

# Electie

## Future of Voting

Tímový projekt 2021

Tím 17

Marek Ceľuch, Libor Duda, Lucia Janíková, Denis Klenovič, Timotej Králik, Adam Slatinský, Matúš Staš

Jaroslav Erdelyi

Elektronické riešenie prezenčných volieb. Cieľom je navrhnúť a implementovať elektronický systém, pomocou ktorého by bolo možné voliť vo volebných miestnostiach za použitia dotykovej obrazovky namiesto dnešného spôsobu - volebných hárkov.

# Big picture

## Cielené využitie

### Celoštátne voľby

Najkomplexnejším využitím nášho riešenia sú všetky druhy celoštátnych volieb, kedy je potrebné pokryť celú krajinu a aspoň po skončení volieb zrátať všetky výsledky na jednom centrálnom serveri. Kvôli tomuto je nutné zabezpečiť bezpečnú komunikáciu medzi volebnou miestnosťou a centrálnym serverom, aby nemohlo dôjsť k falšovaniu výsledkov volieb.

Medzi takéto celoštátne voľby patria najmä voľby do NRSR, voľby prezidenta SR, Voľby do Európarlamentu, komunálne voľby primátorov, starostov a poslancov mestského alebo obecného zastupiteľstva alebo zastupiteľstva mestskej časti. Okrem toho je ale bez problémov možné použiť Electie aj pre referendum.

### Lokálne hlasovania

Existujú ale aj prípady menších lokálnych volieb či už v rôznych organizáciách, domových zastupiteľstvách, univerzitách, fakultách a podobne. Dokonca, rovnako je možné použiť Electie aj pre rôzne prípady anonymných prieskumov, lokálnych referend alebo ankiet.

## Ako by mal vyzeráť volebný proces

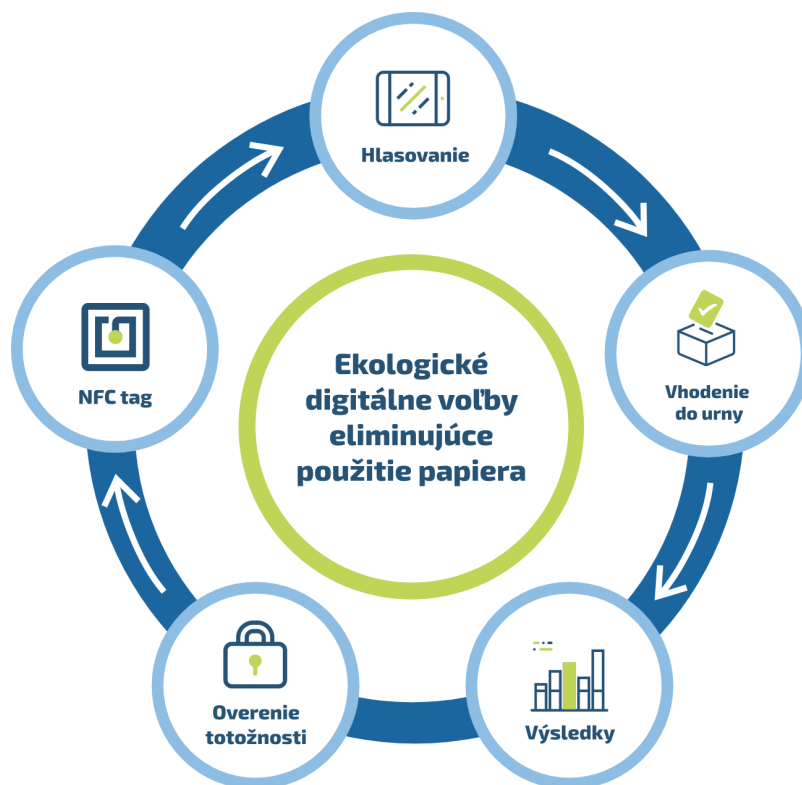
Volič príde klasicky do volebnej miestnosti a volebná komisia overí jeho totožnosť pomocou občianskeho preukazu a zoznamu voličov.

