

Fakulta informatiky a informačných technológií
Slovenská technická univerzita

Group de'Cider

Dokumentácia k tímovému projektu časť 1
(Riadenie projektu)

Tím: číslo 1, sElectors

Pedagogický vedúci: Ing. Ondrej Kaššák

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Akademický rok: 2017/2018

Úvod	2
1 Role členov tímu a podiel práce	3
1.1 Podiel práce na dokumentácii	4
2 Aplikácie manažmentov	5
2.1 Manažment agilného riadenia softvérového vývoja	5
2.2 Manažment verzií	5
2.3 Manažment chýb	5
2.4 Manažment komunikácie	6
3 Sumarizácia šprintov	7
3.1 Šprint 1 (5.10. - 19.10.)	7
3.2 Šprint 2 (19.10. - 2.16.)	12
3.3 Šprint 3 (2.11. - 16.11.)	17
3.4 Šprint 4 (16.11. - 30.11.)	22
3.5 Šprint 5 (30.11. - 7.12.)	28
4 Globálna retrospektíva	32
4.1 Retrospektíva k šprintu 1 (26.10.2017)	32
4.2 Retrospektíva k šprintu 2 (9.11.2017)	33
4.3 Retrospektíva k šprintu 3 (16.11.2017)	33
4.3 Retrospektíva k šprintu 4 (28.11.2017)	35
Záver	37
Príloha A Export evidencie úloh	38
Šprint 1 - Ananás	38
Šprint 2 - Banán	39
Šprint 3 - Citrón	41
Šprint 4 - Datle	43
Príloha B Metodiky	46

Úvod

Dokument obsahuje podrobnú správu tímu sElectors po prvých 5 šprintoch v rámci predmetu Tímový projekt. V tomto súbore sa nachádzajú informácie k časti riadenia projektu, popisy šprintov, podiel úloh práce členov tímu, metodiky k jednotlivým častiam a pohľad na jednotlivé retrospektívy počas šprintov. Rovnako tak táto správa obsahuje aj export úloh z nástroja Jira použitého pri riadení tímu.

Celý tím mal minimálne skúsenosti s aplikovaním agilného vývoja na riešenie problémov, čo sa podpísalo na úvodných fázach riešenia projektu.

1 Role členov tímu a podiel práce

Ing. Ondrej Kaššák, PhD.

Vedúci tímu sElectors. Jeho úlohou bolo pripraviť product backlog a inštruovať nás k úspešnému zvládnutiu vývoja projektu. Počas sedení prezentuje svoje predstavy o finálnom produkte z pohľadu role product ownera. Pri nekorektnom postupe v práci nás vždy usmerní správnym smerom.

Bc. Lukáš Babula

Lukáš je zodpovedný za kontinuálne nasadzovanie. Počas vývoja produktu je potrebné manažovať procesy ohľadom kontinuálneho nasadzovania a administrovať nástroj pre kontinuálne nasadzovanie. Náš tím za týmto účelom používa Jenkins. Lukáš nastavil tento nástroj a v dohľadnej dobe sa bude naďalej venovať jeho správe a konfigurácii. Pre potreby kontinuálneho nasadzovania je tiež potrebné spravovať integračné testovanie na úrovni serveru, o čo sa stará tiež.

Bc. Pavel Balashov

Pavel je zodpovedný za administráciu nástroja určeného na podporu agilného vývoja v tíme. V našom prípade sa jedná o nástroj JIRA. Jeho úlohou bolo nasadenie tohto nástroja na náš server. Tento proces bol spojený s kompletnou konfiguráciou a manažovaním, čo zahŕňa pravidelné aktualizovanie úloh počas šprintu. Okrem toho sa Pavel podieľal na vývoji frontend-ovej časti projektu.

Bc. Michal Dolnák

Michal je zodpovedný za manažment dokumentácie v rámci tímu. Michal nastavil celé prostredie na vývoj frontendu v Angulare a spolupracoval pri procese nasadenia na server. Podieľal aj na návrhu volaní pre komunikáciu medzi backend-om a frontend-om. Implementoval Docker riešenie pre automatické testovanie aplikácie počas kontinuálnej integrácie

Bc. Michal Hucko

Michal je zodpovedný za manažment testovania v tíme. Túto oblasť zastrešuje najmä nad backendovou časťou projektu, kde sa venuje implementácii funkcionality. Väčšinu šprintov sa venoval vývoju REST API pre aplikáciu. Táto časť projektu bola naprogramovaná nad rámcom Django Rest. Taktiež sa podieľal na implementácii Docker riešenia pre automatizované nasadenie a nastaveniu Apache reverse proxy na školskom serveri.

Bc. Miroslav Valčíčák

Miroslav je zodpovedný najmä za integráciu aplikácie v rôznych prostrediach a za bezproblémovú komunikáciu frontendu s backendom. Počas šprintov sa venoval najmä nastaveniu dockera a nginxu, ktorých nastavenia sú rôzne v jednotlivých prostrediach. Okrem toho sa podieľal na úspešnom nastavení SSL certifikátu a implementovaní Google autorizačných služieb.

Bc. Andrej Vítek

Andrej je zodpovedný za správu tímovej stránky. Počas šprintov sa venoval upravovaniu obsahu a aktualizovaniu informácií. Andrej sa ďalej značne podieľal na úprave procesu verziovania a na implementácii funkcionality na backend-e. Tento proces zahŕňal aj niektoré kľúčové skripty potrebné na testovanie. Okrem iného sa podieľal na návrhu grafických rozhraní frontend-u.

Bc. Elena Štefancová

Ela je zodpovedná za návrh grafických rozhraní frontend-u. Táto úloha je spojená s prezentovaním mockupov a zapracovaním častých pripomienok tímu. Ela sa ďalej významne podieľala na organizácii externej komunikácie tímu, čo zahŕňa vypracovávanie dokumentov, písanie správ a podobne, rovnako ako internej komunikácii, napríklad spravovaním spoločného emailu aj nástroja Slack. Ela tiež pracuje na implementácii frontend-ových riešení v Angulari.

1.1 Podiel práce na dokumentácii

Tab. 1: Prehľad podielu práce na dokumentácii k riadeniu (%)

	Lukáš	Pavel	Mišo D	Mišo H	Miro	Andrej	Ela
Úvod	10%	0%	0%	90%	0%	0%	0%
Role členov tímu a podiel práce	13%	13%	13%	22%	13%	13%	13%
Aplikácie manažmentov	10%	55%	0%	35%	0%	0%	0%
Sumarizácie šprintov	5%	5%	5%	40%	5%	5%	35%
Globálna retrospektíva ZS/LS	5%	25%	15%	30%	0%	25%	0%
Motivačný dokument	30%	5%	5%	20%	5%	5%	30%
Metodiky	10%	10%	10%	10%	10%	10%	40%
Export evidencie úloh	0%	95%	0%	5%	0%	0%	0%

Tab. 2: Prehľad podielu práce na dokumentácii k inžinierskemu dielu (%)

	Lukáš	Pavel	Mišo D	Mišo H	Miro	Andrej	Ela
Úvod/Záver	5%	0%	0%	5%	0%	90%	0%
Globálne ciele	5%	0%	0%	0%	0%	90%	5%
Celkový pohľad na systém	0%	0%	20%	15%	0%	65%	0%
Modul hlas. stratégií	90%	0%	0%	0%	0%	10%	0%

Modul hlasovania	5%	5%	60%	0%	5%	25%	0%
Modul autentifikácie	5%	0%	0%	0%	95%	0%	0%
Prílohy	5%	0%	25%	0%	0%	70%	0%

2 Aplikácie manažmentov

2.1 Manažment agilného riadenia softvérového vývoja

V rámci tímového projektu sme sa pravidelne stretávali v štvrtky od 12:00 spolu s produktovým vlastníkom. Tieto stretnutia boli v miestnosti s veľkou obrazovkou a tabuľou na písanie, ktoré sme aktívne využívali. Postupom času sme si osvojili zopár pravidiel, ktoré každý musí počas stretnutia dodržiavať:

- Meškanie nie je povolené.
- Notebooky a mobily sú zakázané (okrem prezentačného a zapisovateľského notebooku).

Stretnutia boli vedené Scrum mastrom, ktorý pred každým napísal na tabuľu program v bodoch. V zásade sa postupovalo podľa dvoch formátov: vyhodnotenie šprintu + plánovanie (planning) a stretnutie v strede šprintu.

Počas plánovania sme si vybrali používateľské príbehy z backlogu a nasledoval plánovací poker. V tejto časti každý ohodnotil používateľský príbeh počtom story pointov, následne sme sa dohodli buď na rozbití používateľského príbehu na menšie, alebo sme prešli na delenie úloh. Ak sme sa pri hlasovaní nezhodli, prebehla krátka debata, po ktorej sa hlasovanie opakovalo. Takto sme postupovali, až kým sme sa nezhodli. Po tomto procese bol úlohám pridelený za zodpovedný člen. Každý z nich sme evidovali v systéme JIRA, kam má každý člen prístup. V prípade stretnutia v strede šprintu sme na začiatku prešli stav jednotlivých úloh a vyťaženie členov tímu. Ak sa našiel niekto, kto mal problém so svojou úlohou, snažili sme sa mu čo najefektívnejšie pomôcť.

Okrem štandardných stretnutí v čase predmetu sme si každý druhý týždeň vymedzili čas na retrospektívu. Termín tohto stretnutia bol v nasledujúcom týždni. Retrospektíva prebiehala v troch kolách, pred ktorými mal každý člen čas na rozmyslenie. V prvom kole sme sa vyjadrovali k tomu, čo sa nám páčilo. V druhom sme odpovedali na otázku toho čo sa nám na šprinte nepáčilo a v poslednom sme predostreli riešenie. Počas jednotlivých kôl mohol rozprávať len jeden človek a reakcie boli povolené až po skončení kola. Viac detailov tohto postupu obsahuje **metodika pre agilné riadenie softvérového vývoja**.

2.2 Manažment verzií

Pre správu verzií kódu sme použili GIT a GitLab. Každý člen tímu počas vývoja pracoval s vetvami v tomto nástroji a vytvoril niekoľko commitov. Viac o tom možno nájsť v **metodike práce s verziami**.

2.3 Manažment chýb

GitLab sme taktiež použili pre manažovanie chýb (angl. issue tracking). Ak pri práci na úlohe člen tímu zistí, že aplikácia alebo jej časť nefunguje správne, ale v minulosti funkčná bola alebo bola za funkčnú označená, toto chybné správanie podrobne opíše do GitLab issues. Počas prvých dvoch šprintov sme podcenili potrebnosť tohto nástroja, ale od tretieho šprintu sme ho začali pravidelne používať.

2.4 Manažment komunikácie

Jedným z kľúčových prvkov manažmentu je komunikácia. V rámci projektu ju vieme rozdeliť na internú (medzi členmi), kde okrem osobných stretnutí používame ako nástroj Slack a Appear.in (hromadné video chaty), a externú (medzi vedením a potenciálnymi zákazníkmi), kde postačuje email na gmaily. Na komunikáciu s produktovým vlastníkom používame taktiež Slack.

Počas priebehu práce na projekte prešla najmä interná komunikácia niekoľkými vývojovými štádiami, ktoré sa ju snažia viac zefektívniť.

2.5 Manažment dokumentácie

Na zladenie členov tímu pri písaní a dokumentovaní zdrojového kódu slúži manažment dokumentácie. Pre dokumentovanie frameworku Django Rest sme zvolili nástroj Swagger. Na dokumentáciu Angularu sme použili nástroj Compodoc. Počas vývoja sme uvažovali o nasadení týchto dokumentácií na náš server vo forme HTML stránok, ale nakoniec sme ostali pri ich lokálnom používaní, lebo naša aplikácia nie je open source.

Podrobnosti nájdete v metodike pre dokumentáciu.

3 Sumarizácia šprintov

3.1 Šprint 1 (5.10. - 19.10.)

Úvodný šprint bol najmä o zoznamovaní sa s agilným vývojom pomocou SCRUMU. V tomto šprinte sme story pointy pridelovali jednotlivým úlohám, čo sa v neskoršej fáze zmenilo. Planning poker prebiehal ku každému jednému z nich.

Definícia POST/GET

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Michal Hucko	Áno

Definovanie rozhrania a procesu tvorby rozhrania medzi frontend-om a backend-om.

Návrh DB schémy

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

Navrhnutie dátového modelu pre funkcionality a jeho vizualizácia pomocou UML diagramu.

Implementácia backend v Django REST - jednoduché hlasovanie

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Implementácia základného vytvárania udalostí spolu s funkcionalitou hlasovania. Bude sa dať vytvoriť hlasovanie spolu s množinou možností. Následne je možné anonymne hlasovať za jednu z nich. Táto úloha je spojená aj s úvodnou inicializáciou projektu nad frameworkom.

Docker setup

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

Vytvorenie konfiguračných súborov pre docker-compose a jednotlivé moduly.

Frontend - infraštruktúra

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Ela	Áno

Vytvorenie nového projektu a nastavenie technológie. Pridanie prekladov, volania backend-u. Nastavenie ciest v aplikácii.

Frontend - zaučenie, zdokumentovanie

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Ela	Áno

Podpora ostatných členov tímu pri svojich úlohách. Predanie svojich vedomostí o technológii.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Nie

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností. Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Dôvod nesplnenia: Úloha nebola splnená z dôvodu komplikácií pri integrácii frontend a backend.

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Dôvod nesplnenia: Úloha nebola splnená z technických príčin - Elenin operačný systém nebol kompatibilný s požiadavkami implementačných nástrojov.

Nasadenie JIRA

Zodpovedný	Overovateľ	Splnené
Andrej	Pavel	Áno

Nasadenie kolaboračného systému JIRA na pridelený server.

Požiadat' o navýšenie RAM na tímovom serveri

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Požiadat' P. Lacka o zvýšenie parametrov servera.

Migrovanie úloh

Zodpovedný	Overovateľ	Splnené
Pavel	Ela	Áno

Migrovanie úloh z Asana. Zahŕňa prepísanie úloh do JIRA a kontrolu ďalším členom tímu.

Prihlásenie sa do TP Cup

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Ela	Áno

Vytvorenie prihlášky na TP cup, ktorá bude zahŕňať všetky body zo stránky predmetu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi. Po finalizácii sa projekt riadne odovzdá do AISu.

Metodika - dokumentovanie frontend kódu

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Vytvorenie metodiky pre dokumentovanie frontend-u.

Metodika - práca s GitLab

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Vytvoriť metodiku ku práci s GitLabom:

- vetvy a ich názvy
- komentáre ku commit-u
- merge request a komentáre

Inštalácia a konfigurácia Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Áno

Nainštalovanie Jenkins na server. Nastavenie a zabezpečenie Jenkins. Nastavenie Gitlab notifikácie pre Jenkins. Vytvorenie Jenkins projektu. Nastavenie SCM v projekte. Rozšírenie Slacku o Jenkins agenta. Integrácia Slack kanála do Jenkins.

Vytvorenie tímového webu

Zodpovedný	Overovateľ	Splnené
Andrej, Lukáš	Pavel	Áno

Vytvorenie tímovej stránky:

- rozloženie elementov
- informácie o tíme a projekte
- šprinty
- dokumenty
- kontakt

Prieskum ohľadom stratégií hlasovania

Zodpovedný	Overovateľ	Splnené
Lukáš	Pavel	Nie

Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.

Dôvod nesplnenia: Analýza mnohých používaných a zaujímavých stratégií bola časovo výrazne náročnejšia, než bolo odhadované. Na konci šprintu bola preto úloha dokončená len na približne 60 %.

Vytvorenie mockupov pre aspoň 5 ďalších stratégií

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.

Dôvod nesplnenia: Úloha nebola splnená, pretože priamo nadväzuje na úlohu *Prieskum ohľadom stratégií hlasovania*, ktorá nebola dokončená.

Návrh grafických rozhraní

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvoriť prvotné mockupy pre rozhrania + vygenerovať CSS.

Dôvod nesplnenia: Úloha sa odložila na ďalší šprint, keďže nebola sfinalizovaná vízia produktu. Bola vykonaná len analýza v oblasti podobných webov.

3.2 Šprint 2 (19.10. - 2.16.)

Na predchádzajúcom šprinte sa prejavila neznalosť agilného vývoja, čo zapríčinilo nesprávny odhad úloh, ktorých sme si dali viac ako sme boli schopní zvládnuť. Všetky nedokončené úlohy sa preniesli do tohto šprintu.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Nie

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností.

Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Dôvod nesplnenia: Úloha nebola splnená z dôvodu komplikácii pri integrácii frontend a backend.

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Dôvod nesplnenia: Technické problémy pretrvávali, z týchto dôvodov bolo nutné používať na vykonávanie úlohy VirtualBox, na ktorého efektívne využitie nemá Elena vhodný počítač. Z tohto dôvodu bola síce funkcionálna naprogramovaná, ale nestihli sa napísať testy.

Nasadzovanie projektu cez Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Nie

Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.

Dôvod nesplnenia: Problémy s nastavením adekvátnych práv pre komunikáciu Jenkins a Docker, čo výrazne oneskorilo ich integráciu. To bolo jednou z príčin nepreviazania Jenkins nástroja s projektom. Ďalšou príčinou bola nepripravenosť a neúplnosť častí projektu, ktoré mali byť použité na testovanie a nasadenie. Na konci šprintu preto nebolo možné odhadnúť, na koľko percent bola úloha splnená, pretože pokrok nebolo možné kvantifikovať.

Vytvorenie mockupov pre aspoň 5 ďalších stratégií

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Nie

Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.

Dôvod nesplnenia: Stále neboli vybrané stratégie, úloha sa teda odložila na neurčito.

Návrh grafických rozhraní

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Vytvoriť prvotné mockupy pre rozhrania + vygenerovať CSS.

Poznámka: Po doplnení prvého náčrtu finálnej vízie produktu boli využité poznatky z analýzy a úloha dokončená.

Prieskum ohľadom stratégií hlasovania

Zodpovedný	Overovateľ	Splnené
Lukáš	Pavel	Áno

Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.

Metodika testovania backend-u

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

V tejto metodike pokryjeme všetky detaily spojené s testovaním backend-u. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.

Metodika - dokumentovanie backend kódu

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

V tejto metodike pokryjeme všetky detaily spojené s dokumentáciou backend-u. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.

Metodika - testovanie frontend-u

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Vytvorenie metodiky pre testovanie frontend-u.

Vyjednať navýšenie RAM + certifikát

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Napísať Peťovi Lackovi o zvýšenie RAM na serveri a zistiť či je možné získať SSL certifikát pre projekt. SSL je potrebný pre integráciu modulu prihlasovania cez Google+.

Dokonfigurovať Docker

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Dokonfigurovanie docker na serveri. Tento krok predchádza samotnému spusteniu dockeru na serveri. Je potrebné doinštalovať docker-compose. Nastaviť práva pre skupinu docker a otestovať spustenie kontajneru.

Backend - Úprava voľby hlasovania

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Používateľský príbeh: Úprava voľby akéhokoľvek používateľa

Backend pre zmenu hlasovania používateľa cez HTTP PUT metódu.

Backend - Ukončenie hlasovania

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Používateľský príbeh: Ukončenie hlasovania

Umožniť ukončenie hlasovania cez backend:

- úprava modelu hlasovania
- vytvorenie URL pre administráciu
- zabezpečiť, že sa nedá hlasovať na uzavretom hlasovaní

Metodika - dokumentovanie frontend kódu

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Andrej	Áno

Vytvorenie metodiky pre tvorbu a dokumentáciu frontend-ového kódu.

Metodika - vývoj a integrácia

Zodpovedný	Overovateľ	Splnené
Miroslav	Pavel	Nie

Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.

Dôvod nesplnenia: Až do konca šprintu nebol istý konkrétny postup pri vývoji kvôli problémom s prípravou jednotlivých prostredí a nástrojov.

Nastavenie Apache na serveri

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Nie

Nastavenie Apache na serveri, aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.

Dôvod nesplnenia: Nesplnenie 2 frontend-ových úloh, na ktorých bola úloha priamo závislá: hlasovacia tabuľka a zobrazenie výsledkov, vytvorenie hlasovacieho formulára.

Merge do mastra

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Úloha zahŕňa zodpovednosť nad tým aby sa stihli spojiť kódy frontend-u a backend-u. V prípade akýchkoľvek problémov zodpovedná osoba poskytne pomoc.

Frontend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Nie

Používateľský príbeh: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google OAuth:

- vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu
- odhlásenie používateľa
- udržiavanie stavu prihlásenia na strane klienta
- vytvorenie sekcie prístupnej len pre prihláseného používateľa
- integrácia riešenia s backend-om

Dôvod nesplnenia: Úloha sa počas analýzy a implementácie ukázala zložitejšia ako pôvodný odhad z dôvodu integrácie viacerých technológií.

Backend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Nie

Používateľský príbeh: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google

- vytvorenie používateľa cez Google účet (OAuth2)
- pridanie nášho hostname do overených hostanemov (Google)
- vytvorenie tajných kľúčov (Google)
- vygenerovanie SSL certifikátu
- nakonfigurovanie Apache s certifikátom
- nakonfigurovanie nginx
- vytvorenie autentifikačného modulu na backend-e

Dôvod nesplnenia: Úloha sa počas analýzy a implementácie ukázala zložitejšia ako pôvodný odhad z dôvodu integrácie viacerých technológií.

3.3 Šprint 3 (2.11. - 16.11.)

V poslednom šprinte boli všetky nedokončené úlohy z minulých šprintov prenesené. V tomto šprinte sme sa snažili dotiahnuť problémy s nasadením a frontend-ové úlohy.

Frontend - hlasovacia tabuľka a zobrazenie výsledkov

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Áno

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Implementácia frontend-u pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností.

