

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Monitorovanie a vyhodnocovanie fyziologických procesov človeka [StressMonitor]

Dokumentácia k riadeniu

Vedúci tímu: Ing. Katarína Jelemenská, PhD.

Členovia tímu: Bc. Martin Nemček, Bc. Oliver Moravčík, Bc. Miloš Pallo,
Bc. Dániel Papp, Bc. Barbora Pavlíková, Bc. Martin Šrank

Školský rok: 2016/2017

OBSAH

1	Úvod.....	1
2	Slovník pojmov.....	2
3	Predstavenie členov tímu a ich role	3
3.1	Bc. Martin Nemček.....	3
3.2	Bc. Oliver Moravčík.....	3
3.3	Bc. Miloš Pallo	3
3.4	Bc. Dániel Papp	3
3.5	Bc. Barbora Pavlíková.....	4
3.6	Bc. Martin Šrank.....	4
4	Používané manažmenty	5
4.1	Manažment komunikácie.....	5
4.1.1	Team Foundation Server (TFS).....	5
4.1.2	Trello	5
4.1.3	Slack	5
4.1.4	Iné sociálne siete (Facebook, Skype)	6
4.1.5	Gmail.....	6
4.1.6	Tímové stretnutia.....	6
4.2	Manažment dokumentácie	6
4.3	Manažment plánovania a rizík.....	7
4.4	Manažment vývoja	8
5	Metodiky	10
5.1	Metodika písania zdrojového kódu.....	10
5.1.1	Pravidlá pre jazyk Python	10
5.1.2	Pravidlá pre jazyk Java.....	11
5.2	Metodika používanie manažovacieho nástroja.....	12
5.2.1	Štruktúra nástroja	12
5.2.2	Označovanie kariet.....	13
5.2.3	Časové odhady a strávený čas	14
5.3	Metodika práce s Git-om a prehliadky zdrojového kódu	14

5.3.1	Pravidlá pre prácu s verziovacím systémom	14
5.3.2	Pravidlá pre vykonávanie prehliadok zdrojového kódu	15
5.4	Metodika tvorby dokumentácie	15
5.4.1	Dokumenty k inžinierskemu dielu a riadeniu tímu	15
5.4.2	Zápisnice zo stretnutí	16
5.4.3	Metodiky	17
6	Sumarizácia šprintov	18
6.1	Sumarizácia šprintu č. 1	18
6.1.1	Opis šprintu a úloh	18
6.1.2	Retrospektíva	18
6.2	Sumarizácia šprintu č. 2	19
6.2.1	Opis šprintu a úloh	19
6.2.2	Retrospektíva	19
6.3	Sumarizácia šprintu č. 3	20
6.3.1	Opis šprintu a úloh	20
6.3.2	Retrospektíva	20
6.4	Sumarizácia šprintu č. 4	21
6.4.1	Opis šprintu a úloh	21
6.4.2	Retrospektíva	22
7	Retrospektíva zimného semestra	23
7.1	Komunikácia	23
7.2	Manažovací nástroj	23
7.3	Agilný vývoj	24
7.4	Zhrnutie	24

ZOZNAM PRÍLOH

PRÍLOHA A - Burdown grafy

PRÍLOHA B - Exporty úloh

1 ÚVOD

Hlavným cieľom predmetu tímový projekt je vyskúšať si prácu v tíme a naučiť sa tímovo myslieť. Počas dvoch semestrov študenti agilne vyvíjajú softvér. Výstupom predmetu teda nie je len samotná aplikácia, ale aj záznamy z testovania a dokumentácia. Tím postupuje podľa agilnej techniky Scrum a pracuje v dvojtýždňových šprintoch, ktoré začínajú plánovaním a končia retrospektívou.

Tento dokument mapuje postup študentov pri práci v tíme. Okrem opisu jednotlivých členov tímu tu nájdeme aj špecifikáciu pridelených rolí, spôsoby komunikácie v tíme a metodiky, ktoré tím postupne zavádzal do svojho fungovania.

Samostatná kapitola tohto dokumentu je venovaná sumarizácii jednotlivých šprintov a globálnej retrospektíve celého semestra.

2 SLOVNÍK POJMOV

Pojem	Vysvetlenie
Obslužná úloha	Z angl. Utility Task – úloha, ktorá nie je zaradená pod konkrétny Epic. Týka sa väčšinou správneho chodu tímu, napr.: Vytvorenie metodiky pre prácu s gitom
Frontend aplikácie	
Backend aplikácie	

3 PREDSTAVENIE ČLENOV TÍMU A ICH ROLE

Identifikácia jednotlivých rolí a následné priradenie členom tímu prebehla počas prvých dvoch týždňoch semestra.

3.1 Bc. Martin Nemček

Absolvent bakalárskeho štúdia na Fakulte informatiky a informačných technológií v Bratislave (FIIT). Pre tím sú veľkým prínosom najmä jeho pracovné skúsenosti. V praxi sa stretol s vývojom väčších projektov, pričom mal na starosť databázovú časť systému. Nakoľko pracuje v malom tíme, často musí zastávať viacero úloh, napr. aj komunikáciu so zákazníkom. Ovláda viacero programovacích jazykov, pričom najväčšie skúsenosti má s jazykom C#.

Rola v tíme: databázový špecialista – vyplýva z jeho skúseností a najmä z potreby udržiavať databázu v konzistentnom stave. Vzhľadom na množstvo dát tiež potrebujeme efektívne vykonávanie dopytov. Martin sa okrem iného veľkou časťou podieľal aj na programovaní serverovej časti aplikácie.

3.2 Bc. Oliver Moravčík

Bakalárske štúdium absolvoval taktiež na FIIT, pričom vo svojej bakalárskej práci sa venoval, predikcii vyrobenej elektriny z obnoviteľných zdrojov na základe predošlých nameraných dát. Preto je Oliverova skúsenosť so strojovým učením významným doplnením nášho tímu. Okrem toho je veľmi vhodný jeho zápal pre štúdium, ktorý využil pri oboznamovaní sa s biznis logikou v doméne merania stresu z odporu kože.

Rola v tíme: Product Owner – bola pridelená Oliverovi pre jeho chuť oboznamovať sa s problematikou, a tiež mal možnosť využiť kreativitu pri vytváraní produktového backlogu. Okrem toho sa tiež podieľal na programovaní mobilnej aplikácie pre koncového používateľa a tvorení tímovej dokumentácie.

3.3 Bc. Miloš Pallo

Taktiež absolvent bakalárskeho štúdia na FIIT. Jeho bakalárska práca je z medicínsko-informatickej oblasti, čo znamená, že v tejto problematike vie prispieť určitými vedomosťami. Skúsenosťou s Android aplikáciami a programovacím jazykom Java výrazne prispieva do developer-skej časti tímu.

Rola v tíme: Java programátor – skúsenosti s Androidom v kombinácii s jazykom Java Miloš výrazne využil pri tvorení mobilnej aplikácie. Taktiež sa najmä v začiatkoch projektu podieľal na skúmaní problematiky určovania stresových situácií na základe odporu kože.

3.4 Bc. Dániel Papp

Bakalárske štúdium absolvoval na FIIT. V rámci súkromných projektov navrhol niekoľko používateľských rozhraní a svoje skúsenosti s návrhom uplatnil v našom tímovom projekte.

Predstavenie členov tímu a ich role

Okrem návrhu rozhraní má skúsenosti aj s tvorbou webových stránok, pričom sa primárne zameriava na HTML a CSS.

Rola v tíme: UX dizajnér – Danovou úlohou bolo najmä navrhnuť a tiež zabezpečiť dizajn používateľských rozhraní. Okrem toho je Dano tvorcom a zároveň správcom webovej stránky tímu. Taktiež má podiel na tvorbe backendovej časti mobilnej aplikácie.

3.5 Bc. Barbora Pavlíková

Bakalársky titul získala na FIIT. Významné pre tím sú najmä jej skúsenosti s agilným vývojom, ktoré získala v praxi. Okrem toho ovláda programovanie najmä v jazyku Java a skúsenosti má so skriptovacím jazykom Javascript a tvorbou webu (HTML, CSS).

Rola v tíme: Scrum Master – vyplýva najmä zo skúsenosti s agilným vývojom a Scrumom. Úlohou Scrum Mastera je dohliadať na dodržiavanie agility, podporovať a zabezpečovať komunikáciu v tíme. Barbora taktiež zodpovedala za dokumentovanie a správne fungovanie celého tímu. Z programovacích úloh sa podieľala na tvorbe mobilnej aplikácie.

3.6 Bc. Martin Šrank

Taktiež absolvent bakalárskeho štúdia na FIIT, pričom za svoju prácu Automatizovaná distribúcia a testovanie softvérových balíkov získal pochvalný list dekanke za vynikajúco vypracovanú bakalársku prácu. Martin má niekoľkoročné skúsenosti s programovaním najmä webových aplikácií. Na pokročilej úrovni ovláda HTML, CSS, Python a PHP, pričom je veľmi všestranný programátor a vie veľmi rýchlo uchopiť akýkoľvek problém.

Rola v tíme: Programátor s orientáciou na server – vyplynulo z Martinových dlhoročných programovacích skúseností. Okrem zodpovednosti za serverovú časť aplikácie tiež dohliadal na správne používanie verziovacieho nástroja Git.

4 POUŽÍVANÉ MANAŽMENTY

Pre správne fungovanie v tíme a udržiavania práce na projekte sme aplikovali niekoľko manažmentov, ktoré sú bližšie opísané v tejto kapitole. Takisto sme v rámci nich zaviedli viacero metódik, ktoré nájdeme v kapitole 5.

4.1 Manažment komunikácie

Pri práci v tíme je komunikácia jednou z najdôležitejších vecí. Na komunikáciu sme vyskúšali viacero rôznych kanálov a nástrojov. V tejto podkapitole ich bližšie opíšeme, akým spôsobom sme ich používali, a či sa pri našej práci osvedčili alebo nie.

4.1.1 Team Foundation Server (TFS)

Je systém, ktorý poskytuje sadu nástrojov pre spoluprácu pri vývoji softvéru. Hlavným cieľom používania TFS bolo vytváranie úloh, s ktorým sme následne pracovali podľa agilného prístupu. Tento nástroj poskytoval výhodu v integrácii s ďalšími systémami ako Git alebo Slack. Taktiež plne podporoval všetky potreby pre agilný vývoj, aj následné retrospektívy (generovanie grafov a reportov).

V prvých týždňoch práce s TFS bol tím značne nespokojný. Systém síce podporoval všetky potrebné nástroje, no práca s ním nám pripadala zbytočne zložitá a ťažkopádna. Takisto pri tvorení produktového backlogu sme boli striktne obmedzený týmto nástrojom, pričom sme museli z každého Epic vytvoriť Feature a z nej User Story, čo bolo nepraktické najmä pri potrebe úloh, ktoré neboli zaradené pod konkrétny Epic, ale patrili pod tzv. obslužné úlohy.