Volič si náhodne vyberie jeden z ponúkaných NFC tagov z misy. Tento NFC tag slúži pre autorizáciu nasledujúcej voľby pri volebnom termináli, aby nebolo možné hlasovať viackrát alebo bez povolenia. Na NFC tagu je tak nahratý jedinečný token, ktorý je možné v danej volebnej miestnosti použiť iba raz. Informácie uložené na tagu nie sú žiadnym spôsobom spájané s identitou voliča, čím je zaručená anonymita voľby.

Volič pristúpi k volebnému terminálu, do ktorého vloží NFC Tag, čím mu je umožnené odvoliť. Podľa pokynov na obrazovke vyberie a potvrdí svoju voľbu. Každý typ volieb má inú volebnú schému a preto aj konkrétny postup výberu voľby je vždy rozdielny.

Volebný terminál vytlačí potvrdenie o voľbe alebo volebný lístok. Vytlačený volebný lístok volič hodí do urny. Tým je umožnená kontrola hlasov v prípade problémov alebo sťažností. Oproti terajšiemu spôsobu voľby však stačí vytlačiť iba malý kúsok papiera so zoznamom zvolených kandidátov a nepotrebujeme mať predpripravené háčky pre každú stranu. Tým pádom je toto riešenie ekologickejšie.

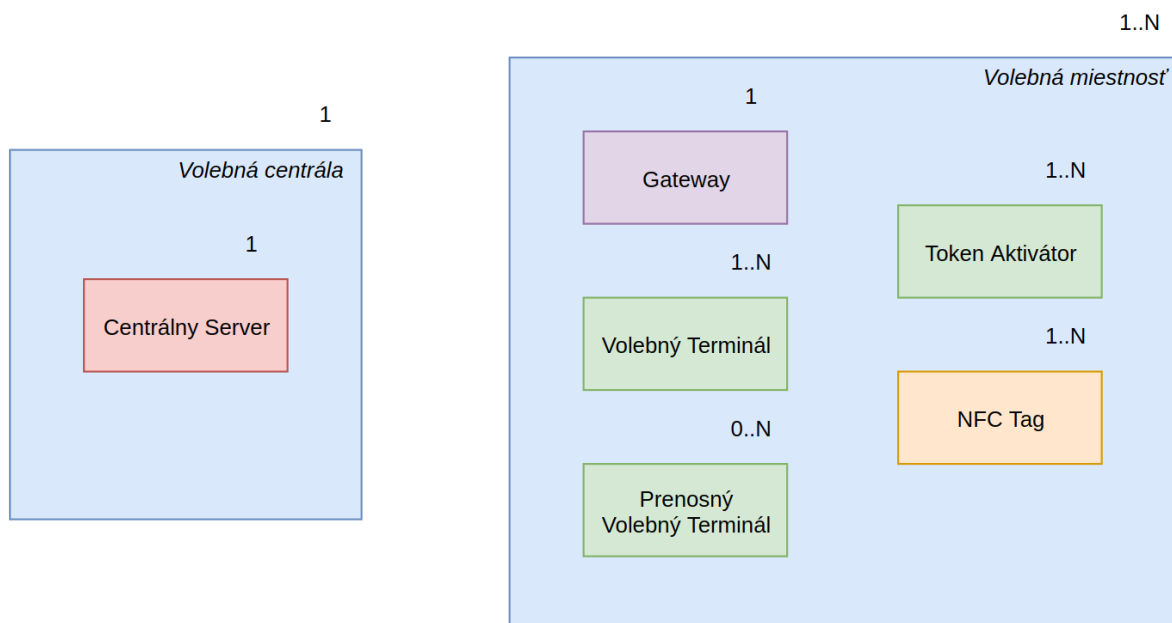
Po vhození potvrdenia do urny môže volič opustiť volebnú miestnosť. Výsledky môžu byť po uzavretí miestností dostupné vďaka nášmu riešeniu oveľa skôr ako býva zvykom pri manuálnom prepočítavaní.