Prvky grafického rozhrania:

- tabuľka
- statický text pre možnosti
- textové pole pre meno
- hlasovacie tlačidlá pre možnosti
- tlačidlo pre potvrdenie hlasovania

Frontend - vytvorenie hlasovacieho formulára

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Používateľský príbeh: Vytvorenie jednoduchého hlasovania

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- Názov
- Možnosti hlasovania v textovom bloku pod sebou
- API volanie po potvrdení
- Vygenerovanie URL na admin a hlasovacie rozhrania

Frontend

Zodpovedný	Overovateľ	Splnené
Elena	Michal Dolnák	Áno

Používateľský príbeh: Uzavrieť hlasovanie

Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre:

- API volania pre prerušenie hlasovania
- Nová admin obrazovka (názov hlasovania, jeho prerušenie)
- Testy

Nasadzovanie projektu cez Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Nie

Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.

Dôvod nesplnenia: Tesne pred dokončením vyplávala na povrch skutočnosť, že na serveri nie je možné spustiť end to end testy pre frontend. Kým teda neprebehne konfigurácia testov so serverovým prostredím, nebude možné vykonávať tieto testy pomocou Jenkins. Úloha je preto síce teoreticky splnená, ale po praktickej stránke je závislá na serverovom testovacom prostredí, takže kým nebude táto skutočnosť vyriešená, nie je možné považovať ju za splnenú, pretože riešenie počas testovania nevykonáva end to end frontend testy.

Nastavenie Apache na serveri

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Nastavenie Apache na serveri, aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou. Je potrebné otestovať komunikáciu s vytvoreným kontajnerom.

Frontend - úprava voľby hlasovania

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Dolnák	Áno

Používateľský príbeh: Úprava voľby akéhokoľvek používateľa

Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutia voľby do stavu upravovania a uloženia zmien.

Frontend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Používateľský príbeh: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google OAuth:

- vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu
- odhlásenie používateľa
- udržiavanie stavu prihlásenia na strane klienta
- vytvorenie sekcie prístupnej len pre prihláseného používateľa
- integrácia riešenia s backend-om

Backend - Vytvorenie prihlasovania

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Hucko	Áno

Používateľský príbeh: Vytvorenie prihlasovania

Vytvorenie prihlasovania cez Google:

- vytvorenie používateľa cez Google účet (OAuth2)
- pridanie nášho hostname do overených hostanemov (Google)
- vytvorenie tajných kľúčov (Google)
- vygenerovanie SSL certifikátu
- nakonfigurovanie Apache s certifikátom
- nakonfigurovanie nginx
- vytvorenie autentifikačného modulu na backend-e

Frontend - Voľba hlasovacej stratégie

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Pavel	Áno

Používateľský príbeh: Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.

Používateľ si po vybratí názvu hlasovania a zadania možností vyberie, ktorú z dostupných hlasovacích stratégií si chce vybrať.

- Pridanie kroku zvolenia stratégie do procesu vytvárania hlasovania.
- Automatická zmena popisu hlasovacej stratégie.

Backend - Voľba hlasovacej stratégie

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Andrej	Áno

Používateľský príbeh: Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.

Doplní sa funkcionalita pri vytváraní hlasovania, ktorá používateľovi umožní výber konkrétnej hlasovacej stratégie. V tomto prípade ide o nastavenie backend-ovej časti.

Backend - Skript testovacej databázy

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Vytvorenie príkazov na vytvorenie testovacej databázy pre frontend s možným zálohovaním a obnovením databázy.

Metodika - vývoj a integrácia

Zodpovedný	Overovateľ	Splnené
Miroslav	Pavel	Áno

Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.

Metodika - agilný vývoj

Zodpovedný	Overovateľ	Splnené
Pavel	Elena	Áno

Vytvorenie metodiky pre riadenie vývoja softvéru, ktoré je založené na metodológii Scrum.

- role a zodpovednosti
- udalosti
- artefakty

Vytvoriť metodiku pre metodiky

Zodpovedný	Overovateľ	Splnené
Pavel	Elena	Áno

Vytvorenie metodiky pre zjednotenie vzhľadu a obsahu ostatných metodík.

Časti:

- Definícia a účel
- Štýl
- Obsah

Vytvoriť šablónu pre metodiky

Zodpovedný	Overovateľ	Splnené
Elena	Pavel	Áno

Vytvorenie LaTeX šablóny pre metodiky:

- Titulná strana
- Jazykové nastavenia
- Rozloženia a formátovanie

Návrh UX

Zodpovedný	Overovateľ	Splnené
Elena	Andrej	Áno

- Návrh farebnej schémy
- Pozadie pre aplikáciu
- Bez rozloženia elementov
- Branding

3.4 Šprint 4 (16.11. - 30.11.)

V tomto šprinte boli všetky nedokončené úlohy z minulých šprintov dokončené. Taktiež sme sa posunuli ďalej v návrhu UX a dokončili doterajšiu dokumentáciu.

Vytvoriť šablónu pre metodiky

Zodpovedný	Overovateľ	Splnené
Elena	Pavel	Áno

Vytvorenie LaTeX šablóny pre metodiky

- Titulná strana
- Jazykové nastavenia
- Rozloženia a formátovanie

Nasadzovanie projektu cez Jenkins

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Áno

Používateľský príbeh: Nasadenie

Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.

Vytvorenie dokumentu k mentoringu

Zodpovedný	Overovateľ	Splnené
Elena	Lukáš	Áno

Vytvorenie dokumentu, ktorý nás prezentuje, k mentoringu TP Cup.

Dokument má časti

- agile
- biznis

Nasadiť Apache

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Používateľský príbeh: Spojenie celej infraštruktúry serveru s okolím cez Apache ako http proxy server.

Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.

Doplnenie úloh do JIRY

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Pridanie nových úloh do JIRY.

Frontend

Zodpovedný	Overovateľ	Splnené
Elena	Pavel	Áno

Používateľský príbeh: Ukončenie hlasovania

- API volania pre prerušenie hlasovania
- Nová admin obrazovka (názov hlasovania, jeho prerušenie)
- Testy

Vytvorenie wireframov

Zodpovedný	Overovateľ	Splnené
Elena	Andrej	Áno

Používateľský príbeh: Úprava dizajnu používateľského rozhrania

Vytvorenie wireframov

- wireframe 1
 - pred formulárom pre stratégiu bude jej popis
 - carousel
 - obrázok + popis + kedy sa hodí
- wireframe 2
 - širšie tabky s vysvetľujúcim obrázkom

Úprava loga

Zodpovedný	Overovateľ	Splnené
Miroslav	Elena	Áno

Používateľský príbeh: Úprava dizajnu používateľského rozhrania

Zpracovanie pripomienok k logu zo stretnutia.

Konfigurácia Apache a Dockerfiles (develop, test, deploy)

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Používateľský príbeh: Konfiguračné súbory pre Jenkins

Úprava Apache konfigurácie pre presmerovávanie stránok /test/ pre test a /app/ pre produkciu.

Dockerfile pre Develop (mapovanie portov pre vývoj na localhoste).

Dockerfile pre Test (mapovanie portov pre testovanie, rozdielne od produkcie, použitie testovacej databázy).

Dockerfile pre Deploy (mapovanie portov produkcie, odstránenie všetkých zbytočností z konfiguračného súboru).

Konfigurácia Django nastavení a testovacej databázy

Zodpovedný	Overovateľ	Splnené
Miroslav	Michal Dolnák	Áno

Používateľský príbeh: Konfiguračné súbory pre Jenkins

Konfigurácia Django nastavení pre vývoj Angularu na localhoste.

Konfigurácia Django nastavení pre použitie testovacej databázy v testovacom prostredí.

Nastavenie testovacej databázy na serveri.

Rozšírenie Jenkins konfigurácie pre testovanie

Zodpovedný	Overovateľ	Splnené
Lukáš	Miroslav	Áno

Používateľský príbeh: Konfiguračné súbory pre Jenkins

Konfigurovať Jenkins pre testovanie na novo-nastavenom testovacom prostredí.

Prispособenie Protractor testov

Zodpovedný	Overovateľ	Splnené
Pavel	Lukáš	Áno

Používateľský príbeh: Konfiguračné súbory pre Jenkins

Úprava frontend testov pre použitie sériovo za sebou na testovacej databáze.

Protraktor v Dockri

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Lukáš	Áno

Používateľský príbeh: Konfiguračné súbory pre Jenkins

Nasadenie nástroja Protractor na Docker. Presunutie testov z kontajnera Frontend do kontajnera Protractor. Konfigurácia nastavení pre Develop testovanie na localhoste a testovanie na serveri, ktoré sa dopytuje na stránku /test/.

Spojenie metodík testovania

Zodpovedný	Overovateľ	Splnené
Michal	Michal	Áno

Používateľský príbeh: Odovzdanie riadenia - metodiky

Spojenie metodík testovania frontendu a backendu do jednej metodiky.

Spojenie metodík

Zodpovedný	Overovateľ	Splnené
Elena	Lukáš	Áno

Používateľský príbeh: Odovzdanie riadenia - metodiky

Vloženie metodík do vytvorenej šablóny a pridanie do príloh v dokumentácii riadenia.

Spojenie metodík dokumentácie

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Michal Dolnák	Áno

Používateľský príbeh: Odovzdanie riadenia - metodiky

Spojenie metodík dokumentácie frontendu a backendu do jednej metodiky.

Export evidencie úloh

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Odovzdanie riadenia - ostatné

Burndown chart pre jednotlivé šprinty a export úloh po jednotlivých týždňoch.

Doplniť retrospektívu 3

Zodpovedný	Overovateľ	Splnené
Pavel	Lukáš	Áno

Používateľský príbeh: Odovzdanie riadenia - ostatné

Skontrolovať a pridať retrospektívu 3 do dokumentácie riadenia.

Popis úloh

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Lukáš	Áno

Používateľský príbeh: Odovzdanie riadenia - ostatné

Pre každý task (každý zodpovedá za svoje) dopísať popis (ak chýba) a zdôvodnenie v prípade, keď úloha nebola splnená.

Doplniť user story k taskom

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Lukáš	Áno

Používateľský príbeh: Odovzdanie riadenia - ostatné

Pre jednotlivé tasky (každý má zodpovednosť za svoje) doplniť k akému story patrí.

Rozdelenie aplikácie manažmentov

Zodpovedný	Overovateľ	Splnené
Pavel	Ela	Áno

Používateľský príbeh: Odovzdanie riadenia - ostatné

Rozdeliť aplikácie manažmentov podľa zápisnice zo 6. stretnutia. Prípadné rozšírenie dokumentu o ďalšie manažmenty.

Modul hlasovacích stratégií

Zodpovedný	Overovateľ	Splnené
Lukáš	Andrej	Áno

Používateľský príbeh: Odovzdanie inžinierskeho diela

Pridať analýzu, návrh, implementáciu a testovanie pre modul.

Modul hlasovania

Zodpovedný	Overovateľ	Splnené
Michal	Lukáš	Áno

Používateľský príbeh: Odovzdanie inžinierskeho diela

Pridať analýzu, návrh, implementáciu a testovanie pre modul.

Modul autentifikácie

Zodpovedný	Overovateľ	Splnené
Miroslav	Lukáš	Áno

Používateľský príbeh: Odovzdanie inžinierskeho diela

Pridať analýzu, návrh, implementáciu a testovanie pre modul.

Celkový pohľad - frontend

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Lukáš	Áno

Používateľský príbeh: Odovzdanie inžinierskeho diela

Rozpísať časť celkového pohľadu na frontend. Poprípade pridať diagram.

Zhodnotenie a vízia

Zodpovedný	Overovateľ	Splnené
Andrej	Michal Hucko	Áno

Používateľský príbeh: Odovzdanie inžinierskeho diela

Pridanie zhodnotenia a vízie do dokumentácie.

3.5 Šprint 5 (30.11. - 7.12.)

V poslednom šprinte bola dokončená dokumentácia za zimný semester a posuny ako v implementácii, tak aj v UX.

Export burndown chartu z JIRY

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Vytvorenie obrázku burndown chartu a vloženie do dokumentu riadenia.

Export evidencie úloh z JIRY

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Export úloh z JIRY z nových šprintov.

Prepis úloh do dokumentu

Zodpovedný	Overovateľ	Splnené
Elena	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Vloženie úloh z nových šprintov do dokumentu riadenia.

Úprava metodiky testovania

Zodpovedný	Overovateľ	Splnené
Lukáš	Michal Dolnák	Áno

Používateľský príbeh: Dokumentácia riadenia

Úprava metodiky podľa vykonaných zmien:

- pridať Protractor
- pridať Jenkins link
- pridať opis špecifických operácií

Úprava metodiku vývoja a integrácií

Zodpovedný	Overovateľ	Splnené
Miroslav	Ela	Áno

Používateľský príbeh: Dokumentácia riadenia

Úprava podľa nových zmien.

Pridať popisy k úlohám

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Pridať popisy k úlohám, poprípade napísať, prečo úloha nebola splnená.

Prečítanie a aktualizácia

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Prečítanie a úprava dokumentu podľa zmien v nových šprintoch.

Úprava metodiky verzií

Zodpovedný	Overovateľ	Splnené
Lukáš	Elena	Áno

Používateľský príbeh: Dokumentácia riadenia

Pridanie a spracovanie pripomienok.

Revízia metodík

Zodpovedný	Overovateľ	Splnené
Elena	Michal Hucko	Áno

Používateľský príbeh: Dokumentácia riadenia

Finálna revízia metodík ako celku.

Úprava diagramov a obrázkov

Zodpovedný	Overovateľ	Splnené
Andrej Vítek	Michal Hucko	Áno

Používateľský príbeh: Dokument inžinierskeho diela

Zmena a prídanie diagramov a obrázkov podľa zmien v nových šprintoch.

Dokumentácia zo Swaggera

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Miroslav	Áno

Používateľský príbeh: Dokument inžinierskeho diela

Analýza riešenia technickej dokumentácie. V princípe ide o rozhodnutie či opraviť CSS na stránke alebo spraviť export PDF.

Prečítanie a aktualizácia

Zodpovedný	Overovateľ	Splnené
Pavel	Michal Hucko	Áno

Používateľský príbeh: Dokument inžinierskeho diela

Prečítanie a úprava dokumentu podľa zmien v nových šprintoch.

Natáčanie a testovanie

Zodpovedný	Overovateľ	Splnené
Pavel	Elena	Áno

Používateľský príbeh: Testovanie farieb a opisov

Otestovanie farieb a opisu aplikácie pomocou 30 sekundového testu a zhodnotenie.

Natáčanie a testovanie

Zodpovedný	Overovateľ	Splnené
Miroslav	Elena	Áno

Používateľský príbeh: Testovanie farieb a opisov

Otestovanie farieb a opisu aplikácie pomocou 30 sekundového testu a zhodnotenie.

Natáčanie a testovanie

Zodpovedný	Overovateľ	Splnené
Elena	Pavel	Áno

Používateľský príbeh: Testovanie rozmiestnenia

Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.
Príprava podkladov, scenára, konečná revízia.

Natáčanie a testovanie

Zodpovedný	Overovateľ	Splnené
Lukáš	Elena	Áno

Používateľský príbeh: Testovanie rozmiestnenia

Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.
Otestovanie na 2 ľuďoch.

Natáčanie a testovanie

Zodpovedný	Overovateľ	Splnené
Andrej	Elena	Áno

Používateľský príbeh: Testovanie rozmiestnenia

Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.
Otestovanie na 2 ľuďoch.

Zabezpečenie JIRY

Zodpovedný	Overovateľ	Splnené
Michal Dolnák	Michal Hucko	Áno

Používateľský príbeh: Zabezpečenie serverových aplikácií

Úprava Apache súborov a reštart JIRY.

Zabezpečenie Jenkins

Zodpovedný	Overovateľ	Splnené
Michal Hucko	Michal Dolnák	Áno

Používateľský príbeh: Zabezpečenie serverových aplikácií

Úprava Apache súborov, dodtatočná konfigurácia prostredia Jenkins a jeho reštart.

Pridanie úloh do JIRY

Zodpovedný	Overovateľ	Splnené
Andrej Vítek	Michal Hucko	Áno

Pridanie nových úloh do JIRY.

4 Globálna retrospektíva

Na lepšie pochopenie zmien v tíme a samotnej globálnej retrospektívy poskytujeme podrobný prehľad ku každému šprintu.

4.1 Retrospektíva k šprintu 1 (26.10.2017)

Problémy:

1. Máme vedúcich pre oblasti/sme rozdelení na podtímy (frontend, backend).
2. Hovoríme príliš dlho k jednotlivým bodom, aj keď už je zrejmá pointa.
3. Odbočujeme od témy.
4. Dlho riešime veci ohľadom scrumu.
5. Chaos ohľadom story pointov.
6. V Jire je veľa taskov bez opisu.
7. Namiesto zápisnice písať veci do Jiry.
8. Zlé odhadovanie taskov.
9. Nie sme jedna masa. Nie sme *my*, ale každý je sám na probléme.
10. Chýba nám tímové cítenie.
11. Chýba nám disciplína. Častokrát meškáme.
12. Dôslednejšie a presnejšie metodiky a chaos v slacku.
13. Nestíha sa obed po tímovom stretnutí

Možné riešenia:

- Nebudeme sa tak oslovovať [k bodu 1].
- Gesto zdvihnutie ruky, aby niekto skončil/predtým ako bude niekto rozprávať dostane časový limit [k bodu 2].
- Napíšeme metodiku [všeobecne].
- Zápisnica je užitočná a ostáva [k bodu 7].
- Časom sa to zlepší. Nie je, že *ja* som nestihol, ale že *my* sme nestihli [k bodu 9 a 10].
- Pivo aspoň raz za mesiac [k bodu 10].
- Tresty (nosenie jedla za meškanie) [k bodu 11].
- Používať channely a thready [k bodu 12].
- Ješť na FIIT [k bodu 13].

Ďalšie pripomienky:

- Nebáť sa prehovoriť aj keď nevieme, aký je problém.

4.2 Retrospektíva k šprintu 2 (9.11.2017)

Pozitíva:

- Prijali sme problémy z minula.
- Zlepšenie disciplíny. Že pracujeme všetci.
- Neodbočujeme od tém.
- Spolupráca v tíme. Prejavilo sa to hlavne v integrácii.
- Spôsob komunikácie.

- Prístup k tomuto predmetu. Neberieme to ako nutné zlo.
- SSL certifikát.
- Ondrejov čas na retrospektívy (extra). Vedenie tímu.

Problémy:

1. Chýbajú priority taskov.
2. Chýbajú popisy taskov.
3. Neprečítali sme celú stránku.
4. Scrum poker je o bodoch.
5. Treba si písať viac progres do Jiry.
6. Utorok a Sobota statusy do Slacku (niekedy sa pozabudne).
7. Dlho čakali merge requesty.
8. Neskoro chodíme na stretnutia.
9. Závidíme si story pointy.

Možné riešenia:

- Mali by sme viac myslieť na ostatných pri vykonávaní svojich úloh (niekto na nás môže čakať) [všeobecne].
- Presunieme retrospektívu na štvrtok hneď po šprinte [všeobecne].
- Každý si bude viac upravovať tasky [k bodu 1 a 2].
- Budeme viac písať issues (gitlab) [všeobecne].
- Pripomienky do gmailu o statusoch [k bodu 6].
- Scrum master bude písať statusy ako prvý (pokúsi sa), pridá @channel [k bodu 6].
- Scrum master rozdá cukríky každému kto ho predbehne po 23:00 so statusom v utorok a sobotu [k bodu 6].
- Pri merge requeste budeme posúvať zodpovednosť ďalej (hľadať ďalšiu osobu) [k bodu 7].

4.3 Retrospektíva k šprintu 3 (16.11.2017)

Pozitíva:

- Sústredili sme sa na to, aby sme stihli šprint.
- Ela včas nahlásila to, že nestíha dokončiť svoju úlohu kvôli chorobe.
- Paľo venuje veľa času nastavovaniu a aktualizovaniu JIRA.
- Andrej inžinierske dielo.
- Miro ochotný komunikovať.
- Jenkins vyzerá sľubne, Lukáš aktívne komunikuje.
- Dokumentácia projektu od chalanov.
- Tomuto projektu venujeme veľa času.
- Zodpovedne makáme.
- Makáme jeden za druhého.
- Andrej stihol spraviť všetky svoje úlohy o dosť skôr ako na konci šprintu.
- Veľké množstvo pripomienok k metodike riadenia agilného vývoja od Ely.
- Zmena vnímania ohľadom písania metodík, zlepšujeme sa.
- Andrej je dobrý spolupracovník pri práci na UX.
- Rozpracovanosť metodík.
- Začali sme logovať prácu v JIRA.

Problémy:

1. Cítim sa ako keby som pracoval s ľuďmi v Amerike, lebo pracujeme v iných časových intervaloch (niekto cez deň, niekto neskoro v noci alebo skoro ráno).
2. Niektorí začínajú robiť na svojich úlohách až na konci šprintu.
3. Máme problém odkomunikovať prekážky v rámci našich úloh.
4. Pred implementáciou úloh nerobíme návrh.
5. Naháňame SP.
6. Treba komunikovať pri práci na úlohách, ktoré na seba nadväzujú.
7. Tomuto projektu venujeme veľa času.
8. Nevieme dobre naplánovať prácu na úlohách.
9. Neuvedomujeme si to, čo obnášajú niektoré úlohy.
10. Máme problémy so spoluprácou.
11. Hneď sme sa vrhli do implementácie.
12. Nenastavili sme si procesy.
13. Definition of done je dosť implicitný.
14. Chýba nám dokumentovanie úloh.
15. Potrebujeme sa lepšie pripravovať na stretnutia.
16. Pdlíšné vnímanie zodpovednosti JIRA administrátora.
17. Odlíšné vnímanie toho, čo má obsahovať metodika pre agilné riadenie softvérového vývoja.
18. Nevydaril sa nám pokus s použitím papierikov počas plánovania šprintu.
19. vysoká technické požiadavky pre prácu s Angluarom spôsobujú neefektivitu pri vývoji na Windows platforme.
20. Na začiatku šprintu bol problém s reakciou na merge requesty.
21. Predmet TP nevnímame všetci rovnocenné.
22. Úprava cudzieho kódu podliehajúcemu aktívnemu vývoju autora.
23. Nemáme dobré premyslený model použitia vetiev.

