

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Projektová dokumentácia k riadeniu projektu

(Tím 03 - SmartStore)

Akademický rok:	2016/2017
Predmet:	Tímový projekt
Členovia tímu (študenti):	Bc. Matej Červenka Bc. Mária Dragúňová Bc. Ondrej Kudláč Bc. Matúš Salát Bc. Martin Šidlo Bc. Lukáš Šimek
Vedúci tímu:	Ing. Peter Krátky
Vlastník produktu:	Ing. Ján Sukeník

Obsah

1 ÚVOD	1
2 ROLY ČLENOV TÍMU A PODIEL PRÁCE	3
2.1 MANAŽÉRSKE ČINNOSTI	3
2.1.1 Manažovanie vývoja metódou SCRUM	3
2.1.2 Posudzovanie kvality kódu	3
2.1.3 Manažovanie plánovania úloh	3
2.1.4 Manažovanie komunikácie	3
2.1.5 Písanie projektovej dokumentácie	3
2.1.6 Testovanie funkcionality produktu	4
2.1.7 Poskytovanie technickej podpory	4
2.1.8 Manažovanie verziovania	4
2.1.9 Manažér dizajnu	4
2.1.10 Manažovanie rizík	4
2.2 PODIEL PRÁCE NA DOKUMENTÁCII	5
3 APLIKÁCIE MANAŽMENTOV	7
3.1 MANAŽOVANIE VÝVOJA METÓDOU SCRUM.....	7
3.2 MANAŽOVANIE PLÁNOVANIA ÚLOH	7
3.3 MANAŽOVANIE KOMUNIKÁCIE	8
3.4 MANAŽOVANIE VERZIOVANIA	8
3.5 MANAŽMENT KVALITY KÓDU	8
3.6 MANAŽOVANIE TESTOVANIA.....	9
3.6.1 Úprava špecifikácie	9
3.6.2 Implementácia novej funkcionality.....	9
3.6.3 Reagovanie v prípade neúspešného testu.....	9
3.6.4 Reakcia v prípade, že boli testy vykonané bez chyby.....	9
3.7 MANAŽOVANIE RIZÍK	9
4 SUMARIZÁCIE ŠPRINTOV	13
4.1 1. ŠPRINT - E.T.....	13
4.1.1 Retrospektíva.....	13
4.1.2 Burndown chart	14
4.1.3 Cumulative flow	14
4.1.4 Zhodnotenie prvého šprintu.....	15
4.2 2. ŠPRINT - SPOCK.....	15
4.2.1 Retrospektíva.....	15
4.2.2 Burndown chart	16
4.2.3 Cumulative flow	16
4.2.4 Zhodnotenie druhého šprintu	16
4.3 3. ŠPRINT - KAL-EL.....	17
4.3.1 Retrospektíva.....	17
4.3.2 Burndown chart	17
4.3.3 Cumulative flow	18
4.3.4 Zhodnotenie tretieho šprintu.....	18
4.4 ŠPRINT 4: CHEWBACCA	18
4.4.1 Retrospektíva.....	18
4.4.2 Burndown chart	19
4.4.3 Cumulative flow	19
4.4.4 Zhodnotenie šprintu	19
4.5 ŠPRINT 5: YODA	20
4.5.1 Retrospektíva.....	20

4.5.2	Burndown chart	20
4.5.3	Cumulative flow	21
4.5.4	Zhodnotenie šprintu	21
4.6	ŠPRINT 6: HAN SOLO	21
4.6.1	Burndown chart	22
4.6.2	Cumulative flow	22
4.6.3	Zhodnotenie šprintu	22
4.7	ŠPRINT 7: ALIEN	23
4.7.1	Burndown chart	23
4.7.2	Cumulative flow	24
4.7.3	Zhodnotenie šprintu	24
4.8	ŠPRINT 8: ALF.....	24
4.8.1	Burndown chart	25
4.8.2	Cumulative flow	25
4.8.3	Zhodnotenie šprintu	25
4.9	ŠPRINT 9: PRAWN	26
4.9.1	Burndown chart	26
4.9.2	Cumulative flow	27
4.9.3	Zhodnotenie šprintu	27
4.10	ŠPRINT 10: YODA	27
4.10.1	Burndown chart	28
4.10.2	Cumulative flow	28
4.10.3	Zhodnotenie šprintu	28
5	POUŽÍVANÉ METODIKY.....	29
5.1	PÍSANIE KÓDU V RUBY ON RAILS	29
5.2	OVEROVANIE KÓDU	29
5.3	VERZIOVANIE PROJEKTU SMARTSTORE.....	29
5.4	EVIDENCIA ÚLOH.....	29
5.5	TESTOVANIE	29
5.6	PÍSANIE SKRIPTOV DÁTOVEJ ANALÝZY V R	29
5.7	DOKUMENTOVANIE	30
6	GLOBÁLNA RETROSPEKTÍVA ZS	31
7	GLOBÁLNA RETROSPEKTÍVA LS	32
8	PRÍLOHY.....	33

1 Úvod

Projektová dokumentácia k riadeniu projektu opisuje tím a zachytáva spôsob celkového riadenia tímu v rámci predmetu Tímový projekt, konkrétne pre tím číslo 3. Našou hlavnou témou, ktorou sa zaoberáme je vymyslieť inteligentný spôsob riadenia skladu pričom úzko spolupracuje so zástupcom zo spoločnosti Martinus, ktorý v procese riadenia vystupuje ako vlastník produktu.

Dokument opisuje jednotlivé manažérske činnosti, ktoré v rámci tímu používame na dosahovanie stanovených cieľov, popisujeme spôsob komunikácie medzi jednotlivými členmi tímu, či spravovanie rozdelených úloh. V dokumente tiež rozoberáme používané metodiky a postupy pri odovzdávaní vypracovaných úloh.

V kapitole 2 opisujeme jednotlivé roly členov nášho tímu a k jednotlivým manažérskym činnostiam uvádzame tiež zodpovedné osoby. V kapitole 3 jednotlivé manažérske činnosti bližšie špecifikujeme a popisujeme ich praktické aplikovanie v procese riadenia projektu. Kapitola 4 obsahuje podrobné výsledky, ku ktorým sme sa dopracovali počas jednotlivých dvojtýždenných šprintov, pričom k lepšiemu prehľadu o úspešnosti prispievajú retrospektívy vypracované na konci každého šprintu. Kapitola 5 ponúka stručný prehľad o používaných metodikách a v záverečnej kapitole 6 uvádzame celkové zhrnutie doterajšej práce na projekte, spolu s vypracovanou retrospektívou pre dané obdobie.

2 Roly členov tímu a podiel práce

2.1 Manažérske činnosti

2.1.1 Manažovanie vývoja metódou SCRUM

Náplň práce: Zodpovedná osoba si udržuje dobrý prehľad o stave úloh jednotlivých členov tímu, upozorňuje na prípadné problémy alebo odchýlky od metodiky SCRUM-u, pátra po prípadných príčinách nespokojnosti jednotlivých členov tímu a zabezpečuje kroky na ich odstránenie. Na tímových stretnutiach určuje štruktúru stretnutia a následnosť jednotlivých diskusií.

Zodpovedná osoba: Mária Dragúňová

2.1.2 Posudzovanie kvality kódu

Náplň práce: Stanovenie metodiky pre písanie kódu, kontrola kódu jednotlivých programátorov podľa stanovených pravidiel pre písanie kódu v danom programovacom jazyku, vykonávanie prehliadok zdrojového kódu novej verzie aplikácie pred spojením s ostatnými modulmi.

Zodpovedné osoby: Matej Červenka, Matúš Salát

2.1.3 Manažovanie plánovania úloh

Náplň práce: Zodpovedné osoby vystupujú pri plánovaní jednotlivých šprintov, v rámci ktorých sú vytvárané user stories a úlohy. Pri plánovaní pritom upozorňujú na prípadné riziká, ktoré by mohli ovplyvniť dodanie plánovanej časti produktu na konci šprintu.

Zodpovedné osoby: Matej Červenka, Mária Dragúňová

2.1.4 Manažovanie komunikácie

Náplň práce: Zodpovedná osoba vystupuje ako zástupca tímu v komunikáciách s externými osobami, napríklad poskytovateľmi softvérových nástrojov alebo pedagógmi, podporuje komunikáciu aj vo vnútri tímu návrhom vhodných komunikačných kanálov.

Zodpovedná osoba: Matej Červenka

2.1.5 Písanie projektovej dokumentácie

Náplň práce: Sledovanie požiadaviek na dokumentáciu projektu, návrh a vytvorenie šablóny pre jednotlivé dokumenty, vytvorenie štruktúry dokumentov. Zodpovedné osoby musia dbať na pravidelné vytváranie zápisníc zo stretnutí tímu, retrospektív šprintov a exportov z nástroja manažmentu úloh.

Zodpovedné osoby: Mária Dragúňová, Lukáš Šimek

2.1.6 Testovanie funkcionality produktu

Náplň práce: Vytváranie integračných testov na zabezpečenie overenia funkčnosti spojenia jednotlivých modulov do celku ale aj vytváranie testov pre overenie funkčnosti jednotlivých modulov.

Zodpovedná osoba: Ondrej Kudláč

2.1.7 Poskytovanie technickej podpory

Náplň práce: Priebežné sledovanie behu aplikácie na serveri, riešenie prípadných problémov. Zodpovedná osoba má za úlohu inštalovať na server potrebné komponenty a spravovať ich.

Zodpovedná osoba: Matúš Salát

2.1.8 Manažovanie verziovania

Náplň práce: Návrh práce s verziovacím systémom git, správa vytvoreného repozitára, prepájanie jednotlivých modulov implementovaných členmi tímu do ucelenej časti. Zodpovedná osoba je oprávnená prepájať jednotlivé verzie systému.

Zodpovedná osoba: Martin Šidlo

2.1.9 Manažér dizajnu

Náplň práce: Návrh dizajnu prezentačného materiálu tímu, najmä však webového sídla tímu a dizajnu výsledného produktu. Udržiavanie a správa webového sídla.

Zodpovedná osoba: Martin Šidlo

2.1.10 Manažovanie rizík

Náplň práce: Odhalenie a identifikácia rizík, ktoré môžu ovplyvniť plynulý beh vývoja produktu a návrh metód, ktorými je možné rizikám predchádzať.

Zodpovedná osoba: Lukáš Šimek

2.2 Podiel práce na dokumentácii

Meno	Vypracované časti dokumentácie	Celkový percentuálny podiel
Bc. Matej Červenka	Riadenie, metodiky, moduly	14%
Bc. Mária Dragúňová	riadenie, inžinierske dielo, preberací protokol	25%
Bc. Ondrej Kudláč	Riadenie, metodiky	14%
Bc. Matúš Salát	Riadenie, metodiky, moduly	14%
Bc. Martin Šidlo	Riadenie, metodiky, moduly	14%
Bc. Lukáš Šimek	Riadenie, metodiky, moduly	19%

3 Aplikácie manažmentov

3.1 Manažovanie vývoja metódou SCRUM

Keďže na vývoj produktu používame metódu SCRUM, na začiatku prvého šprintu bolo našou úlohou vybrať prvého *Scrum Mastera*. Vzhľadom k záujmu vyskúšať si túto manažérsku úlohu, sa prvý kandidát prihlásil dobrovoľne. Pri riadení projektu *Scrum Master* vykonáva nasledujúce úlohy:

- sledovanie postupu jednotlivých členov na ich úlohách, pravidelné komunikovanie s členmi tímu, či stíhajú včas dokončiť prácu, ktorú si vzali na starosti,
- komunikácia s jednotlivými členmi tímu
- identifikácia problémov a ich odstránenie – napríklad problém v komunikácii medzi niektorými členmi tímu.

Pri manažovaní je dôležité komunikovať s jednotlivými členmi a diskutovať ich prípadné pripomienky a problémy s riadením projektu, ktoré ich vyrušujú pri práci na projekte a následne pracovať na tom, aby boli čo najskôr odstránené. Riešenie problémov častokrát zahŕňa otvorenú diskusiu s členom tímu, ktorý je pôvodcom nespokojnosti iných členov a vytvorenie dohody, dodržanie ktorej bude viesť k zmierneniu alebo odstráneniu problémov alebo nespokojnosti.

3.2 Manažovanie plánovania úloh

Pre spravovanie úloh sme zvolili nástroj *YouTrack*¹. Ide o nástroj od spoločnosti *JetBrains*². Všetku prácu, na ktorej plánujeme počas nasledujúceho šprintu pracovať, si rozdeľujeme do jednotlivých *user stories*. Vytvorené *user stories* ohodnotíme počtom bodov, v závislosti od zložitosti danej úlohy. Ku každej *user story* je potrebné napísať podrobný opis, aby bolo jasné a zrejmé, čo daná úloha zahŕňa.

Po vytvorení všetkých *user stories* si rozdelíme prácu a určíme, kto sa danej *user story* bude venovať a kto bude za jej dokončenie zodpovedný. V *YouTracku* rozlišujeme šesť stavov, ktoré môžu úlohy nadobudnúť. *Open* pre úlohy, ktoré nie sú rozpracované, *In Progress* pre úlohy, na ktorých sa pracuje, *Code Review* pre úlohy, ktorým je potrebné spraviť prehliadku kódu, *To Test* pre úlohy určené na otestovanie, *To Verify* pre úlohy, ktoré sú dokončené a je potrebné odsúhlasenie zo strany majiteľa produktu a *Done* pre úlohy plne dokončené a odsúhlasené.

¹ <https://www.jetbrains.com/youtrack/>

² <https://www.jetbrains.com>

3.3 Manažovanie komunikácie

Pre spravovanie oficiálnej tímovej komunikácie medzi jednotlivými členmi tímu, ako aj vedúcimi tímu sme sa rozhodli pre nástroj *Slack*¹. Nástroj máme rozdelený do niekoľkých hlavných kanálov, ktoré slúžia na jednoduché sprostredkovanie a vyhľadanie informácií, týkajúcich sa danej témy. Samozrejme kanály nie sú jediný spôsob, ktorým je možné komunikovať, ale je umožnená komunikácia aj priamo medzi jednotlivými členmi tímu.

Na neformálne komunikácia využívame predovšetkým uzavretú skupinu na *Facebooku*². Dohadujeme sa tu na stretnutiach a posúvame sem dôležité informácie, ktoré sa nachádzajú aj na *Slacku* a je potrebné ich skoré doručenie k ostatným členom.

Ďalší spôsob komunikácie je e-mail, na čo slúži náš tímový e-mail *tim3fit@googlegroups.com*. Tento e-mail je využívaný hlavne na externú komunikáciu mimo tímu

3.4 Manažovanie verziovania

Pre manažment správy verzii sme sa po diskusii s vedúcim na prvom stretnutí dohodli, na používaní GIT-u s webovým nástrojom *GitHub*³. Tento nástroj nám poskytuje grafické rozhranie, ktoré poskytuje rôzne prehľady o činnostiach používateľov a taktiež jednoduchú a účinnú správu verzii projektu.

Okrem iného vieme riadiť prístup do produkčnej verzii projektu, čo nám zabezpečí taktiež ochranu pred potencionálne spôsobenou chybou ľudským faktorom. Tento nástroj vie okrem zobrazenia repozitáru aj vytvárať základné grafy, štatistiky či ohlasovať chyby.

3.5 Manažment kvality kódu

Kvalita kódu je kontrolovaná na niekoľkých úrovniach. Prvou z nich je kontrola syntaxe a rozloženia kódu v zdrojových súborov počas prehliadok kódu. Prehliadky sa dejú po dokončení zmien na zdrojovom kóde vo verziovacom systéme *GitHub* vytvorením pull requestu a priradením recenzenta kvality kódu. Recenzent kvality kódu a vývojár sa v tomto prípade riadia metodikami *Písanie kódu v Ruby on Rails* a *Overovanie kódu*. Kvalita kódu sa overuje aj testovaním, ktoré rieši metodika *Testovanie a Manažovanie testovania*. Overenie kvality kódu končí akceptačným overením produktového vlastníka na pravidelnom stretnutí.

¹ <https://slack.com>

² <https://facebook.com>

³ <https://github.com>

3.6 Manažovanie testovania

3.6.1 Úprava špecifikácie

Každý člen tímu pred rozpracovaním svojej *user story* si premyslí, či nedošlo/nedôjde k zmene špecifikácie a nie je potrebné na to reagovať zrušením alebo prepísaním časti testovacej sady. Jedná sa najmä o zmeny v modeli alebo rozšírenie/zmenenie už fungujúcej a otestovanej funkcionality. V prípade, že došlo/dôjde k zmene, je potrebné kontaktovať člena tímu zodpovedného za vytváranie testov a o tejto skutočnosti ho informovať.

3.6.2 Implementácia novej funkcionality

Pred implementáciou novej funkcionality je povinný každý člen tímu kontaktovať testera a dohodnúť s ním podrobnosti na úrovni bielej skrinky k príprave základných testov. V tejto fáze sa dohodnú validné vstupy/výstupy. Po implementácii vývojár vznesie požiadavku na testera a požiada o vytvorenie testov. Tester pripraví testovaciu sadu. Programátor zodpovedný za danú funkcionality je povinný pred *pull requestom* pustiť predom pripravenú testovaciu sadu k funkcionalite, ale aj ostatné testy k overeniu, že neprišlo k nežiadanej ovplyvneniu behu programu mimo novú implementáciu.

3.6.3 Reagovanie v prípade neúspešného testu

V prípade, že po spustení testov niektorý test neprejde, je potrebné spraviť revíziu kódu a hľadať dôvod, prečo test neprešiel. Hľadanie chyby začína v kóde, ktorý má priamo túto funkcionality na starosti. V prípade, že sa jedná o maličkosť alebo preklep môže chybu opraviť priamo tester čo púšťal testovaciu sadu. V prípade nutnosti väčšieho zásahu sa kód vracia priamo jeho autorovi. V prípade, že sa nejedná o nesprávne správanie, ale je test zastaraný alebo napísaný zle, je potrebné vrátiť na prepracovanie test jeho autorovi.

3.6.4 Reakcia v prípade, že boli testy vykonané bez chyby

Po spustení testov a prípade, kedy kód prejde cez všetky testy, tak je kód uvoľnený k prechodu do ďalšej fázy.

3.7 Manažovanie rizík

Za riziko sa pokladá činnosť, ktorej výsledok môže byť neistý a v prípade neúspechu hrozia následky, ktoré môžu mať negatívny vplyv.

Pri analýze rizík, je potrebné určiť pravdepodobnosť, že riziko nastane, ako aj rozsah dopadu. Spolu s tým súvisia opatrenia, ktoré je potrebné podstúpiť, aby riziko nenastalo a veľmi dôležitou časťou je zmiernenie negatívnych vplyvov, ak sa riziku nepodarí zabrániť.