Obrázok 1: Ilustrácia volebného procesu

## High-level architektúra

V rámci prvého šprintu sme si na začiatok navrhli takúto high-level architektúru aj s naznačenou kardinalitou komponentov:



Obrázok 2: High-level architektúra

Pri voľbách s viacerými volebnými miestnosťami existuje volebná centrála, kde sa nachádza centrálny server, ktorý dostáva výsledky zo všetkých volebných miestností.

Volebných miestností môže byť ľubovoľný počet. V každej z nich sa nachádza jeden Gateway, čo je zariadenie riadiace túto miestnosť - taký lokálny server. Stará sa najmä o generovanie a validovanie tokenov, spracovanie hlasov z volebných terminálov, generovanie zápisnice, komunikáciu s centrálnym serverom a podobne. Ďalej sa v miestnosti musí nachádzať aspoň jeden volebný terminál - zariadenie s dotykovou obrazovkou, na ktorom volič vykonáva voľbu. Token aktivátor je zariadenie starajúce sa o nahrávanie NFC tagov. Pokojne ale môže byť súčasťou Gatewayu. NFC tagy sú potrebné na prenos volebných tokenov, s ktorými je možné odoslať hlas z volebného terminálu na Gateway. Každý token je po spracovaní jeho hlasu deaktivovaný a už sa nedá použiť. Niektoré volebné miestnosti môžu obsahovať aj prenosné volebné terminály, čo sú zariadenia s dotykovou obrazovkou, na ktorých je možné voliť mimo volebnej miestnosti - funkcionality ako prenosná volebná urna.

Krajská volebná centrála je miesto, kde sa bude nahrávať konfigurácia na gateway pre volebné miestnosti. Krajské volebné centrály by sa nachádzali na úrade samosprávneho kraja a odtiaľto by sa rozposielali zariadenia do jednotlivých volebných miestností. Rovnako predpokladáme, že tu budú pracovať zaškolení administrátori z oblasti IT, ktorí by sa starali o nahrávanie konfigurácie.

# User stories

## 1. Nastavenie volebných údajov

Požiadavky:

- Vygenerovanie komunikačných kľúčov kvôli bezpečnej komunikácii *Gatewaya* so *Serverom*
- Administrátor
  - Nastavenie typu volieb
  - Nahratie kľúčov na zariadenia
  - Nastavenie zoznamu volebných miestností
  - Nastavenie údajov (zoznam strán, poslancov,...)
    - Zatiaľ predpokladáme konfiguračný súbor so zoznamom na vstupe, v prípade veľa času spravíme admin frontend
- Identifikovať jednotlivé kľúče ( ku vygenerovanému kľúču pár pre dešifrovanie konkrétnej miestnosti )
  - Z kľúča musí byť možné identifikovať volebnú miestnosť

## 2. Člen komisie overí voliča a dovoľí mu odvoliť

Požiadavky:

- Analýza, výber zariadenia na prácu s NFC tagmi
- Analýza, výber *Gateway* zariadenia
- Nastavenie *Gateway*
- Nastavenie zariadenia na čítanie a zapisovanie token na NFC Tagy
- Nabitie NFC Tagov, manažment
- Elektronické vyhľadávanie

## 3. Volič úspešne odvolí

Predpoklady:

- Generovanie kľúčov *Gatewayom*
- Vytvorenie a nastavenie *Gatewaya*
- Nabitie, manažment NFC Tagov

Požiadavky:

- Analýza potrebných zariadení, návrh siete
- Vytvorenie *Klientskej aplikácie*
- Vytvorenie *Volebného terminálu*: monitor, čítačka NFC, tlačiareň

Východisko:

- Volič má platný NFC Tag, chce voliť

Postup:

- Volič vloží NFC Tag do *Volebného terminálu*
- Volič vykliká svoju voľbu
- NFC Tag zostáva vo *Volebnom termináli*
- *Gateway* overí token

V prípade prázdneho NFC Tagu:

- Vypíše hlášku o probléme

Inak:

- Vymaže token z NFC Tagu

- Oznámi započítanie hlasu

#### 4. Volič úspešne odvolí doma

Predpoklady:

- Volenie vo volebnej miestnosti

Požiadavky:

- Analýza, výber zariadení (s ohľadom na výdrž batérie, veľkosť pamäti)
- Offline validácia kľúčov, ukladanie informácií a odosielanie hlasov
- Cache kľúčov na lokálne zariadenie (dodatkový pamäťový modul pre trvalé ukladanie kľúčov)

#### 5. Používateľ si zobrazí výsledky volieb

Požiadavky:

- Prijatie, spracovanie hlasu z *Gatewaya* na *Server*
- Automatické aktualizovanie výsledkov
- Zverejnenie výsledkov na webe, štatistiky, vizualizácie
- Zvládanie záťaže (generovať reporty pravidelne, negenerovať jednotlivo pre každý dopyt od klienta)
- UX testovanie

#### 6. Spustenie a ukončenie volebnej činnosti vo volebnej miestnosti

Požiadavky:

- Pri spustení volieb sa musia *Volebné terminály* synchronizovať s *Gatewayom* a automaticky budú voľby spustené v stanovený čas
- Automatické zneplatnenie všetkých kľúčov
- *Gateway* automaticky vygeneruje zápisnicu z volieb: Zoznam členov komisie, koľko ľudí volilo, výsledky
- Pokyn pre *Volebné terminály* (ukončenie činnosti)
- Obnova činnosti *Volebných terminálov* iba špeciálnym master kľúčom
- Offline scenár v prípade nedostupnosti internetu
  - Nahranie zápisnice cez inú volebnú miestnosť s prístupom na internet

# Analýza

Počas prvých 3 šprintov sme vykonali viacero analýz, ktoré nám mali slúžiť ako pevný podklad pre ďalšiu prácu na projekte. Začali sme analýzou [HW zariadení](#) aby sme našli optimálne zariadenia na kúpu do volebných miestností. Vzhľadom k tomu, že pri našom riešení bude spolu komunikovať veľa zariadení spoločne, bolo potrebné vykonať [analýzu sieťových možností](#). Časom sme sa potrebovali presunúť z týchto high level analýz do low level, aby sme mohli začať s implementáciou projektu. Vykonali sme teda [analýzu funkcionality a dátového modleu pre G](#). Obdobný typ analýzy sme spravili pre [server](#). Najväčším úskalím nášho projektu je bezpečná komunikácia a časť z nej sa analyzovala v [analýze zabezpečenia komunikácie medzi G a S](#). Vykonali sme aj menšiu analýzu nad možnými šifrovacími algoritmami.

# Návrh

Počas prvých troch šprintov sa nám podarilo navrhnuť kompletne používateľské rozhranie pre [volebnú aplikáciu](#). Vykonali sme aj návrh používateľského rozhrania na gateway pre administrátorov. Začali sme pomaly aj s implementačnými úlohami a tak sme vykonali návrh dátových modelov a softvérových architektúr pre [G](#) a [S](#).

# Riadenie projektu

## Organizácia

### Rozdelenie úloh

Všimli sme si, že k deleniu úloh v tíme pristupujeme rozdielne v porovnaní s inými tímami v tomto ročníku. Scrum master je u nás vždy niekto iný pre každý jeden šprint. Takto si každý člen tímu vyskúša túto úlohu a zodpovednosť a odnesie si tak z predmetu o to viac do reálneho života. Navrhol to náš vedúci a my sme nemali dôvod protestovať.

Je zjavné, že niektorí členovia tímu väčšinou pri rozdeľovaní taskov inklinujú k nejakej svojej obľúbenej oblasti, no fungujeme skôr ako startup a teda každý môže robiť v tíme všetko. Pri samotnom rozdeľovaní nových taskov na začiatku šprintu praktizujeme pull systém, teda každý člen tímu si vytiahne to, na čo má v tom šprinte chuť. Ak nastanú nejaké kolízie, tak sa už nejako dohodneme alebo to rozhodne nový scrum master.

### Stretnutia

Náš tím sa pravidelne stretáva s cieľom zhodnotenia postupu na jednotlivých úlohách, rovnako tu riešime aj inicializáciu a retrospektívu šprintu. Stretnutia s vedúcim sú v utorok o 11-tej a trvajú spravidla okolo 3 hodín. Na začiatku stretnutia sa určí zapisovateľ, ktorý vytvorí zápisnicu, ktorú po stretnutí zverejňujeme na webe. Zapisovatelia sa striedajú na každom stretnutí. Takýmto spôsobom si každý vyskúša túto prácu a zdokonalí si svoje schopnosti.