Možné riešenia:

- Mohli by sme skúsiť zaviesť nový typ tímového stretnutia, aby sme mali viac času osobne rozobrať riešenie úloh [k bodom 3, 6, 10].
- Nemali by sme riešiť TP po polnoci [k bodu 1].
- Mohli by vyskúšať párové programovanie [k bodom 6, 22].
- Mali by sme dohodnúť si deadline na časti tasku [k bodu 2].
- Všetci si máme vyskúšať úlohu scrum mastera [všeobecne].
- Ľudia zodpovední za úlohy v rámci jedného používateľského príbehu budú priebežne spolupracovať na dokumentovaní postupu riešenia [k bodom 6, 10, 22].
- Prioritizácia backlogu [všeobecne].
- Rozloženie práce na úlohách podľa *definition of ready* a *definition of done* [k bodom 2, 8, 12].
- Scrum master by mal viac komunikovať s členmi tímu, kontrolovať pokrok pri riešení úloh a pomáhať s odstraňovaním prekážok pri práci na nich, s tým, že by dostával menej implementačných úloh [k bodom 2, 3].
- Zlepšiť komunikáciu medzi ľuďmi (frontend, backend) [k bodom 6, 22].
- Je potrebné zjednotiť sa a nastaviť rovnaké priority [všeobecne].
- Je potrebné zaviesť development vetvu [k bodu 23].
- Je potrebné nahlasovať issues a nebáť sa o to, že to nie je issue [všeobecne].

- Zmeniť charakter úloh pre ďalší šprint – viac riešiť funkčnosť, menej integráciu [všeobecne].
- Treba vyriešiť frontend testy - najprv treba spustiť v Docker na serveri [všeobecne].

4.4 Retrospektíva k šprintu 4 (28.11.2017)

Plusy:

- Zlepšuje sa komunikácia, a zlepšuje sa prehľad o tom, čo robia ostatní.
- Našli sme hranicu koľko zvládneme.
- Nenecháme problém vyhniť.
- Piškóty sú dobré, noste ich aj naďalej.
- Zlepšuje sa spolupráca.
- Vieme si navoliť správne množstvo práce.
- Dodržiavame statusy a ostatné vlastne zadefinované rituály.

Negatíva:

1. Málo implementujeme.
2. Sme málo kritickí. Kvôli nedostatku kritiky ide bokom implementácia.
3. Nevieme si zadefinovať smer ktorým máme ďalej ísť.
4. Príliš sa bojíme MTS, príliš veľa sa babreme s dokumentáciou.
5. Generujeme príliš veľa dokumentácie.
6. Máme roztrieštenú funkčnosť z user stories cez rôzne šprinty.
7. Zbytočne sa diskutujeme na stretnutiach a potom nestíhame na prednášku.
8. Nemali by sme cez prednášku tímový projekt.
9. Chýba dokumentácia k Jenkinsu.
10. Veľa času sa mýňa na dokumentáciu, testy a integráciu.
11. Chýbajú popisy úloh.

Riešenia:

- Navrhli by sme si funkčnosť viac času dopredu a potom budeme implementovať. Zaviedlo by sa pravidlo o tom, koľko času zo šprintu budeme implementovať. Napr. Budeme neimplementovať max. 40% šprintu [k bodom 1, 2].
- Budeme si dávať viac spätnej väzby. Na Lukáša treba byť horší. Chýba mu kritika na jeho osobu [k bodu 2].
- Na budúcom stretnutí bude namiesto plánovania retrospektíva [k bodu 2].
- Používať gesto, keď bude niekto otravovať ohľadom tímového projektu [k bodu 8].
- Treba vypracovať dokumentáciu k niektorým nástrojom ako napr. Jenkins [k bodu 9].

18.11.2017 budeme mať stretnutie k plánovaniu šprintom v letnom semestri.

4.5 Retrospektíva k šprintu 5 (7.12.2017)

Plusy :

- Podarilo sa nám dokončiť všetky tasky.
- UI testy
- Príjemná komunikácia.

- Pohli sme sa v UX ďalej.
- Dobré plánovanie.
- Prestali sme opravovať problémy v noci.
- Kratší šprint.
- Dobré pridelovanie story pointov.
- Ľudia sa obetovali a riešili veci, ktoré nie sú pre nich zaujímavé.

Negatíva:

1. Vznikajú konflikty v tíme.
2. Málo implementovania.
3. Problémy z normálnou diskusiou a ustúpením.
4. Málo sa hádame.
5. Začíname riešiť nové veci, kým staré nie sú rozobraté.
6. Nepresúvanie vecí v JIRE.
7. Mali by sme si viac riadiť diskusiu.
8. Práca na veciach, s ktorými nebol stotožnení.
9. Veľa taskov, ktoré bolo potrebné vyriešiť na začiatku stretnutia na jednej osobe.
10. Neriešili sme veci priamou komunikáciou.
11. Strach experimentovať a testovať nápady.
12. Neriešime veci verejne.
13. Byť k sebe viac kritický.
14. Nový druh tasku, ktorý sme neodhadli.

Možné riešenia:

- iný spôsob zapracovania a robenia komentárov - napr. komentár si môže zrušiť len autor [k bodu 1, 3, 7, 8, 10, 12].
- slušná komunikácia [k bodu 1, 3, 7, 8, 10, 12].

Záver

Po uplynutí ďalších dvoch šprintov sa situácia so Scrumom výrazne zlepšila. Do riadenia sme zapracovali návrhy na zlepšenia z posledných retrospektív. Hlavnou zmenou bol scrum poker, kde sme story pointy pridelovali na úrovni používateľských príbehov, úlohy sme si potom rozdelili. V týchto šprintoch sme zlepšili naše logovanie času čo pomohlo pri monitorovaní vyťaženia členov tímu.

Zlepšenia sa prejavili aj pri práci s Jirou, kde sme zjednotili deň zapisovania používateľských príbehov. Taktiež sme obmedzili množstvo práce na šprint, aby sa to dalo stíhať. Toto viedlo k úspešnému zvládnutiu šprintu 4 a 5 bez značného presúvanie úloh do ďalších šprintov. Všetky retrospektívy sa uskutočnili na extra stretnutí, kde mal každý dostatok priestoru vyjadriť svoje návrhy a dotazy.

V ďalších šprintoch sa plánujeme viac sústrediť na implementáciu funkcionality, keďže doposiaľ sme sa orientovali hlavne na architektúru systému a riadenie. V Jire presunieme začiatok šprintu na deň keď už budú príbehy v šprinte, aby sme tak upravili burndown chart. Taktiež chceme popracovať na vyberaní úloh, keďže väčšinou tento proces prebieha na stretnutiach a zaberá podstatne veľa času.

Príloha A Export evidencie úloh

Šprint 1 - Ananás

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Nasadenie JIRA	GD-15	Task	Done	Andrej Vitek	Nasadenie kolaboračného systému JIRA na pridelený server.		3
Požiadať o navýšenie RAM na tímovom serveri	GD-16	Task	Done	Andrej Vitek	Požiadať P. Lacka o zvýšenie parametrov servera.		0.5
Metodika - práca s GitLab	GD-32	Task	Done	Andrej Vitek	Vytvoríť metodiku ku práci s GitLabom * vetvy a ich názvy * komentáre ku commit-u * merge request a komentáre		2
Návrh grafických rozhraní	GD-3	Task	Done	Elena Stefancova	2 TP Vytvoríť prvotné mockupy pre rozhrania + vygenerovať css.		2
Frontend - vytvorenie hlasovacieho formulára	GD-9	Task	In progress	Elena Stefancova	Prevedenie formulára na vytvorenie nového hlasovania. Jeho parametre: * Názov * Možnosti hlasovania v textovom bloku pod sebou * API volanie po potvrdení * Vygenerovanie URL na admin a hlasovacie rozhrania	Vytvorenie jednoduchého hlasovania	3
Vytvorenie mockupov pre aspoň 5 ďalších stratégií	GD-14	Task	In progress	Elena Stefancova	3 SP Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.		3
Prieskum ohľadom stratégií hlasovania	GD-13	Task	In progress	Lukas Babula	Analýza stratégií hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.		3
Inštalácia a konfigurácia Jenkins	GD-35	Task	Done	Lukas Babula	Nainštalovanie Jenkins na server. Nastavenie a zabezpečenie Jenkins. Nastavenie Gitlab notifikácie pre Jenkins. Vytvorenie Jenkins projektu. Nastavenie SCM v projekte. Rozšírenie Slacku o Jenkins agenta. Integrácia Slack kanála do Jenkins.		8
Definícia POST/GET	GD-4	Task	Done	Michal Dolnak	Definovanie rozhrania a procesu tvorby rozhrania medzi frontendom a backendom.		2
Frontend - infraštruktúra	GD-8	Task	Done	Michal Dolnak	Vytvorenie nového projektu a nastavenie technológie. Pridanie prekladov, volania backendu. Nastavenie ciest v aplikácii.		3

Export pre šprint 1 - Ananás (časť 1).

Frontend - zaučenie, zdokumentovanie	GD-11	Task	Done	Michal Dolnak	Podpora ostatných členov tímu pri svojich úlohách. Predanie svojich vedomostí o technológií.		2
Metodika - dokumentovanie frontend kódu	GD-31	Task	Done	Michal Dolnak	2 SP		2
Implementácia backend v Django REST	GD-6	Task	Done	Michal Hucko	Implementácia základného vytvárania udalostí spolu s funkcionalitou hlasovania. Bude sa dať vytvoríť hlasovanie spolu s množinou možností. Následne je možné anonymne hlasovať za jednu z nich.	Vytvorenie jednoduchého hlasovania	13
Prihlásenie sa do TP Cup	GD-24	Task	Done	Michal Hucko	Vytvorenie prihlášky na TP cup, ktorá bude zahŕňať všetky body zo stránky predmetu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.		5
Návrh DB schémy	GD-5	Task	Done	Miroslav Valcicak	Navrhnutie dátového modelu pre funkcionalitu a jeho vizualizácia pomocou UML diagramu.		1
Docker setup	GD-7	Task	Done	Miroslav Valcicak	Vytvorenie konfiguračných súborov pre docker-compose a jednotlivé moduly.		8
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-10	Task	In progress	Pavel Balashov	Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností. Prvky grafického rozhrania: * tabuľka * statický text pre možnosti * textové pole pre meno * hlasovacie tlačidlá pre možnosti * tlačidlo pre potvrdenie hlasovanie	Vytvorenie jednoduchého hlasovania	8
Migrovanie úloh z Asana do JIRA	GD-17	Task	Done	Pavel Balashov	Migrovanie úloh z Asana. Zahŕňa prepísanie úloh do JIRA a kontrolu ďalším členom tímu.		1
Vytvorenie tímového webu	GD-67	Task	Done	Lukas Babula	Vytvorenie tímovej stránky * rozloženie elementov * informácie o time a projekte * šprinty * dokumenty * kontakt		1

Export pre šprint 1 - Ananás (časť 2).

Vytvorenie tímového webu	GD-66	Task	Done	Andrej Vitek	Vytvorenie tímovej stránky * rozloženie elementov * informácie o time a projekte * šprinty * dokumenty * kontakt		1
--------------------------	-------	------	------	--------------	---	--	---

Export pre šprint 1 - Ananás (časť 3).

Šprint 2 - Banán

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Backend - Úprava voľby hlasovania	GD-53	Task	Done	Andrej Vítek	Backend pre zmenu hlasovania používateľa cez HTTP PUT metódu.	Úprava voľby akéhokoľvek používateľa	3
Backend - Ukončenie hlasovania	GD-69	Task	Done	Andrej Vítek	Umožniť ukončenie hlasovania cez backend * úprava modelu hlasovania * vytvorenie url pre administráciu * zabezpečiť, že sa nedá hlasovať na uzavretom hlasovaní	Ukončenie hlasovania	3
Frontend - vytvorenie hlasovacieho formulára	GD-70	Task	In progress	Elena Stefancova	* API volania pre prerušenie hlasovania * Nová admin obrazovka (názov hlasovania, jeho prerušenie) * Testy	Vytvorenie jednoduchého hlasovania	3
Návrh grafických rozhraní	GD-3	Task	Done	Elena Stefancova	2 TP		
Vytvorenie mockupov pre aspoň 5 ďalších stratégií	GD-14	Task	In progress	Elena Stefancova	Vytvoriť prvotné mockupy pre rozhrania + vygenerovať css. 3 SP		2
Nasadzovanie projektu cez Jenkins	GD-48	Task	In progress	Lukas Babula	Vytvorenie mockupov obrazoviek ku hlasovaniu pomocou rôznych stratégií.		3
Prieskum ohľadom stratégie hlasovania	GD-13	Task	In progress	Lukas Babula	Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počítavania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.		13
					Analýza stratégie hlasovania a vytvorenie dokumentu s opisom zaujímavých stratégií.		3

Export pre šprint 2 - Banán (časť 1).

Metodika - testovanie frontendu	GD-29	Task	Done	Michal Dolnak	Vytvorenie metodiky pre testovanie frontendu.		2
Frontend - Vytvorenie prihlasovania	GD-58	Task	In progress	Michal Dolnak	Vytvorenie prihlasovania cez Google Oath: - vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu - odhlásenie používateľa - udržiavanie stavu prihlásenia na strane používateľa - vytvorenie sekcie prístupnej len pre prihláseného používateľa	Vytvorenie prihlasovania	8
Metodika - testovanie backendu	GD-26	Task	Done	Michal Hucko	V tejto metodike pokryjeme všetky detaily spojené s testovaním backendu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.		2
Metodika - dokumentovanie backend kódu	GD-30	Task	Done	Michal Hucko	V tejto metodike pokryjeme všetky detaily spojené s dokumentáciou backendu. Táto úloha zahŕňa aj zapracovanie pripomienok po iteráciách jednotlivými členmi.		2
Nastavenie Apache na serveri	GD-49	Task	In progress	Michal Hucko	Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.		8
Vyjednať navýšenie RAM + certifikát	GD-50	Task	Done	Michal Hucko	Napísať Peťovi Lackovi o zvýšení RAMky na servery a zistiť či je možné získať SSL certifikát pre projekt.		1
Dokonfigurovať Docker	GD-51	Task	Done	Michal Hucko	Zodpovedná osoba dokonfiguruje docker na serveri. Tento krok predchádza samotnému spusteniu dockeru na serveri.		3
Merge do master	GD-47	Task	Done	Michal Hucko	Úloha zahŕňa zodpovednosť nad tým aby sa stihli spojiť kódy frontendu a backendu. V prípade akýchkoľvek problémov zodpovedná osoba poskytne pomoc.		2

Export pre šprint 2 - Banán (časť 2).

Metodika - vývoj a integrácia	GD-34	Task	In progress	Miroslav Valčičak	Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.		3
Backend - Vytvorenie prihlasovania	GD-57	Task	In progress	Miroslav Valčičak	Vytvorenie prihlasovania cez Google * vytvorenie používateľa cez Google účet (OAuth2) * pridanie nášho hostname do overených hostanemov (Google) * vytvorenie tajných kľúčov (Google) * vygenerovanie SSL certifikátu * nakonfigurovanie Apache s certifikátom * nakonfigurovanie nginx * vytvorenie autentifikačného modulu na backende	Vytvorenie prihlasovania	13
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-54	Task	In progress	Pavel Balashov	Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutie voľby do stavu upravovania a uloženie zmien.	Vytvorenie jednoduchého hlasovania	3

Export pre šprint 2 - Banán (časť 3).

Šprint 3 - Citrón

Summary	Issue key	Issue Type	Status	Assignee	Description	Epic name	SP
Backend - Skript testovacej databázy	GD-87	Task	Done	Andrej Vitek	Vytvorenie prikazov na vytvorenie testovacej databázy pre frontend s možným zálohovaním a obnovením databázy.		8
Vytvoriť šablónu pre metodiky	GD-76	Task	Done	Elena Stefančova	Vytvorenie latex šablóny pre metodiky * Titulná strana * Jazykové nastavenia * Rozloženia a formátovanie		3
Návrh UX	GD-88	Task	Done	Elena Stefančova	* Návrh farebnej schémy * Pozadie pre aplikáciu * Bez rozloženia elementov * Branding		13
Frontend - vytvorenie hlasovacieho formulára	GD-70	Task	Done	Elena Stefančova	* API volania pre prerušenie hlasovania * Nová admin obrazovka (názov hlasovania, jeho prerušenie) * Testy	Vytvorenie jednoduchého hlasovania	3
Nasadzovanie projektu cez Jenkins	GD-48	Task	In progress	Lukas Babula	Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počúvania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.		13
Frontend - Voľba hlasovacej stratégie	GD-92	Task	Done	Michal Dolnak	Používateľ si po vybratí názvu hlasovania a zadania možnosti vyberie, ktorú z dostupných hlasovacích stratégií si chce vybrať. * Pridanie kroku zvolenia stratégie do procesu vytvárania hlasovania * Automatická zmena popisu hlasovacej stratégie.	Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.	8
Frontend - Vytvorenie prihlasovania	GD-58	Task	Done	Michal Dolnak	Vytvorenie prihlasovania cez Google Oath: - vytvorenie prihlasovania pomocou kliknutia v hornom pravom rohu - odhlásenie používateľa - udržiavanie stavu prihlásenia na strane používateľa - vytvorenie sekcie prístupnej len pre prihláseného používateľa	Vytvorenie prihlasovania	8
Backend - Voľba hlasovacej stratégie	GD-91	Task	Done	Michal Hucko	Doplní sa funkcionálna pri vytváraní hlasovania, ktorá používateľovi umožní vybrať konkrétnu hlasovacu stratégiu. V tomto prípade ide o nastavenie backendovej časti.	Možnosť vybrať hlasovaciu stratégiu pri vytváraní hlasovania.	8
Nastavenie Apache na serveri	GD-49	Task	Done	Michal Hucko	Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.		8
Metodika - vývoj a integrácia	GD-34	Task	Done	Miroslav Valčík	Opis postupu pri vyvíjaní samostatnej funkcionality od návrhu až po jej nasadenie do produkcie.		3
Backend - Vytvorenie prihlasovania	GD-57	Task	Done	Miroslav Valčík	Vytvorenie prihlasovania cez Google * vytvorenie používateľa cez Google účet (OAuth2) * pridanie nášho hostname do overených hostnameov (Google) * vytvorenie tajných kľúčov (Google) * vygenerovanie SSL certifikátu * nakonfigurovanie Apache s certifikátom * nakonfigurovanie nginx * vytvorenie autentifikačného modulu na backende	Vytvorenie prihlasovania	13

Export pre šprint 3 - Citrón (časť 1).

Metodika - agilný vývoj	GD-33	Task	Done	Pavel Balashov	Vytvorenie metodiky pre riadenie vývoja softveru založeného na metodológii Scrum. * roli a zodpovednosti * priebeh stretnutí * práca s JIRA		3
Vytvoriť metodiku pre metodiky	GD-77	Task	Done	Pavel Balashov	Vytvorenie metodiky pre zjednotenie vzťahu a obsahu ostatných metodík. Časť: * Definícia a účel * Štýl * Obsah		2
Frontend - hlasovacia tabuľka a zobrazenie výsledkov	GD-54	Task	Done	Pavel Balashov	Implementácia frontend pre zobrazenie hlasovacej tabuľky a možnosti zahlasovať za jednu z možností. Prvky grafického rozhrania: * tabuľka * statický text pre možnosti * textové pole pre meno * hlasovacie tlačidlá pre možnosti * tlačidlo pre potvrdenie hlasovanie	Vytvorenie jednoduchého hlasovania	3
Frontend - úprava voľby hlasovania	GD-54	Task	Done	Pavel Balashov	Implementácia frontend pre úpravu voľby akéhokoľvek používateľa v hlasovacej tabuľke. Zahŕňa možnosť prepnutie voľby do stavu upravovania a uloženie zmien.	Úprava voľby akéhokoľvek používateľa	3

Export pre šprint 3 - Citrón (časť 2).

Šprint 4 - Datle

Summary	Issue id	Parent id	Issue Type	Status	Assignee	Description	Custom field (Story Points)
Odozdvanie riadenia - ostatné	10104		Story	Done	Michal Hucko	Pripravenie dokumentácie riadenia na odozdvanie.	20
Odozdvanie inžnierskeho diela	10110		Story	Done	Michal Hucko	Pripravenie dokumentácie inžnierskeho diela na odozdvanie.	20
Odozdvanie riadenia - metodiky	10100		Story	Done	Michal Hucko	Pripravenie metodík na odozdvanie a vloženie do dokumentácie riadenia.	13
Doplnenie úloh do JIRY	10123		Task	Done	Andrej Vitek	Pridanie nových úloh do JIRY.	1
Úprava loga	10117	10116	Sub-task	Done	Miroslav Valcicak	Zapracovanie pripomienok k logu zo stretnutia.	
Nasadiť Apache	10048		Task	Done	Michal Hucko	Nastavenie Apache na serveri aby bol schopný pracovať so statickou stránkou a zároveň aby fungoval ako reverse proxy pre docker s aplikáciou.	8
Vytvorenie wireframov	10202	10116	Sub-task	Done	Elena Stefancova	Vytvorenie wireframov * wireframe 1 ** pred formulárom pre strategiu bude jej popis ** carousel ** obrazok + popis + kedy sa hodi	
Nasadzovanie projektu cez Jenkins	10047		Task	Done	Lukas Babula	Integrácia Jenkins a Docker - konfigurácia Docker socketu, nastavenie práv pre Jenkins, nastavenie komunikácie z hľadiska Jenkins rozhrania. Nastavenie Jenkins počívania na konkrétnu vetvu. Nastavenie krokov buildovania a testovania. Vytvorenie a nastavenie vetvy pre nasadzovanú verziu. Nasadenie a spustenie verzie na serveri.	13
Konfiguračné súbory pre Jenkins	10119		Story	Done	Lukas Babula		20
Rozšírenie Jenkins konfigurácie pre testovanie	10122	10119	Sub-task	Done	Lukas Babula	Konfigurovať Jenkins pre testovanie na novo-nastavenom testovacom prostredí.	
Prispôbenie Protractor testov	10200	10119	Sub-task	Done	Pavel Balashov	Úprava frontend testov pre použitie sériovo za sebou na testovacej databáze.	