Označenie	#01
Dopad:	stredný
Pravdepodobnosť:	stredná
Názov	Člen tímu je zo zdravotných dôvodov neschopný pracovať
Spúšťač	Ochorenie
Súvislosť	Neschopnosť práce na projekte
Dôsledok	Väčšie zaťaženie ostatných členov
Opatrenie	Teplé oblečenie, vitamíny
Zmiernenie	Prebratie úloh inými členmi

Označenie	#02
Dopad:	vysoký
Pravdepodobnosť:	stredná
Názov	Člen tímu nie je schopný dokončiť úlohu
Spúšťač	Nedostatok vedomostí
Súvislosť	Úloha je rozpracovaná, ale člen tímu ju nevie dotiahnuť do úspešného konca
Dôsledok	Chybný modul pre výsledný projekt
Opatrenie	Vzájomná diskusia, študovanie problematiky
Zmiernenie	Pomoc od skúsenejšieho člena tímu v danej problematike

Označenie	#03
Dopad:	stredný
Pravdepodobnosť:	vysoká
Názov	Nesprávne rozvrhnutie času pre danú úlohu
Spúšťač	Zlé odhadnutie náročnosti úlohy
Súvislosť	Úloha sa začne riešiť neskoro a do požadovaného termínu sa nedá stihnúť
Dôsledok	Oneskorené dodanie úlohy
Opatrenie	Včasná práca na zadanej úlohe
Zmiernenie	Veľké nasadenie pri dokončovaní úlohy

Označenie	#04
Dopad:	stredný
Pravdepodobnosť:	Nízka
Názov	Absencia na dôležitých stretnutiach
Spúšťač	Ochorenie, zdravotné problémy, iný dôvod
Súvislosť	Vyskytne sa udalosť alebo ochorenie z dôvodu ktorého sa člen tímu nemôže zúčastniť na stretnutí
Dôsledok	Člen tímu nemá informácie o aktuálnosti projektu
Opatrenie	Teplé oblečenie, vitamíny, plánovanie iných aktivít na čas mimo stretnutí
Zmiernenie	Live komunikácia cez Skype/Slack, včasné informovanie o stretnutí

Označenie	#05
Dopad:	Vysoký
Pravdepodobnosť:	nízka
Názov	Zlyhanie softvéru
Spúšťač	Nesprávna manipulácia
Súvislosť	Pri práci na zadaní sa môže nedopatrením pokaziť nainštalovaný softvér, poprípade iné softvérové súčasti a člen tímu nebude schopný ďalej v úlohe pokračovať
Dôsledok	Neskoré dokončenie úlohy
Opatrenie	Používať zavedené postupy, vyhýbať sa vírusom
Zmiernenie	Preinštalovanie postihnutého softvéru

Označenie	#06
Dopad:	Vysoký
Pravdepodobnosť:	nízka
Názov	Zlyhanie hardvéru
Spúšťač	Nesprávna manipulácia, pokazenie hardvéru
Súvislosť	Môže nastať samovoľné pokazenie počítaču
Dôsledok	Úplná neschopnosť pracovať na projekte, strata dát
Opatrenie	Zálohovanie dát, starostlivosť o hardvér
Zmiernenie	Dočasné požičanie notebooku

Označenie	#07
Dopad:	stredný
Pravdepodobnosť:	stredná
Názov	Zlyhanie pripojenie na internet
Spúšťač	Nestabilné pripojenie
Súvislosť	Neschopnosť komunikácie s tímom, neschopnosť odovzdanie úloh, neschopnosť práce na úlohách
Dôsledok	Neinformovanosť, neodovzdané úlohy
Opatrenie	Mobilný internet v zálohe
Zmiernenie	Zvýšiť úsilie pri dokončovaní úloh

Označenie	#08
Dopad:	Vysoký
Pravdepodobnosť:	Vysoká
Názov	Nesplnenie úloh pre daný šprint
Spúšťač	Na úlohách sa dostatočne nepracovalo, vyskytli sa iné neočakávané problémy
Súvislosť	Stav projektu v záveru šprintu nie je taký, ako sme si predstavovali
Dôsledok	Presunutie úloh do ďalšieho šprintu, nedodanie potrebnej funkcionality
Opatrenie	Včasná práca na úlohách, komunikácia v tíme
Zmiernenie	Prebratie úloh do ďalšieho šprintu a mať v ňom viac práce, ako za normálnych okolností

4 Sumarizácie šprintov

V tejto časti dokumentácie zhrnieme jednotlivé šprinty. Pridáme prehľadné tabuľky, ktoré budú obsahovať informácie o *user stories* a retrospektívny pohľad na každý šprint. Keďže sa náš tím volá *Martians*, názvy šprintov sme sa rozhodli vytvárať na základe mien známych mimozemšťanov.

4.1 1. šprint - E.T

Prvý šprint sa niesol v znamení práce na jednotlivých častiach projektu, kde nikto nebol od nikoho závislý. Bolo potrebné nainštalovať server a následne naň nasadiť tímovú stránku. Ďalej sme sa venovali prvotnej analýze dát, ktoré nám boli poskytnuté od Martinusu. Vytváraniu základného GUI pre výsledný produkt a v neposlednom rade vytváraniu dátových modelov a integrácii dátovej analýzy a aplikácie.

User Story	Zodpovednosť
Inštalácia na server	Matúš Salát
Import dát z Martinusu	-
Integrácia programu na dátovú analýzu do aplikácie	Ondrej Kudláč
Prvotná verzia GUI	Matej Červenka
Nasadenie tímovej stránky na server	Martin Šidlo
Dátová analýza	Lukáš Šimek
Vytvorenie dátového modelu	Mária Dragúňová

Tabuľka č.1 *User stories, ktoré boli navrhnuté a ohodnotené v rámci plánovania 1. šprintu*

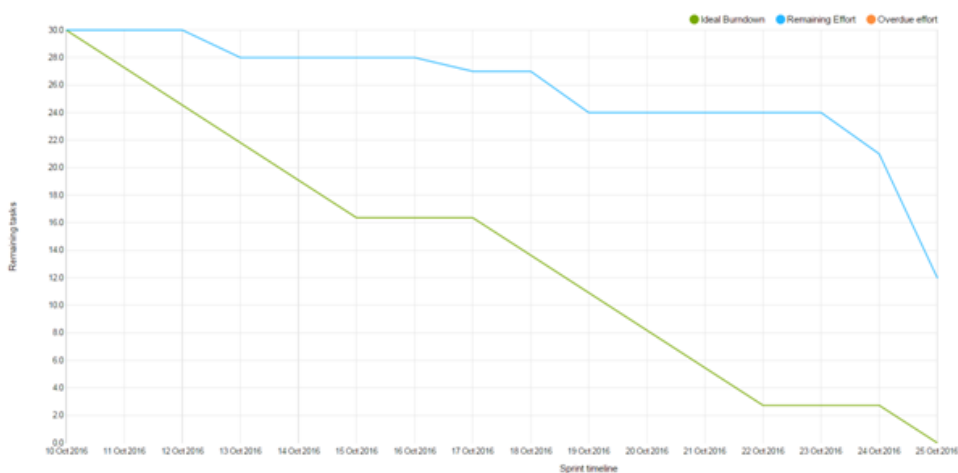
4.1.1 Retrospektíva

START	Pribežné spravovanie svojej user story v YouTracku Pribežné pracovanie na svojej user story Zlepšenie práce na dokumentácií Zlepšenie popisu user story pri vytváraní
KEEP	Komunikácia v tíme Komunikácie s vlastníkom produktu Vytváranie zápisníc
STOP	Komentovanie kódu a commitov v slovenskom jazyku

Tabuľka č.2 *Tabuľka zobrazujúca prehľad retrospektívy 1. šprintu*

4.1.2 Burndown chart

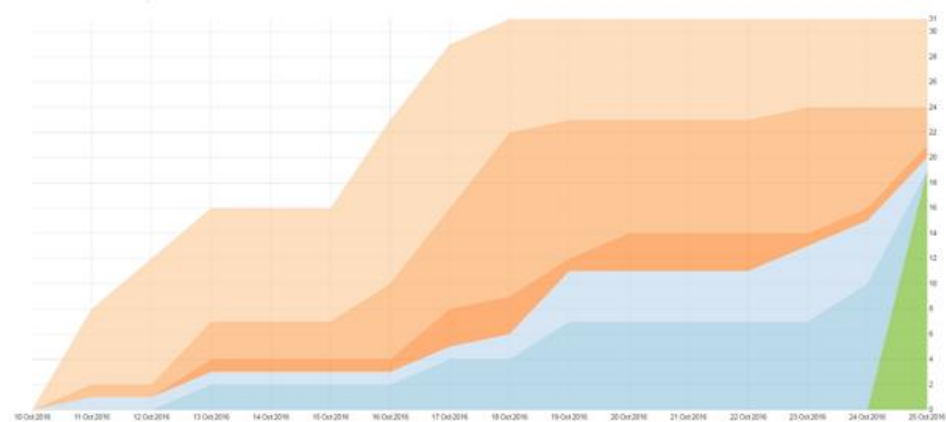
Zelená krivka znázorňuje ideálny priebeh a progres šprintu. Modrá krivka predstavuje náš reálny progres, ktorým sa naša práca počas prvých dvoch týždňov uberala. Pri plánovaní prvého šprintu sme presne neodhadli počet *user stories*, ktoré sme schopní úspešne zvládnuť. Jedna *user story* zostala v štádiu rozpracovania, a preto sa modrá krivka nedostala na úplný spodok grafu.



Obrázok č.1 Burndown chart zobrazujúci priebeh 1.šprintu

4.1.3 Cumulative flow

Dve najbledšie oranžové farby reprezentujú *tasky*, na ktorých sa stále pracuje. Bledomodrá farba predstavuje *tasky*, ktoré sú pripravené na otestovania a zaradenie do sekcie *Done*. Tmavšia modrá predstavuje *tasky*, ktoré je potrebné verifikovať zo strany majiteľa produktu. Posledné zelená farba predstavujú *tasky*, ktoré sú splnené. Ako už bolo spomenuté, neporadilo sa nám zvládnuť jednu *user story*, preto sa zelená časť grafu nedostala na úplný vrchol.



Obrázok č.2 Cumulative flow vytvorený počas 1. šprintu

4.1.4 Zhodnotenie prvého šprintu

Počas šprintu sa nám podarilo dosiahnuť takmer všetky vytýčené ciele. Pracovali sme hlavne na jednotlivých častiach produktu, takže zatiaľ nebol nikto závislý na výsledkoch druhého člena. Neúspešné zvládnutie jednej user story nepredstavuje veľký problém, pretože jej dokončenie je možné v nasledujúcom šprinte, bez akéhokoľvek zdržania pri iných úlohách. Ostatné user story sa podarilo úspešne zvládnuť.

4.2 2. šprint - Spock

Cieľom druhého šprintu bolo vyhodnotiť výsledky, ktoré sa nám na základe dátovej analýzy podarilo dosiahnuť. Vytvoriť a dokončiť funkčné GUI pre zobrazovanie predikovaných informácií, zabezpečiť import dát z Martinusu a dokončiť úpravy v dátovom modeli.

User Story	Zodpovednosť
Vyhodnotenie súčasnej implementovanej metódy na dátovú analýzu	Lukáš Šimek
Dopárovať názov produktu k jeho ID	Mária Dragúňová
Integrácia programu na dátovú analýzu do aplikácie	Ondrej Kudláč
Dokončenie GUI pre zobrazenie predikcií	Matej Červenka
Import dát z Martinusu	Martin Šidlo

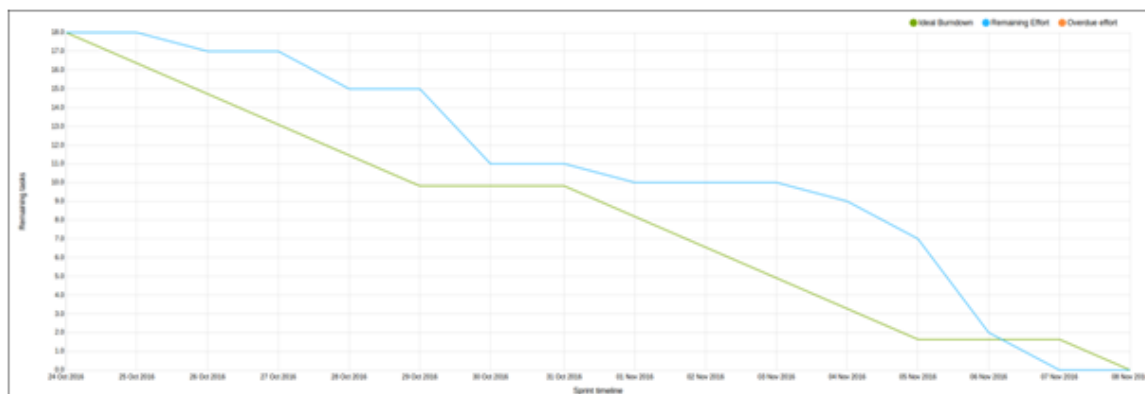
Tabuľka č.3 *User stories vytvorené počas plánovania 2.šprintu*

4.2.1 Retrospektíva

START	Dokončovanie taskov skôr ako v posledný deň (ideálne posielanie v sobotu na revíziu kódu) Pravidelné <i>team buildingy</i> Podávať správu o svojej práci v polke šprintu Lepšie využívanie <i>Slacku</i> Používanie tagov šprintu na <i>Githube</i>
KEEP	Odhodlanie
STOP	Menej konverzácie na <i>Facebooku</i>

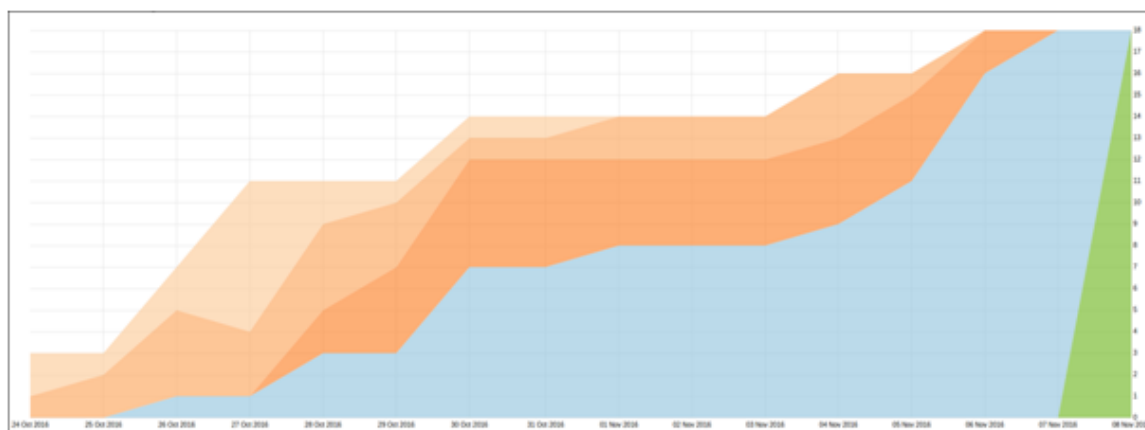
Tabuľka č.4 *Prehľad retrospektívy 2. šprintu*

4.2.2 Burndown chart



Obrázok č.3 Burndown chart zobrazujúci priebeh 2.šprintu (opis grafu: sekcia 4.1.2)

4.2.3 Cumulative flow



Obrázok č.4 .Cumulative flow zobrazujúci 2. šprint (opis grafu: sekcia 4.1.3)

4.2.4 Zhodnotenie druhého šprintu

Počas celého šprintu sme pracovali na svojich *taskoch* a podarilo sa nám dosiahnuť všetky vytýčené ciele. Hlavným cieľom bolo vytvoriť pospájaný produkt z jednotlivých súčastí z minulého šprintu. Tento cieľ sa nám podarilo z väčšej časti dokončiť a úspešne sme zvládli druhý šprint. Jeho výstupom bol celistvý produkt, ktorý dokáže spracovať získané dáta a predikované hodnoty zobrazovať v prehľadnej tabuľke.

4.3 3. šprint - Kal-El

Cieľom tohto tretieho šprintu bola úprava používateľského rozhrania čo zahŕňalo, filtrácia pre rôznych pracovníkov, filtrácia podľa dodávateľov a zobrazovanie grafov predajov. Ďalej sa vytvorili testy, zdokumentoval sa kód, upravil sa import produktov a aplikácia začala automaticky spracovávať všetky údaje, ktoré nám chodia každý deň z Martinusu.

User Story	Zodpovednosť
Graf predajov	Martin Šidlo
Import predajov	Martin Šidlo
Dostať dáta na produkciu	Mária Dragúňová
Filtrovanie	Matej Červenka
Dokumentácia	všetci

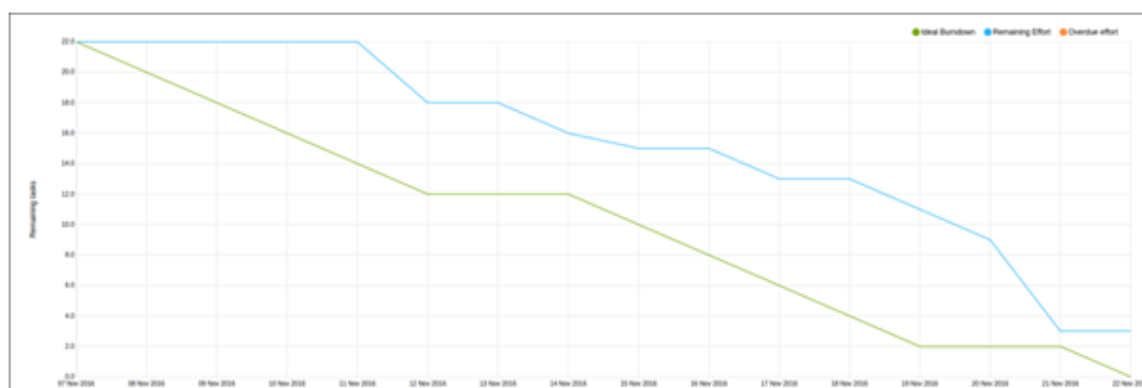
Tabuľka č.5 *User stories vytvorené počas plánovania 3. šprintu*

4.3.1 Retrospektíva

START	Vytvárať zálohu databázy (dump) Lepšie komentovanie vlastného kódu Zaznamenávať bugy v <i>YouTracku</i> (Uncategorized) Priebežne hovoriť o problémoch pri <i>tasku</i> , nečakať na tímové stretnutia Priradovať človeka na revíziu kódu pri <i>pull requeste</i>
KEEP	Zvýšená komunikácia na <i>Slacku</i>
STOP	<i>Pull requesty</i> na <i>Githube</i> na poslednú chvíľu

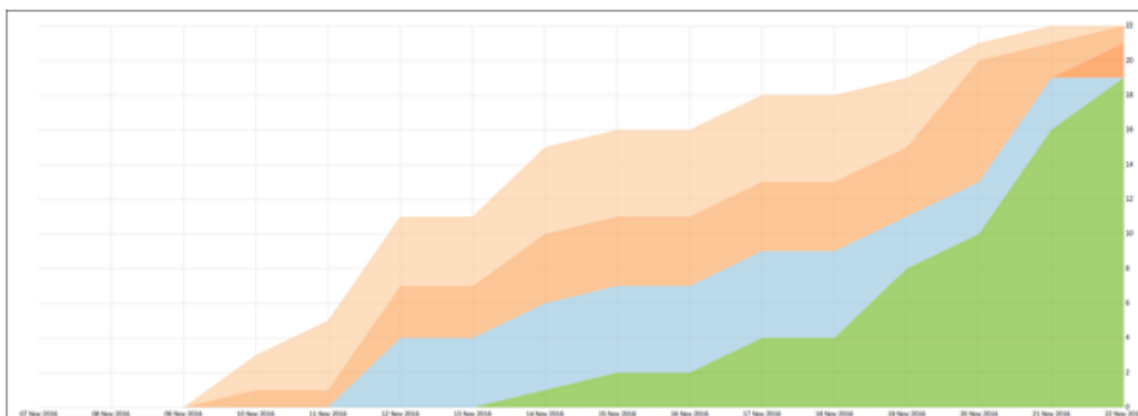
Tabuľka č.6 *Prehľad retrospektívy 3. šprintu*

4.3.2 Burndown chart



Obrázok č.5 *Burndown chart zobrazujúci priebeh 3. šprintu (opis grafu: sekcia 4.1.2)*

4.3.3 Cumulative flow



Obrázok č.6 Cumulative flow zobrazujúci 3. šprint (opis grafu: sekcia 4.1.3)

4.3.4 Zhodnotenie tretieho šprintu

Počas celého šprintu sme pracovali na svojich *taskoch* a podarilo sa nám dosiahnuť všetky vytýčené ciele, spojené s rozširovaním aplikácie. *User story* spojený s testami sa nám nepodarilo úplne dokončiť, ale hlavné ciele šprintu sme zvládli dotiahnuť do úspešného konca. Celkovo bol tretí šprint úspešný, pretože sme dosiahli úroveň aplikácie, ktorá je použiteľná na prvé testovanie v praxi a náš produkt bude v najbližších dňoch predstavený pracovníkom skladu.