4.1.2 Trello

Ako druhý manažovací nástroj sme si vybrali Trello. Dôvodom bola jeho jednoduchosť. Čo sa týka agility, rozšírenia do prehliadača pokryli všetky naše požiadavky. Zobrazovanie grafov a generovanie reportov poskytovalo veľa možností a boli jednoduché na prevedenie. Okrem toho je výhodou tohto nástroja, že umožňuje priradiť jednu úlohu viacerým členom tímu.

Nakoľko Trello sa stalo základným komunikačným nástrojom pre informácie o tok, kto akú úlohu splnil, bolo nutné zaviesť metodiku na prácu s manažovacím nástrojom, ktorej sa bližšie venujeme v podkapitole.

4.1.3 Slack

Taktiež bolo nutné zaviesť neformálnu komunikáciu. Zvolili sme si jeden z najpoužívanejších nástrojov na komunikáciu v tíme – Slack. Tento nástroj sa osvedčil a absolútne pokryl naše potreby tímovej komunikácie. Počas celej tvorby systému sme vytvárali potrebné kanály, kde sme združovali správy s rovnakou tematikou. Nepotrebné kanály sme následne archivovali. Taktiež sme vytvorili niekoľko permanentných kanálov:

- General – správy typu „rozličné“, ktoré nebolo možné zaradiť do iného kanálu.

Používané manažmenty

- Stretnutia – správy týkajúce sa pravidelných tímových stretnutí. Tu sa konzultoval plán na stretnutia, zápisnice zo stretnutí a i.
- Stand-up – nakoľko tím považoval za nutné podávať informácie o aktuálnej práci častejšie ako raz za týždeň na osobných stretnutiach, rozhodli sme sa zaviesť „virtuálny“ stand-up. Každý štvrtok večer sa členovia tímu stretli na Slack-u a napísali, čo urobili, na čom pracujú, prípadne čo im bráni v práci, a aké majú problémy. Keďže sa stávalo, že tím na tento termín zabúdalo, využili sme výhodu tohto nástroja a vytvorili sme si automatického robota, ktorý nám pripomínal stretnutie v dátume, ktorý sme mu nastavili.

4.1.4 Iné sociálne siete (Facebook, Skype)

Ďalším komunikačným nástrojom, ktorý sme v tíme vyskúšali bol Facebook prostredníctvom skupinovej konverzácie. Správy sa tu ale nedali zatriediť podľa tematiky, a tiež bolo náročné ich dohľadať. Tento spôsob komunikácie sa teda neosvedčil a využívali sme ho iba do druhého týždňa semestra, kým sme nezriadili konto na Slack-u.

V prípade, že členovia tímu potrebovali komunikovať a nebolo možné sa v danej chvíli stretnúť, častým riešením bolo telefonovanie prostredníctvom služby Skype. Tieto hovory sme využívali taktiež na konzultáciu používaných agilných techník s externým človekom z praxe.

4.1.5 Gmail

Keďže sme potrebovali komunikovať aj formálne, zriadili sme si tímovú e-mailovú adresu. Túto adresu sme využívali najmä na komunikáciu s vedúcou tímu. Všetky správy sa automaticky preposielali každému členovi tímu. Aby sa predišlo komunikačným nedorozumeniam, zodpovedný člen tímu za toto e-mailové konto bola Barbora Pavlíková.

4.1.6 Tímové stretnutia

Nevyhnutné pre správne fungovanie tímu boli osobné stretnutia. Podľa zvoleného rozvrhu sa konali pravidelne v pondelok v čase cca od 14:00 do 17:00. V polovici semestra bolo nutné stretnutia presunúť na štvrtok kvôli zmenám v rozvrhu vedúcej tímu. Z každého stretnutia vznikla zápisnica, ktorá obsahovala úlohy vyplývajúce z diskusií a tiež členov tímu, ktorí boli za ne zodpovední.

Okrem pravidelných stretnutí sme sa niekoľkokrát stretli aj mimo dohodnutého času. Bolo to z rôznych dôvodov, najmä na začiatku práce, keď sme sa navzájom oboznamovali s rôznymi konvenciami a metodikami.

4.2 Manažment dokumentácie

Počas celej tvorby projektu vznikalo viacero druhov dokumentácií. Dve najväčšie dokumentačné diela sú Dokumentácia k riadeniu a Dokumentácia inžinierskeho diela. Taktiež z každého

stretnutia vznikla zápisnica. Metodiky, ktoré tím pri tvorbe projektu používal boli pre prehľadnosť a ľahkú dostupnosť spísané do samostatných dokumentov a sprístupnené na webovej stránke tímu.

Nakoľko dokumentov týkajúcich sa aplikácie je niekoľko a na ich tvorbe sa podieľali všetci členovia tímu, bolo potrebné túto tvorbu manažovať. Úlohou manažéra dokumentácie je navrhnuť šablónu dokumentov a pravidiel, podľa ktorých sa bude postupovať v záujme udržania konzistencie jednotlivých dokumentov. Jednotlivé pravidlá sú bližšie opísané v Metodike tvorby dokumentácie (5.4). Manažér je taktiež zodpovedný za dodržiavanie jednotlivých pravidiel a revíziu dokumentov.

4.3 Manažment plánovania a rizík

Pre neustále napredovanie projektu je veľmi dôležité plánovanie. Pri našej práci sme z časového hľadiska rozlišovali dva druhy plánovania – dlhodobé a krátkodobé.

Dlhodobé plánovanie sa začalo na začiatku tvorby projektu. Najskôr bolo nutné si ujasniť predstavy vlastníka produktu o výsledku našej práce. Po vyslovení vlastníkových požiadaviek sme pomocou Epic-ov rozdelili projekt na menšie časti. Na jednom Epic-u sa potom pracovalo aj niekoľko šprintov. Dlhodobý plán sa postupne menil podľa jednotlivých výsledkov v šprintoch, a tiež podľa meniacich sa požiadaviek na výsledný produkt.

Krátkodobé plánovanie sa udialo každý druhý týždeň, vždy na konci šprintu. Aktívne sa ho na spoločných stretnutiach zúčastňoval celý tím aj s jeho vedúcou. Priebeh plánovania usmerňoval aktuálny Scrum Master spolu s vedúcou tímu. Pri tomto druhu plánovania sa museli najskôr zohľadniť nesplnené úlohy z predchádzajúceho šprintu. Tie mali v nasledujúcom šprinte najvyššiu prioritu. Taktiež bolo nutné uvažovať aktuálnu kapacitu tímu, ktorá sa menila najmä podľa aktuálneho vyťaženia jednotlivých členov na iných predmetoch. Podľa času, ktorý zostal nevyužitý, sme do šprintového backlogu pridávali ďalšie User Story. V prípade, že sa počas šprintu identifikoval nesprávny odhad na niektorého člena, jeho úlohy boli rozdelené medzi menej vyťažených členov. Taktiež sa muselo promptne konať v prípade, že niektorá z úloh musela byť vykonaná prednostne. Vtedy tiež dochádzalo k zámenám úloh tak, aby žiadny iný člen tímu nemusel byť vo svojej práci blokovaný.

Spolu s plánovaním úzko súvisia aj riziká projektu. V Tab. č. 1 sú stručne a prehľadne spísané riziká, ktoré sme identifikovali pro tvorbe nášho projektu. Pre každé riziko sme určili následku, ktoré budú mať dopad na projekt, ak sa dané riziko stane reálnym. Aby identifikácia rizík mala svoj zmysel, bolo nutné pre každé tiež určiť spôsob prevencie, ktorým sa danému riziku s najväčšou pravdepodobnosťou vyhneme.

Riziko	Následky	Prevencia
Plánovanie malej kapacity do šprintu	Pomalé napredovanie, oneskorenie termínu odovzdania aplikácie	Plánovanie v tíme, nie jednotlivito, porovnávanie s predchádzajúcimi šprintami, používanie Trela
Plánovanie priveľkej kapacity do šprintu	Klient bude očakávať stav aplikácie, ktorý nie sme schopný dodať	Plánovanie v tíme, lepšie odhadovanie kapacity šprintu, používanie Trela
Nedostatočná komunikácia	Jednotliví členovia tímu nevedia o svojej práci, chaos v projekte	Používanie komunikačných nástrojov Slack, Skype, aktívna účasť na tímových stretnutiach
Nedostatočná prioritizácia	Nepodstatné úlohy budú splnené a tie podstatné nebudú	Komunikácia s vlastníkom produktu, dobrá znalosť biznisovej logiky
Nedodržovanie metodiky tvorby kódu	Neudržiavaný, chaotický kód, ťažké udržiavanie projektu aj jeho rozvoj	Prehliadky kódu podľa pravidiel
Nedostatočné alebo iba malé testovanie	Aplikácia pri odovzdaní nebude správne fungovať a vlastník produktu ju nemusí akceptovať	Vytvorenie testovacích scénárov, unit testy pre menšie časti projektu
Riziko nesplnenia požiadaviek	Aplikácia nebude vykonávať požadovanú funkcionality a vlastník produktu ju nemusí akceptovať	Dôkladné testovanie biznisovej logiky a čiastkové akceptovanie jednotlivých častí produktu počas celej tvorby systému

Tab. č. 1: Riziká identifikované v našom projekte

4.4 Manažment vývoja

Celý proces tvorby softvérového produktu vieme rozdeliť do dvoch rozsiahlych častí – vývoj (návrh, implementácia, testovanie...) a dokumentovanie (dokumentácia k riadeniu, dokumentácia k inžinierskemu dielu, technická príručka...).

Používané manažmenty

Kvôli dôležitosti samotného vývoja je nutné túto činnosť dôkladne manažovať. Za manažovanie vývoju v našom projekte zodpovedá Bc. Martin Šrank. Úlohou manažéra vývoja je dohliadať na implementáciu softvéru a činnosti s ňou súvisiace. To zahŕňa tiež vytvorenie Metodiky písania zdrojového kódu (Metodika písania zdrojového kódu 5.1) a Metodiky práce s Git-om a prehliadky kódu (5.3). Na oboch metodikách sa aktívne podieľali aj zvyšní členovia tímu v podobe revízie a tímovej diskusie na stretnutí.

Okrem tvorby metodík je úlohou manažéra vývoja tiež dohliadať na dodržiavanie týchto pravidiel. Nakoľko sme sa ako tím rozhodli, že prehliadky kódu si budeme robiť navzájom a budú do nich zainteresovaní všetci členovia tímu, manažér dohliada najmä na dodržiavanie Metodiky práce s Git-om.

5 METODIKY

V tejto kapitole sú spísané metodiky používané počas celej práce na tímovom projekte. Za dodržiavanie jednotlivých metodík sú zodpovední manažéri. Na tvorbe každej metodiky mali členovia tímu nasledujúci podiel:

- *Metodika písania zdrojového kódu* – Bc. Martin Šrank (Python), Bc. Martin Nemček (Java),
- *Metodika používania manažovacieho nástroja* – Bc. Oliver Moravčík,
- *Metodika práce s gitom a prehliadky zdrojového kódu* – Bc. Martin Šrank,
- *Metodika tvorby dokumentácie* – Bc. Barbora Pavlíková.