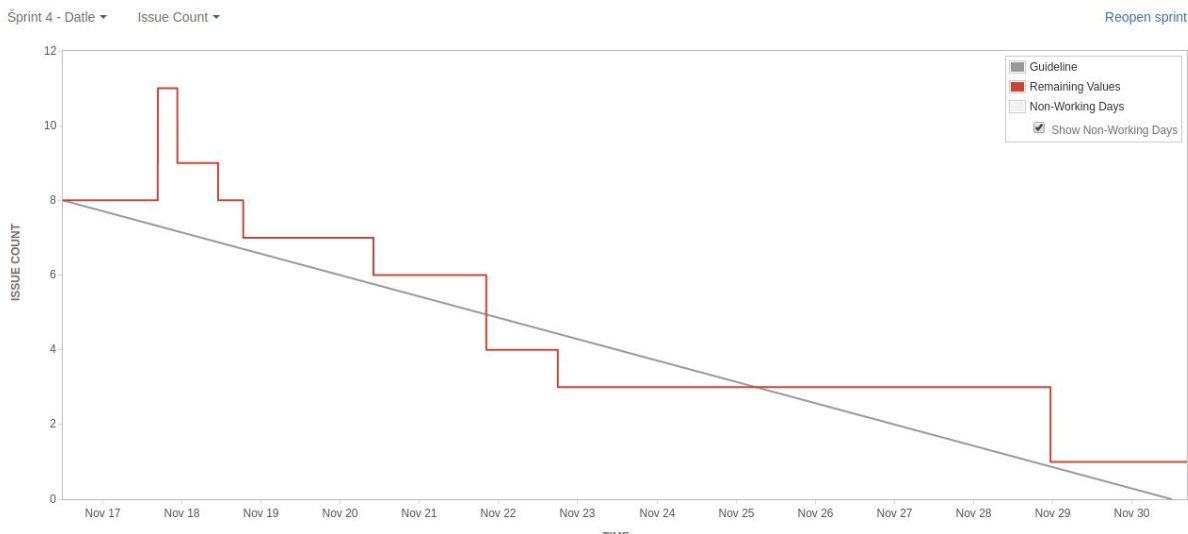
Export pre šprint 4 - Datle (časť 1).

Protractor v Dockri	10201	10119	Sub-task	Done	Michal Dolnak	Nasadenie nástroja Protractor na Docker. Presunutie testov z kontajnera Frontend do kontajnera Protector. Konfigurácia nastavení pre Develop testovanie na localhoste a testovanie na serveri, ktoré sa doplytuje na stránku /test/.	
Konfigurácia Apache a Dockerfiles (develop, test, deploy)	10120	10119	Sub-task	Done	Michal Hucko	Úprava Apache konfigurácie pre presmerovávanie stránok /test/ pre test a /app/ pre produkciu. Dockerfile pre Develop (mapovanie portov pre vývoj na localhoste). Dockerfile pre Test (mapovanie portov pre testovanie, rozdielne od produkcie, použitie testovacej databázy). Decokerfile pre Deploy (mapovanie portov produkcie, odstránenie všetkých zbytočnosti z konfiguračného súboru).	
Konfigurácia Django nastavení a testovacej databázy	10121	10119	Sub-task	Done	Miroslav Valcicak	Konfigurácia Django nastavení pre vývoj Angularu na localhoste. Konfigurácia Django nastavení pre použitie testovacej databázy v testovacom prostredí. Nastavenie testovacej databázy na serveri.	
Modul autentifikácie	10113	10110	Sub-task	Done	Miroslav Valcicak	Pridať analýzu, návrh, implementáciu a testovanie pre modul.	
nasadenie - vytvor. prihlasovania	10084	10048	Sub-task	Done	Miroslav Valcicak	Po implementácii funkcionality sa zabezpečí úspešné nasadenie na serveri.	2
Vytvorenie dokumentu k mentoringu	10124		Task	Done	Elena Stefancova	Vytvorenie dokumentu, ktorý nás prezentuje, k mentoringu TP Cup. Dokument má časti * agile * biznis	1
Modul hlasovania	10112	10110	Sub-task	Done	Michal Dolnak	Pridať analýzu, návrh, implementáciu a testovanie pre modul.	
Celkový pohľad - frontend	10114	10110	Sub-task	Done	Michal Dolnak	Rozpísať časť celkového pohľadu na frontend. Popríklad pridať diagram.	
Rozdelenie aplikácie manažmentov	10105	10104	Sub-task	Done	Pavel Balashov	Rozdeliť aplikácie manažmentov podľa zápisnice zo 6. stretnutia. Prípadne rozšírenie dokumentu o ďalšie manažmenty.	
Export evidencie úloh	10109	10104	Sub-task	Done	Pavel Balashov	Burndown chart pre jednotlivé šprinty a export úloh po jednotlivých týždňoch.	
Spojenie metodík	10101	10100	Sub-task	Done	Elena Stefancova	Vloženie metodík do vytvorenej šablóny a pridanie do príloh v dokumentácii riadenia.	

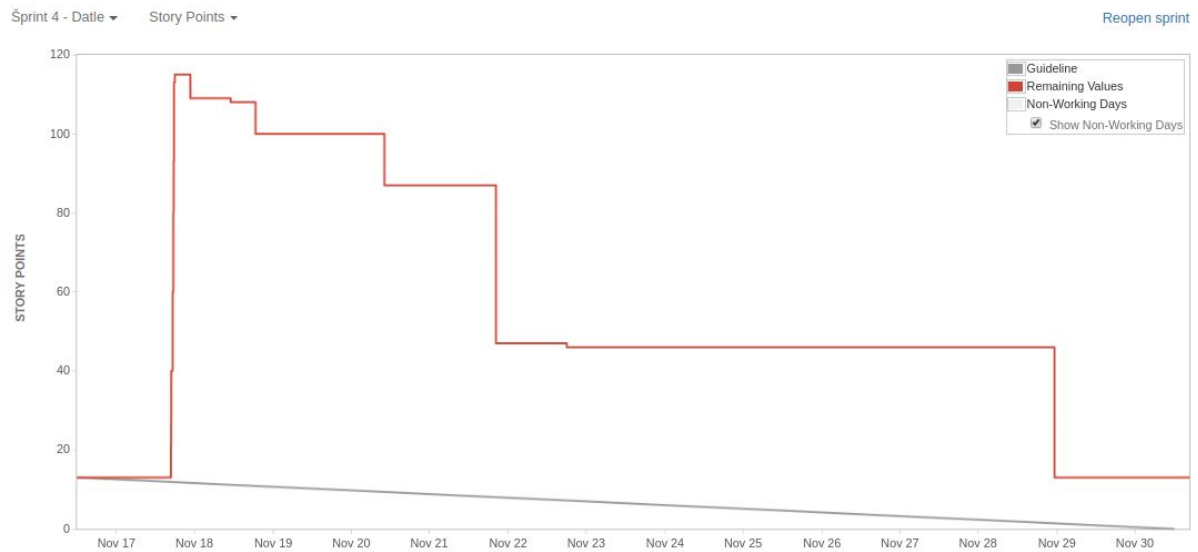
Export pre šprint 4 - Datle (časť 2).

Popis úloh	10107	10104	Sub-task	Done	Michal Hucko	Pre každý task(každý zodpovedá za svoje) dopísať popis(ak chýba) a zdôvodnenie v prípade, keď úloha nebola splnená.	
Doplniť user story k taskom	10106	10104	Sub-task	Done	Michal Hucko	Pre jednotlivé tasky(každý má zodpovednosť za svoje) doplniť k akému story patrí.	
Zhodnotenie a vízia	10115	10110	Sub-task	Done	Andrej Vítek	Pridanie zhodnotenia a vízie do dokumentácie.	
Modul hlasovacích stratégií	10111	10110	Sub-task	Done	Lukas Babula	Pridať analýzu, návrh, implementáciu a testovanie pre modul.	
Doplniť retrospektivu 3	10108	10104	Sub-task	Done	Pavel Balashov	Skontrolovať a pridať retrospektivu 3 do dokumentácie riadenia.	
Nasadenie - ukončen. hlasovania	10085	10048	Sub-task	Done	Michal Hucko	Po implementácii funkcionality sa zabezpečí úspešné nasadenie na serveri.	1
Nasadenie - uprava voľby	10083	10048	Sub-task	Done	Michal Hucko	Po implementácii funkcionality sa zabezpečí úspešné nasadenie na serveri.	1
Spojenie metodík dokumentácie	10102	10100	Sub-task	Done	Michal Hucko	Spojenie metodík dokumentácie frontendu a backendu do jednej metodiky.	
Spojenie metodík testovania	10103	10100	Sub-task	Done	Michal Dolnak	Spojenie metodík testovania frontendu a backendu do jednej metodiky.	
Úprava dizajnu	10118	10116	Sub-task	To Do	Andrej Vítek	Zapracovanie pripomienok: * zvýraznenie tlačidiel * preloženie do angličtiny * povinné texty s hviezdíčkou * odstránenie okraju * zjednotenie fontu loga a písma * výzor chybovej hlášky	
frontend	10069		Task	Done	Elena Stefancova	* API volania pre prerušenie hlasovania * Nová admin obrazovka (názov hlasovania, jeho prerušenie) * Testy	3
Vytvoriť šablónu pre metodiky	10075		Task	Done	Elena Stefancova	Vytvorenie latex šablóny pre metodiky * Titulná strana * Jazykové nastavenia * Rozloženia a formátovanie	3
Úprava dizajnu používateľského rozhrania	10116		Story	To Do	Andrej Vítek	Zapracovanie pripomienok zo stretnutia.	13

Export pre šprint 4 - Datle (časť 3).



Burndown chart pre šprint 4 - Datle (počet dokončených úloh).



Burndown chart pre šprint 4 - Datle (počet story points pre dokončené úlohy)

Naša velocity pre 4. šprint (2 týždne) bola:

- počet dokončených úloh: 10
- počet SP pre dokončené úlohy: 115

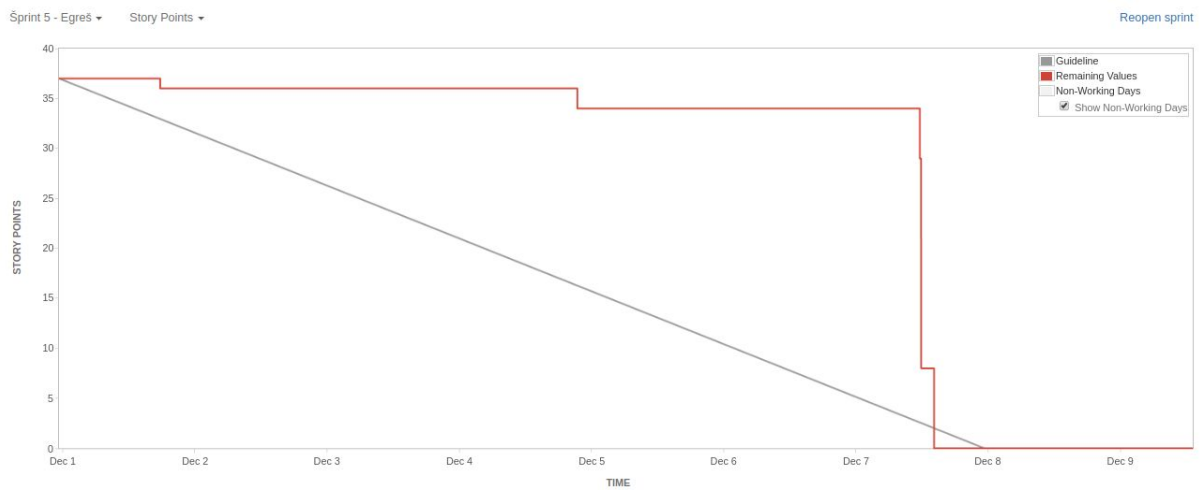
Šprint 5 - Egreš

Summary	Issue id	Parent id	Issue Type	Status	Assignee	Description	Custom field (Story Points)
Testovanie rozmiestnenia	10216		Story	Done	Elena Stefancova		8
Natáčanie a testovanie	10220	10216	Sub-task	Done	Lukas Babula	Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.	
Natáčanie a testovanie	10221	10216	Sub-task	Done	Andrej Vitek	Otestovanie na 2 ľudoch.	
Úprava metodiky testovania	10207	10203	Sub-task	Done	Lukas Babula	Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.	
Úprava metodiky verzii	10226	10203	Sub-task	Done	Lukas Babula	Úprava metodiky podľa vykonaných zmien: * pridať protractor * pridať Jenkins link * pridať opis špecifických operácií	
Dokumentácia riadenia	10203		Story	Done	Michal Hucko	Pridanie a spracovanie pripomienok.	
Pridať popisy k úlohám	10209	10203	Sub-task	Done	Pavel Balashov		13
Prečítanie a aktualizácia	10225	10203	Sub-task	Done	Pavel Balashov	Pridať popisy k úlohám, poprípade napísať, prečo úloha nebola splnená.	
Dokument inžinierskeho diela	10210		Story	Done	Michal Hucko	Prečítanie a úprava dokumentu podľa zmien v nových šprintoch.	
Prečítanie a aktualizácia	10213	10210	Sub-task	Done	Pavel Balashov		8
Testovanie farieb a opisov	10215		Story	Done		Prečítanie a úprava dokumentu podľa zmien v nových šprintoch.	
Natáčanie a testovanie	10218	10215	Sub-task	Done	Miroslav Valcicak		5
Natáčanie a testovanie	10217	10215	Sub-task	Done	Pavel Balashov	Otestovanie farieb a opisu aplikácie pomocou 30 sekundového testu a zhodnotenie.	
Úprava metodiku vývoja a integrácií	10208	10203	Sub-task	Done	Miroslav Valcicak	Úprava podľa nových zmien	

Export pre šprint 5 - Datle (časť 1).

Export evidencie úloh z JIRY	10205	10203	Sub-task	Done	Pavel Balashov	Export úloh z JIRY z nových šprintov.	
Export burndown chartu z JIRY	10204	10203	Sub-task	Done	Pavel Balashov	Vytvorenie obrázku burndown chartu a vloženie do dokumentu riadenia.	
Natáčanie a testovanie	10219	10216	Sub-task	Done	Elena Stefancova	Otestovanie rozloženia prvkov aplikácie pomocou wireframov a vytvorenie zhodnotenia.	
Úprava diagramov a obrázkov	10211	10210	Sub-task	Done	Andrej Vitek	Príprava podkladov, scenára, konečná revízia.	
Pridanie úloh do JIRY	10214		Task	Done	Andrej Vitek	Zmena a pridanie diagramov a obrázkov podľa zmien v nových šprintoch.	
Dokumentácia zo Swaggera	10212	10210	Sub-task	Done	Michal Hucko	Pridanie nových úloh do JIRY.	1
Zabezpečenie serverových aplikácií	10222		Story	Done		Analýza či opraviť CSS na stránke alebo spraviť export PDF.	
Zabezpečenie JIRY	10223	10222	Sub-task	Done	Michal Dolnak		2
Zabezpečenie Jenkins	10224	10222	Sub-task	Done	Michal Hucko	Úprava apache súborov a restart JIRY.	
Revízia metodík	10227	10203	Sub-task	Done	Elena Stefancova	Úprava apache súborov a restart Jenkins.	
Prepis úloh do dokumentu	10206	10203	Sub-task	Done	Elena Stefancova	Finálna revízia metodík ako celku.	
						Vloženie úloh z nových šprintov do dokumentu riadenia.	

Export pre šprint 5 - Datle (časť 2).



Burndown chart pre šprint 5 - Datle (počet story points pre dokončené úlohy)

Naša velocity pre 5. šprint (1 týždeň) bola 37 SP.

Príloha B Metodiky

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Tímový projekt
Group de'Cider
Metodiky

Tím: číslo 1, sElectors

Vedúci projektu: Ing. Ondrej Kaššák, PhD.

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Zodpovedná osoba: Bc. Elena Štefancová

Posledná aktualizácia: 03.12.2017

Obsah

Zoznam pojmov a skratiek	3
1 PÍSANIE METODÍK	5
1.1 Role	5
1.2 Definícia, účel a platnosť	5
1.3 Štýl	5
1.4 Štruktúra dokumentu	6
2 AGILNÉ RIADENIE SOFTVÉROVÉHO VÝVOJA	7
2.1 Role	7
2.2 Udalosti	8
2.2.1 Šprint	8
2.2.2 Hlásenie statusu	8
2.2.3 Tímové stretnutia	8
2.2.4 Plánovanie nového šprintu	8
2.2.5 Stretnutie v polovici šprintu	9
2.2.6 Vyhodnotenie šprintu	9
2.2.7 Retrospektíva šprintu	9
2.3 Artefakty	9
2.3.1 Produktový backlog	9
2.3.2 Backlog pre šprint	10
2.3.3 Položka backlogu	10
2.3.4 Cieľ šprintu	10
3 PRÁCA S VERZIAMI	11
3.1 Role	11
3.2 Vetvy	11
3.3 Commit-y	12
3.4 Merge request-y	12
4 VÝVOJ A INTEGRÁCIA	14
4.1 Role	14
4.2 Opis procesov	14
4.3 Záver	16
5 DOKUMENTÁCIA	17
5.1 Role	17
5.2 Predpoklady	17
5.2.1 Spustenie frontendu	17
5.2.2 Spustenie backendu	18
5.3 Opis procesu	19
5.3.1 Frontend	19
5.3.2 Backend	19

5.4	Záver	20
6	TESTOVANIE	21
6.1	Role	21
6.2	Definition of done z pohľadu testovania	21
6.3	Predpoklady	21
6.3.1	Frontend	21
6.3.2	Načítanie testovacích údajov cez docker	22
6.3.3	Backend	22
6.4	Opis procesu	22
6.4.1	Frontend	22
6.4.2	Backend	23
6.5	Automatizované testovanie	24
6.6	Záver	25

Zoznam pojmov a skratiek

backend serverová časť webovej aplikácie

backlog usporiadaný zoznam úloh, ktoré ešte nie sú vyriešené

camel case spôsob písania názvov bez medzier, kde prvý výraz začína malým písmenom každé nové slovo veľkým (napr. camelCaseWriting)

code review prehliadka kódu, statická analýza kódu - proces, kedy zodpovedná osoba skontroluje kód, tento proces je súčasťou definition of done

commit vytvorenie novej verzie s vybranými zmenami

cors celým názvom ang. *cross-origin resource sharing*, mechanizmus umožňujúci zdieľanie zdrojov webovej stránky pre aplikáciu na inej doméne

definition of done zoznam bodov, ktoré je potrebné splniť na dokončenie úlohy

docker nástroj, ktorý sa používa pri nasadzovaní riešenia do produkcie¹

e2e (end to end) test, ktorý pokrýva všetky časti aplikácie

frontend vizuálna časť webovej aplikácie

kontinuálna integrácia spôsob vývoja, pri ktorom všetci vývojári zdieľajú svoj kód vo vzdialenom repozitári, kód sa automaticky zostaví a otestuje

kontinuálne nasadzovanie rozšírenie kontinuálnej integrácie o automatické nasadenie do produkčného prostredia po úspešnom otestovaní

merge request žiadosť o spojenie vetiev

mockup používateľské rozhranie slúžiace len na demonštráciu bez implementovania funkčnej logiky

ng angular klient

npm celým názvom ang. *node package manager*, správca balíčkov uzla

osx operačný systém pre počítače spoločnosti Apple

používateľsky príbeh (angl. *user story*) opis časti funkcionality aplikácie z pohľadu interakcie medzi používateľom a aplikáciou

REST celým názvom ang. *representational state transfer*, spôsob ako jednoducho vytvoriť, čítať, editovať alebo zmazať informácie zo serveru pomocou jednoduchých volaní

¹<https://www.docker.com/>

Scrum metodológia agilného vývoja softvéru

Selenium rámec na testovanie frontendu

snake case spôsob písania názvov bez medzier, kde prvý výraz začína malým písmenom každé nové slovo malým, slová sú rozdelené podtržníkmi (napr. snake_case_writing)

Swagger nástroj na dokumentovanie REST rozhraní²

tag tzv. značka, pomocné označenie, ktoré zlepšuje čitateľnosť

URL celým názvom ang. *uniform resource locator*, jednotný vyhľadávač prostriedku

view časť backendu, ktorá posiela dáta na frontend v určenom formáte

wireframe bezfarebný náčrt štruktúry, obsahu a rozmiestnenia prvkov grafického používateľského rozhrania

²<https://swagger.io/>

Kapitola 1

PÍSANIE METODÍK

Zodpovedná osoba: Bc. Pavel Balashov

Posledná úprava: 02.12.2017, verzia 1.1

Tato metodika je určená pre členov tímu sElectors, ktorý pracuje na projekte Group de'Cider v rámci predmetu Tímový projekt.