4.4 Šprint 4: Chewbacca

Cieľom štvrtého šprintu bolo vylepšiť súčasný prototyp na predikciu objednávok *SmartOrder* a začať s návrhom produktu *SmartCollect* a vytvoriť pre tento produkt prvý prototyp.

User Story	Zodpovednosť
Reprezentácia skladu, aby sa v ňom dala hľadať dobrá cesta	Matúš Salát
Zoradiť objednávky podľa najlepšej trasy v sklade	Martin Šidlo
Analýza predikcie pri veľkých zmenách v počte objednávok	Lukáš Šimek
Modifikácia metódy predikcie	Ondrej Kudláč

Tabuľka č.7 *User stories* vytvorené počas plánovania 4. šprintu

4.4.1 Retrospektíva

START	Zapisovanie stráveného času pri taskoch Priebežná práca na dokumentácií
KEEP	Komunikácia v tíme Komentáre ku kódu
STOP	Pull Requesty na githubu na poslednú chvíľu

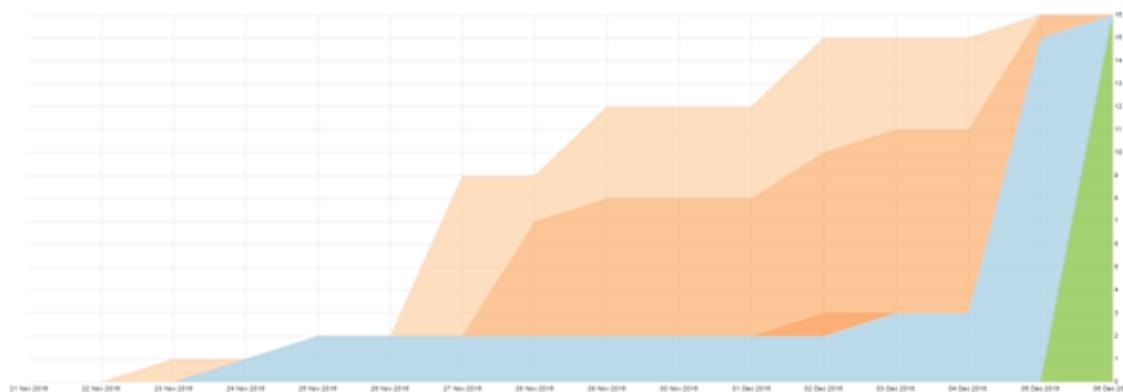
Tabuľka č.8 *Prehľad retrospektívy 4. šprintu*

4.4.2 Burndown chart



Obrázok č.7 Burndown chart zobrazujúci priebeh 4. šprintu (opis grafu: sekcia 4.1.2)

4.4.3 Cumulative flow



Obrázok č.8 Cumulative flow zobrazujúci 4. šprint (opis grafu: sekcia 4.1.3)

4.4.4 Zhodnotenie šprintu

Počas celého šprintu sme pracovali na svojich taskoch a podarilo sa nám dosiahnuť takmer všetky vytýčené ciele, spojené s rozširovaním aplikácie o novú funkcionality. Konkrétne sa jedná o implementovanie zobrazenia najrýchlejšej nožnej cesty na vybavenie objednávky. Podarilo sa nám úspešne reprezentovať sklad a jeho rozmiestnenie. Taktiež sme pracovali na vylepšovaní doterajšej predikcie a vytváraní vyhodnocovacích skriptov, určených na porovnanie.

4.5 Šprint 5: Yoda

Cieľom piateho šprintu bolo úspešne zakončiť semestrálnu prácu na našom projekte tým, že dokončíme automatizované testovanie a budeme pokračovať vo vylepšovaní predikcie a navigácii pracovníka v sklade.

User Story	Zodpovednosť
Automatizované testovanie scenárov	Matej Červenka
Implementácia API služby	Martin Šidlo
Modifikácia reprezentácie skladu	Matúš Salát
Analýza predikcie pri veľkých zmenách v počte objednávok	Lukáš Šimek
Modifikácia metódy predikcie	Ondrej Kudláč

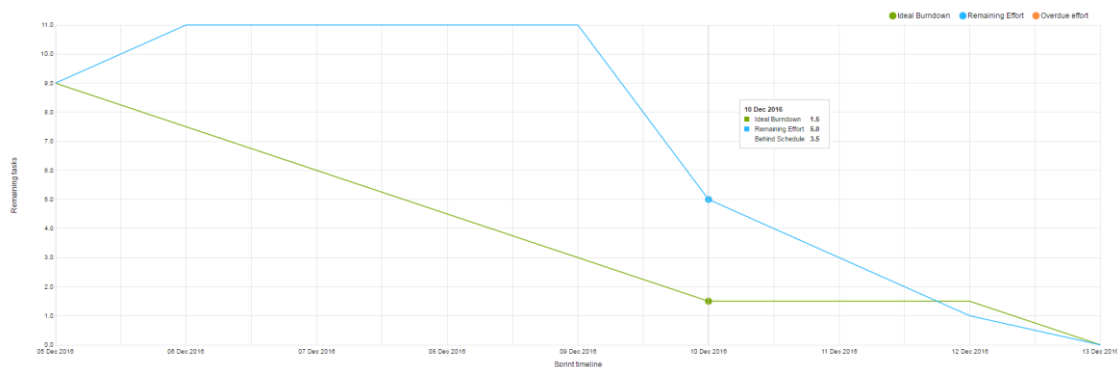
Tabuľka č.9 *User stories vytvorené počas plánovania 5. šprintu*

4.5.1 Retrospektíva

START	Vyskúšať písať dokumentáciu vo OneDrive Výber nástroja na kontinuálnu integráciu Konfigurácia systémovej integrácie Využívať "Code reviewer" - nová funkcionlita na GitHube
KEEP	Pokračovať v písaní testov
STOP	Prestať používať WordOnline

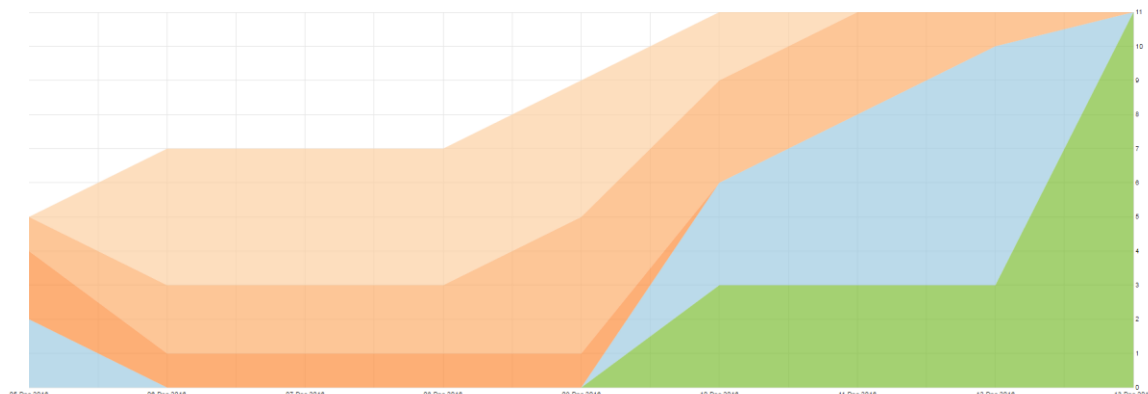
Tabuľka č.10 *Prehľad retrospektívy 5. šprintu*

4.5.2 Burndown chart



Obrázok č.9 *Burndown chart zobrazujúci priebeh 5. šprintu (opis grafu: sekcia 4.1.2)*

4.5.3 Cumulative flow



Obrázok č.10 Cumulative flow zobrazujúci 5. šprint (opis grafu: sekcia 4.1.3)

4.5.4 Zhodnotenie šprintu

Išlo síce len o týždňový, ale nijako sa to neodrazilo nad kvalitou v dodanej iterácii. Počas šprintu sme všetci pracovali na svojich taskoch a úspešne sme ich zvládli všetky dokončiť. Vo veľkej miere sa pracovali hlavne na vytváraní testov a správnym a hlavne rýchlim nájdením ideálnej cesty v sklade. Obe tieto primárne úlohy boli zvládnuté. Hľadanie cesty v sklade bola predvedené product ownerovi a schválené z jeho strany. Celkovo tento krátky šprint považujeme za veľmi úspešný.

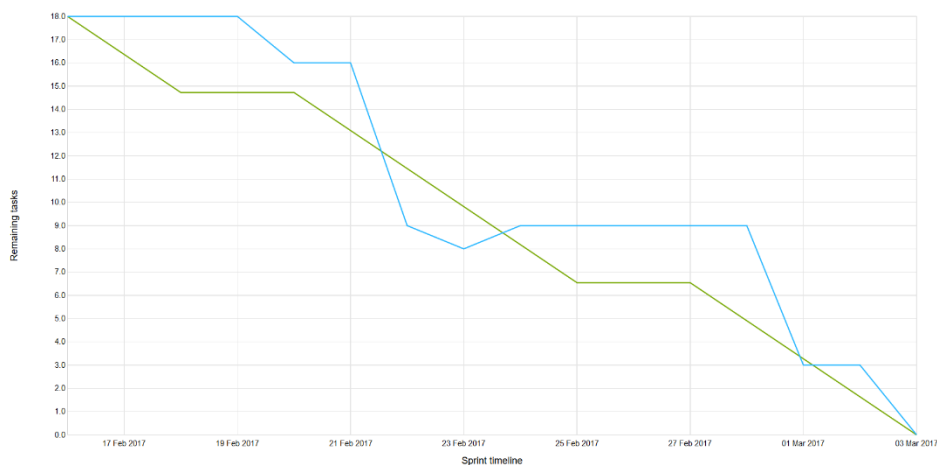
4.6 Šprint 6: Han Solo

Cieľom šprintu bolo prvotné navrhnutie výpočtu popularity produktu a vytvorenie služby na zistenie sektora produktu. V module na predikciu sme pokračovali vo vyhodnocovaní našej metódy.

User Story	Zodpovednosť
SmartOrder: Porovnanie našej metódy s Martinusom	Lukáš Šimek
SmartCollect: Výpočet popularity produktu	Mária Dragúňová
SmartCollect: Zistenie polohy produktu	Matej Červenka
SmartCollect: Vylepšenie dostupnosti služieb	Matúš Salát
SmartCollect: Výpis celej cesty	Martin Šidlo
SmartCollect: Služba na zistenie sektora daného produktu	Ondrej Kudláč

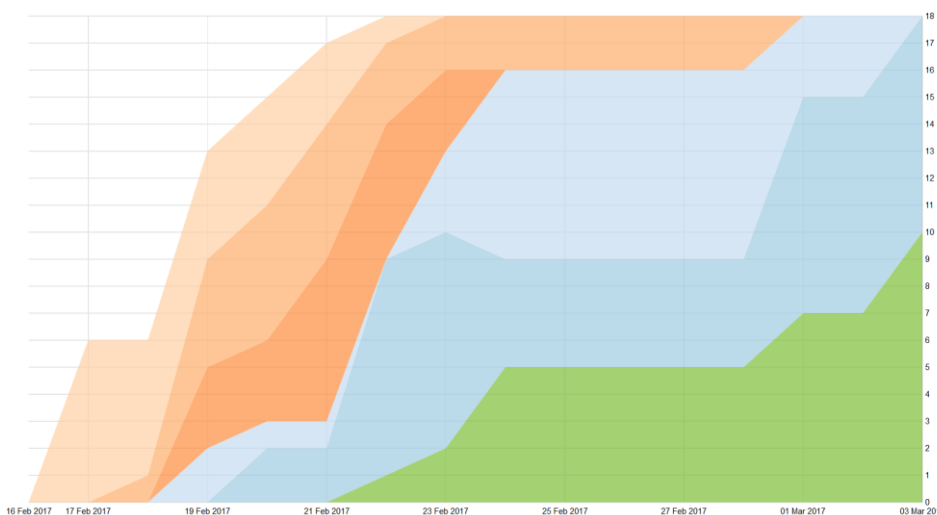
Tabuľka č.11 User stories vytvorené počas plánovania 6. šprintu

4.6.1 Burndown chart



Obrázok č.11 Burndown chart zobrazujúci priebeh 6. šprintu (opis grafu: sekcia 4.1.2)

4.6.2 Cumulative flow



Obrázok č.12 Cumulative flow zobrazujúci 6. šprint (opis grafu: sekcia 4.1.3)

4.6.3 Zhodnotenie šprintu

Prvý šprint nového semestra bol zameraný predovšetkým na SmartCollect. Zamerali sme sa hlavne na počiatočné úlohy spojené s rozmiestnením produktov v sklade, ako určenie populárnejších produktov, či vytvorenie nových služieb pre túto časť projektu. Všetky dôležité úlohy sme v čas dokončili a úspešne začali novú etapu šprintov.

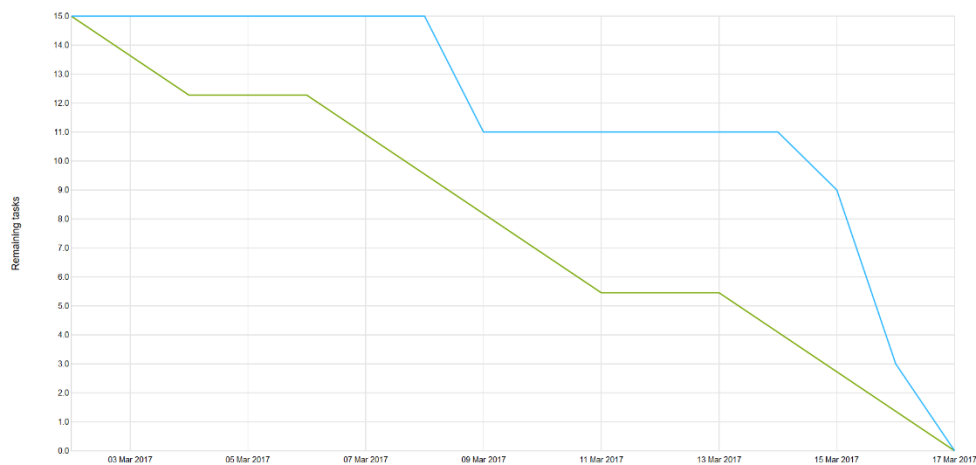
4.7 Šprint 7: Alien

Cieľom siedmeho šprintu bolo spraviť prvotnú analýzu a návrh výpočtu dátovej analýzy nákupného košíka, preskúmať možné knižnice v jazyku R alebo prípadné alternatívy. Chceli sme tiež pokračovať v module na výpočet najkratšej cesty a urobiť vhodnú vizualizáciu tejto cesty sklalom.

<i>User Story</i>	<i>Zodpovednosť</i>
SmartCollect: Dátová analýza nákupného košíka	Lukáš Šimek, Mária Dragúňová, Matej Červenka, Ondrej Kudláč
SmartCollect: Nasadenie a monitoring služby na získanie sektora	Matúš Salát
SmartCollect: Refaktoring vzdialeností medzi policami	Matúš Salát
SmartCollect: Vizualizácia cesty v sklade	Martin Šidlo
Modifikácia metódy predikcie	Ondrej Kudláč

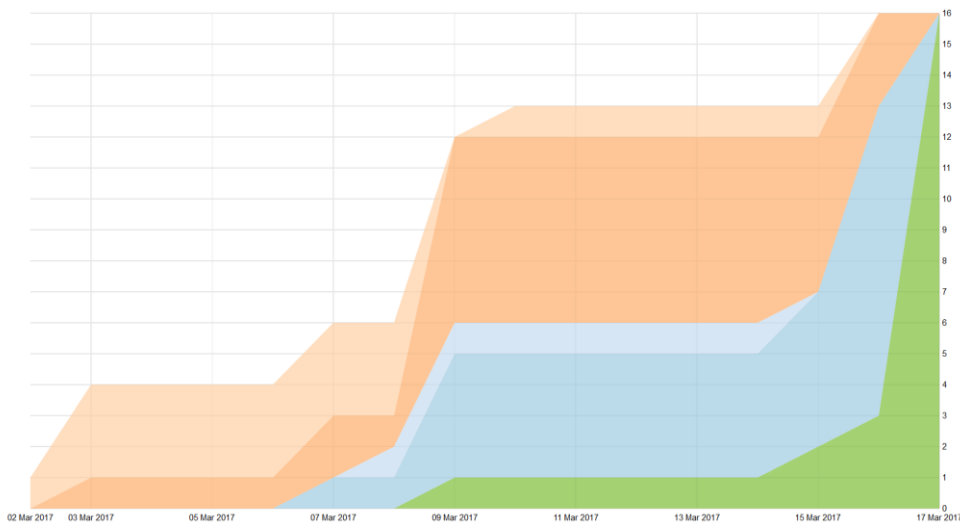
Tabuľka č.12 *User stories* vytvorené počas plánovania 7. šprintu

4.7.1 Burndown chart



Obrázok č.13 *Burndown chart* zobrazujúci priebeh 7. šprintu (opis grafu: sekcia 4.1.2)

4.7.2 Cumulative flow



Obrázok č.14 Cumulative flow zobrazujúci 7. šprint (opis grafu: sekcia 4.1.3)

4.7.3 Zhodnotenie šprintu

Daný šprintu odštartoval úlohy spojené s analýzou nákupného košíka. Nakoľko sme testovali rôzne možnosti, do danej úlohy bolo zapojených viacero ľudí. V závere sme vybrali najvhodnejší spôsob, ktorým používame na nájdenie spolu predávaných produktov. V tomto šprinte sme sa taktiež venovali vizualizácii cesty v sklade, ktorá bude tvoriť podstatnú časť pri vybavovaní objednávok a uľahčení práce zamestnancom. Všetky úlohy sa nám podarilo dokončiť včas a v požadovanej kvalite.