Po spísaní jednotlivých pravidiel každá metodika prešla tímovou diskusiou a po zapracovaní pripomienok ju tím schválil. Všetky metodiky sú tiež ako samostatné dokumenty dostupné na webovej stránke tímu.

5.1 Metodika písania zdrojového kódu

Zdrojový kód aplikácie je výsledným spoločným dielom celého tímu. Metodika písania zdrojového kódu obsahuje pravidlá, podľa ktorých by sa mali developeri pri tvorbe kódu riadiť. Celý náš systém sa skladá z dvoch častí: server a mobilná aplikácia. Nakoľko server je tvorený a obsluhovaný v jazyku Python a mobilná aplikácia v jazyku Java pravidlá pre obe časti sme museli zdefinovať zvlášť. Dodržiavanie jednotlivých pravidiel je kontrolované v rámci prehliadok kódu, ktoré sa taktiež riadia pravidlami spísanými v časti Pravidlá pre vykonávanie prehliadok zdrojového kódu 5.3.2.

5.1.1 Pravidlá pre jazyk Python

1. Používať výhradne verziu *Python 3*.
2. Súbory kódovať v UTF-8.
3. Používať štyri medzery pre *indentation level*.
4. Dĺžka riadku by nemala presiahnuť 80 znakov, nesmie presiahnuť 120 znakov.
5. Importovanie modulov písať v oddelených riadkoch.
6. Definície funkcií oddeľovať dvoma prázdnyimi riadkami.
7. Pri definovaní reťazca znakov prioritne používať úvodzovky, apostrofy iba ak si to situácia vyžaduje.
8. Dokumentačné komentáre uzavrieť tromi párami úvodzoviek.
9. Každá metóda musí mať dokumentačný komentár.
10. Moduly pomenovávať malými písmenami.
11. Triedy pomenovávať vo formáte *UpperCamelCase*.
12. Premenné, metódy a funkcie pomenovávať malými písmenami a slová oddeľovať podčiarkovníkmi.
13. V definícii argumentov inštančných metód je argument *self* vždy na prvej pozícii.

14. V definícii argumentov statických metód je argument *cls* vždy na prvej pozícii.
15. Konštanty pomenovávať veľkými písmenami a slová oddeľovať podčiarkovníkmi.

5.1.2 Pravidlá pre jazyk Java

1. Všetky názvy premenných, tried aj všetky komentáre písať výhradne v anglickom jazyku.
2. Triedy pomenovávať vo formáte *UpperCamelCase*.
3. Metódy pomenovávať vo formáte *lowerCamelCase*.
4. Polia triedy definovať na vrchu definície triedy a dodržiavať pravidlá pre ich pomenovanie:
 - i. Polia pomenovať vo formáte *lowerCamelCase*.
 - ii. Polia s modifikátormi prístupu *private*, *non-static*, pomenovať s písmenom *m* na začiatku.
 - iii. Polia s modifikátormi prístupu *private*, *static*, pomenovať s písmenom *s* na začiatku.
 - iv. Polia s modifikátormi prístupu *final*, *static*, (čiže konštanty) pomenovať veľkými písmenami s podčiarkovníkom pre oddelenie slov (*ALL_CAPS_WITH_UNDERSCORES*).
5. Vždy špecifikovať modifikátor prístupu.
6. Pri pomenovaní písať akronymy podľa vyššie definovaných formátov (*XmlHttpRequest*).
7. Bloky kódu odsadzovať na začiatku riadku štyrmi medzerami.
8. Presahujúce riadky odsadzovať na začiatku nového riadku ôsmimi medzerami.
9. Otváracie zátvorky v definícii tried, metód a blokov kódu písať na konci daného riadku.
10. Časti tried písať v poradí :
 - i. Konštanty
 - ii. Polia
 - iii. Konštruktory
 - iv. “Override” metódy a “callbacks”
 - v. Verejné metódy
 - vi. Public metódy
 - vii. Private metódy
 - viii. Vnorené triedy a rozhrania
11. V parametroch funkcie definovať *Context* na prvej pozícii a *Callback* na poslednej pozícii.
12. Dĺžka riadku by nemala presiahnuť 100 znakov.
13. Komentáre písať vo formáte *Javadoc*.

5.2 Metodika používanie manažovacieho nástroja

Pri práci v tíme je nutné úlohy jednotlivých členov udržiavať na jednom mieste. Nakoľko pracujeme agilne, aj manažovací nástroj a metodiku na jeho používanie sme museli prispôbiť týmto princípom. Nasledujúca kapitola obsahuje pravidlá a princípy pre prácu s manažovacím nástrojom. Na dodržiavanie týchto pravidiel dohliada Scrum Master.

Po spustení nástroja Trello sa zobrazí hlavná obrazovka. Budeme mať len jednu hlavnú obrazovku, takže nemusíme riešiť na každý šprint kopírovanie produktového backlogu, vytváranie listov, pomenovávanie farebných labelov. Budeme mať všetko na jednom mieste a ľahko dostupné v prípade potreby dohľadania úloh v histórii. Grafy tak môžeme pozeráť na jednej obrazovke z akéhokoľvek časového úseku.

5.2.1 Štruktúra nástroja

Hlavná obrazovka obsahuje zvislé zoznamy. Zvislé zoznamy zoskupujú úlohy vytvorené podľa agilných princípov. Tieto úlohy sú reprezentované kartami.

Bude 6 základných, vždy aktuálnych zoznamov: Product Backlog, Sprint Backlog, In progress, Review, Test, Closed

Zoznam **Product Backlog** obsahuje Epics a User Stories, ktoré celkovo plánujeme zapracovať do produktu (aj v ďalekej budúcnosti).

Do zoznamu **Sprint Backlog** dávame na začiatku šprintu na spoločnom stretnutí User Stories zo zoznamu Product Backlog. Jednotlivé Stories rozdelíme na menšie úlohy Tasky. V prípade, že na konci šprintu zostanú nesplnené úlohy, zostanú v zozname Sprint Backlog. Všetky Tasky zostávajú v tomto zozname, kým na nich niektorí z členov nezačne pracovať. Na plánovaní spoločne Taskom priradíme členov tímu, ktorí zaň zodpovedajú. Člen tímu, ktorý má na zodpovednosti najviac Taskov z určitej Story sa zároveň stáva zodpovedným aj za celú Story. Zodpovedný člen si následne zvolí časový odhad náročnosti každého Tasku.

Do zoznamu **In progress** presunie úlohu člen tímu, keď na nej začne pracovať. Podľa náročnosti úlohy, buď priebežne alebo po jej ukončení člen tímu zaznamená reálny čas strávený touto úlohou.

Do zoznamu **Review** presunie úlohu člen tímu, keď je hotová a pripravená na prehliadku zdrojového kódu. Úloha v tomto zozname zostáva ž kým nie je schválená a pripravená na test funkčnosti.

Do zoznamu **Test** úlohu presunie člen tímu po vykonaní prehliadky kódu. Úloha v tomto zozname zostáva, kým ju testujúci člen tímu neposúdi ako splnenú.

Metodiky

Do zoznamu **Closed** úlohu presunie developer, ktorý ju implementoval. Presunutím úlohy vyjadruje, že je nie len zapracovaná, ale aj absolvovala prehliadku kódu, bola riadne otestovaná a posunutá do správnej vetvy vo verziovacom systéme.

Na konci každého šprintu sa aktuálny zoznam Closed uzavrie a premenuje na Closed Sprint XY, kde XY vyjadruje číslo končiacieho sa šprintu. Následne sa vytvorí nový, prázdny zoznam Closed. Stories sa považujú za uzatvorené, keď všetky príslušné Tasky sú uzatvorené. Uzatvorená story sa nachádza v zozname uzatvoreného šprintu, ktorý obsahuje posledný uzatvorený Task z tejto Story,

5.2.2 Označovanie kariet

Zoznamy sú tvorené kartičkami. Každá kartička predstavuje úlohu definovanú podľa agilných princípov.

Druh úlohy je rozlíšený farbou a popisom (Obr. č. 1). Názov Epicu sa začína označením EX, kde X je poradové číslo v zozname Product Backlog. Názvy Stories sa začínajú označením EX_SY, kde X je poradové číslo Epicu, ku ktorému patria a Y je poradové číslo danej Story v Epicu. Rovnakým označením sú tie označené všetky Tasky prislúchajúce k danej Story. Úlohu označenú ako Team Utility nie je možné zaradiť pod konkrétnu Story, a preto označenie nepotrebuje. Team Utility sú dokumentačné úlohy a úlohy potrebné pre správny chod tímu.



Obr. č. 1: Farebné označenie jednotlivých druhov úloh

Každý úlohe je možné priradiť Tag. V zozname Product backlog používame Tagy na vyjadrenie priority jednotlivých úloh. Pre naše potreby sme si zadefinovali nasledujúce priority:

- Critical – aplikácia danú funkcionálnosť musí obsahovať a je nutné ju zapracovať čo najskôr.
- Needed – aplikácia danú funkcionálnosť musí obsahovať, ale má nižšiu prioritu.
- Wanted – aplikácia danú funkcionálnosť nemusí obsahovať, prichádza na rad, keď sú zapracované všetky nutné časti.
- Nice to have – úlohy s úplne najnižšou prioritou.

Keď už je úloha ako Task zaradená do šprintu, Tagy vyjadrujú, či sa daná úloha týka mobilnej aplikácie (`#andriod`) alebo serverovej časti (`#server`). Nástroj podporuje vytváranie vlastných Tagov.

5.2.3 Časové odhady a strávený čas

Na plánovaní, resp. pri vytvorení úlohy sa zadá časový odhad jej zapracovania do poľa označeného ako E (estimate). Odhadujú sa iba Tasky, Buggy a Team Utility. Tento časový odhad sa už ďalej nemení. Po ukončení práce na úlohe (aj čiastočnej), člen tímu zaznamená, koľko času na nej strávil do poľa označeného ako S (spent). V prípade, ak to člen považuje za vhodné, pridá k danej úlohe komentár.

5.3 Metodika práce s Git-om a prehliadky zdrojového kódu

Pri tvorení zdrojového kódu je veľmi dôležité verziovanie. Nakoľko sa na tvorbe systému podieľa celý tím, bolo nutné ustanoviť pravidlá pre prácu s verziovacím systémom. Na dodržiavanie týchto pravidiel dohliada manažér vývoja Bc. Martin Šrank.

Zvolený verziovací nástroj Github podporuje okrem iného praktické prehliadky zdrojového kódu, preto sme sa ho rozhodli využiť aj na tieto účely. V tejto podkapitole sú spísané pravidlá pre prácu s verziovacím systémom, a tiež pravidlá pre vykonávanie prehliadok zdrojového kódu.

