky nebola dokončená (úloha rozpracovaná)
- Vytvorenie testov - nepochopená požiadavka Product Ownera (úloha rozpracovaná)

4.1.2 Retrospektíva

V tomto šprinte sa retrospektíva vynechala, ale sa navrhol spôsob akým bude v nasledovných sprintoch prebiehať.

4.2 Šprint č.2

Šprint sa niesol v duchu dokončovania úloh z predošlého prvého šprintu, nakoľko sa v priebehu 1. šprintu vyskytlo množstvo komplikácií a zdržaní napr. s prístupom do systému Jira, ktorý bol poskytnutý od product ownera(Kistler) alebo odhalením komplikáciami s hardvérom od product ownera, ktorý bolo nutné refaktorovať. Vzhľadom k tomu bolo viacero úloh prenesených z určitej časti aj do druhého šprintu.

4.2.1 Úlohy

V druhom šprinte sa pokračovalo na úlohách z prvého šprintu:

- Vytvorenie metodík - nebola spísaná jedna metodika (úloha rozpracovaná)
- Načítanie programu do RTU na BeagleBone Black - v tomto sprinte sa zistilo že okrem problémov z prvého sprintu sa vyskytli aj mnohé ďalšie (úloha rozpracovaná)
- Komunikácia medzi CPU a RTU - analyzovala sa iba akým spôsobom bude prebiehať komunikácia (úloha rozpracovaná)
- Analyzovanie dosky - analýza dosky bola spísaná (úloha splnená)
- Vytvorenie testov - upresnili sa požiadavky Product Ownera pre túto úlohu, ale akceptačné kritéria neboli splnené (úloha rozpracovaná)

4.2.2 Retrospektíva

V tomto šprinte sa zistili problémy s chýbajúcimi úlohami Product Ownera v Jire a limitácie nástroja Jira, nakoľko sa používa pod licenciami spoločnosti Kistler. Tieto limitácie spočívajú v tom, že táto spoločnosť má vlastné nastavenia nástroja a museli sa prispôbiť potrebám tímu v projekte. Product Owner po upozornení už začal pridávať úlohy do Jire.

4.3 Šprint č.3

V 3. šprinte sa zameriava na implementáciu jednotlivých častí IoT systému: 1. BeagleBone, 2. Robot Framework, 3. REST API. Pre jednotlivé časti sme definovali zodpovedné osoby a rozdelili sa do 3 tímov, podľa definovanej obtiažnosti úloh. Hlavným cieľom úloh v 3. sprinte je zabezpečiť vzdialenú komunikáciu so zariadením Como Neo, ktoré slúži na samotné testovanie. Testy budú iniciované systémom Robot framework, pričom daný systém bude testy aj automaticky vyhodnocovať, čiže bude vykonávať akceptačné testy, čo je hlavným dôvodom jeho použitia. Ide o jednoduchý model, pričom sa bude na Como Neo odosielať digitálny signál v hodnote 0/1.

4.3.1 Úlohy

V treťom sprinte sa dokončili úlohy z druhého sprintu a pridali aj nové, čím sa zvýšila produktivita tímu:

- Vytvorenie metodík - metodiky vytvorené (úloha splnená)
- Načítanie programu do RTU na BeagleBone Black - akceptačné kritéria splnené (úloha splnená)
- Komunikácia medzi CPU a RTU - akceptačné kritéria splnené (úloha splnená)
- Vytvorenie testov - akceptačné kritéria splnené (úloha splnená)
- Vytvorenie knižnice v Robot Framework - akceptačné kritéria splnené (úloha splnená)
- REST API prototyp - akceptačné kritéria splnené (úloha splnená)
- Pripravenie dokumentov pre návrh dosky - neboli pripravené všetky dokumenty (úloha rozpracovaná)

4.3.2 Retrospektíva

V tomto sprinte dostal tím pripomienky od Product Ownera súvisiace s prácou na projekte, s cieľom aby sa zvýšila produktivita tímu. Tieto pripomienky boli akceptované a zapracované, čo ako za následok malo zvýšenie produktivity tímu.