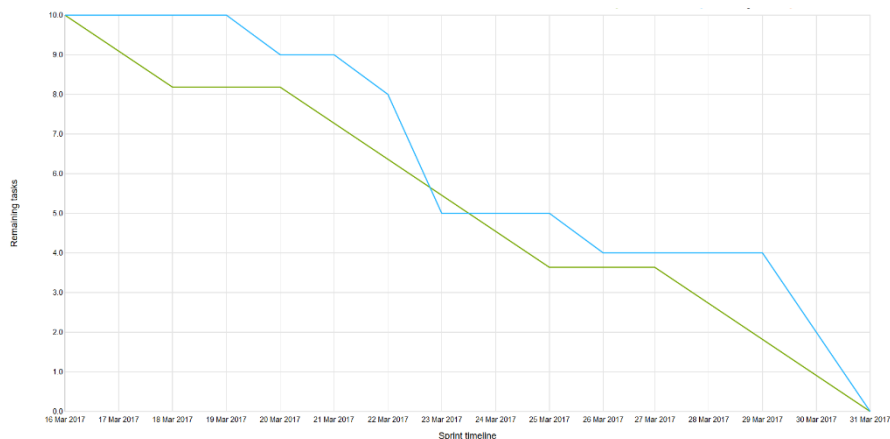
4.8 Šprint 8: Alf

Cieľom ôsmeho šprintu bolo dokončiť v požadovanom formáte vykresľovanie cesty v sklade a vytvorenie mobilnej aplikácie na prezentačné účely na IIT.src. Tiež sme sa chceli ďalej sústrediť na vybranú metódu dátovej analýzy nákupného košíka.

<i>User Story</i>	<i>Zodpovednosť</i>
SmartCollect: Vykresľovanie cesty	Matúš Salát
SmartCollect: Mobilná aplikácia na skenovanie kódu	Martin Šidlo
SmartCollect: Analyza rozdelenia sektorov	Mária Dragúňová, Matej Červenka
Analýza nákupného košíka	Lukáš Šimek, Ondrej Kudláč

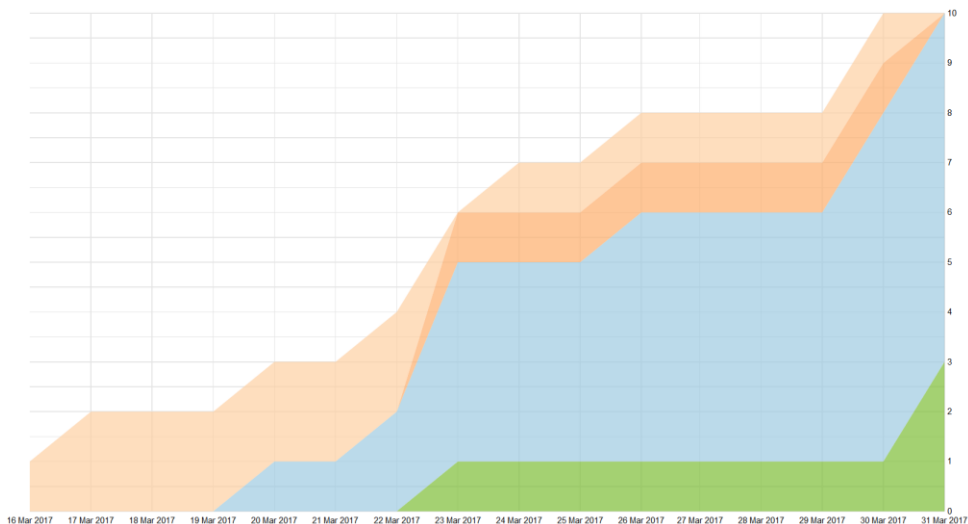
Tabuľka č.13 User stories vytvorené počas plánovania 8. šprintu

4.8.1 Burndown chart



Obrázok č.15 Burndown chart zobrazujúci priebeh 8. šprintu (opis grafu: sekcia 4.1.2)

4.8.2 Cumulative flow



Obrázok č.16 Cumulative flow zobrazujúci 8. šprint (opis grafu: sekcia 4.1.3)

4.8.3 Zhodnotenie šprintu

Šprint bol zameraný hlavne na vizuálnu časť a pokračovanie v analýze nákupného košíka. Vykresľovanie cesty ako aj navrhnutie a vytvorenie mobilnej aplikácie sme úspešne zvládli. Analýza nákupného košíka sa ukázala ako zložitejším problémom, ktorý sme nakoniec riešili viacero šprintov. Ale vo výsledku sme ju úspešne zvládli a navrhli možné riešenie

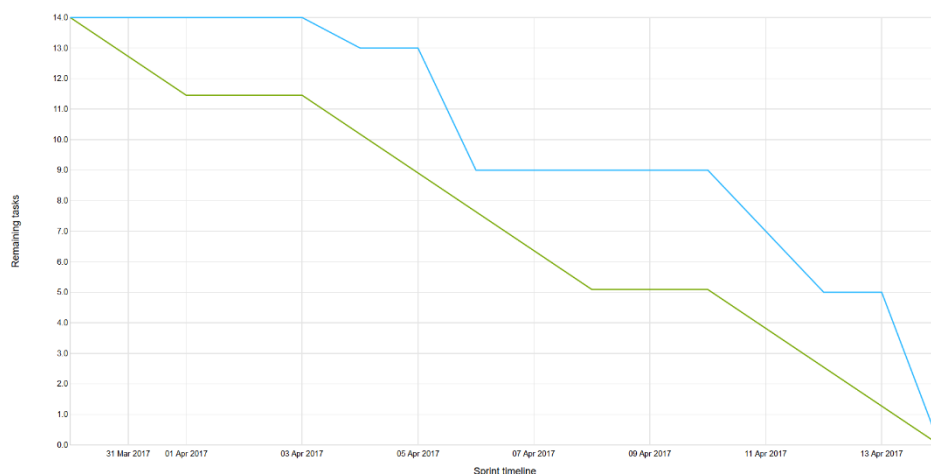
4.9 Šprint 9: Prawn

Cieľom deviateho šprintu bolo pokračovať v dlhodobej úlohe na optimálne rozmiestnenie skladu. Naplánovali sme ciele na presnejšie umiestnenie produktu v rámci skladu a vypísanie všetkých produktov, ktoré sú najviac nevhodne umiestnené.

<i>User Story</i>	<i>Zodpovednosť</i>
SmartCollect: Presnejšie umiestnenie produktu v rámci sektora	Lukáš Šimek, Ondrej Kudláč
SmartCollect: Vypis všetkých zle umiestnených produktov	Mária Dragúňová, Matej Červenka, Matúš Salát
SmartCollect: Evaluacia rozmiestnenia produktov	Martin Šidlo
SmartCollect: Vizualizacia umiestnenia produktu	Ondrej Kudláč

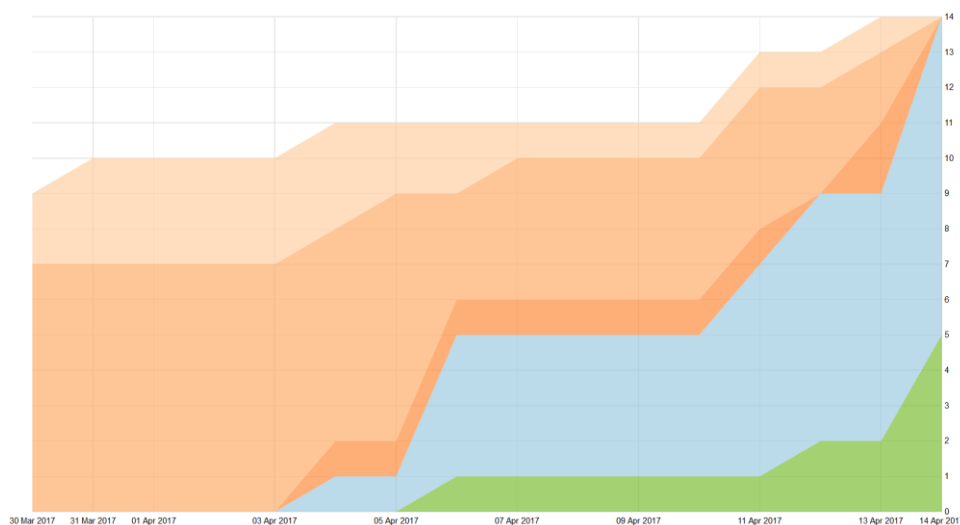
Tabuľka č.14 *User stories* vytvorené počas plánovania 9. šprintu

4.9.1 Burndown chart



Obrázok č.17 *Burndown chart* zobrazujúci priebeh 9. šprintu (opis grafu: sekcia 4.1.2)

4.9.2 Cumulative flow



Obrázok č.18 Cumulative flow zobrazujúci 9. šprint (opis grafu: sekcia 4.1.3)

4.9.3 Zhodnotenie šprintu

Celý šprint bol zameraný výlučne na rozmiestnenie a uloženie produktov v sklade. Venovali sme problémom od vytvorenie vizualizácie umiestnenia produktov, cez ich evaluáciu, až po výpis všetkých nesprávne umiestnených produktov. V závere sme boli úlohy schopní v čas vyriešiť

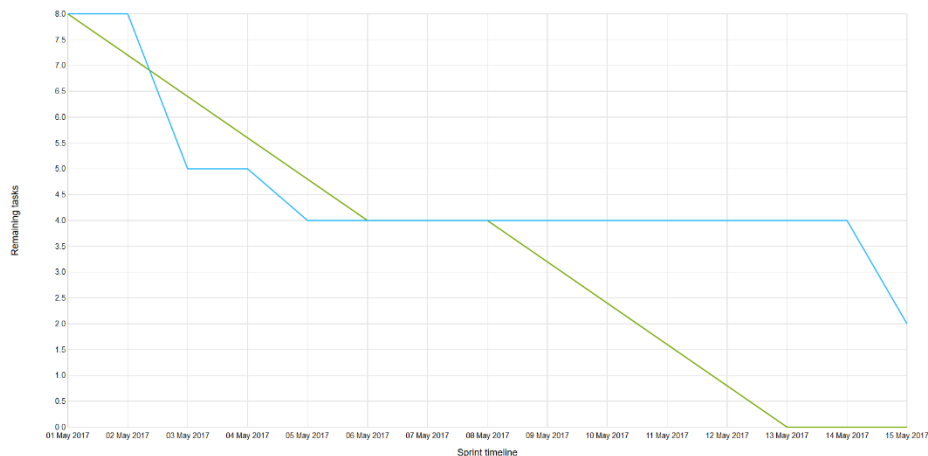
4.10 Šprint 10: Yoda

Cieľom tohto šprintu bolo úspešné odprezentovanie nášho projektu na konferencii IIT.src v rámci súťaže TPCup. Na to, aby bola prezentácia úspešná, venovali sme čas príprave plagátu aj článkom na zverejneným na web stránkach. Cieľom bolo tiež dopracovať a vyhodnotiť prínos optimálneho rozloženia skladu

<i>User Story</i>	<i>Zodpovednosť</i>
IIT.src: Dotazníky a prezentácie	Lukáš Šimek, Mária Dragúňová, Matúš Salát
SmartCollect: Vyhodnotenie a vizualizácia spolupredávaných produktov	Lukáš Šimek, Ondrej Kudláč
SmartCollect: Optimálne rozmiestnenie spolupredávaných produktov	Matej Červenka
SmartCollect: Vyhodnotenie skrátenia cesty	Martin Šidlo
SmartStore: Dokončenie mobilnej aplikácie	Martin Šidlo

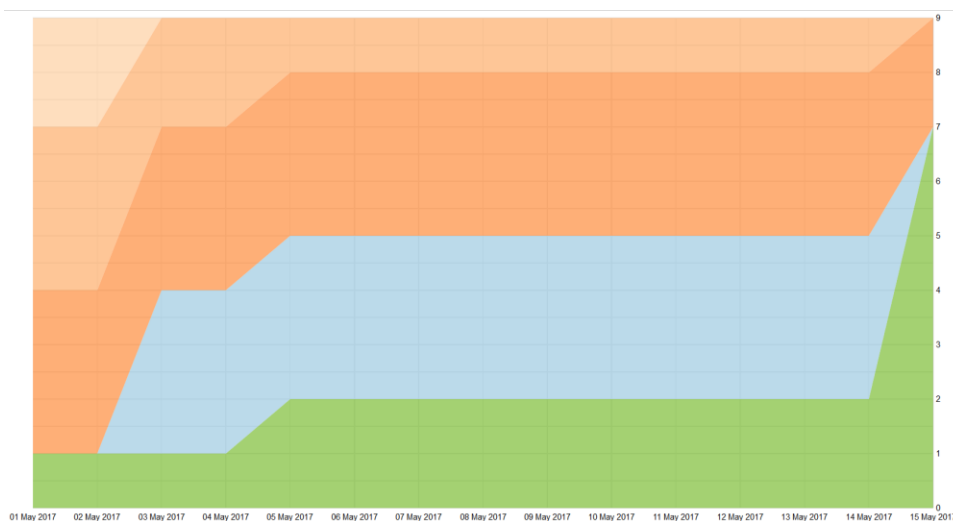
Tabuľka č.15 User stories vytvorené počas plánovania 10. šprintu

4.10.1 Burndown chart



Obrázok č.19 Burndown chart zobrazujúci priebeh 10. šprintu (opis grafu: sekcia 4.1.2)

4.10.2 Cumulative flow



Obrázok č.20 Cumulative flow zobrazujúci 10. šprint (opis grafu: sekcia 4.1.3)

4.10.3 Zhodnotenie šprintu

V poslednom, záverečnom šprinte sme dokončili takmer všetky rozpracované problémy, doladenie chýba len vizualizácii optimálnej pozície produktu.

5 Používané metodiky

Pri písaní metodík sme prihliadali na potreby projektu a činnosti spojené s kolaboratívnym spôsobom vývoja. Všetky metodiky dopodrobna popisujeme v *Prílohe B*.

5.1 Písanie kódu v Ruby on Rails

Metodika sa zaoberá písaním zdrojového kódu vo zvolenej technológii Rails v jazyku Ruby. Metodika rieši rozloženie zdrojového kódu, syntax a jednoduché pravidlá, ktoré je dôležité dodržiavať, aby bol zdrojový kód čitateľný pre každého člena tímu.

5.2 Overovanie kódu

Metodika sa zaoberá tým, ako sa kód overuje z pohľadu vývojára a recenzenta kvality kódu. Kontrola kvality kódu pozostáva z inicializácie, samotnej kontroly a ukončenia overovania kódu.

5.3 Verziovanie projektu SmartStore

Pri projektoch väčšieho rozsahu je zvykom na zdrojovom kóde pracovať paralelne a zmeny postupne zapracovávať do produkčnej aplikácie. V projekte je aplikovaný verziovací systém GIT. Na spravovanie verzií a organizovanie zdrojového kódu sa aplikujú pravidlá práve tejto metodiky, ktoré úzko súvisia aj s overovaním kódu, ktoré je inicializované vybranými mechanizmami verziovania.

5.4 Evidencia úloh

Pri aplikácii agilného vývoja v praxi je potrebné mať ujasnené, akým spôsobom sa úlohy plánujú a manažujú. Metodika evidencie úloh sa podrobne zaoberá správou úloh a softvérových nástrojov, ktoré používame v tíme.

5.5 Testovanie

Metodika sa zaoberá technológiami použitými pri automatizácii testovania aplikácie a základmi práce s týmto testovacím rámcom.

5.6 Písanie skriptov dátovej analýzy v R

Pri písaní skriptov na dátovú analýzu a na celkovú prácu s predikčným modulom bola spísaná metodika pre písanie skriptov v technológii R. Podobne ako pri písaní zdrojového kódu v Ruby on Rails je potrebné dodržiavať doménovo špecifické pravidlá a obmedzenia.

5.7 Dokumentovanie

Pri písaní dokumentácie k riadeniu a k inžinierskemu dielu sa riadime metodikou dokumentovania, v ktorej popisujeme, ako správne formátovať a vypracovať dokumenty tak, aby nebolo nutné ich ďalšie upravovanie. V metodike tiež popisujeme používané procesy na vypracovávanie jednotlivých typov dokumentov a ich uchovávanie.

6 Globálna retrospektíva ZS

Počas zimného semestra sme dosiaľ postupovali podľa vytýčených cieľov, ktoré sme popísali v kapitole 2 projektovej dokumentácie k inžinierskemu dielu. Sústredili sme sa na problém optimalizácie skladu pomocou predpovedania predajov na základe dát z uplynulého obdobia. V posledných dvoch šprintoch sme však riešili aj požiadavku na optimalizáciu trasy pracovníka v priestoroch skladu.

Počas prvých troch šprintov sa nám podarilo vytvoriť funkčný prototyp webovej aplikácie, ktorý je možné prezentovať pracovníkom skladu, aby sme získali spätnú väzbu od skutočných používateľov a tým budeme môcť určiť ďalšie smerovanie a vývoj tejto aplikácie. Napriek počiatočnému čiastočnému neúspechu v prvom šprinte, kedy sme nestihli dokončiť všetky *user stories* sa nám podarilo tento trend v ďalších dvoch šprintoch napraviť.

Po neúspechu sme identifikovali problémy, ktoré čiastočne pramenili z nedostatočnej komunikácie a zlého odhadu času a úsilia, ktoré je na prácu potrebné, a už v nasledujúcom druhom šprinte sa nám podarilo nedokončené úlohy dodať spolu s ďalšími, ktoré boli naplánované. Vzhľadom k tomu, že sa nám podarilo dostatočne rozpracovať prototyp predikcie objednania tovaru, už v strede tretieho šprintu nás vlastník produktu podrobnejšie oboznámil s ďalším cieľom, ktorým je optimalizácia pohybu pracovníka v priestoroch skladu.

Po analýze, bližšej špecifikácii problému s vyhľadáním optimálnej trasy a zadefinovaní štruktúry skladu našim produktovým vlastníkom sme v šprinte 4 a 5 riešili aj problém optimalizácie trasy – tzv. *SmartCollect*.

Ak to globálne zhrnieme, chceme pokračovať v nastolenom trende dodávania naplánovaných úloh včas na konci každého šprintu, pričom sa chceme zlepšiť v pokrytí implementovanej funkcionality testami. Tiež by sme chceli pracovať na vyváženosti rozdelenia úloh medzi jednotlivých členov tímu tak, aby sme minimalizovali prípady, kedy musia niektorí členovia dokončovať alebo opravovať prácu niekoho iného. Veríme, že pravidelné tímové stretnutia mimo školy môžu zlepšiť súdržnosť tímu a posilnia tímového ducha.

7 Globálna retrospektíva LS

Plánovanie práce na letný semester už bolo viac ovplyvnené aj záujmami tímu a ciele boli stanovené diskusiou s produktovým vlastníkom. Počas letného semestra na nám podarilo všetky stanovené ciele splniť a teda značne sme vylepšili produkt *SmartCollect*, ktorý už dokáže stanoviť optimálne miesto pre každý produkt, ktorý je práve na sklade.

Počas prvých šprintov sme pokračovali v práci, ktorou sme končili v zimnom semestri a diskusiou sme postupne získavali predstavu o ďalších požiadavkách zákazníka na modul na optimalizáciu rozloženia skladu. Pri vývoji tohto modulu sa nám práve agilný spôsob vývoja osvedčil, keďže podmienky a požiadavky boli pravidelne upravované a do celkovej podoby bolo nutné urobiť niekoľko iterácií návrhu aj implementácie tohto modulu. Pri práci na projekte už členovia tímu lepšie poznali silné a slabé stránky ostatných, a preto vznikalo menej nervozity z dodania požadovanej funkcionality načas.

V práci sme pokračovali počas celého semestra, kedy sme aj vyhodnotili, aký prínos má naše riešenie. Vypracovanie a vyhodnotenie nášho projektu sme v záverečných týždňoch semestra odprezentovali na študentskej vedeckej konferencii IIT.src, kde sa nám podarilo postúpiť do semifinále v súťaži TPCup.