5.3.1 Pravidlá pre prácu s verziovacím systémom

1. Vždy na začiatku práce na konkrétnej úlohe aktualizovať lokálnu verziu základnej vetvy (*master branch*) s kódom.
2. Z aktuálneho stavu základnej vetvy odvodiť novú vetvu (*branch*) pre danú úlohu, do ktorej bude úloha zapracovaná.
3. Každú vetvu výstižne pomenovať podľa pridanej funkcionality s predponou “dev-”.
4. Prírastky na danej úlohe pridávať (*commit*) len do danej vetvy, nikdy nie do základnej vetvy.
5. Pridávať *commit* primerane často.
6. *Commit* vždy doplniť stručným a výstižným komentárom, ktorý obsahuje definovanie oblasti, ktorej sa *commit* týka, nasledované popisom prírastku alebo zmeny.
7. Po pridaní podstatnej časti danej úlohy do vetvy, potlačiť vetvu (*push*) na GitHub a otvoriť k nej *Pull Request*. Pridať odkaz na *Pull Request* do kartičky s úlohou v nástroji Trello.
8. Pokračovať s pridávaním *commit*-ov, kým nie je úloha dokončená.
9. Vykonať prehliadku kódu.
10. Po doplnení zmien na základe prehliadky kódu spojiť (*merge*) *Pull Request* s *Master Branch*.

11. Ak nie je možné spojiť vetvy automaticky, vyriešiť konflikty vo vetve ručne, spojiť vetvy lokálne a následne vykonať *push* na GitHub.
12. Nikdy nepoužívať *force push*.
13. Pridávať prírastky na úlohe vykonaním *commit* radšej často a po malých častiach než menej často a po veľkých častiach.
14. Komentáre pri *commit*, názvy vetiev a iné písať vždy v anglickom jazyku.
15. Vytvárať *Pull Request* čo najksôr, a pravidelne dopĺňať zmeny, aby bol kód úlohy prístupný na prehliadku kódu.
16. Spojenú vetvu z GitHubu vymazať.

5.3.2 Pravidlá pre vykonávanie prehliadok zdrojového kódu

1. Prehliadku kódu danej úlohy prioritne vykonáva člen tímu, ktorý nie je autorom daného kódu a má dostatočné skúsenosti pre posúdenie kódu.
2. Pre vykonanie prehliadky kódu používať GitHub, kde je kód prístupný a umožňuje pridávať komentáre k riadkom kódu.
3. K nájdenému problému v kóde pridať komentár, ktorého obsahom je výstižný popis problému alebo spresňujúca otázka.
4. Pri prehliadke kódu sa zameriavať na nasledujúce:
 - i. Dodržiavanie stanovených konvencií písania zdrojového kódu.
 - ii. Logická a implementačná správnosť riešenia danej úlohy.
 - iii. Pamäťová a výpočtová efektívnosť riešenia danej úlohy.
5. Komunikovať s autorom zdrojového kódu úlohy ohľadne nájdených problémov, snaha o nájdenie riešenia problémov.
6. Kontrolovať zmeny v zdrojovom kóde vykonané na základe prehliadky kódu.
7. Informovať autora kódu o začatí prehliadky kódu aj o jej ukončení a finálnom schválení zdrojového kódu.

5.4 Metodika tvorby dokumentácie

Ako samotný systém tak aj práca v tíme vyžadujú dokumentáciu. Počas celej tvorby aplikácie vzniklo viacero dokumentov, ktoré boli tvorené viacerými členmi tímu. Aby pri dokumentovaní nevznikal zmätok, zaviedlo sa niekoľko šablón a pravidiel, podľa ktorých sa jednotlivé dokumenty tvorili. Zodpovednosť za dodržiavanie pravidiel dokumentovania prebrala Bc. Barbora Pavlíková. V tejto podkapitole sú zhrnuté pravidlá pre písanie dokumentov, ktoré boli rozdelené do troch skupín podľa použitých šablón: dokumenty k inžinierskemu dielu a riadeniu tímu, zápisnice zo stretnutí a metodiky.

5.4.1 Dokumenty k inžinierskemu dielu a riadeniu tímu

Z práce v tíme a tvorbe systému vzišli dva rozsiahle dokumenty: dokumentácia k riadeniu a dokumentácia k inžinierskemu dielu. Oba dokumenty majú podobnú štruktúru, a preto sa tím rozhodol, že pre ich tvorbu bude použitá taktiež rovnaká šablóna.

Oba dokumenty obsahujú titulný list a štruktúrovaný obsah dokumentu. Ďalej nasleduje úvod, slovník pojmov a zvyšné kapitoly, ktoré obsahovo patria do jednotlivých dokumentácií. Každý dokument sa končí záverom, po ktorom sú zaradené príslušné prílohy.

Každá kapitola je očíslovaná arabským číslovaním, pričom číslujeme maximálne do tretej úrovne. V prípade, že je nutné rozčleniť aj tretiu úroveň, použijeme iba jednoduchý nadpis, formátovaný tučným rezom písma – bez číslovania.

7 NADPIS PRVEJ ÚROVNE

7.1 Nadpis druhej úrovne

7.1.1 Nadpis tretej úrovne

Obr. č. 2: Štýly nadpisov kapitol v rôznych úrovniach

V celom dokumente je použité riadkovanie veľkosti 1,3 násobku riadku pre uľahčenie čítania a sprehľadnenie textu. Celý text je zarovnaný do bloku. Všetky tabuľky použité v dokumente sú rozťahnuté na celú šírku strany a majú jednotné formátovanie zobrazené na Tab. č. 2.

Prvý stĺpec	Druhý stĺpec
Obsah prvého stĺpca	Obsah druhého stĺpca
Druhý riadok	Ďalší stĺpec
Tretí riadok	Ešte ďalší riadok

Tab. č. 2: Vzor formátovania tabuliek

Každý obrázok, tabuľka, či graf sú očíslované, zarovnané na stred a vo svojej spodnej časti obsahujú popis toho, čo znázorňujú, taktiež zarovnaný na stred.

5.4.2 Zápisnice zo stretnutí

Zápisnice zo stretnutí vznikajú pravidelne každý týždeň a sú uverejňované na webovej stránke tímu. Pre lepšiu orientáciu a prehľadnosť bola zavedená rovnaká šablóna pre všetky zápisnice. Každá strana zápisnice má hlavičku obsahujúcu logo tímu a základné informácie o ňom (Obr. č. 3).

Obr. č. 3: Hlavička zápisnice

Pod nadpisom nasleduje miesto, dátum a čas konania a zoznam jednotlivých účastníkov stretnutia. Ďalšími povinnými časťami sú program, priebeh stretnutia a úlohy vyplývajúce zo stretnutia.

Program obsahuje body, ktoré sa účastníci rozhodli prediskutovať. Definovanie týchto bodov má na starosti Scrum Master. Ten dostatočne včas pred stretnutím zozbiera témy od všetkých členov tímu a následne ich zaradí do programu, aby za sebou logicky nasledovali.

Priebeh stretnutia stručne opisuje diskusie jednotlivých bodov, prípadne zachytáva nové poznatky, ktoré zo stretnutia vzišli a je nutné ich podložiť písomne, aby sa predišlo riziku nesprávneho pochopenia jednotlivých tém.

Každý zápis vo svojom závere obsahuje úlohy vyplývajúce zo stretnutia, ktoré môžu ale nemusia byť explicitne uvedené v manažovacom nástroji. Sú to úlohy, ktoré je nutné splniť do najbližšieho stretnutia. Každá úloha má okrem popisu tiež priradeného člena tímu, ktorý je zodpovedný za jej splnenie.

Po spísaní zápisnice je táto odoslaná na revíziu jednotlivým členom. V prípade, že členovia odsúhlasia jej obsah, je následne uverejnená na webovej stránke tímu.

5.4.3 Metodiky

V našom projekte sme identifikovali dva druhy metodík. Prvý druh je koncipovaný ako súvislý text, druhý tvoria prevažne pravidlá, ktorá sú spísané do očíslovaných bodov.

Metodiky ako súvislý text majú jednoduchú štruktúru. Názov dokumentu je formátovaný typom písma Arial, veľkosť 26. Kapitoly obsahujú iba jednu úroveň číslovania formátovanú rovnakým typom písma, no veľkosťou 16. Celý text je zarovnaný do bloku.

Metodiky ako zoznam pravidiel obsahujú stručný obsah dokumentu, ktorý charakterizuje, čoho sa daná metodika týka. Nasleduje Zoznam pravidiel, ktorý je číslovaný arabskými číslicami. V prípade potreby ďalšej úrovne číslovania sa použijú rímske číslice formátované malými písmenami.

6 SUMARIZÁCIA ŠPRINTOV

V tejto kapitole nájdeme zosumarizované jednotlivé šprinty. Každá podkapitola obsahuje určitý opis predpokladaných úloh a výsledku šprintu. Taktiež sú tu vygenerované reporty z manažovacieho nástroja a retrospektíva šprintu z pohľadu tímu.

6.1 Sumarizácia šprintu č. 1

6.1.1 Opis šprintu a úloh

Úlohou prvého šprintu bolo najmä nájsť spôsob ako spolupracovať v tíme, ktorý by čo najviac vyhovoval všetkým členom. Na začiatku šprintu sme si definovali funkcionality aplikácie, ktoré by mali byť na konci prvého šprintu implementované. Aplikácia mala byť schopná prijímať dáta zo zariadenia a odosielať ich na server. Taktiež mala zbierať dodatočné dáta o telefonátoch, sms správach, polohe, údaje z krokomeru a tiež dáta o udalostiach v kalendári. Nakoľko zariadenie nemáme k dispozícii, súčasťou šprintu bola tiež úloha na vytvorenie generátora falošných dát pre testovacie účely aplikácie.

Keďže sme ako tím pracovali prvýkrát, bolo nutné tiež zadefinovať viaceré metodiky a pravidlá, ktoré postupne vyplývali zo vzájomnej spolupráce a potreby prehľadnosti a informovanosti o projekte. V rámci toho tiež začali vznikať viaceré kapitoly dokumentácie o riadení projektu.

6.1.2 Retrospektíva

Počas celého šprintu sme detegovali a následne riešili problémy týkajúce sa tímovej spolupráce. Prvé spoločné stretnutie malo chaotickú tendenciu, pretože sa diskusia na danú tému otvorila a skôr ako sa uzavrela, zmenila sa téma rozhovoru. Prvé stretnutie tak nakoniec bolo zdĺhavé a tím prišiel iba na málo konštruktívnych záverov. Zo stretnutia bola cítiť mierna frustrácia. Následne, po zadefinovaní rolí jednotlivých členov tímu bolo jasne badať zlepšenie aj v komunikácii na stretnutiach a jednotlivé diskusie prebiehali riadene.