Tato metodika bola vytvorená pre upresnenie toho, čo je metodika a na čo sa používa. Taktiež v tejto metodike sa opisuje štýl písania a štruktúra, ktoré musia dodržiavať metodiky vytvorené a používané členmi tímu sElectors v rámci projektu Group de'Cider.

1.1 Role

Autor metodiky - člen tímu, ktorý napísal danú metodiku. Iba autor metodiky má možnosť meniť a dopĺňať metodiku.

1.2 Definícia, účel a platnosť

Metodika je dokument ktorý zachytáva skúsenosti a znalosti jedného alebo viacerých členov tímu v rámci konkrétnej oblasti vývoja, riadenia vývoja alebo technológií. Je potrebná pre interné štandardizovanie procesov, ktoré môžu alebo musia byť vykonané v rámci projektu. Metodika slúži pre zaistenie kvality vykonania procesov, bez ohľadu na to, kto ho vykonáva, a je nápomocná najmä v prípade, že existujú alternatívne možnosti a postupy¹.

V rámci tímu sElectors, ktorý pracuje na projekte Group de'Cider, metodika nadobúda účinnosť po ukončení šprintu, v rámci ktorého bola schválená. Po nadobudnutí účinnosti je metodika záväzná pre všetkých členov tímu.

Všetky návrhy pre zmenu alebo doplnenie metodiky sa píšú do komentárov. Ak sa nejedná o opravu gramatickej chyby, tieto zmeny a doplnenia musia byť následne schválené tímom. Po schválení ich autor metodiky doplní do metodiky a zvýrazní žltou farbou, čo znamená, že ešte nie sú platné. Tieto zmeny a doplnenia sa stávajú platnými po ukončení šprintu, v rámci ktorého boli schválené. Po nadobudnutí platnosti autor odstráni zvýraznenie.

1.3 Štýl

Metodika je formálny technický dokument, ktorý má obsahovať:

- jednotný strohý technický presný jazyk
- jednoznačné, zrozumiteľné a explicitné formulácie

¹<http://www2.fkit.stuba.sk/bielik/courses/msi-slov/metodika/metodika2012-13.pdf>

Metodika by mala mať odporúčací alebo normatívny charakter. Pri písaní metodiky by sa mal autor úplne odosobniť, čo znamená:

- nepoužívať vlastné hodnotenia
- nepoužívať osobné zámená (ja, my, vy atď.)
- používať neurčitky a neosobné zámená

Metodika musí byť zrozumiteľná aj čitateľom s nižšou úrovňou znalostí než akou disponuje jej autor. Podľa možnosti, autor by mal uviesť príklad správneho a nesprávneho vykonávania konkrétneho procesu a pridať aj snímky obrazoviek.

Nakoniec, metodika nemá obsahovať analýzy problémov, názory a hodnotenia, avšak môže sa odkazovať na dokumenty, ktoré ich obsahujú².

1.4 Štruktúra dokumentu

1. titulná strana
2. obsah
3. zoznam pojmov a skratiek
4. nadväznosť na iné metodiky
5. úvod
 - (a) cieľová skupina - stanovuje pre koho je metodika určená
 - (b) ciele - vymedzenie cieľov metodiky
6. role a ich zodpovednosti
7. opis procesov
 - (a) opis postupu – opis všetkých podstatných povinných a podmienených krokov, ktoré je alebo môže byť potrebné vykonať
 - (b) vyhodnotenie kvality – ako zistiť, že opísaný postup bol správne dodržaný
 - (c) doplňujúce zdroje informácií – napr. odkazy na detailnejší opis riešenia problému na internete alebo dokumentáciu konkrétnej časti technológie
 - (d) hľadanie a odstránenie závad (ang. *troubleshooting*) – detaily toho, kedy môže nastať konkrétna záhada a ako sa dá vyriešiť
8. zdroje

²<http://www2.fkit.stuba.sk/bielik/courses/msi-slov/metodika/metodika2012-13.pdf>

Kapitola 2

AGILNÉ RIADENIE SOFTVÉROVÉHO VÝVOJA

Zodpovedná osoba: Bc. Pavel Balashov

Posledná úprava: 02.12.2017, verzia 2.0

Táto metodika je určená pre členov tímu sElectors, ktorý pracuje na projekte Group de'Cider.

V rámci tímu bolo potrebné sa zhodnúť na prístupe k riadeniu projektu založenom na metodológii agilného vývoja softvéru Scrum. Najmä vznikla potreba jednoznačne definovať role, udalosti, artefakty a pravidlá podľa uvedenej metodológie, ako aj rozšíriť ich o ďalšie detaily, ktoré sú špecifické pre tím sElectors a projekt Group de'Cider.

2.1 Role

Produktový vlastník -

- vízia a definícia produktu¹
- manažment položiek produktového backlogu
- prioritizácia položiek produktového backlogu
- akceptácia implementácie

Scrum master - rola scrum master je pridelená jednému z členov tímu. Pred začiatkom nového šprintu sa táto rola prideluje inému členovi.

- dodržiavanie Scrum a agilných princípov
- identifikácia a odstránenie problémov, ktoré ovplyvňujú výkon tímu
- sledovanie progresu vývoja
- organizácia a vedenie stretnutí

Tím -

- spôsob vývoja produktu
- tvorba plánu šprintu
- odhad náročnosti práce
- predvedie výsledkov šprintu
- zlepšenia procesu

¹<http://scrum.sk/o-agile-a-scrum/2-scrum-a-roly/>

2.2 Udalosti

Všetky udalosti sú časovo ohraničené.

2.2.1 Šprint

Šprint (angl. *sprint*) je udalosť, počas ktorej sa vyvíja ďalší plne funkčný a použiteľný inkrement funkcionality produktu. V rámci šprintu sa uskutočňujú ďalšie udalosti ako tímové stretnutia a hlásenia statusu.

Šprint má fixnú dĺžku 2 týždne. Pred šprintom prebehne plánovanie šprintu, po ukončení ktorého začne šprint. Počas šprintu:

- nemôže sa meniť cieľ šprintu
- nemôžu byť vykonané zmeny, ktoré ohrozia cieľ šprintu
- backlog pre šprint môže byť upravený po prediskutovaní s produktovým vlastníkom

Šprint končí vyhodnotením šprintu.

2.2.2 Hlásenie statusu

Hlásenie statusu je časový úsek, kedy každý člen tímu musí oznámiť pokrok na svojich úlohách a problémy, ktoré ho aktuálne brzdia. Je alternatívou pre denné scrumy (angl. *daily scrums*).

Hlásenie statusu sa koná minimálne 2 krát do týždňa - v utorok a sobotu. V stanovených dňoch je každý člen tímu povinný do 23:00 napísať do Slack kanálu #status:

- názvy úloh na ktorých pracuje
- vyjadrenie pokroku (0 % - 99 %, ready for review, done)
 - ak je pokrok uvedený v percentách a nie je 0 %, je potrebné do zátvorky krátko opísať čo ešte ostalo dorobiť
 - ak niektorá úloha nadväzuje na inú, je vhodné jej status doplniť o informáciu, na ukončenie ktorej úlohy pod jeho vedením sa čaká

2.2.3 Tímové stretnutia

Tímové stretnutia vyžadujú prítomnosť všetkých členov tímu a produktového vlastníka. Neúčast je povolená iba vo výnimočných situáciách a musí byť vopred oznámená všetkým členom tímu.

Trestom za nedochvilnosť je povinnosť priniesť chutnú odmenu pre ostatných členov tímu na ďalšom stretnutí.

Scrum master vytvorí súpis tém, ktoré sa budú preberať na tímovom stretnutí a zverejní ho ostatným členom tímu najneskôr deň pred stretnutím. Toto neplatí pre retrospektívu šprintu.

Počas stretnutia jeden člen tímu má na starosti vedenie zápisnice, ktorá obsahuje krátky opis priebehu stretnutia. Ďalší člen tímu má na starosti zapisovanie ďalších informácií, ktoré nepatria do zápisnice (napr. zapisovanie detailov jednotlivých úloh). Pre zvyšných členov tímu platí prísny zákaz používania elektroniky bez súhlasu ostatných členov tímu.

2.2.4 Plánovanie nového šprintu

Počas plánovania nového šprintu (angl. *sprint planning*) tím spoločnými silami rozvrhne prácu pre ďalší šprint. Toto stretnutie zahŕňa 3 podstatné činnosti:

- stanovenie cieľa šprintu
- výber úloh z produktového backlogu, ktoré budú vykonané počas nového šprintu
- ako bude prebiehať práca na vybraných úlohách

Pri výbere úloh do backlogu nového šprintu berie tím ohľad na: prioritu úloh, rýchlosť tímu a predchádzajúcu skúsenosť členov tímu. O počte úloh, ktoré budú vybrané do nového šprintu, rozhoduje tím.

Po výbere úloh do backlogu nového šprintu vytvorí tím plán práce. Tento plán bude obsahovať nadväznosť úloh a približný odhad kedy sa bude pracovať na ktorej úlohe. Pre všetky úlohy bude vytvorený detailný opis, ktorý musí obsahovať kroky vývoja a akceptačné kritériá.

Po vytvorení opisov tím vyhodnotí náročnosť každej položky backlogu pre nový šprint. Pre tento účel prebehne plánovací poker (angl. *planning poker*). Nakoniec každej úlohe bude pridelený člen tímu, ktorý bude za ňu zodpovedný.

Plánovanie nového šprintu prebieha každý druhý štvrtok o 12:00, spolu s vyhodnotením predchádzajúceho šprintu. Spolu tieto činnosti môžu trvať maximálne 3 hodiny.

2.2.5 Stretnutie v polovici šprintu

Stretnutie v polovici šprintu sa využíva pre:

- riešenie prekážok, ktorým čelia členovia tímu počas práce na položkách backlogu pre aktuálny šprint
- prediskutovanie úpravy backlogu pre aktuálny šprint s produktovým vlastníkom
- ďalšie činnosti, ktoré vyžadujú vstup od všetkých členov tímu a ktoré nie je možné alebo sa nedajú efektívne vykonať mimo stretnutia

Toto tímové stretnutie prebieha každý druhý štvrtok o 12:00 a môže trvať maximálne 3 hodiny.

2.2.6 Vyhodnotenie šprintu

Na začiatku vyhodnotenia šprintu (angl. *sprint review*) sa posúdi práca, ktorá bola vykonaná počas šprintu a prezentovanie funkčného inkrementu, ktorý mal vzniknúť na konci šprintu. Ak na konci šprintu ostali nesplnené niektoré úlohy, počas tohto stretnutia sa posúdi ich stav a potrebnosť ich úpravy.

Potom tím spolu s produktovým vlastníkom vyhodnotí stav produktového backlogu a dohodne sa na ďalšej práci a smerovaní projektu. Výsledkom tohto stretnutia by mal byť revidovaný produktový backlog, ktorý je pripravený pre plánovanie ďalšieho šprintu.

Vyhodnotenie šprintu prebieha každý druhý štvrtok o 12:00, spolu s plánovaním nového šprintu. Spolu tieto činnosti môžu trvať maximálne 3 hodiny.

2.2.7 Retrospektíva šprintu

Retrospektíva šprintu (angl. *sprint retrospective*) slúži pre vyhodnotenie pozitívnych a negatívnych momentov šprintu, ktorý bol práve ukončený, s ohľadom na ľudí, vzťahy, činnosti a nástroje. Na základe nich tím spoločne sformuluje možné úpravy procesu riadenia a vývoja projektu, ktoré sa bude snažiť vykonať v ďalších šprintoch. Ako osnova slúžia tri otázky, na ktoré odpovie každý z členov tímu:

- Čo sa mi v tomto šprinte páčilo?
- Čo sa mi v tomto šprinte nepáčilo?
- Aké zmeny by som navrhol pre vylepšenie budúcich šprintov?

Retrospektíva šprintu prebieha každý druhý štvrtok o 18:40, v rovnaký deň ako prebehlo vyhodnotenie šprintu. Toto stretnutie trvá maximálne 2 hodiny.

2.3 Artefakty

2.3.1 Produktový backlog

Produktový backlog (angl. *product backlog*) je usporiadaný zoznam všetkých úloh, ktoré musia byť vykonané. Je to zdroj všetkých požiadaviek a zmien, ktoré musia byť zahrnuté do produktu. Keďže požiadavky na produkt sa stále menia, produktový backlog sa môže meniť. Podľa potreby, úlohy v produktovom backlogu môžu byť rozbité na menšie úlohy.

2.3.2 Backlog pre šprint

Backlog pre šprint (angl. *sprint backlog*) pozostáva z položiek produktového backlogu, ktoré boli vybrané tímom pre aktuálny šprint, a plánu práce pre dodanie ďalšieho inkrementu a dosiahnutie cieľu šprintu. Informácie pre položky tohto backlogu sa postupne dopĺňajú počas šprintu.

2.3.3 Položka backlogu

Každá položka backlogu definuje úlohu, ktorá musí byť vykonaná.

Pre všetky úlohy sú definované podmienky, ktoré musia byť splnené pred tým ako úloha môže byť považovaná za hotovú (tzv. *definition of done*). Tieto podmienky sú vymenované vo forme krokov v časti 4.2.

Taktiež každá úloha môže mať špecifické akceptačné kritéria, ktoré musia byť splnené pred tým, ako sa úloha bude považovať za hotovú.

2.3.4 Cieľ šprintu

Cieľ šprintu (angl. *sprint goal*) je stanovený na to, aby členovia tímu nepracovali na vlastných úlohách izolovane, ale spolupracovali pre splnenie spoločného cieľa. Počas šprintu je tento cieľ realizovaný úlohami, ktoré boli vybrané do backlogu pre aktuálny šprint a ďalšími súvislosťami, ktoré motivujú tím pracovať spolu.

Kapitola 3

PRÁCA S VERZIAMI

Zodpovedná osoba: Bc. Andrej Vitek

Posledná úprava: 02.12.2017, verzia 2.0

Cieľom tejto metodiky je definovať spôsob práce s git repozitárom v rámci tímového projektu a tým dosiahnuť prehľadné verziovanie kódu nezávisle od člena tímu, ktorý zmeny vykonáva. Metodika sa zaoberá spôsobom práce s vetvami, tvarom commit správ a spôsobom ako spájať už dokončenú funkcionálnosť vo vetve s hlavnou vetvou.

3.1 Role

Zodpovedný člen tímu - osoba ktorá je zodpovedná za vykonané zmeny.

Reviewer - osoba zodpovedná za code review.

3.2 Vetvy

Pre zjednodušenie vývoja sa budú pre nové funkcionality a riešenia v projekte vytvárať vetvy. Úlohy, pre ktoré sa budú vytvárať nové vetvy, by mali byť na úrovni „story“ alebo náročnejších úloh. Názvy vetiev sa budú skladať z 3 častí:

- typ funkcionality - podľa riešenej úlohy obsahuje buď „feature“ alebo „bug_fix“,
- časť projektu - ktorá časť projektu je riešená (backend, frontend, development, ...),
- názov - napísané malým písmenom v angličtine a v prípade viacerých slov, budú slová oddelené podčiarkovníkom (znak "_").

Názov vetvy sa bude skladať spojením týchto 3 častí pričom časti budú oddelené lomítkom (znak "/"). Vzor názvu vetvy: typ_funkcionality/časť_projektu/názov

Vetvy sa prehadzujú pomocou príkazu „git checkout nazov_vetvy“ a vytvárajú pomocou príkazu "git checkout -b nazov_vetvy", pričom vytváraná vetva sa vytvorí na základe aktuálne vybranej vetvy. Nové vetvy je potrebné vytvárať na základe hlavnej vetvy.

V prípade, že sa pokračuje v práci s vetvou, ktorá už bola spojená s hlavnou vetvou, tak je potrebné pracovať podľa nasledujúcich krokov.

1. Prejsť do vetvy, ktorá sa ide meniť pomocou príkazu "git checkout nazov_vetvy"
2. Natiahnuť si najnovšiu verziu aktuálnej vetvy príkazom "git pull origin nazov_vetvy"
3. Natiahnuť si najnovšiu verziu hlavnej vetvy príkazom "git pull origin master"
4. Vykonať zmeny pomocou commit-ov

5. Znovu natiahnuť najnovšiu verziu hlavnej vetvy príkazom "git pull origin master"

Správne názvy:

- bug_fix/backend/<bug_xy>
- feature/frontend/<feature_xy>

Nesprávne názvy:

- feature-BackEnd-Feature-xy (zlý formát)
- frontend/bug_fix/#9 (vymenené časti v názve)
- feature/frontend/tests-rework (_ namiesto -)
- 1-odlisne-nazvy-poli-v-api-dokumentaciach (zlý formát a slovenský jazyk)

Po ukončení práce sa pre vetvu vytvorí merge request.

3.3 Commit-y

Po ukončení práce na časti funkcionality sa vytvorí commit obsahujúci krátku správu popisujúcu, čo bolo zmenené. Správa bude obsahovať do 10 slov v anglickom jazyku v imperatívne (veta bude mať formu príkazu a začína slovesom) v jednoduchom prítomnom čase. Prvé slovo vo vete bude začínať veľkým písmenom a veta bude končiť bodkou.

Commity sa vytvárajú pomocou príkazu "git commit -m <správa>".

Správne názvy:

- Add table xy.
- Fix runtime error while testing with empty DB.

Nesprávne názvy:

- Added (chýba predmet a bodka)
- table_for_database (chýba podmet, malé začiatkové písmeno, zlý formát a bodka)
- basic layout and look for team page (chýba podmet a bodka)
- Updated README.md. (minulý čas slovesa)
- Anonymous user login fix. (zlý slovosled -> nie je príkaz)
- CorsMiddleware fix for development and auth navigation fix (zlý slovosled a bodka)

3.4 Merge request-y

Pred spojením vetvy s dokončenou funkcionalitou s hlavnou vetvou je potrebné postupovať podľa nasledujúcich krokov. V prípade, keď ide o funkcionalitu, ktorá sa vytvára na úrovni viacerých vetví (napr. obsahuje backend aj frontend), tak musí prebehnúť ešte integrácia medzi jednotlivými časťami v novej vetve. Integrácia sa bude konať vo vetve s názvom typ_funkcionality/development/názov podľa pravidiel v časti 3.2. Túto vetvu je potrebné vytvoriť na základe vetvy, ktorá sa bude integrovať, alebo s ňou spojiť ak už existuje. Ak je daná funkcionalita len na úrovni jednej vetvy, tak sa integračná vetva nevytvára. Na koniec je ešte potrebné si natiahnuť poslednú verziu hlavnej vetvy a vyriešiť konflikty.

Spôsob, ako toto spraviť, je použiť príkazy:

- "git checkout nazov_vetvy"

- "git pull origin master"

Po vyriešení konfliktov je potrebné vytvoriť merge request, ktorý musí prejsť prehliadkou. Človek vykonávajúci prehliadku (reviewer) doplní svoje pripomienky pomocou komentárov do merge requestu. V prípade, ak po dokončení prehliadky sú v kóde nejaké pripomienky, tak sa pošlú zodpovednému členovi tímu na prepracovanie. Ak po prehliadke nie sú žiadne pripomienky, tak sa merge request akceptuje.

Kapitola 4

VÝVOJ A INTEGRÁCIA

Zodpovedná osoba: Bc. Miroslav Valčíčák

Posledná úprava: 02.12.2017, verzia 1.1

Metodika je určená všetkým, ktorí sa akýmkoľvek spôsobom podieľajú na tvorbe produktu a jeho nasadení do produkčného prostredia. Pritom nadväzuje na:

- Kapitola 3. Práce s verziami
- Kapitola 5. Dokumentácia
- Kapitola 6. Testovanie

Snahou nášho tímu je, aby proces vývoja a integrácie bol čo najjednoduchší, najefektívnejší, nadmieru nezaťažoval tím a zároveň zaručoval zachovanie vysokej kvality vytváraného produktu. Taktiež sa proces pokúša presunúť čo najviac rutínnej práce na nástroje a systémy podporujúce ich automatizované vykonávanie.

4.1 Role

Databázový návrhár - človek, ktorý navrhuje dátový model a definuje obmedzenia.

Návrhár grafického používateľského rozhrania - jeho úlohou je navrhnuť rozhranie pre používateľa z vizuálneho hľadiska. Zaoberá sa farbami, ale aj rozmiestnením jednotlivých prvkov.

API návrhár - je zodpovedný za dobre definované rozhranie pre komunikáciu medzi klientom a serverom.

Vývojár - implementuje návrhy a odstraňuje implementačné chyby.

Recenzent zdrojového kódu - vykonáva statickú analýzu kódu, za účelom odhalenia chýb, odstraňovania nekonzistencie a nedodržiavania dohodnutých zásad pri programovaní.