V poslednom týždni semestra sme sa tímovom stretnutí zhodnotili prácu počas celého akademického roka a vypočuli sme si hodnotenie produktového vlastníka, ktorý vyhodnotil našu prácu ako veľmi dobrú a presahujúcu jeho očakávania. Veríme, že náš produkt, alebo jeho časť, bude aj po skončení predmetu Tímový projekt využívaný v praxi a prinesie zákazníkovi zlepšenie v dodávaní produktov v čo najkratšom čase.

8 Prílohy

- Príloha A: Preberacie protokoly
- Príloha B: Metodiky
- Príloha C: Export evidencie úloh

Príloha A Preberacie protokoly

V prílohe sa nachádzajú nasledujúce preberacie protokoly:

- Preberací protokol dokumentácie odovzdávanej na konci ZS 2016/2017
- Preberací protokol dokumentácie odovzdávanej na konci LS 2016/2017

PREBERACÍ PROTOKOL

Odovzdávajúci subjekt: Tím 3, Tímový projekt 2016/2017
v z. Bc. Mária Dragúňová

Preberajúci subjekt : Ing. Peter Krátky,
Vedúci Tímu 3, Tímový projekt 2016/2017

Predmety prebratia (zhotovené dielo):

Dokumentácia odovzdávaná na konci ZS predmetu Tímový projekt 2016/2017
zahŕňajúc dokumenty:

Projektová dokumentácia k inžinierskemu dielu
Projektová dokumentácia k riadeniu projektu

Preberajúci subjekt prijíma predmety prebratia s nasledovným vyjadrením:
So zhotovením diela je preberajúci subjekt spokojný, nie je si vedomý žiadnych
námietok proti zhotovenému dielu a preto s odovzdaním súhlasí a toto dielo
preberá.

V , dňa

.....
podpis odovzdávajúceho subjektu

.....
podpis preberajúceho subjektu

PREBERACÍ PROTOKOL

Odvzdávajúci subjekt: Tím 3, Tímový projekt 2016/2017
v z. Bc. Mária Dragúňová

Preberajúci subjekt : Ing. Peter Krátky,
Vedúci Tímu 3, Tímový projekt 2016/2017

Predmety prebratia (zhotovené dielo):

Dokumentácia odovzdávaná na konci LS predmetu Tímový projekt 2016/2017 zahŕňajúce dokumenty:

Projektová dokumentácia k inžinierskemu dielu a projektová dokumentácia k riadeniu projektu zviazané ako jeden dokument

Preberajúci subjekt prijíma predmety prebratia s nasledovným vyjadrením: So zhotovením diela je preberajúci subjekt spokojný, nie je si vedomý žiadnych námietok proti zhotovenému dielu a preto s odovzdaním súhlasí a toto dielo preberá.

V , dňa

.....
podpis odovzdávajúceho subjektu

.....
podpis preberajúceho subjektu

Príloha B Metodiky

V tomto dokumente popisujeme používané metodiky:

1. Písanie kódu v Ruby on Rails
2. Overovanie kódu
3. Verziovanie projektu
4. Evidencia úloh
5. Testovanie
6. Písanie skriptov dátovej analýzy v R
7. Dokumentovanie

1 Písanie kódu v Ruby on Rails

Vo všeobecnosti sa riadime pokynmi tejto metodiky v kombinácii s radami vývojového prostredia, v ktorom pracujeme.

1.1 Štandardné pravidlá

1. Všetko sa píše po anglicky, vrátane komentárov.
2. Z názvu premennej, triedy, súboru, atď. musí byť jasné o čo ide bez študovania funkcionality alebo významu.
3. Na voľný kód sa používa notácia pomocou ‘_’ (pre_men_na). Vo frameworku Rails platí pravidlo „*Convention over Configuration*”, ktoré sa pri pomenovaniach snažíme dodržiavať.
4. Používa sa štandardnú architektúra frameworku Rails. V prípade potreby rozšírenia je dôležité odkonzultovať umiestnenie adresárov.
5. V prípade potreby rozšírenia aplikácie o novú knižnicu je dôležité odkonzultovať rôzne alternatívy a vybrať najvhodnejšiu.
6. Každá metóda musí byť opísaná podľa pravidiel dokumentovacieho komponentu RDoc.
7. Pred komitovaním zmien je potrebné svoj kód okontrolovať na zbytočné alebo neefektívne riadky kódu. Riadime sa pravidlom: „Ak je moje riešenie zreteľne zložité alebo príliš dlhé, spravil som to zle.”
8. Pri vyhľadávaní metód sa hľadajú tie, ktoré robia to, čo je potrebné bez vedľajších alebo nepoznaných efektov.
9. Kód je potrebné aspoň raz spustiť a dôkladne overiť jeho funkčnosť.

1.2 Nastavenia formátu vývojového prostredia

- Ruby - tab size 2, indent 2, continuation indent 2
- CSS - tab size 4, indent - 4, continuation indent - 4
- ERB - tab size 2, indent 2, continuation indent - 4
- HTML - tab size 2, indent, 2, continuation indent - 8
- JavaScript - tab size 4, indent 4, continuation indent - 4
- JSON - tab size 4, indent 2, continuation indent - 8
- SASS - tab size 4, indent 2, continuation indent - 8
- YAML - indent 2

1.3 Rozloženie zdrojového kódu

- Vnorený kód oddeľujeme dvomi medzerami.

```
def index
  do_something
```

end

- Príkazy neoddeľujeme ‘;’, ale novým riadkom.

```
puts ‘kebab’  
puts ‘pizza’
```

- Používame medzery okolo operátorov (s výnimkou **), za čiarkami a na miestach, kde si to čitateľnosť kódu vyžaduje.

```
a = a + b  
[a, b]  
8**2  
metoda(kebab = true, pizza = false)
```

- Zarovnanie pri použití ‘case’:
 - štandardný prípad použitia

```
case var  
when ‘pizza’  
  true  
else  
  false
```

- priradovanie

```
result = case var  
  when ‘pizza’  
    true  
  else  
    false
```

- Zarovnanie pri použití ‘if’:
 - štandardný prípad použitia

```
if pizza  
  true  
else  
  false
```

- priradovanie

```
result = if pizza  
  true  
else  
  false
```

- Zlom riadkov pri dlhom chaine metód.

```
metoda1.metoda2
    .metoda3
```

- Metódy s neprehľadným množstvom parametrov v jednom riadku.

```
Model.new(param: kebab,
           param2: pizza,
           ..)
```

- Písanie komentárov.

```
# kebab
# pizza
```

1.4 Syntax

- Definícia metód.

```
def metoda_bez_parametrov
  do_something
end
def metod_s_parametrami(parameter)
  do_something
end
def self.staticcka_metoda
  do_something
end
```

- Volanie metód.

```
volanie.bez.parametru
volanie.s.parametrami.prehľadne parameter
volanie.s.parametrami(kebab).neprehľadne(pizza)
Staticcka.metoda
```

- Nepovinné argumenty metódy.

```
def metoda(davaj, vzdy, na = true, koniec = 5)
  do_something
end
```

- Priradovanie do premenných.

```
davame = 1
```

```
pod_seba = 2
```

- For cyklus.

```
používaj .each ☺
```

- If podmienka.

```
if true
  puts 'true'
end
if true
  puts 'true'
else
  puts 'false'
end
if true
  puts 'true'
elsif false
  puts 'false'
else
  puts '9'
end
```

```
value = 'kebab' if kebab #nepouzivame else v 1-riadkovom
zapise
```

```
if value1 && value2 || value3
  puts 'true'
end
```

```
unless false #pouzivame iba v pripade nezloženej podmienky a
bez else a namiesto !
```

```
  puts 'true'
end
```

- While cyklus.

```
while kebab > 5
  do_something
end
until kebab > 5
  do_something
```

```
end # podobne ako pri unless, until sa pouziva namiesto ! a
pri nezlozitych podmienkach
```

```
one_line_do_something while condition
loop { do_something } #nekonecny loop
```

- Ternárny operátor
var = value.kebab? 'kebab' : 'pizza'
- Bloky.
viacriadkovy do
do_something
end
jednoriadkovy { do_something }
- Operátory.
a += 5
b ||= object
a is_a?/include?/eql? B #pouzivame namiesto == vsade, kde sa
da
- Vyhybanie sa „Undefined method for nil class“.
try - <http://apidock.com/rails/v3.2.1/Object/try>
snažíme sa rozmýšľať nad kódom a v prípade podozrivého kódu
vymyslieť individuálne riešenie
- Chyby (vytvorenie, handlovanie).
raise Exception, additional_params
begin
do_something
rescue Exception
do_something
ensure
finally_do_something
end
- Súbory.
File.open('w', path) do
end # pouziva automaticky metodu close
- Reťazce.
'jednoduchy retazes'


```
html-atribut="hodnota"  
"interpolacia #{do_somethhing}"  
"vnorene 'uvodzovky'"
```

- Polia.

```
pole = ['a', 3, objekt]
```

- Práca s polom/kolekciou.

- Iterácia.

```
pole.each { |el| puts el }
```

- Prirad'ovanie.

```
pole << novy_prvok
```

- Vytváranie nového pola.

```
pole.map { |el| el.co_chcem_v_novom_poli }
```

- Filter.

```
pole.select { |el| el.podmienka_pre_vyber }
```

- Redukcia.

```
pole.inject(val) { |el, val| el + val }
```

- Hashe.

```
hash = {a: 1, b: 2}
```

```
hash.keys == [:a, :b]
```

```
hash.values == [1, 2]
```

- Práca s hashom (podobne ako pri poli).

- Iterácia

```
hash.each_key { |k| puts k }
```

```
hash.each_value { |v| puts v }
```

```
hash.each { |k, v| puts k, v }
```

- Štruktúra modelov.

```
class Person
  # extend a include najskôr
  extend Module1
  include Module2
  # verejné konštanty
  CONSTANT = :kebab
  # atribúty, vzťahy (has_many, ...), scopes
  attr_reader :kebab
  # validácie
  validates :name
  # callbacky
  after_commit :metoda
  # verejné metódy
  def metoda
  end
  # konštruktor
  def initialize
  end
  # chránené metódy a konštanty
  protected

  def protected_metoda
  end

  # privátne metódy a konštanty
  private

  def private_metoda
  end
end
```

2 Overovanie kódu

2.1 Inicializácia overovania

Recenzent kvality kódu:

Kontrola kódu začína priradením recenzenta ku konkrétnemu *pull requestu* (*Verziovanie projektu Smart Store*) vývojára. Recenzent je povinný v dostatočnom čase upozorniť vývojára na chyby v kóde prostredníctvom komentárov.

Vývojár:

Kontrola kódu prebieha po vytvorení *pull requestu* (*Verziovanie projektu Smart Store*) a priradení jedného z recenzentov kvality kódu na daný *pull request*.

2.2 Overovanie kódu

Recenzent kvality kódu:

Recenzent overuje kód na základe metodiky *Písanie kódu v Ruby on Rails* a svojho vlastného svedomia. Recenzent musí okrem plnenia pravidiel písania kódu brať do úvahy taktiež efektívnosť kódu, ako aj správne využitie technológií.

Vývojár:

V prípade vyžiadania zmien formou komentárov je potrebné kód upraviť. V prípade nehody je nutné počúvať rady recenzenta aj keď sa nezhodujú s pravidlami písania kódu v metodike *Písanie kódu v Ruby on Rails*. Recenzent má vždy pravdu.

2.3 Ukončenie overovania

Recenzent kvality kódu:

Recenzent musí po pozitívnom ukončení revízie kód potvrdiť a vykonané zmeny spojiť so zvyšným zdrojovým kódom na základe metodiky *Verziovanie projektu Smart Store*.

Vývojár:

Po odsúhlasení zmien recenzentom a uzatvorením *pull requestu* sa kontrola považuje za ukončenú. Vykonané zmeny podliehajú potom pravidlám testovania a verziovania, ktoré už s vývojárom nesúvisia.

3 Verziovanie projektu SmartStore

Pre tento projekt sa používa verziovací systém GIT. Ako podporný mechanizmus pre jeho distribúciu je používaný webový nástroj *Github*, ktorý poskytuje aj grafické rozhranie pre efektívnu prácu a ochranu jednotlivých verzií. Ďalej poskytuje rôzne prehľady, grafy aj možnosť vytváranie takzvaných “*issues*”, respektíve sledovanie a oznamovanie problémov a ich jednotlivé priradovanie členom tímu.

3.1 Vytvorenie a správa repozitáru

Vytvorenie a správu repozitáru zabezpečí manažér pre verziovanie. Po vytvorení pridá členom tímu práva, aby mohli pristupovať ku všetkým zložkám a súborom aplikácie. Následne vytvorí 2 základné vetvy pre aplikáciu a nastaví práva pre ochranu pridávania nových častí kódu. Manažér verziovania preberá zodpovednosť za repozitár čo zahŕňa dohľad nad dodržiavaním metodiky. V prípade chybnjej manipulácie s repozitárom, je povinný upozorniť člena tímu a dohliadnuť na jej odstránení, poprípade asistovať na jej odstránení.

3.2 Vetvy aplikácie

Vetvenie aplikácie pozostáva zo základnej štruktúry 2 vetvy. Vetva s názvom *master* je produkčná verzia programu, ktorá je identická s verziou kódu na produkcii. Ďalej vetva *staging* obsahuje časti kódu, ktoré sú otestované a čakajú na aktualizáciu na produkčnú verziu. Ostatné vetvy sa vytvárajú podľa funkcionality, ktorá sa v nej vytvára. Každý si takúto vetvu môže vytvoriť u seba lokálne a následne ju publikovať do *Github*-u, aby bola verejná pre všetkých členov tímu. Tieto vetvy nesú meno s malým začiatočným písmenom, oddelené podčiarkovníkom s názvom funkcionality v angličtine. Celý projekt je programovaný v angličtine, čiže aj verziovanie bude zachovávať konzistenciu s kódom. Príklad názvu vetvy *prediction_of_sales*.

3.2.1 Vetva master

Ide o hlavnú vetvu aplikácie a ako bolo spomenuté vyššie, obsahuje produkčnú verziu kódu. Táto vetva sa zabezpečí proti pridávaniu kódu bez toho, aby bola nová časť preverená členmi tímu zodpovednými za zrecenzovanie, respektíve kvalitu kódu. Pridávanie novej funkcionality sa vykonáva pomocou takzvaných *pull request*, ktoré je podrobne popísané nižšie pri pridávaní novej funkcionality.

3.3 Pridávanie novej funkcionality

Pri práci na novej funkcionalite sa postupuje nasledovne. Ako prvé je potrebné aby si vývojár aktualizoval verziu kódu vo vetve *staging*, ktorú má u seba v počítači a ktorá je na *Github*-e. Následne si z tejto vetvy vytvorí svoju novú lokálnu vetvu pre novú

funkcionalitu, na ktorej bude pracovať. Pomenuje ju ako bolo popísané vo vetvení aplikácie a následne ju nahrá aj na *Github*.

3.3.1 Nové časti kódu

Pridávanie nových častí kódu sa realizuje pomocou tzv. *commit*. Obsah jednotlivých *commit*-ov by nemal byť príliš veľký, ale zároveň ani malý. Granularita by mala byť zvolená podľa logických celkov v kóde, ktoré boli vytvorené. Napríklad po naprogramovaní generovania a zobrazovania elementu na stránke, sa vytvorí *commit*. Jednotlivé *commit*-y a ich názvy sa taktiež vytvárajú v Anglickom jazyku. Tento text pozostáva z nadpisu, ktorý by mal byť výstižný a dostatočne stručný. Nadpis je nasledovaný ďalším textom s podrobnejším popisom v prípade, že si to charakter zmien vyžaduje, respektíve nadpis nie je dostačujúci.

3.3.2 Pull request

Pred pridaním novej funkcionality do vetvy *staging*, je potrebné aby prebehla recenzia kvality kódu. Na toto sa využívajú tzv. *pull request*. Programátor nahrá všetky svoje zmeny do *Github*-u a následne vytvorí *pull request* z jeho vetvy do vetvy *staging*. Názov je opäť v angličtine, avšak popis už môže byť aj v slovenčine. Ďalšia povinnosť je priradenie členov tímu zodpovedných za kontrolu kódu k tomuto *pull requestu*. Iných členov je zakázané pridávať. Výnimku tvoria členovia zodpovední za kvalitu, ktorí môžu na základe svojho uváženia pridať aj iného člena tímu na vykonanie a schválenie recenzie.

V prípade, že je kód overený a schválený, vetva je spojená so *staging*-om a vetva, v ktorej bola nová funkcionality vytvorená sa vymaže. Pôvodný *pull request* je označený ako *closed*, čiže zatvorený a je možné dať ďalší *pull request* na vetvu *master* z vetvy *staging*. Z iných vetvy ako *staging* je zakázané pridávať *pull request* na *master*. Takýto *pull request* na *master* sa po schválení ihneď dáva na server, za čo je zodpovedný člen tímu pre správu servera.

3.3.3 Nahlasovanie a oprava chýb

V prípade, že je objavená chyba v aplikácii, hociktorý programátor môže vytvoriť na *Github*-e tzv. “*issue*”, ktorej dá názov v angličtine, no popis je možné nechať v slovenčine. Nadpis by mal popisovať chybu, ktorá nastala. Napríklad “*Wrong ordering of books*”. Následne je potrebné priradiť člena tímu, ktorý bude opravovať túto chybu. Oprava sa vykonáva ako v prípade vytvárania novej časti kódu. Uzatvorenie tohoto problému sa vykoná až po jeho opravení a nasadení na produkčný server. Zodpovednosť za toto označenie preberá osoba, ktorá bola priradená na riešenie tohoto problému.

3.4 Označovanie verzii

Označovanie verzii programu má na starosti manažér verziovania. Po každej novej verzii, respektíve šprinte, po ktorom sa pridal nejaký nový balík funkcionalít, sa vytvorí tzv. “*tag*” na *Github*-e, ktorý označuje tieto zmeny.

4 Evidencia úloh

V projekte využívame nástroj na evidovanie úloh *YouTrack*, ktorý má podporu pre agilnú metodiku vývoja SCRUM, ktorou sa v rámci tímového projektu riadime. Okrem prehľadu stavu úloh jednotlivých členov tímu nám tento nástroj umožňuje aj ich export ako aj export najrôznejších štatistík napríklad *Burndown grafu*.

4.1 Úlohy

Úlohy sú vytvárané a pridelené jednotlivým členom tímu v rámci plánovania nového šprintu, každé dva týždne. V každej vytvorenej úlohe musíme pri jej vytvorení definovať názov spolu s opisom, typ úlohy, jej stav, zodpovedného človeka, prioritu a prepojenie s inou úlohou ak nejaké existuje.

4.1.1 Typy úloh

V tomto projekte rozoznávame nasledovné typy úloh.