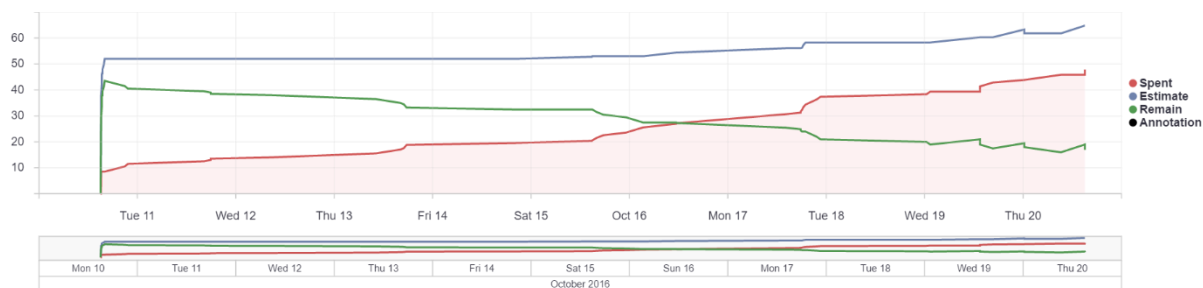
V prvom šprinte sme tiež hľadali najlepší spôsob ako pracovať s manažovacím nástrojom. Už začiatkom šprintu sme pôvodný TFS vymenili za Trello (Kap. 4.1.2). Po zmene nástroja na Trello bolo badať väčšiu ochotu všetkých členov tímu zaznamenávať svoju prácu do tohto nástroja, no aj tu sme pocítovali mierne nedostatky a potrebu zaviesť metodiku pre prácu s ním.

V rámci šprintu sme absolvovali stretnutie, kde sme tiež hľadali najlepšiu cestu ako pracovať s verziovacím nástrojom Git, pričom z tohto stretnutia tiež vznikla metodika na prácu s Gitom.

Výsledný prototyp aplikácie mal určité nedostatky. Nestihla sa vytvoriť zaznamenávanie telefonátov a sms správ. Taktiež sme mali nedostatky vo vytvorení simulátora zariadenia. Viaceré úlohy boli síce hotové, no nestihla sa na nich urobiť prehliadka kódu, čím sme ich nemohli

Sumarizácia šprintov

jednoznačne uzavrieť. Nesplnené úlohy boli z časti spôsobené nesprávnym odhadom na plánovaní a z časti tým, že sme do šprintu zaradili viacero úloh navyše, ktoré bolo potrebné urobiť pre správne fungovanie tímu. Taktiež nakoľko sme pociťovali nedostatky v oblasti merania stresu z odporu kože, venovali sme istý čas naštudovaniu tejto problematiky.



Obr. č. 4: Burndown graf na konci prvého šprintu

6.2 Sumarizácia šprintu č. 2

6.2.1 Opis šprintu a úloh

Hlavnou úlohou tohto šprintu bolo vytvoriť prototyp aplikácie, ktorá bude nielen zbierať dáta, ale ich už aj vhodným spôsobom spracovávať a zobrazovať používateľovi podľa jednotlivých funkcionalít.

Počas plánovania tím navrhol tri hlavné obrazovky, ktorá sa budú používateľovi zobrazovať. Prvou obrazovkou je úvodná obrazovka, ktorá sa zobrazí po spustení aplikácie. Tím po diskusii rozhodol, že obrazovka bude zobrazovať informácie o aktuálnom stave používateľa, vrátane počasia. Ďalšie dve obrazovky mali zobrazovať lineárny graf predstavujúci sumárne informácie o dátach získaných za celý deň, či týždeň. Bližšie grafické prevedenie mal na zodpovednosti člen tímu v roli UX dizajnéra Dániel Papp.

Taktiež sa tím zhodol na tom, že je nutné pokračovať v tvorbe dokumentácie k riadeniu, dielu a v tvorbe metodík.

6.2.2 Retrospektíva

Celý šprint sa aktívne pracovalo na aplikácii i na dokumentácii. Pri spoločných diskusiách sme identifikovali niekoľko problémov týkajúcich sa tímovej spolupráce.

Prvým problémom bolo nevhodné rozdelenie úloh jednotlivých členom, čo malo viaceré negatívne dopady. Najväčším dopadom bola situácia, kedy na seba čakali viacerí členovia tímu, pretože im chýbal určitý modul aplikácie, čím dochádzalo k neefektívnemu využívaniu tímového času. Tento problém sa následne prejavil aj na nižšie opísanom stave aplikácie na konci šprintu.

Ďalším problémom bolo neurčenie zodpovednosti za prehliadky kódu. Podľa Pravidiel pre vykonávanie prehliadok zdrojového kódu bližšie spísaných v kapitole 5.3.2 po dokončení programovacej úlohy presunie programátor príslušnú kartičku do stĺpca Code Review. Následne si iný člen tímu kartičku priradí a urobí na danej časti kódu prehliadku. Metodika ale bližšie nedefinuje, ktorý člen tímu má prehliadku vykonať, a tak vznikla situácia, že na konci šprintu bolo síce veľa úloh naprogramovaných, no neboli na nich vykonané prehliadky a tým pádom neboli označené za vyriešené a zimplementované vo vetve master. Tento problém sme vyriešili tak, že sme metodiku rozšírili o pravidlo, podľa ktorého programátor po presunutí kartičky do stĺpca Code Review túto kartičku zároveň priradí niektorému z členov tímu.

Výsledný stav aplikácie a úloh na konci šprintu tím zhodnotil nasledovne. Dokumentácia k riadeniu bola na konci šprintu takmer hotová a zrevidovaná. Začala sa práca na tvorbe dokumentácie k inžinierskemu dielu.

Dizajn aplikácie bol kompletne navrhnutý a odprezentovaný. Tím urobil niekoľko návrhov na zlepšenie, celkovo však dizajn zhodnotil kladne. Z pohľadu funkcionality aplikácie chýbala dôležitá časť pre komunikáciu aplikácie s databázou. Toto malo za následok, že obrazovky nemohli byť oživené a naplnené testovacími dátami. Nakoľko sa celý šprint týkal práve tohto oživenia, viaceré User Stories nemohli byť uzavreté.

6.3 Sumarizácia šprintu č. 3

6.3.1 Opis šprintu a úloh

Nakoľko v predošlom šprinte č. 2 zostalo neuzavretých viacero User Stories, ktoré sme preniesli do šprintu č. 3, sústredili sme sa práve na tieto úlohy. Boli to prevažne úlohy týkajúce sa použiteľnosti aplikácie. Naším cieľom bolo dopracovať prototyp aplikácie tak, aby bol pripravený na zobrazovanie reálnych dát. Väčšinu úloh tvorila backendová časť aplikácie.

Keďže počas tohto šprintu prebehlo odovzdávanie predbežnej dokumentácie, veľká časť tímového času bola venovaná dopracovaniu dokumentov a ich príprave na odovzdanie, najmä dokumentácie k inžinierskemu dielu.

Samostatnou časťou šprintu bolo tiež naštudovanie možností automatického testovania, ktoré mal na starosti člen tímu Miloš Pallo.

6.3.2 Retrospektíva

Z pohľadu zapracovania úloh bol šprint č. 3 pre tím najúspešnejší. Podarilo sa uzavrieť všetky User Stories zaradené do šprintu, aj celý Epic. Celkovo sme okrem jednej úlohy typu Bug dokázali uzavrieť všetky úlohy v šprinte.

Úspešne sme dokončili tímovú dokumentáciu a odovzdali v termíne.

Sumarizácia šprintov

Člen tímu Miloš Pallo zmapoval možnosti testovania, ktoré následne tímu prezentoval na spoločnom stretnutí.

Aplikácia bola na konci šprintu pripravená na zobrazovanie reálnych dát ako z frontendovej časti, tak aj z backendovej.

Na stretnutí na konci šprintu tím rozhodol, že je nutné čo najskôr začať s analýzou dát zo zariadenia. Na to je potrebné uskutočniť v nasledujúcom šprinte stretnutie, pri ktorom zozbierame testovacie dáta zo zariadenia na meranie stresu.

Najväčšou zmenou pre tím v tomto šprinte bolo začatie používania Story Pointov pri odhadovaní náročnosti úloh a plánovanie pomocou techniky Planning Poker. Táto zmena vyplynula z diskusie na cvičeniach z predmetu Manažment informačných systémov, kedy sme boli upozornení, že táto časť agilného vývoja je povinná a je nutné ju aplikovať v našich tímových projektoch.

Túto techniku sme používali prvýkrát, a preto sme nevedeli odhadnúť, koľko Story Pointov je vhodné zaradiť do šprintu. Po krátkej dohode sme si zvolili referenčnú User Story, podľa ktorej sme následne odhadovali zvyšné Stories.

Kolektívne sme rozhodli, že 40 SP by mohlo byť postačujúcich na zaplnenie šprintu.

Taktiež sme nevedeli, ako skĺbiť logovanie času do Trelly spolu so Story Pointami. To sa odrazilo aj na neúplných a nič nehovoriacich grafoch a exportoch z tohto šprintu. Nakoniec sme sa ako tím dohodli na tom, že User Stories budú mať vlastný odhad v Story Pointoch a do Taskov a Bugov budeme zarátavať odpracovaný čas.

Na tímovej retrospektíve sme vyhodnotili, že sa nám tento spôsob odhadovania nepáči a ani nám nevyhovuje. Zároveň je pre nás oveľa prirodzenejšie odhadovať náročnosť User Stories pomocou hodín. Keďže je ale táto metóda povinná, budeme ju používať ďalej.

Tímová retrospektíva priniesla ďalšie postrehy. Člen tímu Martin Šrank zhodnotil, že vynechané tímové stretnutie spôsobené sviatkom sa prejavilo na tímovej práci negatívne a pracovali sme menej efektívne ako zvyčajne. Tím s týmto názorom súhlasil.

6.4 Sumarizácia šprintu č. 4

6.4.1 Opis šprintu a úloh

V tomto šprinte sme usúdili, že je nutné sa zamerať na aplikáciu z programátorského pohľadu. Keďže za posledné týždne pribudlo množstvo kódu a z hľadiska dlhodobej udržateľnosti je nutné pravidelne vykonávať refaktoring zdrojového kódu.

Okrem toho, že sme menili štruktúru androidovej časti projektu, viaceré globálne premenné sme vyniesli do konfiguračných súborov, aby bola aplikácia jednoduchšie konfigurovateľná.

Sumarizácia šprintov

Veľmi dôležitou časťou boli experimenty so zariadením na meranie stresu a následné vyhodnotenie získaných dát.

Taktiež sme sa rozhodli implementovať automatizované testy lokálnych funkcií.

6.4.2 Retrospektíva

V tomto šprinte sme druhýkrát plánovali úlohy pomocou Story Pointov a Planning Poker. Neskúsenosť v tomto spôsobe plánovania a zároveň viacej školských povinností sa odrazili na viacerých neuzatvorených Stories na konci šprintu.