4.4 Šprint č.4

Úlohou 4. šprintu je integrácia jednotlivých častí systému, ktoré boli vytvorené v predošliých šprintoch. Po úspešnej integrácii by malo byť možné cez zadané testy v Robot Frameworku odoslať konfiguráciu digitálnych signálov na IoTester, ktoré ich má vygenerovať. Tieto vygenerované digitálne signály sa zobrazia v grafickom rozhraní ComoNeo zariadenia, kde sa za pomoci Robot Frameworku zistí, či boli signály vygenerované správne. Do šprintu pribudol aj nový “epic” pod názvom “ComoNeo Analog Inputs”, kde cieľom je nie len testovať digitálne signály, ale aj analógové. Pre tento účel, boli vytvorené prvé úlohy na návrh rozhraní medzi PRU a ARM, taktiež aj rozhranie medzi Web Serverom na IoTester a Serverom (Robot Framework). Tieto rozhrania majú zabezpečiť odosielanie konfigurácií analógových a digitálnych signálov. Druhá úloha bola pre vygenerovanie jednoduchých analógových signálov zo zariadenia IoTester cez PRU.

4.4.1 Úlohy

Vo štvrtom šprinte najdôležitejšia úloha bola integrácia modulov systému a pribudol aj nový “epic” pre testovanie analógových signálov:

- Vytvorenie dokumentácie pre priebežné odovzdanie - dokumentácia vytvorená a odovzdaná (úloha splnená)
- Program pre PRU - program na vygenerovanie signálov bol vytvorený a funkčný (úloha splnená)
- Integrácia modulov pre digitálny signál - kvôli novému Linuxu na IoTTester nebolo možné inštalovať knižnice pre Web Server. Nepredvídaný problém integrácie Web Servera spôsobilo nesplnenie úlohy pre tento šprint (úloha rozpracovaná)
- Návrh rozhraní pre analógové signály - navrhla sa iba časť rozhraní a kvôli prioritám sa nepokračovalo v tejto úlohe (úloha rozpracovaná)
- Analógový výstup z PRU - analyzoval sa spôsob vygenerovania analógových signálov z PRU a implementácia sa nerealizovala (úloha rozpracovaná)

4.4.2 Retrospektíva

Retrospektívna po štvrtom šprinte bola vynechaná, aby sa ušetril čas na globálnu retrospektívu pre zimný semester na poslednom stretnutí po piatom šprinte.

4.5 Šprint č.5

Tento šprint sa odlišoval od druhých šprintov, tým že bol časovo o polovicu kratší ako ostatné (trval jeden týždeň). Cieľom bolo dokončiť integráciu modulov systému a odprezentovať vlastníčkovi produktu prvý funkčný prototyp systému pre digitálne signály. Zároveň pribudla úloha pre dokumentáciu na zimný semester a boli aj pridané rozpracované úlohy zo šprintu 4.

4.5.1 Úlohy

Najväčšiu prioritu majú medzi úlohami majú integrácia modulov systému pre digitálny signál a dokumentácia pre zimný semester.

- Integrácia modulov pre digitálny signál - integrácia úspešná, vlastník produktu potvrdil funkčnosť počas prezentácii (úloha splnená)
- Dokumentácia pre zimný semester - dokumentácia vytvorená (úloha splnená)
- Návrh rozhraní pre analógové signály - kvôli prioritám v tejto úlohe sa nepokračovalo tento šprint (úloha rozpracovaná)
- Analógový výstup z PRU - kvôli prioritám v tejto úlohe sa nepokračovalo tento šprint (úloha rozpracovaná)

4.5.2 Retrospektíva

Na konci tohto šprintu bola globálna retrospektíva tímového projektu pre zimný semester, ktorá je popísaná v kapitole "Globálna retrospektíva".

5 Globálna retrospektíva

Na poslednom stretnutí v zimnom semestri bola spravená globálna retrospektíva. Cieľom bolo, aby každý člen tímu vyjadril svoje pozitívne a negatívne myšlienky o tímovom projekte v tomto semestri. Prebiehalo to tak, že členovia tímu napísali pozitívne a negatívne postrehy, ktoré boli zoskupene do rôznych kategórií.