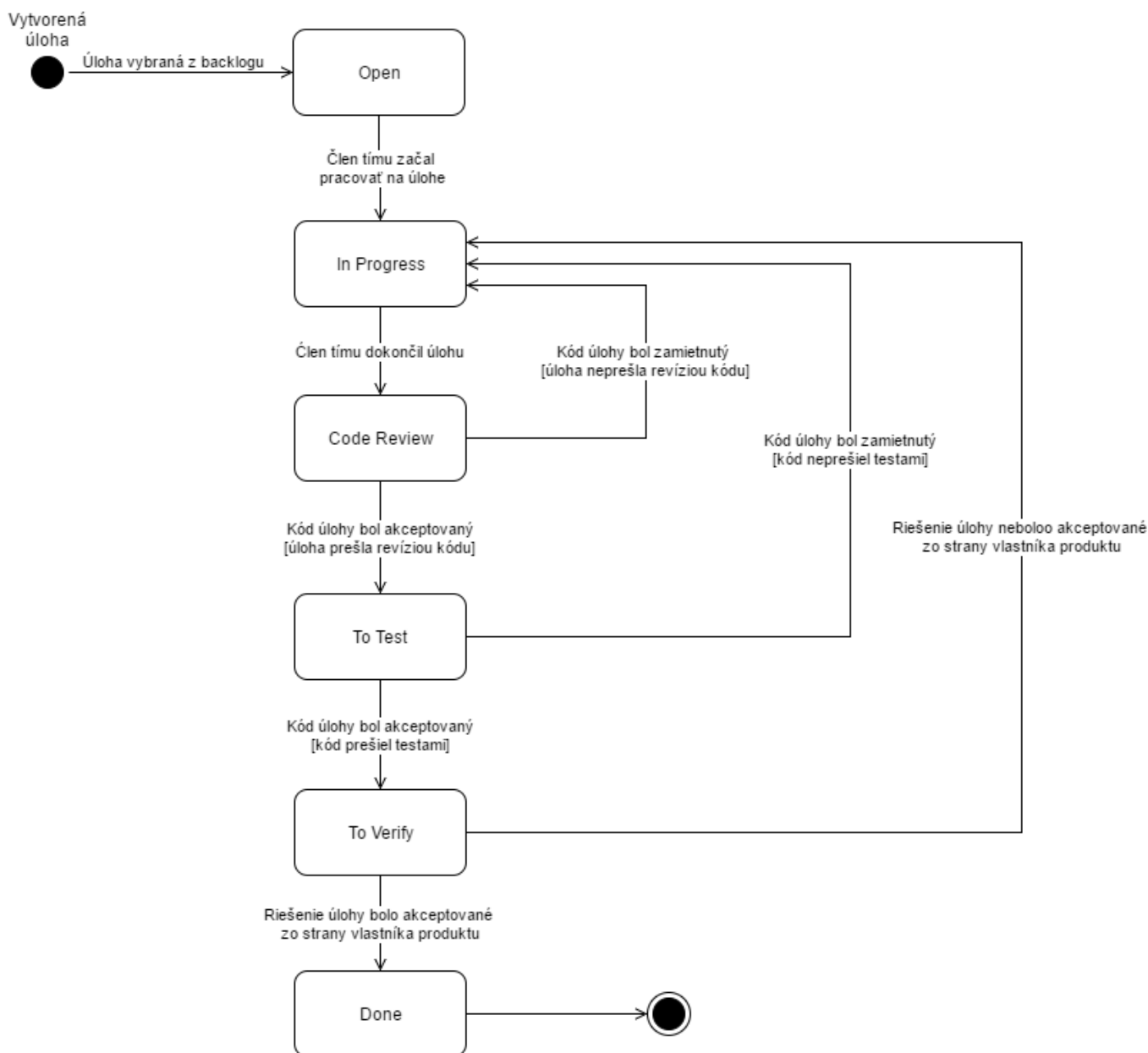
- **User Story**
 - bežná úloha, ktorá má ako celok zmysel pre vlastníka produktu
 - má vždy prideleného človeka, ktorý je za jej organizáciu a dokončenie zodpovedný
 - vytvára úlohy typu *task* (ďalej iba „podúlohy“) patriace a vyplývajúce zo *story* a pridružuje ich buď sebe alebo iným členom tímu po vzájomnej komunikácii
 - vždy je ohodnotená *story pointami*, ktoré členovia tímu odhlasovali na plánovaní
- **Management Story**
 - špecifický typ úlohy vyhradený pre úlohy manažérskeho typu pre vylepšenie riadenia v rámci tímu
 - má vždy prideleného človeka, ktorý je za jej organizáciu a dokončenie zodpovedný
 - vytvára úlohy typu *task* (ďalej iba „podúlohy“) patriace a vyplývajúce zo *story* a pridružuje ich buď sebe alebo iným členom tímu po vzájomnej komunikácii
 - vždy je ohodnotená *story pointami*, ktoré členovia tímu odhlasovali na plánovaní
- **Task**
 - predstavuje podúlohu, ktorá vždy patrí ku konkrétnej *story*
 - má vždy prideleného človeka, ktorý je za jeho organizáciu a dokončenie zodpovedný
 - môže byť vytvorená v priebehu šprintu

- **Bug**
 - oznámenie o chybe v programe
 - po vytvorení nemá prideleného človeka
 - člen tímu, ktorý vie problém vyriešiť alebo ktorý riešil časť programu v ktorom sa chyba vyskytla si ju sám priradí
 - môže byť vytvorená v priebehu šprintu

4.1.2 Stavy úloh

Úlohy prechádzajú počas svojho životného cyklu nasledujúcimi stavmi.

- *Open* - riadne vytvorená úloha, na ktorej sa ešte nezačalo pracovať
- *In Progress* - úloha je rozpracovaná, členom/mi tímu
- *Code Review* - úloha je dokončená a čaká na code review
- *To Test* - úloha čaká na otestovanie funkcionality
- *To Verify* - úloha čaká na schválenie vlastníkom produktu
- *Done* - dokončená a schválená úloha



4.2 Práca s YouTrack-om

4.2.1 Vytvorenie šprintu

V našom projekte okrem čísla šprintu určujeme aj ďalší identifikátor, ktorým je jeho meno. Podmienkou je aby meno bolo zvolené podľa známeho mimozemšťana. Názvy šprintov musia byť v tvare:

- (číslo šprintu).šprint (meno šprintu)

Okrem mena šprintu je nutné zadať jeho trvanie a stručne uviesť cieľ/e, ktoré chceme splniť.

4.2.2 Vytvorenie úlohy

Práva na vytvorenie, upravovanie a mazanie úloh má každý používateľ systému odsúhlasený administrátorom. Úlohy typu *user story* spravidla vytvára *Scrum Master* na plánovaní šprintu, avšak vytvorenie podúloh už rieši každý člen tímu pre jeho pridelenú *user story* sám. Všetky vytvorené úlohy musia slniť vopred dohodnuté pravidlá. Musia mať:

- ponechaný preddefinovaný projekt - SmartStore
- stručný a výstižný názov
- opis úlohy v bodoch pre lepšie priblíženie z čoho úloha pozostáva (platí iba pre úlohy typu *story*)
- určenie typu úlohy
 - pre podúlohy je nutné priradiť rodiča tejto úlohy
- určenie priority úlohy
- stav úlohy ponechať preddefinovaný
- prideliť zodpovedného za úlohu
- zaradiť úlohu do aktuálneho šprintu
- odhadnúť spoločne s tímom množstvo *story pointov* úlohy (výnimku tvoria úlohy podúlohy pre ktoré toto pole ostáva nezadané)
- odhadnúť množstvo hodín pracovania na úlohe
 - platí len pre úlohy typu *Task a Bug*

Okrem vyššie spomenutého je možné k úlohe pridať aj obrázok či súbor, ktorý môže tiež slúžiť ako rozsiahlejší popis úlohy

4.2.3 Organizovanie úloh

O organizáciu *user story* sa stará vždy len k nej pridelený člen tímu. To znamená, že jeho povinnosťami je

- vytvoriť podúlohy, ktoré vychádzajú z tejto *user story*,
- prideliť tieto vytvorené úlohy buď sebe alebo ostatným členom tímu,
- udržiavať aktuálny stav všetkých pridelených úloh a
- evidovať si počet odpracovaných hodín na úlohe.

Úlohy typu *story* menia svoj stav v rámci šprintu menej často než jej podúlohy. Členovia tímu teda presúvajú stav im pridelených *story* iba v dvoch prípadoch a to:

1. Na ľubovoľnej podúlohe určitej *story* sa začalo pracovať -> *story* presúvame do stavu *In Progress*
2. S výnimkou prvého prípadu, je stav *story* vždy totožný so stavom jej pridelených podúloh, ktorá sa nachádza v hierarchii stavov na najnižších pozíciách.

4.2.4 Export úloh

V pravidelných týždňových intervaloch vždy po tímovom stretnutí exportujeme aktuálny stav úloh na *YouTrack*-u. Na exportovanie musíme v nástroji najskôr úlohy vyfiltrovať podľa aktuálneho šprintu a následne ich zoradiť do želaného poradia. O toto sa stará špeciálny príkaz zapísaný do vyhľadávača v *YouTracku*:

```
#{NÁZOV ŠPRINTU} order by: Type,State,Assignee
```

Po vyhľadaní želaných úloh ich exportujeme do CSV súboru.

5 Testovanie

5.1 Nástroje na testovanie

5.1.1 RSpec

Ruby on Rails má v sebe integrovaný testovací framework *RSpec* s integrovanou podporou pre tvorbu *mock* a *stub* objektov pre inteligentné testovanie. Všetky testy a konfiguračné súbory frameworku sú uložené v podpriechínku *./spec*.

Dokumentácia k *RSpec*: <http://rspec.info/documentation/>

5.1.2 Gem FactoryGirl

Gem *FactoryGirl* slúži pre vytvorenie objektu alebo kolekcie objektov s predom preddefinovanými hodnotami.

Dokumentácia ku gemu *FactoryGirl*: http://www.rubydoc.info/gems/factory_girl/4.7.0

5.1.3 Gem Faker

Gem *Faker* slúži pre generovanie náhodných dát predom určeného typu a rozsahu. Výborná pomôcka najmä pri potrebe veľkého množstva rozmanitých testovacích dát.

Dokumentácia ku gemu *Faker*: <http://www.rubydoc.info/gems/faker/1.6.6>

5.2 Rozmiestnenie súborov a názvoslovie

5.2.1 Súborový strom

```
Smart-store-|
              |-spec-|-
                |   |controllers|
                |   |factories  |
                |   |helpers   |
                |   |models    |
                |   |support   |
                |   |jobs      |
                |   |rails_helper.rb
                |   |spec_helper.rb
```

Všetky testy sa nachádzajú v *./spec*. V zložke *factories* sa nachádzajú pripravené *factories*. Všetky modifikácie a pridávanie nových *factories* patria sem.

V zložke *controllers*, *helpers*, *models*, *jobs* sa nachádzajú jednotlivé testy k týmto vrstvám aplikácie.

V zložke *support* je konfiguračný súbor k *database_cleaner*. Momentálne je nastavený tak, aby po každom spustení testu po sebe všetky zmeny udiané počas testu vrátil späť.

Súbory *rails_helper* a *spec_helper* sú konfiguračné súbory k *rspec*.

5.2.2 Názvoslovie

Názov testovacieho súboru odvodíme od toho, na čo v ňom sú písané testy.

Formát je :

- Factories : *nazovfactory.rb*
- Controller: *xyz_controller_spec.rb*, kde *xyz* je názov testovaného controllera
- Job: *xyz_job_spec.rb*, kde *xyz* je názov jobu
- Model : *xyz_spec.rb*, kde *xyz* je názov modelu, ktorý testujeme
- Helper: *xyz_helper_spec.rb*, kde *xyz* je názov hellperu

Všetky testy okrem factories musia končiť *_spec.rb*.

5.3 Generovanie testov, spúšťanie testov a vyhodnocovanie

5.3.1 Generovanie testov

V Railsoch sa testy pre jednotlivé súčasti generujú zadaním do konzoly

```
rails generate rspec:cast nazov
```

Kde časť sa nahradí tým čo chceme generovať (napr. model) a názov súčasti.

```
rails generate rspec:model widget
```

Tento príklad vygeneruje test modelu do *./spec/model/widget_spec.rb*

5.3.2 Spustenie testov

Spustenie celej testovacej sady naraz je jednoduché a vykoná sa zadaním v konzole

```
rspec
```

alebo

```
bundle exec rspec
```

Pred spustením testov je potrebné mať vytvorenú a pripravenú testovaciu databázu pomocou

```
rails db:create
```

```
rails db:migrate
```

V prípade, že chceme pustiť len jeden test použijeme

```
rspec ./cesta_k_testu
```

5.3.3 Vyhodnotenie testov

Po spustení testov sa po ich zbehnutí na konci v konzole zobrazí štatistika s testami, ktoré zbehli korektne, s chybou alebo neboli implementované.

Pri testoch, ktoré zbehli chybou je v konzole napísané o aký test ide a aké dáta do neho prúdili a aké sa očakávali.

6 Písanie skriptov dátovej analýzy v R

Pri písaní skriptov pre dátovú analýzu v jazyku R je potrebné dodržať niekoľko pravidiel, ktoré obsahuje táto metodika

6.1 Umiestnenie skriptov

Dátová analýza:

```
/lib/data_analysis
```

Vyhodnotenie dátovej analýzy:

```
/lib/method_evaluation
```

Grafy:

```
/lib/method_evaluation/evaluation_graphs
```

6.2 Rozdelenie skriptov

Skripty dátovej analýzy primárne rozdeľujeme do dvoch hlavných kategórií. Ide o skripty určené na konkrétnu predikciu a skripty, ktoré slúžia na vyhodnotenie nami navrhnutých metód.

Skripty na predikciu – Ide o skripty, ktorých vstupom sú dáta v podobe CSV súborov. Po kompletnej analýze súboru a použití metód, sú výstupom predikované hodnoty, ktoré sa ukladajú do nového CSV súboru s dátami určenými na uloženie do databázy

Skripty na vyhodnotenie – Ide o skripty, v ktorých sa vytvárajú štatistické tabuľky o úspešnosti predikcie, histogramy a rôzne grafy, pre lepšie znázornenie úspechu, resp. Neúspechu.

6.3 Zdrojový kód

6.3.1 Vkladanie knižníc

Na začiatok zdrojového kóde je potrebné vložiť príkaz

```
library("názov knižnice")
```

6.3.2 Tvorba vlastných funkcií

Vlastné funkcie sú deklarované na začiatku zdrojového kódu s dostatočným okomentovaním hlavnej náplne danej funkcie

```
"názov funckie" <- function() {}
```

6.3.3 Rozloženie zdrojového kódu

- Využívanie medzier okolo operátorov (napríklad: =, +, -, <-)

```
suma <- suma + 1
```

- Využívanie medzier pri použití if, while, for

```
if (true)
```

- Zarovnanie zátvoriek pri použití if, while, for

```
if (true) {  
    suma <- suma + 1  
}
```

- Umiestnenie else, pri použití if

```
If (true) {  
    Suam <- suma + 1  
} else {  
    Suma <- suma + 2  
}
```

- Písanie komentárov

```
# komentár
```

7 Dokumentovanie

7.1 Ukladanie dokumentov v úložiskách

7.1.1 Uloženie projektovej dokumentácie

Na ukladanie projektovej dokumentácie je použitý nástroj *OneDrive*. Na kolaborovanie pri úprave dokumentov je použitý nástroj *Word*. Úložisko na *OneDrive* má nasledujúcu štruktúru, ktorú je potrebné pri pridávaní nových súborov dodržiavať (významné priečinky sú zobrazené **tučným**, dôležité súbory *kurzívou*):

- **Dokumentácia**
 - o **Ostatné** – obsahuje prihlášky, motivačný dokument ai.
 - o **Šablóny** – obsahuje šablóny k jednotlivým dokumentom
 - o *Team03_dokumentacia_produkту_final.docx*
 - o *Team03_dokumentacia_riadenia_final.docx*
- **Pdf** – exporty vo formáte pdf určené na odovzdanie v kontrolných bodoch
- **Obrázky** – Obrázky použité v dokumentácii
- **Vzory** – Vzory dokumentácií k TP z minulého roku
- **Zadanie** – Dokumenty, ktoré definujú štruktúru dokumentov pre TP pre akademický rok 2016/2017.

7.1.2 Uloženie dokumentov pre týždenné reportovanie postupu

Dokumenty pre týždenné reportovanie sú každý týždeň (alebo po skončení šprintu) publikované na stránke tímu. Tieto dokumenty zahŕňajú:

- Zápisnice zo stretnutí tímu
- Exporty z nástroja na manažovanie projektu
- Retrospektívy šprintov

Dokumenty sú uložené v úložisku *Google Drive*. Do úložiska je potrebné dokumenty pridávať s dodržaním podmienok:

- Dokument musí byť vo formáte pdf.
- Zápisnice a exporty musia mať názov v tvare:

 <deň>.<mesiac>.<rok>.pdf
- Dokument je potrebné pridať do správnej zložky úložiska.
- Pre publikovanie dokumentu na stránke tímu treba upovedomiť manažéra dizajnu a webového sídla tímu.

Dokumentové úložisko obsahuje nasledujúce významné priečinky (zobrazené **tučným**):

- **Zápisnice** – zápisnice z týždenných stretnutí

- **Exporty_youtrack** – exporty z nástroja na manažovanie úloh
- **Retrospektívy** – retrospektívy jednotlivých šprintov
- **Stretnutia** – jednoduché zápisky zo stretnutí vo forme poznámok

7.2 Vytvorenie dokumentu

7.2.1 Vytvorenie zápisnice

- *Šablóna: zapisnica_sablona.dotx*
- *Hlavička:* uviesť dátum, čas a prítomné osoby
- *Telo dokumentu:* V tomto dokumente treba popísať priebeh tímového stretnutia a hlavné body diskusie a prípadné navrhnuté riešenia k diskutovaným problémom.

7.2.2 Vytvorenie exportu z nástroja na manažment úloh

- *Šablóna: export_sablona.dotx*
- *Telo dokumentu:* Tabuľka obsahujúca export z nástroja na manažment úloh YouTrack. Vytvorená automaticky.
- *Proces vytvorenia:*
 1. Vytvoriť nový Word dokument.
 2. Nastaviť mód hromadnej korešpondencie na „Adresár“.
 3. Upraviť zoznam príjemcov. Vybrať dokument .csv s exportom z *YouTrack*
 4. Nastaviť aktuálny dátum.
 5. Dokončiť a zlúčiť dokument.

7.2.3 Vytvorenie retrospektívy

- *Šablóna:* -
- *Telo dokumentu:* Popisuje retrospektívu k danému šprintu. Musí obsahovať:
 - *Burndown chart*
 - *Cumulative flow*
 - Slovný popis retrospektívy
 - Tabuľku zobrazujúcu úlohy v šprinte a zodpovedné osoby
 - Tabuľku zobrazujúcu hlavné body, ktoré sme identifikovali ako:
 - *START*
 - *STOP*
 - *CONTINUE*

7.3 Aktualizovanie projektovej dokumentácie

Na aktualizovanie projektovej dokumentácie je potrebné použiť *kolaboratívny prístup*:

1. Zvoliť dokument, ktorý mám záujem upraviť na *OneDrive*:
 - Zvoliť možnosť „*Edit in Word*“. Dbáť na dodržanie definovanej šablóny.
2. Doplniť text do príslušnej časti dokumentu.