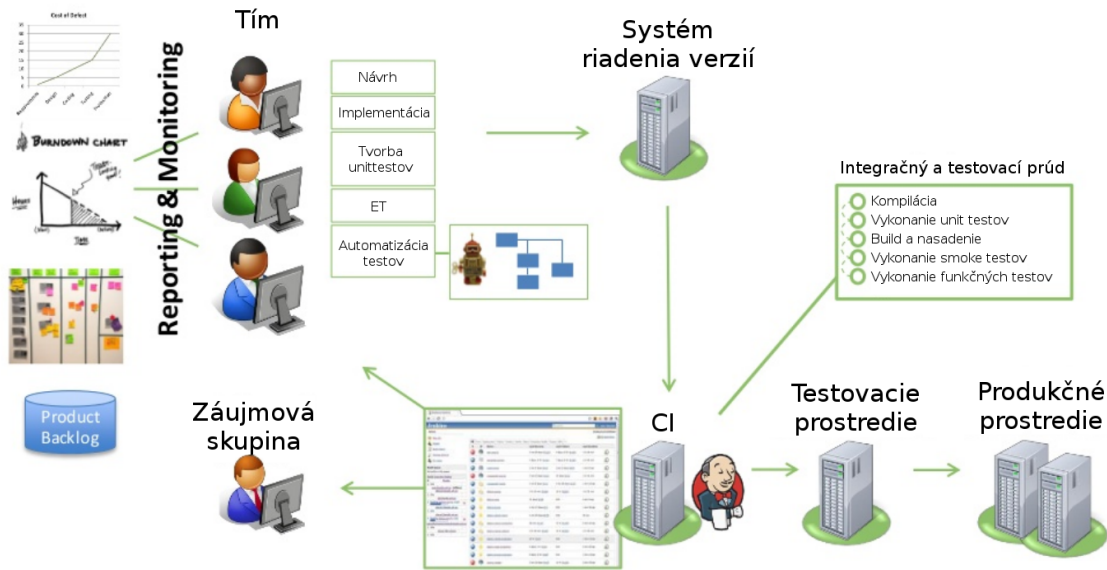
Správca prostredí - osoba zodpovedná za riešenie akýchkoľvek problémov spojených s integráciou a funkčnosťou prostredí

4.2 Opis procesov

Návrh procesu vývoja a integrácie bol výrazne ovplyvnený praktikami kontinuálnej integrácie (ang. *continuous integration*, Obrázok 4.1)¹ a kontinuálneho nasadzovania (ang. *continuous deployment*)². Ich aplikovanie v procese pomáhajú dosiahnuť nasledovné nástroje a služby:

- Git/GitLab (manažment verzií zdrojového kódu)
- Docker (integrácia)
- Jenkins (testovanie, nasadzovanie)

Kontinuálne nasadzovanie



Obr. 4.1: Proces kontinuálneho nasadzovania.

V súčasnosti sa pri vývoji produktu využívajú tri prostredia a to:

- vývojové
- staging
- produkcia

každé prostredie má vlastnú sadu konfiguračných súborov. Vývojové prostredie slúži taktiež ako testovacie prostredie. Staging prostredie slúži tímu v prvom rade na overenie bezproblémového integrovania novej funkcionality do existujúceho systému aj pomocou automatických integračných testov. Rovnako, slúži product ownerovi na sledovanie stavu vývoja funkcionality, ich validovanie a testovanie.

Pri vývoji každej samostatnej funkcionality je potrebné dodržiavať nasledujúcu postupnosť krokov:

1. vyberie sa prípad použitia
2. databázový návrhár vytvorí návrh dátového modelu a jeho vizualizácia UML diagramom na základe definovaných požiadaviek (ak to charakter úlohy umožňuje)
3. návrhár grafického používateľského rozhrania vytvorí wireframe-y a mockup-y pre používateľa systému (ak to charakter úlohy umožňuje)
4. API návrhár definuje rozhrania komunikácie medzi klientskou časťou systému a serverovou časťou - potrebná konzultácia s vývojárom klientskej aj serverovej časti systému
5. vývojári vytvoria osobitné vetvy v systéme na manažment verzií zdrojového kódu
6. vývojári implementujú časti funkcionality
7. vývojári vytvoria testy pre príslušné časti zdrojového kódu

¹<https://codeship.com/continuous-integration-essentials>

²<https://codeship.com/continuous-integration-essentials>

8. vývojári vytvoria dokumentáciu k príslušnej časti zdrojového kódu
9. vývojári okomentujú a odošlú zmeny do vzdialeného repozitára vetvy
10. v prípade, že funkcia nie je úplne implementovaná, pokračovať bodom 6
11. vývojári si k sebe prevezmú zmeny z hlavnej vývojovej vetvy
12. vývojári vyriešia prípadné konflikty
13. vývojári pridajú komentáre a odošlú riešenia konfliktov do vzdialeného repozitára vetvy
14. vývojári vytvoria požiadavky na zlúčenie s hlavnou vývojovou vetvou (ang. *merge request*) v prostredí GitLab-u (iba ak ešte neexistuje)
15. vývojári požiadajú recenzentov zdrojového kódu (sú pridelení pre každú úlohu) o vykonanie code review
16. recenzenti zdrojového kódu si prezrú novovytvorený kód a v prípade potreby pridajú k nemu pripomienky
17. vývojári zapracujú pripomienky vo vlastnej vetve a pokračujú bodom 11 (iba v prípade, že nejaké pripomienky existujú)
18. vývojári zlúčia vývojové vetvy

Po každej zmene v hlavnej vývojovej vetve Jenkins spustí automatizované testy a v prípade ich úspešného ukončenia nasadí novú verziu produktu do staging prostredia. Vývojári sú povinný sledovať stav testovania a v prípade ich neúspešného dokončenia musia chyby odstrániť vo vlastných vývojových vetvách. Bod číslo 2 a 3 sú od seba navzájom nezávislé a môžu sa vykonávať paralelne.

V prípade úspešného ukončenia procesu vývoja, testovania a nasadenia na staging prostredie sa vyzve product owner, aby zvalidoval implementovanú funkcionálnosť a po jej akceptovaní správca prostredia dá pokyn Jenkinsu na jej automatické nasadenie do produkčného prostredia.

Každý člen tímu je povinný v prípade objavenia akejkoľvek chyby počas svojho vývoja, testovania alebo akejkoľvek práce s aplikáciou v staging prostredí túto chybu zdokumentovať a nahlásiť. Proces manažmentu chýb je opísaný v osobitnej metodike.

4.3 Záver

Aby mala metodika procesu vývoja a integrácie reálny pozitívny prínos, je nevyhnutné, aby ju každý člen vývojového tímu rešpektoval a dodržiaval. Jednotlivé kroky vývoja môžu presnejšie upravovať ďalšie metodiky, s ktorými sa je každý člen tímu povinný oboznámiť ešte pred začatím práce na vývoji produktu.

Kapitola 5

DOKUMENTÁCIA

Zodpovedná osoba: Bc. Michal Dolnák, Bc. Michal Hucko

Posledná úprava: 19.11.2017, verzia 2.0

Cieľom dokumentu je objasniť postup pri dokumentovaní. Metodika je záväzná a platí pre všetkých členov tímu 1, ktorí sa podieľajú na tvorbe kódu. Všetky postupy opísané v tomto dokumente sú platné. Táto metodika nadväzuje na metodiku:

- Kapitola 3. Práca s verziami

5.1 Role

Programátor - osoba ktorá je zodpovedná za napísaný kód (či na frontende, alebo backende).

Reviewer - osoba zodpovedná za code review.

5.2 Predpoklady

5.2.1 Spustenie frontendu

Kľúčové závislosti Angularu sú (názov -> odporúčaná verzia):

- npm -> 5.3.0
- nodeJs¹ -> 8.6.0
- ng -> 1.4.4

Odporúča sa pracovať na operačných systémoch založených na UNIX technológii, ako napr. Linux či OSX. Pre používateľov Windows doporučujeme použiť VirtualBox na vytvorenie UNIX virtuálneho stroja.

- npm a nodejs stiahnete pomocou príkazu "\$ apt-get install npm"(Linux) alebo príkazom "\$ brew install npm"(OSX).
- ng nainštalujete pomocou príkazu (Linux, OSX): "\$ npm install -g @angular/cli"

Lokálne spustenie

Postup:

- Lokálne spustíme aplikáciu pomocou príkazu: "\$ ng serve"
- V environment.local.ts si je potrebné nastaviť premennú prostredia apiBase, aby smerovala na backend.

¹<https://nodejs.org/en/>

- Backend musí mať povolené CORS pre prístup z portu, kde je lokálne spustený frontend. V opačnom prípade odmietne prístup a nie je možné lokálne vyvíjať².

Lokálne spustenie (docker)

V súbore dockerfile v priečinku frontendu je nutné spúšťať aplikáciu pomocou príkazu: `"CMD ["npm", "run", "devel"]"`

5.2.2 Spustenie backendu

Na to, aby bolo možné písať dokumentáciu je potrebné aby mal programátor nasledujúce knižnice:

- `django` (`"$ pip install django"`)
- `django-rest-framework` (`"$ pip install djangorestframework"`)
- `django swagger` (`"$ pip install django-rest-swagger"`)

Swagger je nástroj na dokumentovanie. V našom projekte ho používame na dokumentovanie REST rozhrania. Aby správne bežal, musí byť okrem inštalácie pridaný k nainštalovaným aplikáciám³:

```
1 INSTALLED_APPS = [
2     ...
3     'rest_framework_swagger',
4     ...
5 ]
```

Ukážka zdrojového kódu 5.1: Vyhodenie dokumentácie Python

A taktiež musí byť pridaný medzi URL v projekte:

```
1 from django.conf.urls import url
2 from rest_framework_swagger.views import get_swagger_view
3
4 schema_view = get_swagger_view(title='Some cool name')
5
6 urlpatterns = [
7     url(r'^$', schema_view)
8 ]
```

Ukážka zdrojového kódu 5.2: Vyhodenie dokumentácie Python

Lokálne spustenie

Pre lokálne spustenie projektu prejdite do adresára `/backend/group/` v koreňovom adresári projektu a spustite príkaz: `"$ python manage.py runserver"`

Následne beží na `localhost:8000/`.

Lokálne spustenie (docker)

Pre lokálne spustenie pomocou dockeru je potrebné z koreňového adresára projektu spustiť príkaz: `"$ docker-compose up"`

Následne beží na `localhost:8000/`.

²https://en.wikipedia.org/wiki/Cross-origin_resource_sharing

³<https://django-rest-swagger.readthedocs.io/en/latest/>

5.3 Opis procesu

5.3.1 Frontend

Pravidlá:

- Dokumentáciu sa píše k triedam, k funkciám a k zložitejším premenným.
- Pri dokumentácii funkcií, ktoré majú parametre a návratové hodnoty používajte tagy `@param` a `@return` s opisom a použitými dátovými typmi.
- Dokumentáciu píšete po anglicky a vo vetách.
- Ak majú premenné deklarované typy, tak ich píšeme v zátvorkách.

```
1 /**
2 * Function creates http request to the server in a proper format, which is decided by it's
   type.
3 * @param options object with type(string), query(boolean) and body(object) fields
4 * @param callback to call after response is received
5 *
6 * @return callback(err(object)), data(object))
7 */
```

Ukážka zdrojového kódu 5.3: Príklad dokumentácie frontend

5.3.2 Backend

Pre tento účel sa používa nástroj swagger. Aby swagger vedel, čo má dať do dokumentácie, špecifikujú sa detaily už počas vytvárania modelu⁴. Je potrebné uviesť, ktoré polia sú povinné, aký majú typ, správne napísať ich názov, prípadne uviesť základné hodnoty. Príklad:

```
1 class Vote(models.Model):
2     created = models.DateTimeField(auto_now_add=True)
3     username = models.CharField(max_length=100, default="User")
```

Ukážka zdrojového kódu 5.4: Príklad dokumentácie backend

Ďalej je potrebné okomentovať každú metódu vo view, ktorá bude použitá v URL. Swagger používa tieto komentáre ako pole Implementation notes. Je nutné, aby bolo v komentári spomenuté, na čo bude metóda slúžiť. Príklad:

```
1 class EventCreate(generics.GenericAPIView):
2     """
3     Create new Event with given Options.
4     """
5     def get_serializer_class(self):
6         return NewEventSerializer
7
8     def post(self, request, format=None):
9         """
10        Create new Event with given Options.
11        """
```

Ukážka zdrojového kódu 5.5: Príklad dokumentácie backend

Komentáre

V projekte je nutné dodržať štruktúru komentárov. Komentáre dodržiať nasledujúce pravidlá:

- Komentár je písaný po anglicky.
- Komentár začína veľkým písmenom.

⁴<http://www.django-rest-framework.org/topics/documenting-your-api/>

- Komentár končí bodkou.
- Komentár je písaný rozkazovacím spôsobom.

Vhodné príklady:

```
1 "Create new action."
2 "Update action depending on the date."
```

Ukážka zdrojového kódu 5.6: Príklad komentárov backend

Nevhodné príklady:

```
1 "It will create a record"
2 "Some method"
```

Ukážka zdrojového kódu 5.7: Príklad komentárov backend

5.4 Záver

Dokumentácia je jedna z častí definition of done. Programátor je teda zodpovedný za jej obsah. Reviewer je povinný okrem funkcionality kontrolovať aj dokumentáciu, či je v súlade s týmto dokumentom. Ak sa počas vývoja produktu vyskytnú nedostatky tohto dokumentu, je možné ho rozšíriť.

Kapitola 6

TESTOVANIE

Zodpovedná osoba: Bc. Michal Dolnák, Bc. Michal Hucko
Posledná úprava: 02.12.2017, verzia 3.0

Cieľom dokumentu je objasniť postup pri testovaní. Metodika je záväzná a platí pre všetkých členov tímu 1, ktorí sa podieľajú na tvorbe kódu. Všetky postupy opísané v tomto dokumente sú platné. Táto metodika priamo nadväzuje na metodiku:

- Kapitola 5. Dokumentácia

6.1 Role

Programátor - osoba, ktorá je zodpovedná za napísaný kód (či na frontende, alebo backende).

6.2 Definition of done z pohľadu testovania

Každý prípad použitia musí byť napísaný test, ktorý zachytáva korektné správanie aplikácie pri vykonávaní prípadu použitia. Test nepíšeme len v prípade, že ho z objektívnych príčin nie je možné vykonať alebo by jeho výsledok nebol deterministický. Aplikácia sa odovzdáva na posúdenie (ang. *review*), ak sú funkčné všetky existujúce automatické testy.

6.3 Predpoklady

6.3.1 Frontend

- Predpoklad je fungujúca frontendová aplikácia. Návod na spustenie aplikácie nájdete v Kapitole 5, v časti dokumentovanie frontendu, v sekcii Návod na lokálne spustenie projektu.
- Testovací framework sa nachádza v priečinku protractor a obsahuje dva docker súbory - docker-compose.yml a docker-compose.test.yml. Druhý súbor slúži pre automatické testovanie aplikácie pomocou nástroja Jenkins.
- Pre testovanie aplikácie spustenej lokálne v dockeri spustíme v priečinku protractor príkaz „docker-compose up“. Docker stiahne a spustí selenium a chrome a spustí testy aplikácie na url „localhost:4200“.
- Na písanie samotných testov je lepšie nainštalovať si protractor a selenium lokálne. Návod nájdete tu <http://www.protractortest.org/#/>.
- Po spustení lokálneho selenium servera môžete lokálne pustiť testy pomocou príkazu „protractor protractor.local.js“ s tým, že je potrebné nastaviť parameter baseUrl na adresu, kde ste si lokálne spustili aplikáciu.

6.3.2 Načítanie testovacích údajov cez docker

Na načítanie testovacích údajov do databázy, zálohu aktuálnej databázy a jej obnovenie slúžia 3 príkazy:

- „docker-compose run backend python manage.py setupdbfrontend“ - resetuje databázu a vloží testovacie údaje.
- „docker-compose run backend python manage.py setupdbfrontendwithbackup“ - zálohuje aktuálne údaje v databáze, resetuje ju a vloží testovacie údaje.
- „docker-compose run backend python manage.py restorebackup“ - resetuje databázu a načíta zálohované údaje.

Tieto príkazy sa spúšťajú z hlavného adresára projektu, tj. adresár v ktorom je súbor docker-compose.yml.

6.3.3 Backend

Zdrojový kód na backende je napísaný nad frameworkom django-rest. Verzia frameworku django je aspoň 1.11.5 a frameworku django-rest je 3.6.4. Pri implementácii testovania používame štandardné testovanie nad frameworkom Django, ktoré sa dá spustiť z koreňového adresára backendu cez príkaz: "\$ python manage.py test"

6.4 Opis procesu

6.4.1 Frontend

Štruktúra testov

Príklad testu:

```
1 describe('Home page is correct.', () => {
2   let page: HomePage;
3
4   beforeEach(() => {
5     page = new HomePage();
6   });
7
8   it('should display project name in header', () => {
9     page.navigateTo();
10    expect(page.getProjectName()).toEqual('Group de Cider');
11  });
12 });
```

Ukážka zdrojového kódu 6.1: Príklad testu frontend

Testy sa skladajú z 3 častí. Prvou je časť describe, kde je nutné uviesť aká časť systému sa testuje. Druhou sú deklarované komponenty alebo stránky, ktoré sa budú testovať. Tretou sú jednotlivé časti testu zoskupené v podmienkach¹.

Formálne zásady:

- Popisy testu sa píše po anglicky.
- Všetky súbory s testami musia byť nazvané *.e2e.ts a musia byť umiestnené v priečinku protractor/e2e.
- Na testovanie sa používa framework protractor².

¹<http://www.protractortest.org/#/style-guide>

²<http://www.protractortest.org>

Štruktúra stránok

```
1
2 export class HomePage {
3   navigateTo() {
4     return browser.get('/');
5   }
6
7   getProjectName() {
8     return element(by.id('project-name')).getText();
9   }
10
11   getAboutUsButton() {
12     return element(by.id('about-us'));
13   }
14
15   getLoginButton() {
16     return element(by.id('login-button'));
17   }
18 }
```

Ukážka zdrojového kódu 6.2: Príklad testu frontend

Stránky opisujú komponenty aplikácie a vyberajú z nich jednotlivé elementy, ktoré sa následne používajú v testoch. Elementy sa vyberajú na základe atribútov unikátnych identifikátorov (id) a ak to nie je možné, tak na základe atribútov tried.

6.4.2 Backend

Štruktúra testov

Testy organizujeme do logických celkov, ktorými sú triedy. Každá trieda môže dediť z jednej z možností:

- Štandardnej triedy djanga `TestCase` v prípade, že sa jedná o funkcionality mimo frameworku `django-rest`: `"from django.test import TestCase"`
- Triedy `APITestCase` v prípade testovania funkcionality `django-rest` frameworku. Viac v kapitole 6.4.2. `"from rest_framework.test import APITestCase"`

Testovacie scenáre sú ďalej organizované do metód.

Štábná kultúra

Konvencie štruktúry písania názvov V prípade „test case-ov“ použijeme camel-case formát názvu (prvé písmeno veľké a každé nové slovo s veľkým, všetky slová spolu). V prípade špecifických názvov metód využijeme snake-case (každé slovo s malým oddelené podtržníkom). Príklad:

```
1 class MyNewTestCase(APITestCase):
2     def new_test_1_test(self):
```

Ukážka zdrojového kódu 6.3: Príklad testu backend

Konvencie štruktúry obsahu názvov Celý „test case“ je potrebné robiť v kontexte view, na ktorý sa viaže. To znamená, že ak testujem funkcionality view, kde vytvárame Auto, názov bude vytvorený presne podľa toho. Názvy test-casov končia reťazcom „TestCase“. Konkrétne metódy v test case (testy) sú potom pomenúvané podľa špecifickej udalosti, ktorá je testovaná. Názvy testov končia reťazcom „_test“. Nasledujúci príklad ukazuje ako pomenovať test case a testy na vytváranie triedy auta:

```
1 class CreateCarTestCase(APITestCase):
2     def test_create_car_test(self):
3         ...
4     def test_create_car_without_wheels_test(self):
```

Ukážka zdrojového kódu 6.4: Príklad testu backend

```

1 class New_testcase(APITestCase):
2     def testcreatecartest(self):
3         ...
4     def testCreateCarWithoutWheelsTest(self):

```

Ukážka zdrojového kódu 6.5: Nevhodný príklad testu backend

Django-rest APITestCase

Výhodou použitia tejto triedy je fakt, že nám poskytuje parameter “client”, cez ktorý vieme jednoducho poslať dopyty na REST API. Príklad³:

```

1 class CreateCarTestCase(APITestCase):
2     def test_create_car_test(self):
3         response = self.client.post(url, data, format='json')
4         ...

```

Ukážka zdrojového kódu 6.6: Príklad testu backend

6.5 Automatizované testovanie

Automatizované testovanie je vykonávané na základe zmeny vo vetve master nášho git repozitára. Celý proces zastreňuje nástroj pre kontinuálnu integráciu – Jenkins, ktorý sa nachádza na adrese <http://team01-17.studenti.fiit.stuba.sk:8081/>. Po prihlásení je možné vidieť všetky časti projektu s ich aktuálnym stavom. Modrá farba pri časti znázorňuje úspech pri poslednom vykonaní, červená neúspech. Posledných 5 vykonaní je zhrnutých vo forme počasia, ktoré znázorňuje percentuálnu úspešnosť posledných 5 vykonaní. Pre podrobnejšie informácie stačí prejsť na ikonku s počasím danej časti. Z dôvodu nižšej konfiguračnej náročnosti je celý proces rozdelený na 4 časti:

- timak,
- tester,
- pusher,
- deployer.

Lepší náhľad na obsah a tvorbu jednotlivých častí vysvetlí krátky tutoriál pre vytvorenie časti.

timak Časť s názvom „timak“ má na starosti počúvanie a reagovanie na git repozitár nášho projektu. Pri zmene vo vetve master stiahne novú verziu tejto vetvy na náš server.

tester Časť „tester“ je ústredná v celom automatickom testovaní. V jednotlivých krokoch, v súlade s predchádzajúcimi časťami a príkazmi tejto metodiky, volá

- zastavenie starého testovacieho prostredia,
- build novej verzie,
- spustenie testov backend-u,
- nasadenie nového testovacieho prostredia,
- zálohovanie a prípravu testovacej databázy pre integračné testovanie,

³<http://www.django-rest-framework.org/api-guide/testing/>

- spustenie frontend integračných testov,
- obnovenie zálohy testovacej databázy.