Tím spolu s vedúcou v rámci retrospektívy hľadali spôsoby, ako sa takýmto situáciám vyvarovať. Jeden z návrhov, ktoré sa tím rozhodol vyskúšať v budúcom semestri je vyberanie rôznych druhov Stories do šprintu, čím predíde problému, že ak sa čaká na určitú časť systému, nebude kvôli nej stáť tvorba celej aplikácie.

V tomto šprinte bolo významné tiež stretnutie, počas ktorého tím meral dáta zo zariadenia a jednotliví členovia si navzájom navádzali stresové pocity podľa vopred pripravených scenárov.

Nazbierané dáta sme podrobili analýze a skonzultovali s výskumníkom, ktorý sa venuje hľadaniu anomálií v prúde dát. Navrhli sme niekoľko riešení, ktoré sme spísali do dokumentu a uložili do tímového adresára na gite.

Z analýzy vyplynulo, že ďalšie experimenty by sme mali sústrediť na jedného človeka a vykonávať ich dlhší čas bez prerušenia, aby sa na dátach vedeli dostatočne prejaviť anomálie vyvolané stresom. Taktiež je nutné eliminovať veľké množstvo chýb v dátach spôsobených ľudským faktorom. Jeden z návrhov je, že by sme čip na meranie počas experimentu pripevnili na prst lepiacou páskou a tak čo najviac obmedzili jeho pohyb.

Nakoľko ďalší šprint by trval iba 6 dní, tím sa spolu s vedúcou rozhodol, že nasledujúce dni sa bude venovať príprave na prezentáciu riadenia na predmete Manažment informačných systémov.

7 RETROSPEKTÍVA ZIMNÉHO SEMESTRA

V tejto kapitole sa nachádza zhrnutie tímového riadenia a celkového fungovania v tíme počas zimného semestra.

7.1 Komunikácia

Komunikácia je jednou z najdôležitejších aspektov fungovania v tíme. Počas práce na projekte tím vyskúšal viacero komunikačných kanálov.

Na okamžité správy tím využíval službu **Slack**. Tá umožňuje zriadenie kanálov, preto bolo aj dohľadávanie v histórii jednoduchšie, v porovnaní s napr. Facebookom. Jeden z kanálov slúžil na virtuálny stand-up, ktorý sme zaviedli po prvom týždni fungovania, pretože stretnutie raz za týždeň sa nám zdalo príliš málo.

Postupne sa nám podarilo Slack prepojiť s manažovacím nástrojom Trello, aj s gitom, čo nám poskytovalo jasný obraz o tom, kto na čom práve pracuje.

Nakoľko tím komunikoval nie len s vedúcou ale aj s externou firmou, bolo nutné zriadiť oficiálny komunikačný kanál. Tímová **emailová adresa** je team12fiit@gmail.com. Táto emailová adresa slúžila naozaj iba na komunikáciu s tvorcami zariadenia.

Posledným významným spôsobom komunikácie boli **spoločné stretnutia**. Tie najskôr bývali pravidelne v pondelok. V polovici semestra sme museli stretnutia posunúť na štvrtok, čo spôsobilo tiež posun v šprinte.

Prvé stretnutia boli rozpačité, skákalo sa od témy k téme a na konci stretnutia bolo stále mnoho nevyriešených otázok. Postupne tím začal stretnutia plánovať a riadiť, čím sa ich priebeh značne zefektívnil. Pred každým stretnutím scrum master zozbieral témy, ktoré chceli členovia prebrať a zaradil ich do programu. Z každého stretnutia bola spísaná zápisnica, ktorá je uverejnená na webovej stránke tímu.

7.2 Manažovací nástroj

Po dôkladnom prieskume bol prvou voľbou tímu nástroj TFS. Ten ponúkal všetky rozšírenia potrebné pre agilný vývoj. Po nakonfigurovaní nástroja bol pokus integrovať ho s gitom. Tento pokus aj napriek veľkej snahe nevyšiel. Po krátkej práci s týmto nástrojom tím usúdil, že práca s ním je zbytočne zložitá.

Ďalšou voľbou bol nástroj Trello. Trello nie je pôvodne zamýšľané ako pomôcka pri agilnom vývoji, no existujú rozšírenia do prehliadača Google Chrome, ktoré ho podporujú. Pomocou týchto rozšírení je možné nie len logovať čas na jednotlivých úlohách ale aj sledovať ich plnenie pomocou burndown grafov.

Farebné označenia na jednotlivých úlohách umožnili veľmi ľahko rozlíšiť typ danej úlohy. Členovia tímu si museli zaviesť viacero pravidiel, ktoré spísali do Metodiky práce s manažovacím nástrojom, no aj napriek ťažšiemu štartu bol s týmto nástrojom spokojný. Trello používal celý zvyšok semestra.

7.3 Agilný vývoj

Podľa zadania majú jednotlivé tímy pracovať na svojich projektoch agilným vývojom.

Tím dodržiaval pravidelné 2 –týždňové šprinty. Výnimkou bol 2. šprint, ktorý trval dlhšie, pretože sa menil termín pravidelných stretnutí a tím chcel dodržať, že šprint bude končiť práve na tomto stretnutí.

Na konci každého šprintu tím absolvoval retrospektívu, na ktorej jednotliví členovia vyjadrili svoju spokojnosť resp. nespokojnosť s končiacim šprintom. Z týchto retrospektív tiež vzniklo niekoľko návrhov na zlepšenie metódik.

Po retrospektíve nasledovalo plánovanie. Prvé 2 šprinty tím plánoval a odhadoval svoj čas v hodinách nakoľko to všetkým členom tímu prišlo prirodzenejšie. Po diskusii na predmete Manažment informačných systémov tím zistil, že odhadovanie pomocou Story pointov je povinné.

Ďalší šprint sa pokúsil teda plánovať pomocou planning pokru a Story pointov. Celý tím sa zhodol, že tento spôsob plánovania nie je intuitívny a predošlý spôsob pomocou hodín mu vyhovoval viacej. Nakoľko je ale táto časť tímového projektu povinná, tím aj posledný šprint naplánoval týmto spôsobom.

7.4 Zhrnutie

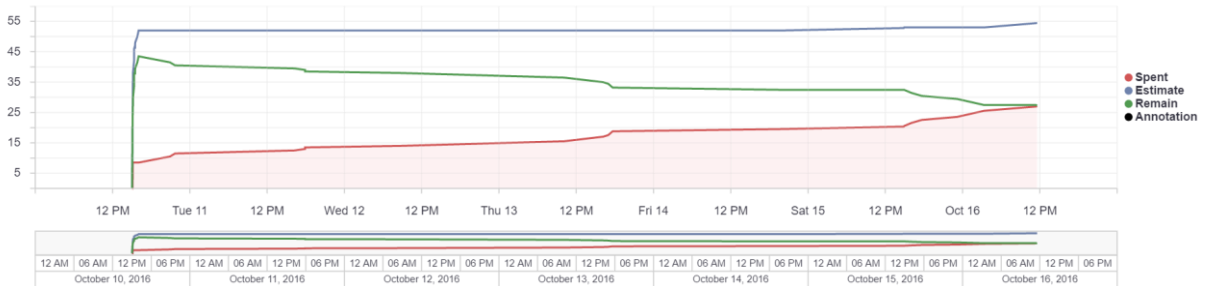
Členovia tímu v takomto zložení spolupracovali prvýkrát. S agilným vývojom sa pred tímovým projektom v praxi stretol iba jeden člen tímu, preto boli začiatky ťažšie. Prevažná časť tímu má stále názor, že ak by aplikáciu mali vyvíjať sami, bola by dávno naprogramovaná.

V závere semestra je vidieť veľký pokrok v komunikácii medzi jednotlivými členmi tímu. Postupne sme si našli spôsob, ako čo najlepšie používať Trello. Na niektorých stretnutiach vznikali rozpory spojené s rôznymi názormi na niektoré časti aplikácie. Aj napriek snahe riešiť tieto konflikty konštruktívne, nakoniec vždy vyhral ten čo sa vydržal hádať najdlhšie.

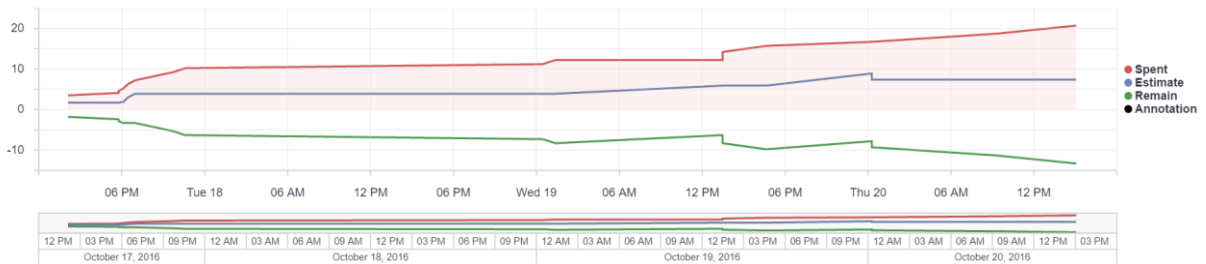
Celkovo ako tím hodnotíme aktuálne fungovanie v porovnaní s prvými týždňami kladne.

PRÍLOHA A - BURNDOWN GRAFY

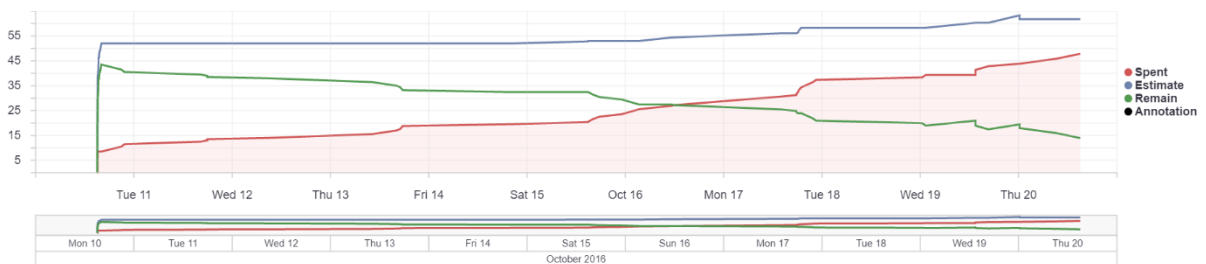
V tejto prílohe sa nachádzajú exporty z manažovacieho nástroja. Podľa pokynov sme každý týždeň vytvárali burndown graf. Tento graf slúžil tímu na vytvorenie obrazu o celkovej práci tímu počas týždňa. Na konci každého šprintu sme vytvorili sumárny graf, ktorý dával tímu obraz o jeho aktivite počas celého šprintu.