3. Aktualizovať obsah dokumentu.
4. Uložiť dokument.

7.4 Základné pravidlá formátovania dokumentov

7.4.1 Pravidlá použitia formátovania písma

- **Kurzíva:**
 - Cudzie názvy, anglické pomenovania.
 - Príklad: *pull request, commit*
 - Názvy nástrojov, technológií
 - Príklad: *Github, Rspec*
 - Na zvýraznenie dôležitých slov a odkazovanie na názvy
 - Príklad: Recenzent overuje kód na základe metodiky *Písanie kódu v Ruby on Rails* a svojho vlastného svedomia.
 - Cesty k dokumentom, url adresy
 - Príklad: *https://www.google.sk/*
- **Tučné:**
 - Podnadpis na nižšej ako tretej úrovni
 - Inak len výnimočne, na vyjadrenie myšlienky autora textu
 - Príklad: ... obsahuje významné priečinky (zobrazené **tučným**)
- **Podčiarknuté:**
 - Silné zdôraznenie, najmä pri normatívom texte
 - Príklad: Súbor musí obsahovať najmenej...
- **Štýl Code:**
 - Ukážky zdrojového kódu alebo ukážky príkazov
 - Príklad: `while, if (TRUE) { }`

7.4.2 Pravidlá formátovania nadpisov kapitol

- **Projektová dokumentácia (Big Picture)**
 - Nadpis 1. úrovne: *Nadpis 1*
 - Nadpis 2. úrovne: *Nadpis 2*
 - Nadpis 3. úrovne: *Nadpis 3*
 - Nadpisy ďalších úrovní: tučným písmom
- **Príloha**
 - Nadpis prílohy: *Priloha*
 - Nadpis 1. úrovne: *Priloha-Nadpis 1*
 - Nadpis 2. úrovne: *Priloha-Nadpis 2*
 - Nadpis 3. úrovne: *Priloha-Nadpis 3*
 - Nadpisy ďalších úrovní: tučným písmom

Príloha C Export evidencie úloh

V tejto prílohe prikladáme export z nástroja na evidenciu úloh z celého akademického roka..

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-3	1.šprint: E.T	Datova analyza	Lukáš Šimek	Done	User Story	Major	8	<ul style="list-style-type: none">- predikovanie datumu, kedy sa minie tovar (metoda frekvencie nakupov)- predikovanie poctu, ktory treba objednat- porovnanie s existujucou metodou- kvantitativne vyhodnotenie nasich metod
SS-2	1.šprint: E.T	Vytvorenie dátového modelu	Mária Dragúňová	Done	User Story	Major	5	<p>Cieľ: Cieľom tejto úlohy je vytvoriť prvý dátový model a do databázy importovať testovacie dáta.</p> <ol style="list-style-type: none">1. Vytvorenie tabuliek podľa existujúceho logického modelu poskytnutého majiteľom produktu. Tabuľky budú vytvorené migráciami.2. Existujúci dataset je potrebné očistiť od identifikovaných chýb. Celkovú veľkosť datasetu zmenšíme.3. Po očistení dát sa vytvoria vstupné csv súbory, z ktorých budú dáta načítané do databázy.

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-7	1.šprint: E.T	Prvotna verzia GUI	Matej Červenka	Done	User Story	Normal	3	- len jednoduchy vypis vysledkov - co bude obsahovat tabulka: <ul style="list-style-type: none">- product_id, pocet ks, datum kedy treba objednat, datum kedy sa zasoba vypreda, aktualny stock_count- zoradit podla datumu, kedy ich treba objednat (staci max tyzden) - zabezpecit menom, heslom (netreba UI, staci vyskakovacim oknom)
SS-5	1.šprint: E.T	Instalacia na server	Matúš Salát	Done	User Story	Critical	5	- konfiguracia firewallu, ssh - instalacia vsetkeho (ruby, rails, dbs, nginx, R (python)...) - https
SS-6	1.šprint: E.T	Nasadenie timovej stranky na server	Martin Šidlo	Done	User Story	Critical	3	- konfiguracia nginx, php - nekopirovat rucne, ale cez git
SS-29	2.šprint Spock	Vyhodnotenie súčasne implementovanej metódy na dátovú analýzu	Lukáš Šimek	Done	User Story	Major	8	- Vyhodnotiť predikciu dátumu, kedy treba objednať a počtu kusov na objednanie. - Vyhodnotenie, koľko "núl" sa nám podarilo odstrániť, porovnávanie počtu, ktorý sme predikovali objednať a ktorý bol reálne objednaný

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-30	2.šprint Spock	Dopárovať názov produktu k jeho ID.	Mária Dragúňová	Done	User Story	Normal	3	(kontrolovať, či sme nepreskladnili sklad). - Treba overiť, či funguje overenie. - Napr: Predikovali sme 15 kusov, ale stačilo reálne 10. - Aby vedel vlastník produktu vyhodnotiť prínos nášho produktu.
SS-28	2.šprint Spock	Dokončenie GUI pre zobrazenie predikcií	Matej Červenka	Done	User Story	Critical	1	- Na základe informácií v XML získame názov produktu - Vytvorenie requestu na zadanú url, spracovanie odpovede (vhodne - SAX, aby sa nezahltla RAM). - Doplnenie informácií do nášho dátového modelu (k produktu) - pribudne autor, názov, atď. - Denný job, ktorý celé toto vykoná. Zmeniť stĺpce v tabuľke, v ktorej budú jednotlivé predikcie. Usporiadať podľa dátumu, kedy treba najbližšie objednať. Môže sa zobrazovať aj 100 riadkov na jednu stranu. Čo zobrazovať: "ID knihy, meno knihy(s odkazom), predikovaný datum kedy

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-8	2.šprint Spock	Import dát z Martinusu	Martin Šidlo	Done	User Story	Normal	5	<p>dojdu zasoby, predikovany datumu kedy objednat, kolko kusov treba objednat, zamestnancov, meno distributora"</p> <p>- Vytvoriť samostatnú routu. - Json súbor (aktuálny sklad, 10-50 MB): údaje budú rozdelené podľa ID po cca 10 000 na json súbory, koľko súborov (requestov) príde, toľko súborov my uložíme (unikátny názov súborov!). Podľa požiadaviek by mali byť vypočítané predikcie ráno dostupné pracovníkom skladu, tj. v noci bude prebiehať výpočet. Pozor na veľkosť súborov! Treba vymyslieť spôsob na na ukladanie do súboru. Musíme odpovedať status 200 OK - funkcia head(200). - Job na import dát do databázy z jsonu (ktorý prišiel). Zapne sa nakonci, keď sa json z requestu uloží do súboru. - Logovať ukladanie do súborov a vykonanie jobu v krokoch.</p>

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-4	2.šprint Spock	Integrácia programu na dátovú analýzu do aplikácie	Ondrej Kudláč	Done	User Story	Critical	8	<ul style="list-style-type: none">- Pripravenie csv s 3 mesiacmi dát v ruby/rails:ch: urobiť select nad databázou, vytvoríme csv (aby toto nebolo nutné riešiť v Rku).- Prevzatie vystupu z algoritmu (v ruby): načítanie "curent_output.csv", po spracovaní sa názov zmení na aktuálny timestamp. Musí sa prečistiť(aktualizovať) celá databáza (na základe dát, ktoré prišli). V databáze nemôžu byť dve rôzne predikcie pre jeden produkt. Implementovať transakčne (viacero insertov vrámci jednej transakcie). Ak sa jeden zápis nepodarí, nech sa rollbackne celá transakcia.- Background processing: Treba spojzdníť na serveri (na produkcii). Jeden job na pustenie Rka. Vyriešiť výnimky v prípade chyby v skripte v Rku. Treba spraviť "cronový".- Doplniť dátový model o "OrderPrediction"

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-51	3.šprint Kal-El	Dostať dáta na produkciu	Mária Dragúňová, Matúš Salát	Done	User Story	Normal	1	<ul style="list-style-type: none"> - Reset DB - Import dát - Spustenie nových jobov - Zmena času vykonania jobu pre import produktov
SS-50	3.šprint Kal-El	Filtrovanie	Matej Červenka	Done	User Story	Normal	2	<ul style="list-style-type: none"> - Ako zamestnanec chcem filtrovať predikcie podľa prideleného zamestnanca a podľa distribútora - Pridať dva filtre k predikciám (filtrovanie zamestnanca a distribútora)
SS-52	3.šprint Kal-El	Import predajov	Martin Šidlo	Done	User Story	Normal	5	<ul style="list-style-type: none"> - Vytvoriť produkt k predaju ak neexistuje - Import dát zo septembra a októbra
SS-55	3.šprint Kal-El	Graf predajov	Martin Šidlo	Done	User Story	Normal	5	<ul style="list-style-type: none"> - Vytvoriť pri každom produkte graf predajov za posledné mesiace
SS-72	4.šprint Chewbacca	Reprezentácia skladu, aby sa v ňom dala hľadať dobrá cesta	Matúš Salát	Done	User Story	Normal	8	<ul style="list-style-type: none"> - statický výpočet grafu - riešenie neznámych políc - priradenie na koniec zoznamu - overenie validity pozície v sklade napr B2L2 - vytvorenie grafu a jeho uloženie do súboru alebo DB a jeho načítanie do pamäte
SS-89	4.šprint Chewbacca	Zoradiť objednávky podľa najlepšej trasy v sklade	Martin Šidlo	Done	User Story	Normal	5	<ul style="list-style-type: none"> - spraviť Dijkstru medzi dvoma vrcholmi

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-87	5.šprint Yoda	Analýza predikcie pri veľkých zmenách v počte objednávok	Lukáš Šimek, Mária Dragúňová	Done	User Story	Normal	3	<ul style="list-style-type: none"> - kombinácie všetkých zoradení kníh a spustiť nad nimi Dijkstru - vytvoriť model - analýza skokov predajov - výpočet rozdielu v počte objednávok v závislosti od času - ako často sa podobné skoky dejú - experimentovať s dĺžkou obdobia, na základe ktorého vypočítavame predikcie
SS-56	5.šprint Yoda	Automatizované testovanie scenárov	Matej Červenka, Ondrej Kudláč	Done	User Story	Normal	8	<ul style="list-style-type: none"> - Testovanie vytvorenia csv, spustenie predikcie, uloženie výsledku predikcie do DB - Testovanie improtu dát produktu z Maritnusu - Testovanie udpatu produktov - Testovanie zbehnutia Jobov
SS-110	5.šprint Yoda	Modifikácia reprezentácie skladu	Matúš Salát	Done	User Story	Normal	5	<ul style="list-style-type: none"> - zjednodušiť sklad na funkčnú jednotku - vytvorenie špeciálnych poličiek
SS-74	5.šprint Yoda	Implementácia API služby	Martin Šidlo	Done	User Story	Normal	3	<ul style="list-style-type: none"> - dohodnúť formáty - autentifikácia - API kľúč a obmedzenie na IP (aj na už existujúcu APIInu - import)
SS-88	5.šprint Yoda	Modifikácia metódy predikcie	Ondrej Kudláč	Done	User Story	Normal	5	<ul style="list-style-type: none"> - uvažovanie posledných troch týždňov v predikcii

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
								<ul style="list-style-type: none"> - porovnanie metód (stará Lukášova vs. nová Lukášova vs. Ondrejova) - doladenie starej metódy výpočtu kusov, ktoré boli preskladnené, resp. ktoré chýbali (o koľko kníh) - vyšetrenie priebehu predajov (kategorizovanie) - opraviť bug
SS-57	3.šprint Kal-El	Dokumentácia	Lukáš Šimek, Mária Dragúňová, Matej Červenka, Matúš Salát, Martin Šidlo, Ondrej Kudláč	Done	Management Story	Normal	13	<ul style="list-style-type: none"> - Napísať metodiky: verziovanie, revízia kódu, písanie kódu, testovanie, riziká, dokumentovanie. - Dokumentácia riadenia - Dokumentácia produktu
SS-12	1.šprint: E.T	Predikcia vypredania zasob	Lukáš Šimek	Done	Task	Major	Not estimated	
SS-19	1.šprint: E.T	Porovnanie s existujúcou metódou	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-13	1.šprint: E.T	Vytvorenie tabuliek	Mária Dragúňová	Done	Task	Major	Not estimated	Migrácie na vytvorenie tabuliek: Stock, Product, Sale, Distributor

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-14	1.šprint: E.T	Import dat z csv do tabuliek v dbs	Mária Dragúňová	Done	Task	Major	Not estimated	Algoritmus na naplnenie tabuliek z cvicneho csv
SS-15	1.šprint: E.T	Očistenie skusobnych dat v csv	Mária Dragúňová	Done	Task	Normal	Not estimated	Odstranenie "nezmyselných nul" Oprava neodpocitaného stock_count
SS-16	1.šprint: E.T	Prihlasovacia obrazovka	Matej Červenka	Done	Task	Normal	Not estimated	
SS-17	1.šprint: E.T	Vytvorenie rozhrania pre vypis dat	Matej Červenka	Done	Task	Normal	Not estimated	
SS-9	1.šprint: E.T	Demonizovanie smart-store RoR app	Matúš Salát	Done	Task	Normal	Not estimated	
SS-10	1.šprint: E.T	Subdomena pre prezentacny web/appku	Matúš Salát	Done	Task	Minor	Not estimated	
SS-18	1.šprint: E.T	Predikcia počtu objednaných kusov	Matúš Salát	Done	Task	Normal	Not estimated	
SS-27	1.šprint: E.T	Logger	Matúš Salát	Done	Task	Minor	Not estimated	Aplikovanie loggeru na log-filey produkčnej aplikácie
SS-11	1.šprint: E.T	HTTPS protokol	Martin Šidlo	Done	Task	Minor	Not estimated	Zavisle od instalacie webovej stránky timu na server
SS-20	1.šprint: E.T	PHP - pre generovanie linkov pre dokumenty	Martin Šidlo	Done	Task	Normal	Not estimated	- generovanie linkov podľa obsahu priečinkov - zobrazenie obsahu príloh
SS-21	1.šprint: E.T	Spustenie web stránky	Martin Šidlo	Done	Task	Normal	Not estimated	- nasadenie stránky na server (team03-16.studenti.fiit.stuba.sk)

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-35	2.šprint Spock	Vyhodnotenie predikcie počtu kusov	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-36	2.šprint Spock	Vyhodnotenie predikcie dátumu	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-38	2.šprint Spock	Overiť vyhodnotenia	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-39	2.šprint Spock	Vytvoriť štatistiky z vyhodnotení	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-43	2.šprint Spock	Doplniť dátový model	Mária Dragúňová	Done	Task	Normal	Not estimated	Doplniť dátový model o: - názov, autor k produktu - meno distribútora, zodpovednú osobu k distribútorovi - odstránenie not null constraints - pridanie Employees
SS-44	2.šprint Spock	Vytvoriť job na denný update produktov	Mária Dragúňová	Done	Task	Normal	Not estimated	
SS-48	2.šprint Spock	Jednorazový import dát do databázy	Mária Dragúňová	Done	Task	Normal	Not estimated	Import dát z csv: - employees - products - distributors
SS-26	2.šprint Spock	Background process	Matúš Salát	Done	Task	Normal	Not estimated	
SS-40	2.šprint Spock	Sidekiq demon skript	Matúš Salát	Done	Task	Normal	Not estimated	
SS-41	2.šprint Spock	Sidekiq cron jobs	Matúš Salát	Done	Task	Normal	Not estimated	Zapnúť cron joby v aplikácii

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-42	2.šprint Spock	Serverový import starých vyčistených dát	Matúš Salát	Done	Task	Normal	Not estimated	
SS-49	2.šprint Spock	Sidekiq konzola	Matúš Salát	Done	Task	Normal	Not estimated	
SS-32	2.šprint Spock	Samostatná route pre Martinus	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-33	2.šprint Spock	joby pre naplnenie dát z Martinusu do databázy	Martin Šidlo	Done	Task	Normal	Not estimated	- doplniť aj logovanie chýb (Rollbar)
SS-34	2.šprint Spock	ukladanie súborov z requestov	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-23	2.šprint Spock	Príprava DB to CSV (ruby)	Ondrej Kudláč	Done	Task	Normal	Not estimated	Príprava údajov do CSV pre predikciu. 90 dni
SS-25	2.šprint Spock	Prevzatie výstupu z ERka	Ondrej Kudláč	Done	Task	Normal	Not estimated	Prevzatie výstupu z ERKA a ďalšie práce s tým
SS-31	2.šprint Spock	Doplniť dátový model	Ondrej Kudláč	Done	Task	Normal	Not estimated	Doplniť dátový model o order prediction
SS-79	3.šprint Kal-EI	Analýza rizík	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-81	3.šprint Kal-EI	Dokument riadenia - vseobecna cast	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-84	3.šprint Kal-EI	Metodika písanie scriptov v R	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-85	3.šprint Kal-EI	Modul predikcie	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-82	3.šprint Kal-EI	Metodika úloh	Matej Červenka	Done	Task	Normal	Not estimated	
SS-83	3.šprint Kal-EI	Modul grafického rozhrania	Matej Červenka	Done	Task	Normal	Not estimated	

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-67	3.šprint Kal-EI	Setup produkcie po 2. šprinte	Matúš Salát	Done	Task	Normal	Not estimated	
SS-68	3.šprint Kal-EI	Zmena času vykonania jobu na sťahovanie produktov	Matúš Salát	Done	Task	Normal	Not estimated	
SS-69	3.šprint Kal-EI	Reset databázy, import spracovaných dát	Matúš Salát	Done	Task	Normal	Not estimated	
SS-75	3.šprint Kal-EI	Komentovanie metód projektu - časť MVC	Matúš Salát	Done	Task	Normal	Not estimated	neokomentované assety a iné detaily
SS-76	3.šprint Kal-EI	Metodika písanie kódu, code review	Matúš Salát	Done	Task	Normal	Not estimated	
SS-78	3.šprint Kal-EI	Komentovanie metód projektu - Assety	Matúš Salát	Done	Task	Normal	Not estimated	
SS-80	3.šprint Kal-EI	Modul - Pracovníci na pozadí	Matúš Salát	Done	Task	Normal	Not estimated	
SS-64	3.šprint Kal-EI	vyťahnutie dát o predajoch	Martin Šidlo	Done	Task	Normal	Not estimated	vyťahnuť dáta z predajov pre každý produkt za posledné 3 mesiace a zgrupiť ich po týždni
SS-65	3.šprint Kal-EI	zobrazovanie grafu	Martin Šidlo	Done	Task	Normal	Not estimated	premietnutie dát pre každý produkt v podobe malého okienka grafu pri prejdení myšky ponad ikonu
SS-66	3.šprint Kal-EI	Vytváranie produktu	Martin Šidlo	Done	Task	Normal	Not estimated	pri neexistujúcom produkte počas importu predajov, vytvoriť tento produkt

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-77	3.šprint Kal-El	Chybný formát dát z martinusu	Martin Šidlo	Done	Task	Normal	Not estimated	Oprava/prispôsobenie
SS-62	3.šprint Kal-El	Metodika testovanie	Ondrej Kudláč	Done	Task	Normal	Not estimated	
SS-63	3.šprint Kal-El	Dokumentácia manažment Testovanie	Ondrej Kudláč	Done	Task	Normal	Not estimated	
SS-94	4.šprint Chewbacca	Doladenie starej vyhodnocovacej metody	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-105	4.šprint Chewbacca	Experimentovanie s dlzkou obdobia, na zaklade ktoreho sa predikuje	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-106	4.šprint Chewbacca	Odhalenie skokovych predajov	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-107	4.šprint Chewbacca	Analyzovanie frekvencie skokov	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-109	4.šprint Chewbacca	Obejdnavky v ramci casu	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-113	4.šprint Chewbacca	Analýza možností predikcie	Mária Dragúňová	Done	Task	Normal	Not estimated	- analyzovať existujúce riešenia na predikovanie objednávania tovaru
SS-95	4.šprint Chewbacca	Vytvorenie modelu reprezentujúceho vrchol grafu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-96	4.šprint Chewbacca	Načítanie skladu zo súbora	Matúš Salát	Done	Task	Normal	Not estimated	