Pozitívne:

1. Komunikácia s product ownerom bola na vysokej úrovni. Product owner bol veľmi ústretový a vždy sme sa mohli spoľahnúť na jeho pomoc, keď sme ju potrebovali.
2. Práca na tímových stretnutiach sa postupne zlepšovala. Keď sa tím začal sústrediť na riešenie problémov na stretnutí tak boli aj výsledky.
3. Tím má veľký potenciál, ku koncu semestra sa tímu podarilo splniť všetky úlohy.

Negatívne:

1. Tím využíval nástroj Jira zriedkavo. Chýbalo pridávanie komentárov a písanie progresu ku každej úlohe.
2. Najväčším problémom v tíme je komunikácia. Slack, ktorý je určený na komunikáciu sa veľmi málo používal. Častokrát sa stávalo, že keď sme sa dohodli na tímovom stretnutí alebo hovore, niektorí členovia tímu toto ignorovali alebo sa vyhovárali, že nestíhajú. Dôležité je dopredu dať vedieť takéto veci tímu.
3. Tímové stretnutia sú určené na prácu na tímovom projekte. Na začiatku semestra sa často krát stalo, že sa na stretnutí riešili veci, ktoré nesúvisia s tímovým projektom. Toto sa ku koncu semestra zlepšilo, keď sa začalo svedomitejšie pracovať.
4. Práca na dokumentácií bola slabá. Jedna osoba musí byť zodpovedná za hľadanie potrebných informácií a materiálov, aby ostatní mohli ľahšie písať svoje časti dokumentu.

Návrhy na zlepšenie:

1. Od nasledovného semestra každý člen bude písať progres aj do systému Jira.
2. Slack bude primárny komunikačný kanál. Dôležité je, aby product owner videl že sa reálne pracuje na projekte, aby nám mohol pomôcť, keď budeme mať problém vyriešiť nejakú úlohu.
3. Tím sa bude musieť stretávať aj mimo vyhradeného času na tímové stretnutia. Takto sa zvýši produktivita tímu a lepšia bude aj komunikácia medzi členmi.
4. Dodržiavať dohodnuté termíny. Ak sa člen tímu nemôže zúčastniť stretnutia alebo hovoru, dôležité je aby napísal na čom pracoval a aké výsledky dosiahol.

6 Záver

V doterajšom priebehu projektu môžeme sledovať rastúcu tendenciu výkonnosti tímu, k čomu prispieva hlavne lepšia kooperácia jednotlivých zložiek tímu a v neposlednom rade otvorenejšia a intenzívnejšia komunikácia v rámci tímu. Vo všeobecnosti môžeme povedať, že adaptovanie sa na daný typ vývoju spolu s neočakávanými komplikáciami, spomalilo predbežne očakávaný štart vývoja projektu.

V zimnom semestri sme sa stretli s problémami spojenými so zhoršenou komunikáciou v tíme ako aj s problémami spojenými s vývojovou doskou BeagleBone Black. Prvé dva šprinty tohto semestra neprebehli podľa našich predstáv avšak predposledný a posledný šprint dokázali, že výsledky tímu sa zlepšujú a veríme, že výstupom projektu bude funkčné zariadenie, ktoré bude spĺňať potrebné kritéria a bude akceptované ako tímom tak aj vedúcim projektu a firmou Kistler.

Príloha A - Metodiky

Metodika je popis postupu, ktorý obsahuje základné prvky metodiky kto, čo, komu, prečo a ako. Inak povedané je to postup krokov pomocou ktorých vieme dosiahnuť stanovené výsledok. Tím si vytvoril niekoľko metodík, aby sa práca uľahčila a zefektívnila. Všetky metodiky sú uvedené v časti Príloha A.

Metodika tvorby webovej stránky

Webové sídlo tímového projektu bude spravované výhradne jednou zodpovednou osobou. Správou webového sídla rozumieme:

- Dostupnosť
- Aktuálnosť
- Kvalita

Pre webové sídlo bude uchovávaná história vývoju a bude zálohované cloudovou kópiou. Pre splnenie tejto požiadavky bude využité tímové sídlo Gitu, ku ktorému majú prístup všetci členovia tímu.