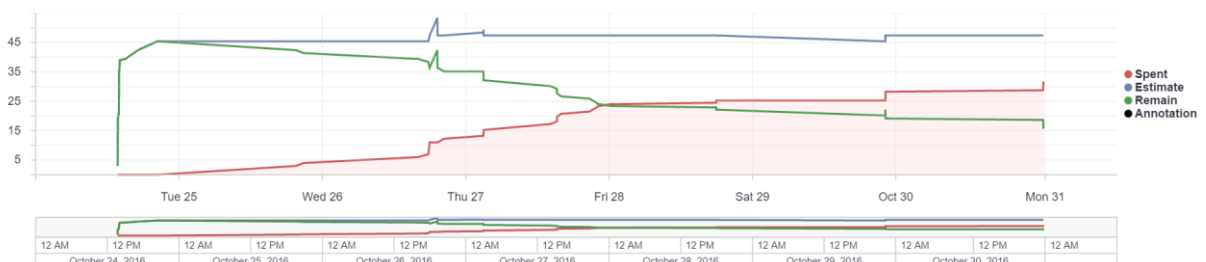
Obr. č. 5: Burndown graf z prvého týždňa prvého šprintu



Obr. č. 6: Burndown graf z druhého týždňa prvého šprintu

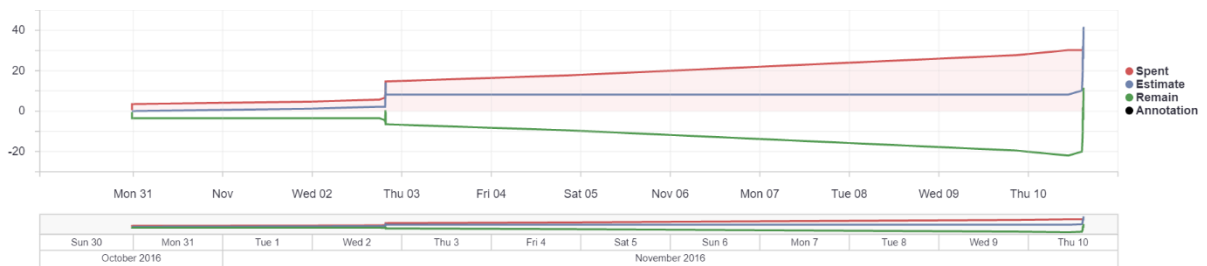


Obr. č. 7: Burndown graf prvého šprintu

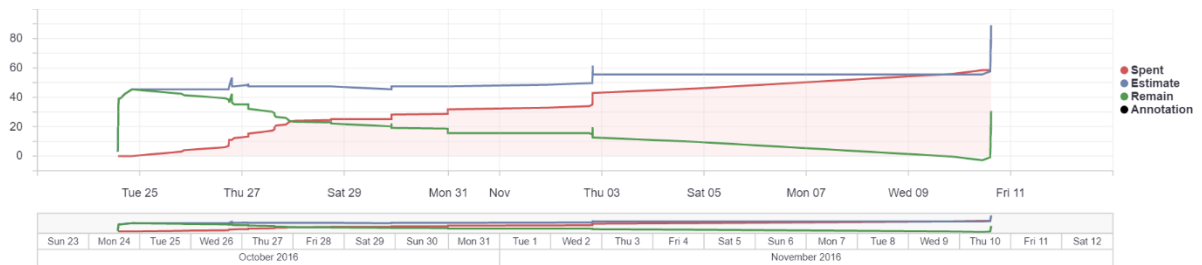


Retrospektíva zimného semestra

Obr. č. 8: Burndown graf z prvého týždňa druhého šprintu



Obr. č. 9: Burndown graf z druhého týždňa druhého šprintu



Obr. č. 10: Burndown graf z druhého šprintu

V ďalších šprintoch sme burndown grafy negenerovali, keďže zobrazované dáta nám nepomáhali v reálnom vyhodnotení aktuálneho stavu šprintu.

PRÍLOHA B - EXPORTY ÚLOH

Nasledujúca tabuľka predstavuje export úloh na konci prvého šprintu.

User	List	Card	S	E 1 st	E
smasty	Sprint Backlog	E1_S1 Asynchrone uklada- nie dat	0	3	3
milospallo	Sprint Backlog	E1_S5 Získať dáta o telefoná- toch a SMS	0	4	4
milospallo	Sprint Backlog	E1_S2 Vytvoriť simulátor na vyžiadanie dát	0	01.5	01.5
milospallo	Sprint Backlog	E1_S1 Vytvoriť simulátor na odosielanie dát	0	02.5	02.5
olivermoravcik	Sprint Backlog	E1_S1 Vytvoriť simulátor na generovanie dát zo zariade- nia	0	3	3
smasty	Review	E1_S1 Ukladanie do databazy	2	0	0
barbora64	Review	E1_S4 Získať dáta z kroko- meru	2	0	0
martin_nemcek	Review	E1_S1 Ukladanie do databazy	0	0	-1.5
smasty	Review	E1_S1 Ukladanie do databazy	1	3	3
martin_nemcek	Review	E1_S5 Povolenia na androide	2	0	0
martin_nemcek	Review	E1_S5 Povolenia na androide	0	2	2
smasty	Review	E1_S1 Server REST API	1	0	0
smasty	Review	E1_S1 Server REST API	1	0	0
smasty	Review	E1_S1 Server REST API	1	0	0
smasty	Review	E1_S1 Server REST API	2	0	0
martin_nemcek	Review	E1_S6 Získať dáta z kalendáru pomocou Android API	1	0	1
martin_nemcek	Review	E1_S6 Získať dáta z kalendáru pomocou Android API	03.5	0	01.7
martin_nemcek	Review	E1_S6 Získať dáta z kalendáru pomocou Android API	0.2	0	0
martin_nemcek	Review	E1_S3 Zbierať dáta o polohe	2	0	0
martin_nemcek	Review	E1_S6 Získať dáta z kalendáru pomocou Android API	0	2	2
martin_nemcek	Review	E1_S3 Zbierať dáta o polohe	0	2	2
martin_nemcek	Review	E1_S1 Ukladanie do databazy	0	01.5	01.5

Retrospektíva zimného semestra

barbora64	Review	E1_S4 Získať dáta z kroko- meru	0	5	5
smasty	Review	E1_S1 Server REST API	0	4	4
smasty	Closed*	E1_S1 Databazova schema	01.5	0	0
barbora64	Closed*	Dokumentacia - manazment dokumentacie	1	0	1
barbora64	Closed*	Vygenerovanie reportov	0.6	0	0.2
barbora64	Closed*	Vytvorit zapis zo stretnutia c. 4	0.5	0	0
barbora64	Closed*	Vygenerovanie reportov	0.6	0	0
barbora64	Closed*	Dokumentacia - manazment dokumentacie	01.4	01.4	01.4
smasty	Closed*	E1_S1 Nakonfigurovať server	1	0	0
barbora64	Closed*	Dokumentacia - predstavenie timu	1	0	0
barbora64	Closed*	Dokumentacia - predstavenie timu	1	0	0
barbora64	Closed*	Dokumentacia - uvod	0.2	0	0.2
barbora64	Closed*	Dokumentacia - uvod	0.8	0.8	0.8
barbora64	Closed*	Vytvorit zapis zo stretnutia c. 3	0.75	0	0
olivermoravcik	Closed*	E1_S1 Analyza posielanych dat	1.25	0	0
milospallo	Closed*	E1_S7 Vytvorenie android projektu	01.5	0	0
olivermoravcik	Closed*	E1_S1 Analyza posielanych dat	1.53	0	0
olivermoravcik	Closed*	E1_S1 Analyza posielanych dat	0.5	0	0
d4no	Closed*	Vytvorit webstránku tímu	0.5	0	0
olivermoravcik	Closed*	E1_S1 Analyza posielanych dat	0.5	0	0
smasty	Closed*	E1_S1 Nakonfigurovať server	1	0	0
smasty	Closed*	Spisat konvencie pre pracu s gitom	1	0	0
d4no	Closed*	Pripomienky k web stranke	2	0	0
barbora64	Closed*	Dokumentacia - predstavenie timu	0	2	2
d4no	Closed*	Pripomienky k web stranke	0	2	2
milospallo	Closed*	E1_S7 Vytvorenie android projektu	0	1	1
barbora64	Closed*	E1_S7 Vytvorenie android projektu	0	0	-1
barbora64	Closed*	E1_S7 Vytvorenie android projektu	0	1	1
olivermoravcik	Closed*	E1_S1 Analyza posielanych dat	0	4	4

Retrospektíva zimného semestra

barbora64	Closed*	Vygenerovanie reportov	0	1	1
smasty	Closed*	Spisat konvencie pre pracu s gitom	0	1	1
smasty	Closed*	E1_S1 Nakonfigurovať server	1	4	4
d4no	Closed*	Vytvorit webstránku tímu	07.5	8	8
barbora64	Closed*	Vytvorit zapis zo stretnutia c. 4	0	0.5	0.5
barbora64	Closed*	Vytvorit zapis zo stretnutia c. 3	0	1	1
smasty	Closed*	E1_S1 Databazova schema	0	1	1
martin_nemcek	Closed*	E1_S1 Databazova schema	0	0	-1
martin_nemcek	Closed*	E1_S1 Databazova schema	0	1	1

Nasledujúca tabuľka predstavuje export úloh na konci druhého šprintu.