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-97	4.šprint Chewbacca	Inicializácia mapy	Matúš Salát	Done	Task	Normal	Not estimated	
SS-98	4.šprint Chewbacca	Kontrola validného pomenovania vrchola grafu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-99	4.šprint Chewbacca	Generovanie väzieb medzi vrcholmi grafu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-102	4.šprint Chewbacca	Grafický pohľad na sklad	Matúš Salát	Done	Task	Normal	Not estimated	
SS-103	4.šprint Chewbacca	Dokumentovanie kódu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-100	4.šprint Chewbacca	Implementácia Dijstrovho algoritmu	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-101	4.šprint Chewbacca	Vytvoriť kombinácie ciest	Martin Šidlo	Done	Task	Normal	Not estimated	- vytvoriť kombinácie ciest pre výber kníh v sklade - spustiť vyhľadávanie pre všetky kombinácie - vybrať najlepšie kombináciu
SS-91	5.šprint Yoda	Nastroj na porovnanie metod predikcie	Lukáš Šimek	Done	Task	Normal	Not estimated	Implementácia nástroja/skriptu na porovnanie metód predikcie implementovaných v Rku
SS-114	5.šprint Yoda	Experimentovať s dĺžkou obdobia pri novej vyhodnocovacej metode	Lukáš Šimek	Done	Task	Normal	Not estimated	

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-115	5.šprint Yoda	Pri skokovych predajov urcit percentualne navysenie (od priemeru)	Lukáš Šimek, Mária Dragúňová	Done	Task	Normal	Not estimated	
SS-58	5.šprint Yoda	Testy pre model	Matej Červenka, Ondrej Kudláč	Done	Task	Normal	Not estimated	
SS-61	5.šprint Yoda	Vytvorenie jednotlivých factory	Matej Červenka, Ondrej Kudláč	Done	Task	Normal	Not estimated	
SS-117	5.šprint Yoda	Testy pre validovanie jobu na predikciu	Matej Červenka	Done	Task	Normal	Not estimated	
SS-111	5.šprint Yoda	Okresaná reprezentácia skladu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-112	5.šprint Yoda	Rozšírená reprezentácia skladu	Matúš Salát	Done	Task	Normal	Not estimated	
SS-116	5.šprint Yoda	Serverová inicializácia	Matúš Salát	Done	Task	Normal	Not estimated	
SS-59	5.šprint Yoda	Integračné testy	Ondrej Kudláč	Done	Task	Normal	Not estimated	
SS-92	5.šprint Yoda	Metoda na predikciu podľa váh dôležitosti na jednotlivé obdobia	Ondrej Kudláč	Done	Task	Normal	Not estimated	Implementacia metody na predikciu v zavislosti roznych vah dolezitosti u sales jedneho produktu

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-93	4.šprint Chewbacca	Pridat filter na minimalny pocet kusov	Matej Červenka	Done	Bug	Normal	Not estimated	
SS-104	4.šprint Chewbacca	SmartOrder: doplnit predpoved poctu kusov na 14 dni	Matej Červenka, Ondrej Kudláč	Done	Bug	Normal	Not estimated	Nasi zamestnanci su zvyknuti objednavat tovar na sklad na 14 dni. To by sme samozrejme chceli zlepšit/zmenit. Na ulahcenie prechodu (alebo na porovnanie) by sa ale zislo mat stlpec, ktory by predpovedal pocet kusov na najblizsich 14 dni. Cize vsetko by som nechal ako je (datumy aj pocet kusov, chceme k tomu smerovat), len pridat este jeden stlpec, kde bude predpoved poctu kusov "natvrdo" na 14 dni, aby si to vedeli porovnat.

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-123	6.šprint Han Solo	SmartOrder: Porovnanie našej metódy s Martinusom	Lukáš Šimek	Done	User Story	Normal	3	- porovnať CSV od Martinusu s našou predikciou
SS-121	6.šprint Han Solo	SmartCollect: Výpočet popularity produktu	Mária Dragúňová	Done	User Story	Normal	8	- rátanie atribútu popularity produktu ako frekvencie predaju (počet predajov za 2 týždne) - na konci vyrátať sektor v sklade na základe popularity - job po importovaní predajov
SS-120	6.šprint Han Solo	SmartCollect: Zistenie polohy produktu	Matej Červenka	Done	User Story	Normal	2	- upraviť datový model (pridať produktu atribút polohu, popularitu, sektor) - upraviť import
SS-118	6.šprint Han Solo	SmartCollect: Vylepšenie dostupnosti služieb	Matúš Salát	Done	User Story	Normal	8	- monitorovanie dostupnosti - automatické riešenie zlyhania - v nastroji monit/god
SS-122	6.šprint Han Solo	SmartCollect: Výpis celej cesty	Martin Šidlo	Done	User Story	Normal	8	- nová služba na vypísanie celej cesty (každého bodu)
SS-119	6.šprint Han Solo	SmartCollect: Služba na zistenie sektora daného produktu	Ondrej Kudláč	Done	User Story	Normal	1	- vypýtať API kľúč - vstup je ID knihy, výstup je sektor
SS-131	6.šprint Han Solo	Vypracovanie výsledkov pre daná obdobie	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-132	6.šprint Han Solo	Porovnanie výsledkov	Lukáš Šimek	Done	Task	Normal	Not estimated	
SS-125	6.šprint Han Solo	Výber a aplikácia monitorovacieho nástroja	Matúš Salát	Done	Task	Normal	Not estimated	
SS-135	6.šprint Han Solo	Monitoring servera, meranie dostupnosti	Matúš Salát	Done	Task	Normal	Not estimated	

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-126	6.šprint Han Solo	Ukladanie cesty pri predpočítavaní	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-127	6.šprint Han Solo	Úprava data modelu	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-128	6.šprint Han Solo	Úprava job-ov pri výpočte cesty	Martin Šidlo	Done	Task	Normal	Not estimated	
SS-129	6.šprint Han Solo	Vytvoriť job na výpočet popularity	Mária Dragúňová	To Verify	Task	Normal	Not estimated	
SS-130	6.šprint Han Solo	Algoritmus na update popularity a sektora	Mária Dragúňová	To Verify	Task	Normal	Not estimated	
SS-124	6.šprint Han Solo	Nasadenie zmien	Matúš Salát	To Verify	Task	Normal	Not estimated	
SS-133	6.šprint Han Solo	Vypýtanie API kľúča	Ondrej Kudláč	To Verify	Task	Normal	Not estimated	
SS-134	6.šprint Han Solo	Služba na zistenie sektora	Ondrej Kudláč	To Verify	Task	Normal	Not estimated	
SS-142	7.šprint: Alien	SmartCollect: Dátová analýza nákupného košíka	Lukáš Šimek, Mária Dragúňová, Matej Červenka, Ondrej Kudláč	Done	User Story	Normal	13	- použitie datetimu namiesto čísla objednávky - koľkokrát sa predali produkty spolu
SS-143	7.šprint: Alien	SmartCollect: Nasadenie a monitoring služby na získanie sektora	Matúš Salát	Done	User Story	Major	3	Nasadenie novej služby na produkciu

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Stav	Typ	Priorita	Body	Opis
SS-152	7.šprint: Alien	SmartCollect: Refaktoring vzdialeností medzi policami	Matúš Salát	Done	User Story	Normal	8			- Prispôbiť krajné body reálnej dĺžke police - Identifikovať prípady, kedy je jedna z hrán (políc) dlhšia - Nanovo vytvoriť sklad - Pytagorova veta
SS-136	7.šprint: Alien	SmartCollect: Vizualizácia cesty v sklade	Martin Šidlo	Done	User Story	Normal	8			- návrh zobrazenia cesty (v elektronickej podobe)
SS-146	7.šprint: Alien	SmartOrder: Vylepšenie predikcie v1	Lukáš Šimek	Done	Task	Normal	Not estimated			-odstránenie "subsetov"
SS-147	7.šprint: Alien	Vylepšenie predikcie v2	Lukáš Šimek	Done	Task	Normal	Not estimated			-použitie "data.table"
SS-157	7.šprint: Alien	Analýza nákupného košíka v2	Lukáš Šimek	Done	Task	Normal	Not estimated			
SS-151	7.šprint: Alien	Dátová analýza v R	Mária Dragúňová	Done	Task	Normal	Not estimated			Vyskúšanie tutorialu http://snowplowanalytics.com/guides/recipes/catalog-analytics/market-basket-analysis-identifying-products-that-sell-well-together.html
SS-148	7.šprint: Alien	Nasadenie zmien	Matúš Salát	Done	Task	Normal	Not estimated			
SS-149	7.šprint: Alien	Monitoring	Matúš Salát	Done	Task	Normal	Not estimated			
SS-150	7.šprint: Alien	Reštartovanie servera	Matúš Salát	Done	Task	Normal	Not estimated			stara IP: 147.175.149.176
SS-158	7.šprint: Alien	Spracovanie šírky menších políc do ceny hrany	Matúš Salát	Done	Task	Normal	Not estimated			
SS-159	7.šprint: Alien	Vytvorenie skladu, evaluácia hrán	Matúš Salát	Done	Task	Normal	Not estimated			

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-144	7.šprint: Alien	X,Y súradnice pre body	Martin Šidlo	Done	Task	Normal	Not estimated	vytvorenie funkcie na výpočet X,Y súradníc pre canvas pri zobrazovaní bodov skladu
SS-145	7.šprint: Alien	Nakreslit cestu	Martin Šidlo	Done	Task	Normal	Not estimated	Nakraslit cestu do canvasu s bodmi v sklade
SS-139	7.šprint: Alien	SmartCollect: Optimalizácia výpočtu popularity	Matúš Salát	Done	Bug	Critical	Not estimated	Vyriešiť problém s narastajúcou veľkosťou procesu
SS-161	8. šprint Alf	SmartCollect: Vykreslovanie cesty	Matúš Salát	Done	User Story	Normal	3	- odstrániť header - zmenšiť mapu - dať do pozadia reálnu mapu skladu
SS-141	8. šprint Alf	SmartCollect: Mobilná aplikácia na skenovanie kódu	Martin Šidlo	Done	User Story	Normal	5	- Naskenovať kód produktu - Zavolať službu a zistiť cestu - Vykresliť cestu na mobilnom telefóne
SS-168	8. šprint Alf	SmartCollect: Analyza rozdelenia sektorov	Unassigned	Done	User Story	Normal	Not estimated	
SS-165	8. šprint Alf	Analýza nákupného košíka	Lukáš Šimek	To Verify	Task	Normal	Not estimated	- úprava výstupu, pre ďalšie spracovávanie
SS-170	8. šprint Alf	Zistenie intenzity predavanie knih spolu	Lukáš Šimek	To Verify	Task	Normal	Not estimated	
SS-173	8. šprint Alf	Umiestnenie produktov do sektorov + odporúčanie pre spoločne produkty	Lukáš Šimek	To Verify	Task	Normal	Not estimated	
SS-166	8. šprint Alf	Prispôsobenie rozhrania	Matúš Salát	To Verify	Task	Normal	Not estimated	

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-167	8. šprint Alf	Vloženie reálnej mapy grafu	Matúš Salát	To Verify	Task	Normal	Not estimated	
SS-174	8. šprint Alf	Fixovanie dotazov zo stand-upu	Matúš Salát	To Verify	Task	Normal	Not estimated	
SS-163	8. šprint Alf	Vytvoriť aplikáciu	Martin Šidlo	To Verify	Task	Normal	Not estimated	Vytvoriť aplikáciu na skenovanie kódov.
SS-153	9. šprint Prawn	SmartCollect: Presnejšie umiestnenie produktu vrámci sektora	Lukáš Šimek, Ondrej Kudláč	Done	User Story	Normal	13	<ul style="list-style-type: none"> - Zohľadniť pri premiestňovaní produktu aj produkty, ktoré sa prejdávajú s ním - Odporúčiť premiestniť produkt k polici X1Y2 - Ku každému produktu máme 3 produkty a 3 váhy - Čo je často predávané? Budeme mať pozíciu produktu, ktorý je najpredavanejší - Doplnenie vstupu a nájdenie správnej limity (počet kusov produktov)
SS-140	9. šprint Prawn	SmartCollect: Výpis všetkých zle umiestnených produktov	Mária Dragúňová, Matej Červenka, Matúš Salát	Done	User Story	Major	5	<ul style="list-style-type: none"> - Urobiť normalizáciu popularity na interval <0,1> - Vypočítať vzdialenosť, kde sa produkt nachádza, vypočítať, kde by sa MAL nachádzať - Spraviť rozdiel vzdialeností a vypísať tieto rozdiely od najväčšieho - Robiť po dávkach = nie SELECT * - Vytvoriť concern, ktorý bude obsahovať metódu zo Šidlovho helpera NodesCoordinates... Táto metóda slúži na získanie koordinátov XY vzhľadom na polohu produktu (Ľavý horný roh je [0,0], Pravý dolný [inf,inf]) - getXAxisRange, getYAxisRange - Zoznam rozdielov pridať na stránku SmartStore - prvých 200 produktov (so stránkovaním) - zoradiť od najhoršie umiestnených - ID, meno, aktuálna pozícia, optimálna pozícia, resp sektor - Pridať pozíciu policky knihy produktu, ktorým sa daný produkt predáva

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-169	9. šprint Prawn	SmartCollect: Evaluacia rozmiestnenia produktov	Martin Šidlo	Done	User Story	Normal	8	- vyhodnotenie vzdialenosti k produktu - vysledky do tabulky, ktora obsahuje rozmiestnenie produktov - porovnanie vzdialenosti - umiestnenie produktov na miesta v sklade podľa popularity a pocu kusov, po kruznici - proste ich ulozieme tak aby sme ich ulozili
SS-176	9. šprint Prawn	SmartCollect: Vizualizacia umiestnenia produktu	Unassigned	Done	User Story	Normal	3	Vyfarbit kruzok okolo idealnej pozicie
SS-164	9. šprint Prawn	Failed test - treba identifikovať príčinu a odstrániť	Ondrej Kudláč	Done	Task	Normal	Not estimated	1) CalculatePopularityJob Job for calculation of product popularity should classify product to right sector Failure/Error: expect(Product.find(100).sector.to_i).to eq 1 expected: 1 got: 3 (compared using ==) # ./spec/jobs/calculate_popularity_job_spec.rb:54:in `block (3 levels) in <top (required)>' Finished in 20.06 seconds (files took 2.01 seconds to load) 36 examples, 1 failure Failed examples:

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
								rspec ./spec/jobs/calculate_popularity_job_spec.rb:40 # CalculatePopularityJob Job for calculation of product popularity should classify product to right sector
SS-177	9. šprint Prawn	Doplnenie vystupu	Lukáš Šimek	To Verify	Task	Normal	Not estimated	
SS-178	9. šprint Prawn	Doplnit viacero stlpcov (spolu predavanych knih)	Lukáš Šimek	To Verify	Task	Normal	Not estimated	
SS-179	9. šprint Prawn	Otestovanie + najdenie idealnej limity	Lukáš Šimek	To Verify	Task	Normal	Not estimated	Hranica, kedy beriem do uvahy, ze knihy sa spolu predavaju
SS-154	9. šprint Prawn	Normalizovať popularitu	Ondrej Kudláč	To Verify	Task	Normal	Not estimated	- na interval <0,1> - použiť z nrmalizáciu
SS-171	9. šprint Prawn	Spracovanie csv s často predávanými produktami	Ondrej Kudláč	To Verify	Task	Normal	Not estimated	- vhdenie do jobu pre predikciu aj r na vypocitanie podobnych - controller pre spracovanie - view pre zobrazenie v nejakej table
SS-180	9. šprint Prawn	Vizualizacia umiestnenia produktu	Ondrej Kudláč	To Verify	Task	Normal	Not estimated	
SS-155	9. šprint Prawn	Výpočet polohy produktu	Unassigned	To Verify	Task	Normal	Not estimated	- vypočítať vzdialenosť medzi pozíciou, kde sa produkt aktuálne nachádza a kde by sa mal nachádzať - selecty robiť po dávkach
SS-156	9. šprint Prawn	Stránka na zobrazenie zle umiestnených produktov	Matej Červenka	To Verify	Task	Normal	Not estimated	- stránka so stránkovaním - zobrazí prvých 200 zle umiestnených produktov - zoradiť podľa najhoršie umeistených
SS-162	9. šprint Prawn	Otestovať performance	Unassigned	To Verify	Task	Normal	Not estimated	
SS-181	10. šprint: Groot	IIT.src: Dotazniky a prezentacie	Lukáš Šimek,	Done	User Story	Normal	9	1. Mrskova - dotaznik do 19.4. 2. robime.it - prezentacia do 20.4

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
			Mária Dragúňová, Matúš Salát					- Obrázok 200x200 3. Poster
SS-183	10. šprint: Groot	SmartCollect: Vyhodnotenie a vizualizácia spolupredávaných produktov	Lukáš Šimek, Ondrej Kudláč	Done	User Story	Normal	7	Vizualizovanie spoluprávaných produktov: - Dotiahnuť vizualizáciu, aby bola zobrazená na 'novom' sklade - Zobrazenie viacerých produktov na premiestnenie Príprava jobu na vytvorenie basket modelu
SS-184	10. šprint: Groot	SmartCollect: Optimálne rozmiestnenie spolupredávaných produktov	Matej Červenka	Done	User Story	Normal	7	Praca s modelom basket, ktorý obsahuje spolupredávané produkty Pri výpočte optimálnej pozície treba vziať do úvahy spolupredávané produkty - Je tam job, na generovanie CSV pre Lukasovu basket analýzu - tu treba spustiť jobom, to vypluje CSV a to sa uloží do Basket modelu. - Pri ukladaní produktov do skladu treba ukladať informácie o zaplčení jednotlivých uzlov. Pri uložení produktu na určitú pozíciu sa pozriem do zoznamu spolupredávaných, ak sa tam nachádza nejaká dvojčička, vezmem ju a skúsím ju uložiť "niekde blízko" = súradnica +- 1,2,..atď. Uložím produkt, a ďalej pokračujem podľa popularity. Pridanie optimálnej pozície do View
SS-182	10. šprint: Groot	SmartCollect: Vyhodnotenie skrátenia cesty	Martin Šidlo	Done	User Story	Major	3	Urobiť migráciu - pridať atribút optimal_position, ktorý sa bude updatovať v jobe na výpočet popularity. Vypočítať dĺžku cesty pre aktuálne rozloženie a pre optimálne rozloženie -> porovnať
SS-190	10. šprint: Groot	SmartStore: Dokoncenie mobilnej aplikácie	Unassigned	Done	User Story	Normal	Not estimated	- vykresliť 3 rôzne obrázky - oskenovanie kódu a vykreslenie realnej a optimálnej pozície danej knižky - switchnúť pozadie aby tam nebol realný sklad

Martians

Id	Šprint	Úloha	Pridelené	Stav	Typ	Priorita	Body	Opis
SS-185	10. šprint: Groot	Vizualizacia spolupredávaných produktov	Ondrej Kudláč	Code Review	Task	Normal	Not estimated	Vizualizacia pomocou nového skladu
SS-187	10. šprint: Groot	Zhrupenie viac predávaných produktov spolu	Ondrej Kudláč	Code Review	Task	Normal	Not estimated	
SS-189	10. šprint: Groot	SmartCollect: Pridanie stlpca	Matej Červenka	Done	Task	Normal	Not estimated	-pridat stlpec s policou (optimalna pozicia)
SS-188	10. šprint: Groot	Dokončenie jobu na výpočet basket	Ondrej Kudláč	Done	Task	Normal	Not estimated	