Všetky tieto kroky sú zhrnuté do 3 celkov: *Building*, *Backend Testing* a *Frontend Testing*. Tieto celky možno vidieť aj po rozkliknutí časti „tester“.

pusher Časť „pusher“ je volaná z časti „tester“ po úspešnom vykonaní všetkých testov. Zabezpečí, aby sa nová funkcionálnosť dostala do vetvy release.

deployer Časť „deployer“ je nastavená tak, aby stiahla obsah vetvy release a spustila novú produkčnú verziu nášho projektu. Jej konfigurácia predpokladá odsúhlasenie nasadenia novej verzie produktovým vlastníkom, a preto je spustenie jej vykonávania možné len manuálne pomocou „Build Now“.

6.6 Záver

Testovanie je jedna z častí definition of done. Programátor je teda zodpovedný za jej obsah. Posudzovateľ je povinný okrem funkcionality kontrolovať aj testy, či sú v súlade s týmto dokumentom. Okrem toho spúšťa testy. Ak počas sa počas vývoja produktu vyskytnú nedostatky tohto dokumentu, je možné ho rozšíriť.

Fakulta informatiky a informačných technológií
Slovenská technická univerzita

Group de'Cider

Dokumentácia k tímovému projektu časť 1
(Inžinierske dielo)

Tím: číslo 1, sElectors

Pedagogický vedúci: Ing. Ondrej Kaššák

Členovia tímu: Bc. Lukáš Babula, Bc. Pavel Balashov, Bc. Michal Dolnák, Bc. Michal Hucko, Bc. Elena Štefancová, Bc. Miroslav Valčíčák, Bc. Andrej Vítek

Akademický rok: 2017/2018

Úvod	2
1 Globálne ciele	3
1.1 Zimný semester	3
2 Celkový pohľad na systém	4
2.1 Backend systému	5
2.2 Frontend systému	5
3 Moduly systému	6
3.1 Modul hlasovacích stratégií	6
3.1.1 Analýza	6
3.1.2 Návrh	6
3.2 Modul hlasovania	6
3.2.1 Analýza	7
3.2.2 Návrh	7
3.2.3 Implementácia	7
3.2.4 Testovanie	7
3.3 Modul autentifikácie	7
3.3.1 Analýza	7
3.3.2 Návrh	8
3.3.3 Implementácia	8
3.3.4 Testovanie	8
Záver	9
Prílohy	10
Príloha A: Dokumentácia volaní z nástroja swagger	10
Volania patriace do modulu hlasovania	10
Volania patriace do modulu autentifikácie	13
Príloha B: Návrhy grafických rozhraní	14
Prvotný návrh grafických rozhraní	14
Druhá iterácia návrhu grafických rozhraní	15
Tretia iterácia návrhu grafických rozhraní	17
Wireframy pre testovanie rozhraní	18

Úvod

Dokument obsahuje informácie o projekte Group de'Cider po prvých 5 šprintoch v rámci predmetu Tímový projekt. Cieľom dokumentu je opísať technickú časť nášho projektu.

Dokument sa skladá z globálnych cieľov nášho projektu, cieľov v rámci zimného semestra a celkového pohľadu na vytváraný systém. Taktiež obsahuje informácie o moduloch systému a ich návrhu, implementácie a spôsobe testovania. V prípade, keď modul vyžadoval aj detailnejšiu analýzu nejakého problému, tak obsahuje aj túto časť.

1 Globálne ciele

Hlasovacích aplikácií existuje veľa. Väčšinou však zvládnu len jednoduché jednorazové úlohy. Rovnako spôsob hlasovania v nich je obmedzený na spočítanie hlasov, ktoré sa nehodí vždy.

V rámci tímového projektu sa snažíme vytvoriť univerzálne riešenie umožňujúce efektívne rozhodovanie v skupinách. Group de'Cider je webová aplikácia poskytujúca:

- tvorbu hlasovaní pre skupiny používateľov
- intuitívne používateľské rozhranie
- výber rozličných hlasovacích stratégií (average without misery, most pleasure, cumulative voting, dictatorship)
- opakované hlasovania (napríklad pre každodenný výber miesta na obed)
- komplexné hlasovania (súčasný výber napríklad času A miesta alebo miesta A aktivity)
- možnosť skrytého hlasovania
- chat pre rýchle vyjasnenie situácií

Vytváraná aplikácia je určená pre aktívnych ľudí komunikujúcich s priateľmi, kolegami či spolužiakmi, ktorí sa chcú vedieť dohodnúť dobre a rýchlo. Používať ju zvládne každý, kto vie ovládať počítač. Vytvoriť hlasovanie a pozvať doň priateľov je úplne jednoduché.

Biznis potenciál nášho riešenia bude môcť spočívať v platenej funkcionalite, napríklad v obmedzení maximálnej veľkosti skupiny pre free verziu, zamknutí zaujímavých hlasovacích stratégií, či obmedzení maximálneho počtu opakovaní hlasovania.

1.1 Zimný semester

Počas zimného semestra sa náš tím sústreďí na navrhnutie a implementáciu základnej funkcionality vytváraného systému, jeho nasadenie a použiteľnosť.

V rámci základnej funkcionality je naším cieľom poskytnúť možnosť vytvorenia hlasovania a opakovaného hlasovania, umožniť používateľovi zahlasovať a taktiež vytvoriť jednoduchú administráciu, pomocou ktorej bude možné upraviť vytvorené hlasovanie. Chceme taktiež umožniť používateľom sa prihlásiť a mať prehľad o všetkých hlasovaniach, ktoré buď vytvorili, alebo do nich patria.

Nasadením je myslené vytvorenie kontinuálnej integrácie pomocou nástroja Jenkins. Taktiež sem patrí vytvorenie Docker obrazu pre náš projekt a samotné nasadenie na server so zabezpečením možného prístupu cez internet.

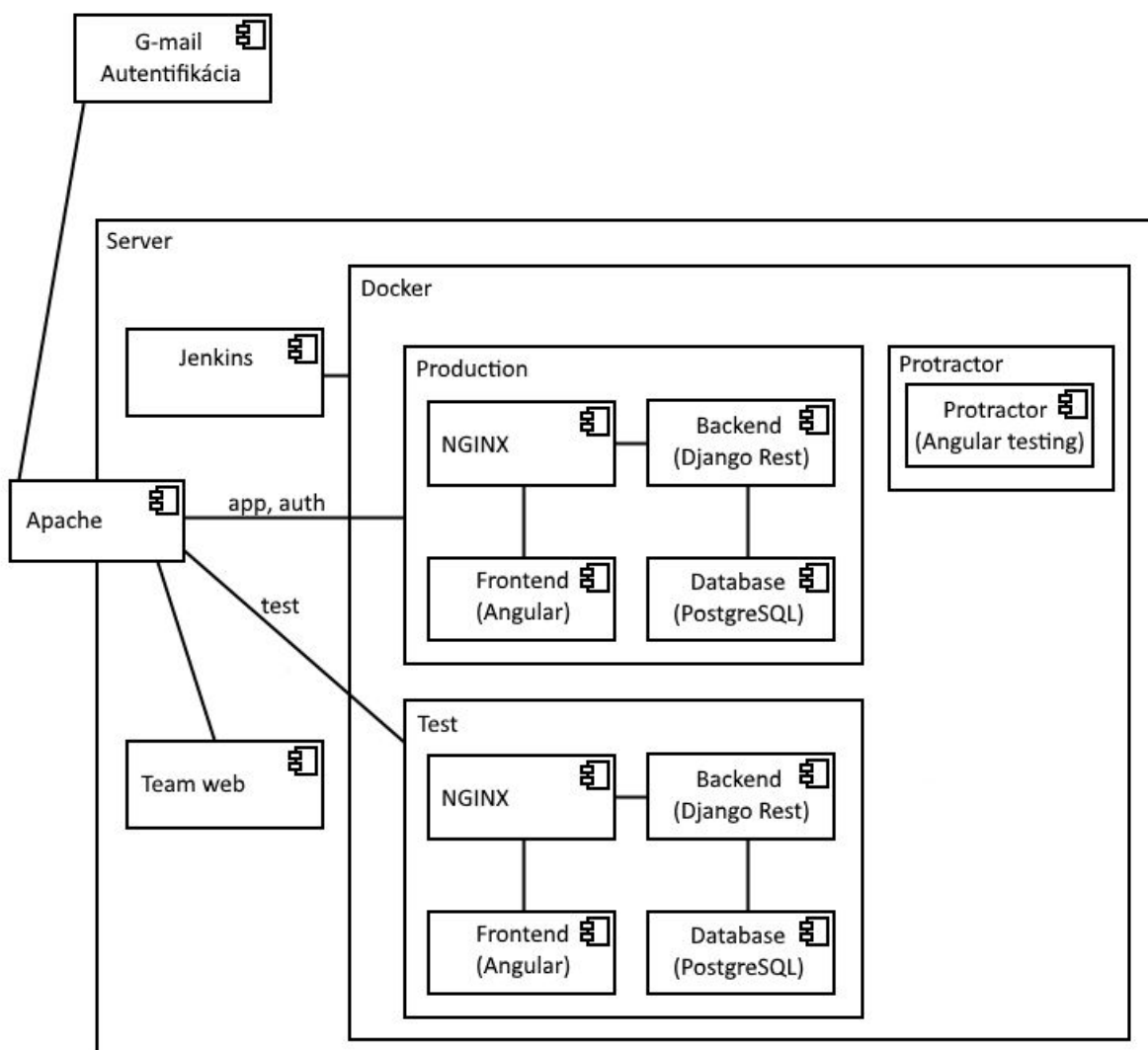
Na zabezpečenie použiteľnosti sa zameriavame na navrhnutie vhodného rozhrania pomocou mockupov.

2 Celkový pohľad na systém

Celá aplikácia pozostáva z docker riešenia, ktoré združuje tri kontainery. Prvým z nich je backend kontainer, ktorý v sebe uchováva django REST API a postgresql databázu. Druhým je frontend s angulárom. Aby tieto 2 kontainery vedeli medzi sebou komunikovať používame tretí s NGINXom, ktorý slúži ako proxy HTTP server. Využitie technológie dockeru značne podporuje integráciu a znovu-nasadenie aplikácie .

Compose spolu s hlavnou web stránkou sme nasadili na pridelený server. Ich komunikáciu s okolím zabezpečujeme pomocou apache HTTP serveru, ktorý funguje ako reverse proxy pre NGINX kontainer. Celkový pohľad na architektúru nášho systému v rámci servera je vidieť na obrázku 1.

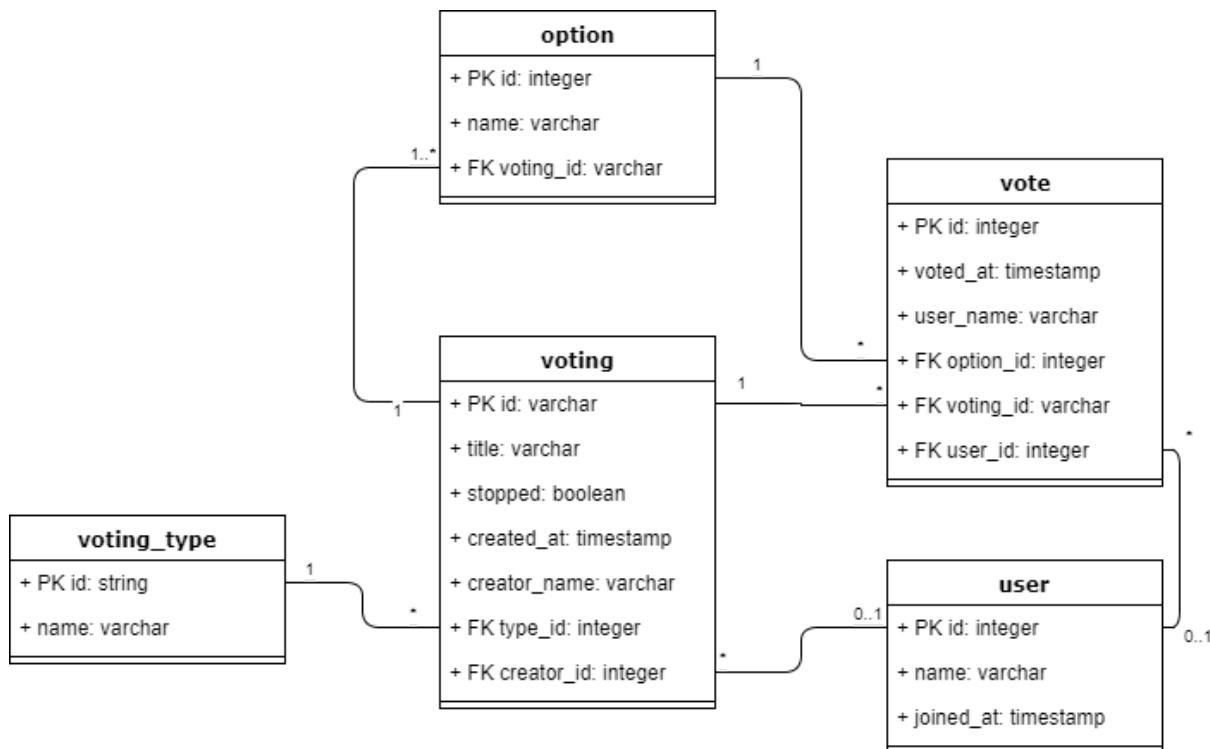
Kód projektu ako aj jednotlivé verzie vyvíjame pomocou systému na kontrolu verzií Git uchováваме vo vytvorenom úložisku od spoločnosti GitLab.



Obr. 1: Ukážka architektúry systému v rámci servera. Apache je použitý na zabezpečenie prístupu do vytvoreného Docker kontajnera s produkčnou a testovaciou verziou projektu a na tímovú stránku. Jenkins zabezpečuje automatizované nasadzovanie a testovanie.

2.1 Backend systému

Do časti backend patrí databáza PostgreSQL a rámec (framework) Django REST, ktorý slúži na vývoj webových aplikácií. Testovanie, správu databázy a vytváranie modelu ma tiež na starosti rámec Django REST a jeho knižnice. Dokumentácia je v rámci tejto časti zabezpečená pomocou nástroja Swagger. Návrh schémy databázy je vidieť na obrázku 2. Hlavnou úlohou tejto časti je komunikácia pomocou API volaní s frontend-om a poskytovanie aplikačnej logiky.



Obr. 2: Návrh schémy pre databázu.

Aby sme umožnili aj vytvorenie anonymných hlasovaní, tak v prípade vytvorenia nového hlasovania (voting) alebo zahlasovania za možnosť (vote), nie je nutné prepojenie s tabuľkou používateľa (user) a v tom prípade sa zaznamená len meno používateľa.

2.2 Frontend systému

Frontend je v rámci nášho projektu oddelený od backend-u a je vytvorený pomocou aplikačného rámca Angular v programovacom jazyku typescript. Na automatické testovanie celej aplikácie sa využíva aplikačný rámec Protractor, ktoré je postavený na nástroji Selenium. Automatické testy sa spúšťajú prostredníctvom nástroja Headless Chromium. Dokumentácia zdrojového kódu sa riadi odporúčaniami projektu TypeDoc.

Do aplikácie sa postupne pridávajú grafické návrhy používateľských rozhraní prostredníctvom jazyka SASS, ktorý sa kompiluje na štandardné CSS.

3 Moduly systému

3.1 Modul hlasovacích stratégií

3.1.1 Analýza

Jedným z cieľov nášho systému je možnosť použiť rôzne zaujímavé hlasovacie stratégie pri vytváraní hlasovania. Kvôli tomu sme v tejto časti robili prieskum v oblasti možností hlasovania a systémov volieb. Mnohé tieto systémy a hlasovania sú v súčasnosti používané v politickej scéne, kde sa pomocou nich vyberajú politickí hodnostári. Sú však aj hlasovacie systémy a stratégie, ktoré majú svoj pôvod pri manažovaní kolaboratívneho výberu a rozhodovania sa. Medzi typické príklady takýchto stratégií patrí napríklad hlasovacia stratégia *Dot voting*.

Počas hľadania a zhromažďovania jednotlivých stratégií bol kladený dôraz aj na vedecké zdroje a publikácie z oblasti modelovania skupín používateľov, rozhodovania sa jednotlivcov v rámci takýchto skupín a možností rozhodovania sa skupiny ako celku. Na základe toho je výsledkom tejto analýzy výber trinástich rôznorodých, pre náš problém použiteľných, hlasovacích stratégií a ich modifikácií, z ktorých sa po spoločnej konzultácii v kruhu členov nášho tímu a produktového vlastníka niektoré vyberú a následne použijú v našom systéme.

Všetky hlasovacie stratégie uvažujú asynchrónny model rozhodovania sa, kedy používateľ hlasuje nezávisle na čase hlasovania ostatných používateľov. Do úvahy však pripadá aj možnosť, kedy by hlasovanie prebiehalo synchronne. V takom prípade by bolo možné dosiahnuť výsledné rozhodnutie hlasovaním v reálnom čase, na čo by bola napríklad vhodná stratégia viackolových volieb. Tento model hlasovania sme však bližšie nerozvíjali a v prípade potreby a záujmu sa mu budeme venovať v neskorších fázach projektu.

3.1.2 Návrh

Návrh tohto modulu je v súčasnosti vo veľmi ranom štádiu. Uvažujeme robustné riešenie na backend-e, ktoré by generalizovalo všetky hlasovacie stratégie pod spoločné rozhranie, kde by pre každú stratégiu existovali iné ohraničenia na základe možností voľby.

Keďže tento modul sa zatiaľ nachádza vo fáze návrhu, tak implementácia ani testovanie nie sú realizované.

3.2 Modul hlasovania

Tento modul sa zaoberá vytvorením hlasovania na základe vybraných možností, hlasovacej stratégie a titulu. Tiež sem patrí aj schopnosť používateľa zahlasovať v už vytvorenom hlasovaní a správa hlasovania cez administráciu, v ktorej je možné upravovať jeho vlastnosti. Po zahlasovaní má používateľ umožnené zmeniť možnosti, za ktoré hlasoval.

3.2.1 Analýza

Uskutočnili sme prieskum iných webových stránok, ktoré poskytujú hlasovania a zhodli sme sa, že by mal byť proces vytvorenia hlasovania z pohľadu používateľa čo možno najjednoduchší. Vytvorenie hlasovania a pridanie hlasu do hlasovania by malo byť umožnené prihláseným aj neprihláseným používateľom. Po vytvorení hlasovania dostane používateľ dve unikátne URL adresy. Pomocou prvej z nich môže administrovať hlasovanie. Druhá slúži na pridávanie hlasov do hlasovania inými používateľmi.

3.2.2 Návrh

Frontend a backend časti sú v našom systéme oddelené a vyžadujú definovať žiadosti (request) a odpovede (response) v API volaniach, ktoré sa používajú na komunikáciu medzi nimi. Pre účel komunikácie medzi backend-om a frontend-om sme zdefinovali API, ktoré umožňuje prenos dát vo formáte JSON.

Vytvoril sa návrh dátového modelu databázy, ktorý je súčasťou backend-u. Ďalej sa navrhli grafické rozhrania vytvorenia hlasovania a pridania nového hlasu do hlasovania, ktoré je možné vidieť v prílohe B.

3.2.3 Implementácia

Na strane backend-u boli implementované volania, ktoré poskytovali funkcionality vytvorenia hlasovania, pridania nového hlasu do hlasovania a získania aktuálneho stavu hlasovania. Dokumentáciu jednotlivých volaní z nástroja Swagger možno vidieť v prílohe A.

Na základe grafického návrhu boli implementované obrazovky vytvorenia hlasovania a pridania nového hlasu do hlasovania. Vzhľadom na to, že išlo o implementáciu prvých rozhraní medzi časťami systému, tak sa vyriešila integrácia všetkých používaných technológií.

3.2.4 Testovanie

Pre tento modul sú vytvorené automatizované testy. Na strane backend-u testy overujú konzistentnosť ukladaných dát. Na strane frontend-u sa testuje úspešné vykonanie všetkých používateľských scenárov automatickou simuláciou prostredníctvom webového prehliadača.

3.3 Modul autentifikácie

Primárnou úlohou modulu je overovať oprávnenosť používateľov vykonávať určité akcie. Slúži na prvotnú registráciu používateľa do nášho systému a overovanie jeho totožnosti pri prihlasovaní sa. Taktiež poskytuje ostatným modulom informáciu o identite aktuálne (ne)prihláseného používateľa.

3.3.1 Analýza

Je v našom záujme, aby sa používateľ cítil pri používaní našej webovej aplikácie čo najpríjemnejšie a nebol zbytočne zaťažovaný nepotrebnými úkonmi. Preto sme sa rozhodli proces registrácie - vkladanie osobných a prihlasovacích údajov a overovanie hesla -

nahradiť autentifikačnou službou tretej strany. Takéto riešenie by malo výrazne skrátiť dobu prvého prihlásenia sa a odbremeniť používateľa od pamätania si ďalších prihlasovacích údajov.

Na základe prieskumu popularity sociálnych sietí sme ako sprostredkovateľa autentifikácie zvolili *Google+ API* autentifikačné služby. Vybraný sprostredkovateľ je v súčasnosti jediný, s ktorým naša aplikácia spolupracuje. Je teda nevyhnutné, aby používateľ, ktorý sa rozhodne registrovať do našej služby, vlastnil nejaký účet (stačí e-mailový) od spoločnosti Google.

3.3.2 Návrh

Registrácia a prihlásenie zo strany používateľa pozostáva len z kliknutia na tlačidlo prihlásenia sa a potvrdenia povolenia poskytnúť osobné údaje používateľa našej službe zo stránky spoločnosti Google. Naša aplikácia následne prevezme potrebné informácie o používateľovi a v prípade, že sa jedná o nového používateľa, vytvorí nové záznamy v databáze.