Predpokladom tvorby kvalitného webového sídla je, že všetci členovia tímu budú prispievať k tvorbe jeho obsahu a to v podobe dokumentov vo formáte (.pdf). Tieto dokumenty budú všetci členovia pridávať samostatne na spoločný server Gitu, pričom po pridaní nového dokumentu upozornenia správcu webu na nový dokument, ktorý je potreba umiestniť na webové sídlo! Správca webového sídla umiestni tento dokument na webstránke a zverejní ho.

Správca webového sídla priebežne/po väčších úpravách zverejňuje notifikáciu (správa do spoločnej skupiny) o úprave/rozšírení webového sídla. Každý člen tímu má možnosť vyjadriť sa k úprave a navrhnúť prípadné zmeny. Ak prípadné zmeny budú schválené väčšinou tímu, správca webu zapracuje tieto zmeny a opätovne zverejní notifikáciu o zapracovaní pripomienok.

Samotný dokument k tvorbe webového sídla je zhrnutý v ucelenom dokumente, ktorý je zverejnený a dostupný na tímovej stránke:

<http://labss2.fiit.stuba.sk/TeamProject/2018/team15iss-it/documents/web.pdf>

Metodika manažmentu úloh

Projekt je riadený agilnou metódou Scrum a kvôli toho pre podporu manažmentu úloh bol zvolený nástroj Jira. Do tohto nástroja má prístup celý tím, v ktorom sú dôležité

informácie ako úlohy jednotlivých členov tímu. Aby sa medzi tímom zvýšila komunikácia, rovnaká štruktúra úloh a aplikoval Scrum, bola navrhnutá metodika manažmentu úloh.

Počas návrhu tejto metodiky sme sa v prvej verzii dokumentu zamerali na niektoré základné princípy, ktoré predpisuje Scrum a prispôbili požiadavkám nášho tímu. Z pohľadu manažmentu úloh bol zadaný "Definition of Done" za pomoci ktorého sa zadal "Definition of Ready" úloh. "Definition of Ready" bol zadaný tak, aby všetky úlohy mali rovnakú štruktúru a rovnako sa manažovali. V dokumente je popísané aké informácie treba vyplniť pre úlohy a ako ich manažovať, aby tím fungoval ako jeden celok.

Samotný manažment úloh je zhrnutý v ucelenom dokumente, ktorý je zverejnený a dostupný na tímovej stránke:

<http://labss2.fiit.stuba.sk/TeamProject/2018/team15iss-it/documents/man.pdf>

Metodika verziovania kódu

Súčasťou projektu je veľké množstvo zdrojových súborov. Projekt sa bude postupne upravovať a je potrebné udržiavať prehľad nad verziami projektu. Pre tento účel je použitý systém GIT. Všetky zdrojové súbory budú v aktuálnej forme uchovávané v tomto systéme. Pri práci so systémom GIT je potrebné dodržiavať nasledujúce pravidlá:

- Každá úloha musí byť riešená v novej vetve.
- Názov vetvy pozostáva z čísla a názvu konkrétnej úlohy („IOT-[číslo úlohy]_[názov úlohy]“). Pozor nie názov User story ale konkrétnej úlohy.
- Je potrebné „commitovať“ malé zmeny v zdrojovom kóde.
- Pri každom „commite“ musí byť vyplnená „commit message“. Obsah „commit message“:
 - ❖ Názov vetvy.
 - ❖ Zmeny vykonané v „commite“.
- Pred každým „Merge“ vetvy A do vetvy B je potrebné vykonať „Pull“ na vetve A. Nie je potrebný „Pull“ na vetve A, ak vetva A je osobná vetva, na ktorej pracujem len ja.
- Do vetiev BBB, RF, Develop a Master je dovolené „merge-ovať“ len spustiteľný kód.
- Do vetvy Develop sa zmeny z vetiev BBB a RF „merge-ujú“ po tom čo Product Owner schváli User story.

Samotné verziovanie kódu je zhrnuté v ucelenom dokumente, ktorý je zverejnený a dostupný na tímovej stránke:

<http://labss2.fiit.stuba.sk/TeamProject/2018/team15iss-it/documents/git.pdf>