Card	List	S	E 1 st	E	R
E2_S3 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za týždeň	Sprint Backlog	0	6	0	0
E2_S3 Odosielanie dát zo servera	Sprint Backlog	0	0	0	0
E1_S1 Ako používateľ chcem aby aplikácia spracovávala a ukladala dáta mimo mobilu aby ho nezaťažovala	Sprint Backlog	0	0	0	0
E2_S1 Ako používateľ chcem aby ma aplikácia privítala peknou úvodnou obrazovkou, zobrazila dôležité informácie a ponúkla prehľadné menu	Sprint Backlog	0	6	0	0
E2_S2 Odosielanie dát zo servera	Sprint Backlog	0	2	2	2
E2_S1 Odosielanie dát zo servera	Sprint Backlog	0	2	2	2
E2_S2 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za deň	Sprint Backlog	0	6	6	6
E2 Zobrazovať používateľovi grafy	Sprint Backlog	0	0	0	0
E1 Vytvorit prvý funkčný prototyp mobilnej aplikácie so serverom	Sprint Backlog	0	0	0	0
E1_S2 Vytvorit simulátor na vyžiadanie dát	Sprint Backlog	0	01.5	01.5	01.5
E1_S2 Vytvorit simulátor na vyžiadanie dát	Sprint Backlog	0	0	0	0
E1_S1 Vytvorit simulátor na odosielanie dát	Review	10	02.5	02.5	-7.5
E2_S3 Implementácia obrazovky s týždenným prehľadom	Review	3	6	6	3
E1_S5 Ako používateľ chcem aby aplikácia zbierala dáta o mojich telefonátoch a SMS aby som vedel kto mi zvýšil stres	Closed*	0	0	0	0
E2_S1 Implementácia úvodnej obrazovky	Closed*	05.5	6	6	0.5
E2_S2 Implementácia obrazovky s denným prehľadom	Closed*	6	6	6	0

Retrospektíva zimného semestra

E2_S2 Návrh obrazovky s denným prehľadom	Closed*	2	4	2	0
E2_S1 Návrh úvodnej obrazovky obrazovky	Closed*	2	3	2	0
E1_S5 Získať dáta o telefonátoch a SMS	Closed*	02.2	05.2	06.2	4
Metodika používania manažovacieho nástroja	Closed*	1	2	2	1
Metodika tvorby dokumentácie	Closed*	01.5	2	2	0.5
Metodika práce s Git-om a prehliadky kódu	Closed*	1.25	2	2	0.75
Konvencie písania zdrojového kódu	Closed*	1	2	2	1
E2_S3 Návrh obrazovky	Closed*	1	4	2	1
Prerobenie krokomeru na akcelometer	Closed*	2	6	6	4
E1_S8 Ako používateľ chcem aby aplikácia zbierala dáta o počasí aby som vedel aké má počasie vplyv na môj stres	Closed*	0	0	0	0
Manažment plánovania a rizík	Closed*	1.75	3	3	1.25
Manažment dokumentácie	Closed*	0.75	3	3	2.25
E1_S8 Sťahovanie aktuálneho počasia na základe reportovanej polohy používateľa	Closed*	1	3	3	2
E1_S1 Asynchrone spracovanie prijatých dát	Closed*	5	4	5	0
Vytvoriť notifikáciu na virtualny stand-up	Closed*	4	02.4	02.4	-1.6
Sumarizácia šprintu č. 1	Closed*	1	2	2	1
E1_S1 Vytvoriť simulátor na generovanie dát zo zariadenia	Closed*	4	3	3	-1
Manažment vývoja	Closed*	0	3	3	3
E1_S4 Získať dáta z krokomeru	Closed*	2	5	5	3
E1_S5 Povolenia na androide	Closed*	2	2	2	0
E1_S6 Získať dáta z kalendáru pomocou Android API	Closed*	04.7	2	04.7	0
E1_S3 Zbierať dáta o polohe	Closed*	2	2	2	0

Nasledujúca tabuľka predstavuje export úloh po prvom týždni tretieho šprintu. Keďže sme po prvýkrát skúšali odhadovanie pomocou Story Pointov, logovanie času sme nebrali do úvahy.

Card	List	S	E 1 st	E	R
E2_S3 Odosielanie dát zo servera	Sprint Backlog	0	0	0	0
Retrospektíva 2. šprintu	Sprint Backlog	0	0	0	0
Priprava testovacích scenárov	Sprint Backlog	0	0	0	0
Zlyhanie aplikácie pri týždennom prehľade	Sprint Backlog	0	0	0	0
E2_S5 APChcem, aby mi aplikácia zobrazila sumár, koľo času za deň/týždeň som a koľno nie som pod stresom	Sprint Backlog	0	8	8	8
E1_S9 Ako používateľ chcem aby aplikácia získavala informácie počas celého dňa	Sprint Backlog	0	8	8	8

Retrospektíva zimného semestra

priebežne, aby som mal vždy aktuálne informácie o mojom strese					
E2_S3 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za týždeň	Sprint Backlog	0	5	0	0
E2_S1 Odosielanie dát zo servera	Sprint Backlog	0	0	0	0
E1_S2 Vytvoriť simulátor na vyžiadanie dát	Sprint Backlog	0	0	-1.5	-1.5
E2_S2 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za deň	Sprint Backlog	0	8	8	8
E2_S1 Ako používateľ chcem aby ma aplikácia privítala peknou úvodnou obrazovkou, zobrazila dôležité informácie a ponúkla prehľadné menu	Sprint Backlog	0	8	8	8
E1_S1 Ako používateľ chcem aby aplikácia spracovávala a ukladala dáta mimo mobilu aby ho nezaťažovala	Review	2	2	2	0
E1_S1 Automatická registrácia aplikácie	Review	0	0	0	0
Nastudovať metodiku testovania	In progress	0	0	0	0
E1_S2 Ako používateľ chcem aby aplikácia sťahovala dáta len keď si ich vyžiadam aby šetrila moje mobilné dáta	In progress	0	1	1	1
E2_S2 Odosielanie dát zo servera	Closed*	0	0	0	0
Testovanie	Closed*	0	0	0	0
Implementácia	Closed*	0	0	0	0
Návrh	Closed*	0	0	0	0
Analýza	Closed*	0	0	0	0
E1_S1 Aktualizácia dátovej schémy	Closed*	0	0	0	0

Nasledujúca tabuľka predstavuje export úloh na konci tretieho šprintu.

Card	List	S	E 1 st	E	R
Zlyhanie aplikácie pri týždennom prehľade	Sprint Backlog	0	0	0	0
Priprava testovacích scenárov	In progress	0	0	0	0
E2 Zobrazovať používateľovi grafy	Closed*	0	0	0	0
E2_S5 APChcem, aby mi aplikácia zobrazila sumár, koľo času za deň/týždeň som a koľko nie som pod stresom	Closed*	8	8	8	0
E2_S3 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za týždeň	Closed*	5	0	5	0
E2_S2 Ako používateľ chcem aby aplikácia zobrazovala prehľad môjho stresu za deň	Closed*	8	8	8	0

Retrospektíva zimného semestra

E2_S1 Ako používateľ chcem aby ma aplikácia privítala peknou úvodnou obrazovkou, zobrazila dôležité informácie a ponúkla prehľadné menu	Closed*	8	8	8	0
E2_S2 Odosielanie dát zo servera	Closed*	0	0	0	0
E1 Vytvoriť prvý funkčný prototyp mobilnej aplikácie so serverom	Closed*	0	0	0	0
Nastudovať metodiku testovania	Closed*	0	0	0	0
E2_S2 Implementácia obrazoviek s detailmi o eventoch (call,message,calendar event)	Closed*	3	3	3	0
E1_S9 Ako používateľ chcem aby aplikácia získavala informácie počas celého dňa priebežne, aby som mal vždy aktuálne informácie o mojom strese	Closed*	8	8	8	0
E2_S5 Implementácia obrazoviek pre súhrn stresu za den/týždeň	Closed*	3	3	3	0
E1_S2 Implementácia	Closed*	0	0	0	0
E1_S1 Automatická registrácia aplikácie	Closed*	0	0	0	0
E1_S2 Ako používateľ chcem aby aplikácia sťahovala dáta len keď si ich vyžiadam aby šetrila moje mobilné dáta	Closed*	1	1	1	0
Retrospektíva 2. šprintu	Closed*	0	0	0	0
E1_S1 Ako používateľ chcem aby aplikácia spracovávala a ukladala dáta mimo mobilu aby ho nezaťažovala	Closed*	2	2	2	0
Testovanie	Closed*	0	0	0	0
Implementácia	Closed*	0	0	0	0
Návrh	Closed*	0	0	0	0
Analýza	Closed*	0	0	0	0
E1_S1 Aktualizácia dátovej schémy	Closed*	0	0	0	0
E1_S2 Vytvoriť simulátor na vyžiadanie dát	Closed*	0	0	-1.5	-1.5

Nasledujúca tabuľka predstavuje export úloh po prvom týždni štvrtého šprintu.

Card	List	S	E 1 st	E	R
Dokumentácie riadenia na odovzdanie	Sprint Backlog	0	0	0	0
Dokumentácie diela na odovzdanie	Sprint Backlog	0	0	0	0
E3_S6 Vytvorenie servisu na odosielanie zozbieraných dát	Sprint Backlog	0	0	0	0
IS3 Konfigurácia testov používateľského rozhrania	Sprint Backlog	0	0	0	0

Retrospektíva zimného semestra

IS3 Vytvorenie testov používateľského rozhrania	Sprint Backlog	0	0	0	0
IS3 Vytvorenie testov bežiacich na JVM	Sprint Backlog	0	0	0	0
IS3 Konfigurácia frameworkov pre testovanie používateľského rozhrania	Sprint Backlog	0	0	0	0
IS2 Konfigurácia lokálnych testov	Sprint Backlog	0	0	0	0
IS2 Vytvorenie lokálnych JUnit testov	Sprint Backlog	0	0	0	0
E3 Prototyp s reálnymi dátami	Sprint Backlog	0	0	0	0
E3_S5 Stiahnutie dát zo servera	Sprint Backlog	0	0	0	0
E3_S5 Zjednotenie modelových tried	Sprint Backlog	0	0	0	0
E3_S4 Stiahnutie dát zo servera	Sprint Backlog	0	0	0	0
E3_S4 Zjednotenie modelových tried	Sprint Backlog	0	0	0	0
E3_S4 Backend k úvodnej obrazovke	Sprint Backlog	0	0	0	0
Analýza získaných dát	Sprint Backlog	0	0	0	0
E3_S5 Ako používateľ chcem aby prehľady ukazovali moje reálne dáta	Sprint Backlog	0	13	13	13
E3_S4 Ako používateľ potrebujem aby domáca obrazovka zobrazovala reálne dáta	Sprint Backlog	0	8	8	8
E3_S6 Ako používateľ potrebujem aby aplikácia ukladala o mne reálne dáta	Sprint Backlog	0	3	3	3
Review šprintu č.3	Sprint Backlog	0	0	0	0
IS4 Ako developer potrebujem pridať do projektu config súbor s konštantami kde budú prístupné pre všetky triedy / funkcie	Sprint Backlog	0	2	2	2
IS3 Ako developer potrebujem vytvoriť testy používateľského rozhrania	Sprint Backlog	0	8	8	8
IS2 Ako developer potrebujem vytvoriť základné lokálne automatizované testy	Sprint Backlog	0	5	5	5
IS4 Zimplementovať konštanty do tried	Review	0	0	0	0
IS4 Vytvoriť konfiguračný súbor	Review	0	0	0	0
E3_S5 Backend k percentuálnemu prehľadu	Review	0	0	0	0
E3_S5 Backend k týždennému prehľadu	Review	1	1	1	0
IS2 Pripravenie prostredia pre lokálne testy	In progress	0	0	0	0
E3_S4 Zobrazenie podrobností podľa typu eventu	In progress	0	0	0	0
Zlyhanie aplikácie pri týždennom prehľade	Closed*	1	1	1	0
Čo všetko máme odovzdať?	Closed*	0.25	0.25	0.25	0
IS1 String resources	Closed*	1.57	1.57	1.57	0
IS1 Pomenovanie tried	Closed*	0.88	0.88	0.88	0
IS1 Organizácia package-ov	Closed*	0.93	0.93	0.93	0
IS1 Ako developer potrebujem refactoring celého projektu a jeho štruktúry, aby bol projekt prehľadný a bol v ňom poriadok	Closed*	0	1	1	1