3.3.3 Implementácia

Registrácia a autentifikácia prebieha v backend časti systému, ktorá komunikuje so službami Google cez poskytnuté rozhranie. Aby bolo možné vyžívať autentifikačné služby Google, bolo potrebné vytvoriť *credentials*. Takisto bol získaný SSL certifikát od certifikačnej authority *Lets's Encrypt* pre náš hostname, pretože ďalšou podmienkou vyžívania Google služieb bolo zabezpečené šifrované spojenie, ktoré implementáciu mierne skomplikovalo a oddialilo jej dokončenie.

Proces komunikácie s Google službami, vytváranie nových používateľov na základe získaných informácií a obnovovanie platnosti autorizačných tokenov prebral na seba modul od externých poskytovateľov *python-social-auth*. Informácie o prihlásení a oprávnenosti používateľov je typu *"session-based"* a ich kontrola prebieha výlučne na backend-e, s čím boli spojené prvotné komunikačné problémy s frontend-om, ktoré zapríčinili značné oneskorenie dokončenia a nasadenia modulu.

3.3.4 Testovanie

Testovanie registrácie a prihlasovania je vykonávané manuálne kvôli náročnej tvorbe testovacieho skriptu, ktorého výsledok je závislý od tretej strany. Avšak, testovanie kontroly oprávnenosti vykonávať akcie vyžadujúce prihlásenie, ako aj prístupnosť verejných funkcií pre anonymných používateľov, sa už vykonáva automaticky pomocou testovacieho skriptu.

Záver

Počas prvých troch šprintov sme sa sústredili na vytvorenie základného hlasovania, návrh používateľských rozhraní a na prípravu prostredia. Aby sme zjednodušili ďalší vývoj na projekte, tak sme vytvorili Docker obraz pre jednotlivé časti projektu a zabezpečili automatické nasadzovanie pomocou aplikácie Jenkins. Návrh používateľských rozhraní prešiel cez viacero iterácií, ako je možné vidieť v prílohe B.

Počas ďalších dvoch šprintov sme sa zamerali na úpravu architektúry spojenú s konfiguráciou Jenkinsu ako nástroja na kontinuálnu integráciu. Tento cieľ sa nám podarilo dosiahnuť. Momentálne máme úspešne nasadené dve prostedia. Jedno z nich je produkcia, ktorú má na starosti samotný Jenkins (po úspešnom vykonaní testov sa v nej prejavia zmeny) a druhou je test prostredie, kde sa testy vykonávajú.

V týchto šprintoch sme sa tiež venovali vývoju UX stránky produktu. Absolvovali sme testovanie s wireframami, farbami a obsahom so skupinou testerov, ktorých návrhy sme zdokumentovali a spoločne prešli. Všetky nadobudnuté poznatky aplikujeme pri implementácii frontendovej časti produktu.

Naším cieľom na ďalšie šprinty je implementácia funkcionality produktu. Chceme sa sústrediť najmä na rôzne stratégie hlasovania spojené s ich vyhodnocovaním. Taktiež chceme používateľom umožniť opakované hlasovanie. Okrem toho plánujeme implementovať funkcionality miestností pre hlasovanie.

Prílohy

Príloha A: Dokumentácia volaní z nástroja swagger

Volania patriace do modulu hlasovania

POST

/event_create/

Create new Event with given Options.

Implementation Notes

Create new Event with given Options.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
data	- title Hlasovanie strategy Dot voting options item 1 Možnosť 1 Delete item Move down item 2 Možnosť 2 Delete item Move up Add item Delete Last item Delete All body Model Example Value <pre>{ "title": "string", "strategy": "string", "options": ["string"] }</pre>			

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

Obr. 3: Dokumentácia pre volanie POST nad stránkou event_create, ktorého úlohou je vytvorenie nového hlasovania.

GET /event/{id}/ Retrieve details about certain voting, create new vote.

Implementation Notes
Retrieve details about certain voting, create new vote.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

[Try it out!](#)

Obr. 4: Dokumentácia pre volanie GET nad stránkou event/<id>, ktorého úlohou je získanie informácií o hlasovaní.

POST /event/{id}/ For certain Event create new Vote and assign its Option.

Implementation Notes
For certain Event create new Vote and assign its Option.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	-		body	Model

Parameter content type:

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

[Try it out!](#)

Obr. 5: Dokumentácia pre volanie POST nad stránkou event/<id>, ktorého cieľom je umožniť zahlasovať za možnosť.

PUT

/event/{id}/

For certain Event change selected Option of given Vote

Implementation Notes

For certain Event change selected Option of given Vote

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	- name Jozef voted_item_id 1		body	Model Example Value <pre>{ "name": "string", "voted_item_id": 0 }</pre>

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 6: Dokumentácia pre volanie PUT nad stránkou event/<id>, ktoré umožňuje zmenu zahlasovaných možností používateľa.

GET

/event/admin/{id}/

Administration of given Event, changing Event attributes.

Implementation Notes

Administration of given Event, changing Event attributes.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 7: Dokumentácia pre volanie GET nad stránkou event/admin/<id>, ktorý vracia informácie o hlasovaní.

POST /event/admin/{id}/ Change attributes of given Event.

Implementation Notes
Change attributes of given Event.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1		path	string
data	- stopped true		body	Model

Parameter content type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201			

Try it out!

Obr. 8: Dokumentácia pre volanie POST nad stránkou event/admin/<id>, ktorý umožňuje zmenu vlastností hlasovania.

Volania patriace do modulu autentifikácie

GET /auth/logout/ Log out currently logged in user.

Implementation Notes
Log out currently logged in user.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
page	1	A page number within the paginated result set.	query	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200			

Try it out!

Obr. 9: Dokumentácia pre volanie GET nad stránkou auth/logout, ktoré slúži na odhlásenie používateľa.

Príloha B: Návrhy grafických rozhraní

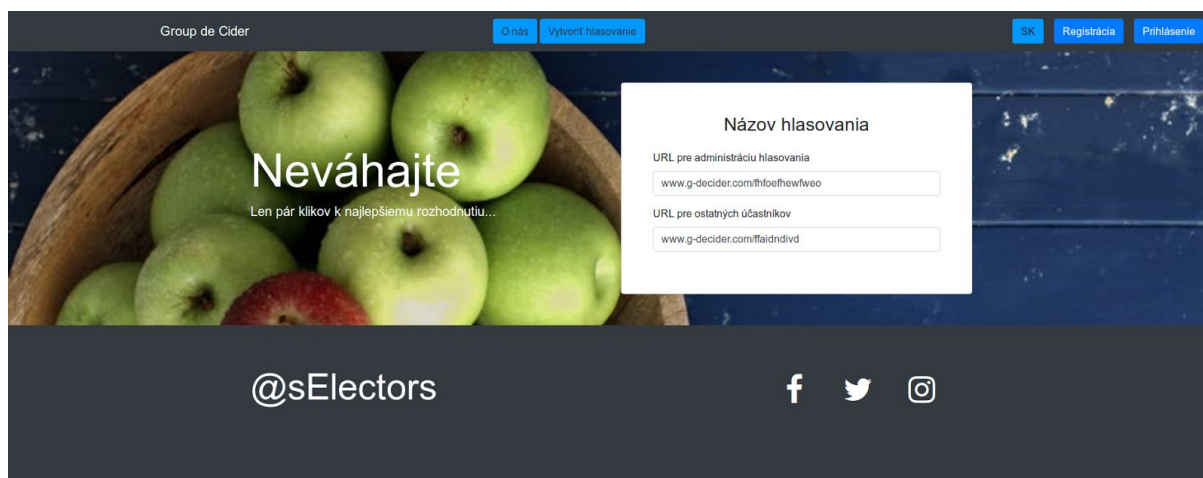
Prvotný návrh grafických rozhraní

The screenshot shows the 'Vytvor hlasovanie' (Create Poll) form in the @sElectors application. The form is titled 'Vytvor hlasovanie' and contains two input fields: 'Názov' (Name) with the placeholder 'Názov podujatia' and 'Možnosti' (Options) with the placeholder 'Možnosť 1; Možnosť 2'. A 'Dalej >>' button is located at the bottom of the form. The background features a wooden bowl of green and red apples. The top navigation bar includes 'Group de Cider', 'O nás', 'Vytvor hlasovanie', 'SK', 'Registrácia', and 'Prihlásenie'. The footer displays '@sElectors' and social media icons for Facebook, Twitter, and Instagram.

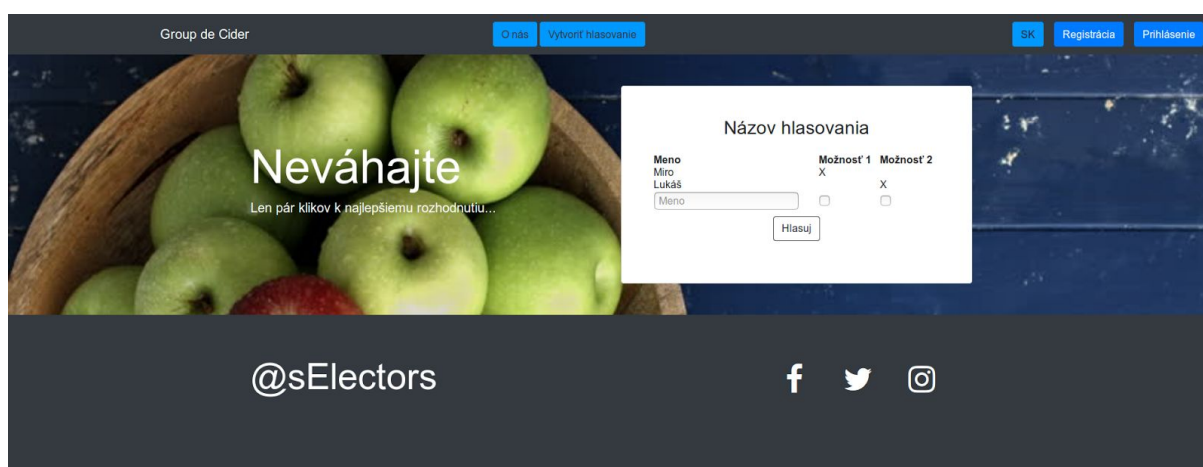
Obr. 10: Vytvorenie hlasovania - 1. obrazovka.

The screenshot shows the 'Vyber spôsob hlasovania' (Choose voting method) form in the @sElectors application. The form is titled 'Vyber spôsob hlasovania' and contains three radio button options: 'Možnosť č.1', 'Možnosť č.2', and 'Možnosť č.3'. A 'Vytvor hlasovanie' button is located at the bottom of the form. The background features a wooden bowl of green and red apples. The top navigation bar includes 'Group de Cider', 'O nás', 'Vytvor hlasovanie', 'SK', 'Registrácia', and 'Prihlásenie'. The footer displays '@sElectors' and social media icons for Facebook, Twitter, and Instagram.

Obr. 11: Vytvorenie hlasovania - 2. obrazovka.

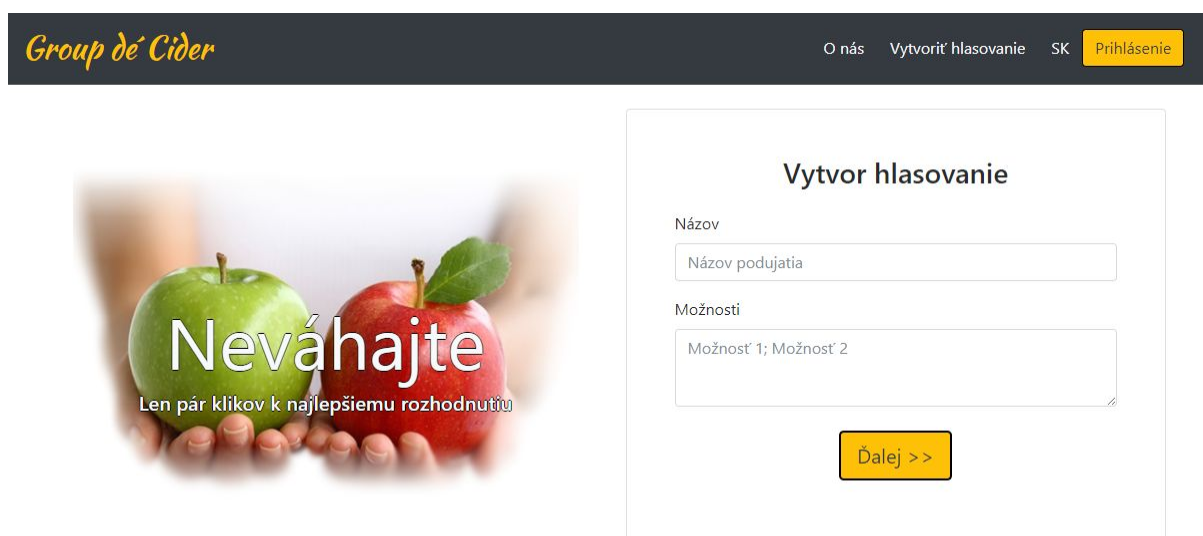


Obr. 12: Vytvorenie hlasovania - 3. obrazovka.

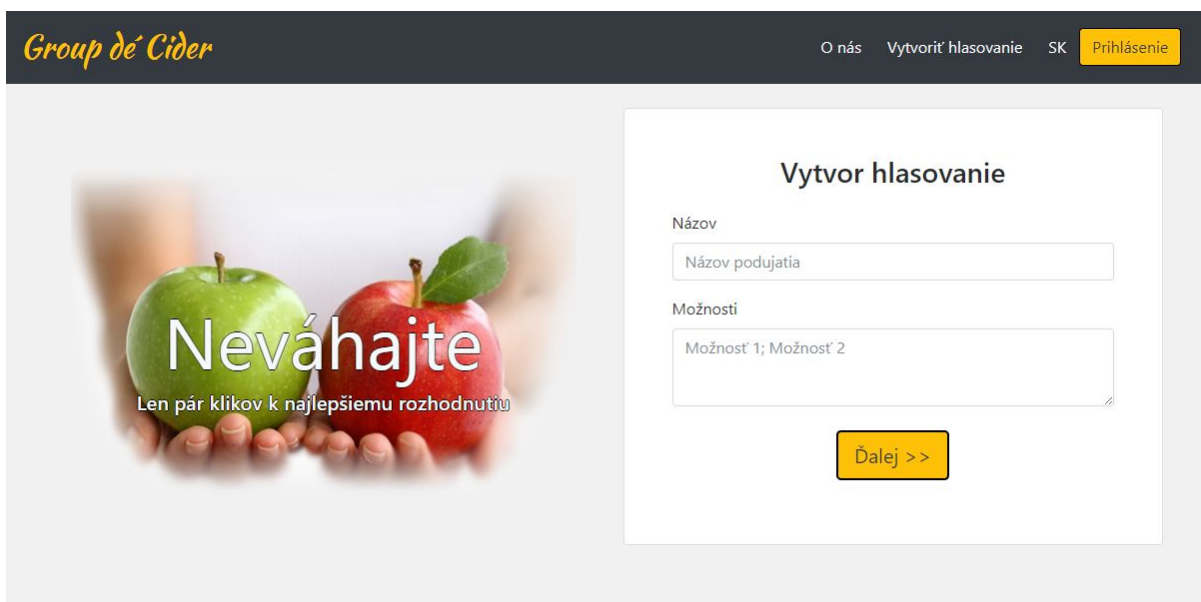


Obr. 13: Pridanie hlasu do hlasovania.

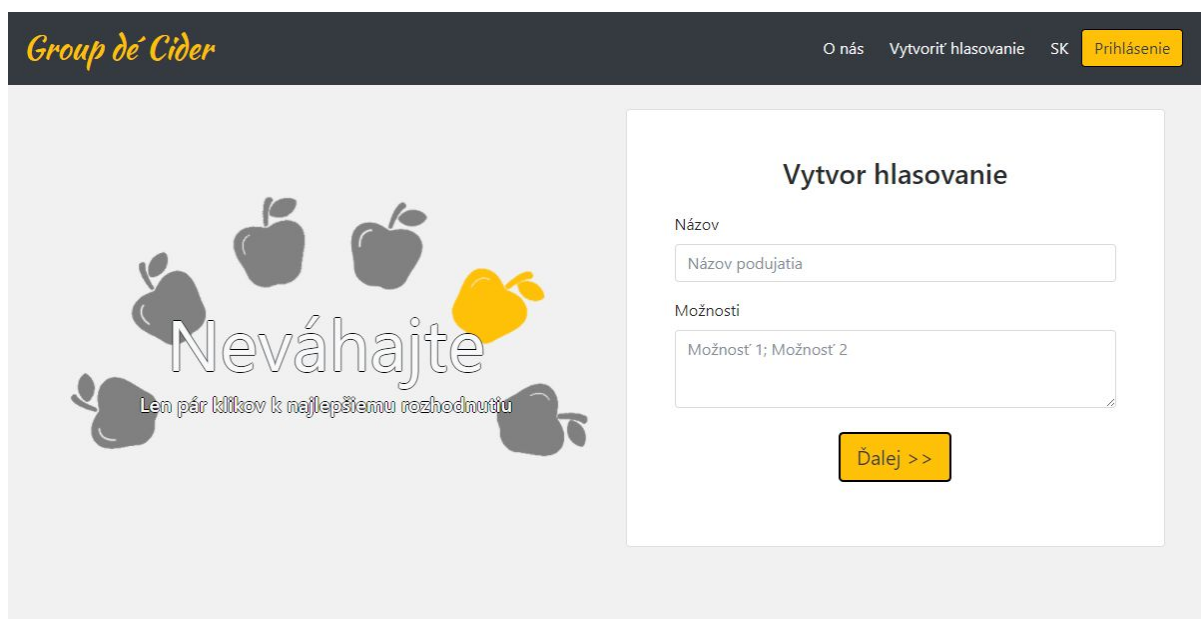
Druhá iterácia návrhu grafických rozhraní



Obr. 14: Mockup pre hlavnú obrazovku - verzia 1.



Obr. 15: Mockup pre hlavnú obrazovku - verzia 2.



Obr. 16: Mockup pre hlavnú obrazovku - verzia 3.

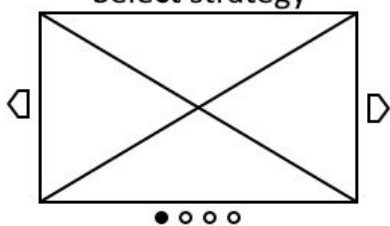
Tretia iterácia návrhu grafických rozhraní



Obr. 17: Mockup pre hlavnú obrazovku s vyberateľnou stratégiou.

MAKE BEST GROUP DECISIONS

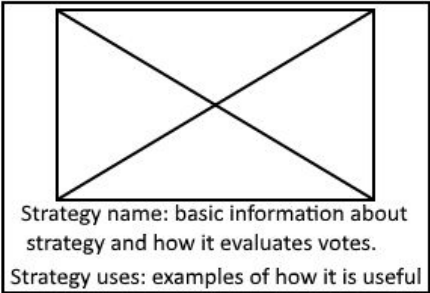
Select your preferred voting style

Select	Ranking	Points	Party
Create a poll Name Options		Select strategy  Strategy name: basic information about strategy and how it evaluates votes. Strategy uses: examples of how it is useful	

Obr. 18: Wireframe pre hlavnú obrazovku - verzia 1a.

MAKE BEST GROUP DECISIONS

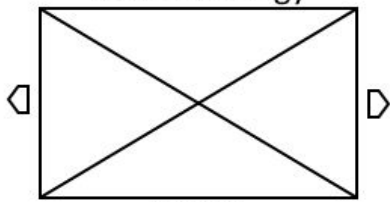
Select your preferred voting style

Select	Ranking	Points	Party
Create a poll Name Options		Select strategy  Strategy name: basic information about strategy and how it evaluates votes. Strategy uses: examples of how it is useful Create	

Obr. 19: Wireframe pre hlavnú obrazovku - verzia 1b.

MAKE BEST GROUP DECISIONS

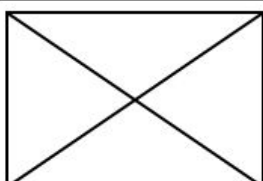
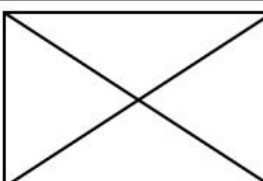
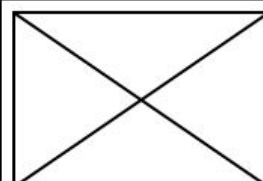
Select your preferred voting style

Select	Ranking	Points	Party
Create a poll Name Options Create		Select strategy  ● ○ ○ ○ Strategy name: basic information about strategy and how it evaluates votes. Strategy uses: examples of how it is useful	

Obr. 20: Wireframe pre hlavnú obrazovku - verzia 1c.

MAKE BEST GROUP DECISIONS

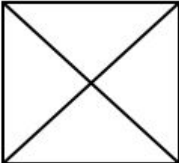
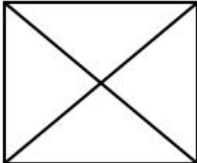
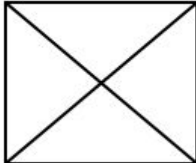
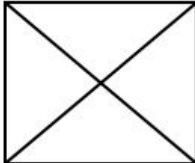
Select your preferred voting style

Select	Ranking	Points	Party
Select strategy			
 Single choice:	 Multi choice:	 Negative:	

Obr. 21: Wireframe pre hlavnú obrazovku - verzia 2.

MAKE BEST GROUP DECISIONS

Select your preferred voting style

Select	Ranking	Points	Party
			
Info about select strategies ...	Info about ranking strategies ...	Info about points strategies ...	Info about party strategies ...

Obr. 22: Wireframe pre hlavnú obrazovku - verzia 3.



Obr. 23: Mockup pre testovanie vzhľadu a opisov hlavnej obrazovky.