Do nástupu online vyučovanie sme sa stretávali aj na fakulte najčastejšie v knižnici aj mimo tohto času a preberali sme problémy priebežne.

### Používané nástroje

Pre komunikáciu, organizáciu úloh, trackovanie, logovanie, manažment kódu a tvorbu analýz a dokumentácie používame viacero nástrojov:

#### Projektový manažment - Jira

Pre manažment úloh a ich trackovanie a logovanie času sme si zvolili známu Jiru. Bol to jediný nástroj, ktorý sme z počutia poznali. Avšak, po troch šprintoch sme už prišli na to, že má pomerne dosť nedokonalostí, a nabudúce by sme si asi spravili lepší research a použili niečo iné.

#### Manažment kódu - GitHub

Zo začiatku sme používali BitBucket, ktorý patrí pod rovnakú spoločnosť ako Jira. Tam sa nám ani po nespočetnom úsilí a komunikácii s ich technickou podporou nepodarilo rozbehať workspace pre viac ako 5 ľudí, čo bol pre nás pochopiteľne značný problém.



Na GitHube máme založenú organizáciu [tp17-2021](#), kde máme organizované repozitáre s našimi kódmi. GitHub sme prepojili s Jirou, takže napríklad id tasku v commit message nám vie pekne automaticky spárovať commit s daným taskom.

## Manažment dokumentácie - G Suite

Všetky zápisnice, analýzy, návrhy, metodiky a iné dokumenty píšeme na G Suite. Má to jednoduché dôvody - všetci máme stuba G Suite účty, neobmedzený disk a vieme tam veľmi jednoducho a pohodlne spolupracovať na tvorbe dokumentov.

## Komunikácia - Messenger

Na komunikáciu v rámci tímu nepoužívame žiadny fancy nástroj ako Slack, MS Teams alebo Discord. Messenger má každý z nás a je zvyknutý ho používať na každom zariadení. Navyše, tím má iba 7 členov, čo je podľa nás ešte v norme pre komunikáciu cez jeden komunikačný kanál. Prácu si vždy rozdelíme na spoločných stretnutiach a pobavíme sa tam o rôznych veciach, ktoré má zmysel riešiť spolu.

Ak pracujeme na niečom vo dvojiciach, vieme komunikovať one to one tiež cez messenger. Ak sa niečo vyskytne aj pre 3 a viac ľudí, dobre nám poslúži už spomínaný G Suite s nejakým ad-hoc Google Meet hovorom.

## Ďalšie

### Figma

Používame na návrh používateľských rozhraní.

### Axure

Použili sme na prototypovanie používateľského rozhrania pre potreby používateľského testovania.

## Ciele

Naším hlavným a globálnym cieľom projektu je vytvoriť inovatívne riešenie poloautomatizovaných volieb. Aplikácia musí byť použiteľná pre každú vekovú skupinu. Nemenej podstatným cieľom projektu je zdokonaľiť svoje manažérske, komunikačné a programátorské schopnosti.

## Prvý míľnik

Cieľom prvého míľnika bolo navrhnutie a ujasnenie architektúry celého systému, aby sa v ďalšej fáze mohlo začať s implementáciou. Ďalším cieľom bol návrh a používateľské testovanie volebnej aplikácie, čo sa nám aj podarilo naplniť.

## Zimný semester

Cieľom po zimnom semestri by malo byť riešenie, ktoré by malo byť podporovať odvolenie a odoslanie hlasu na server cez všetky navrhnuté kroky, ktoré treba vykonať. Predpokladáme, že riešenie bude najprv podporovať iba jeden typ volieb a nasledujúcich fázach postupne doimplementujeme podporu pre ďalšie voľby.

## Letný semester

Cieľom v letnom semestri je dokončenie implementácie celej volebnej aplikácie a hlavne vykonať precízny testing aby sme si mohli byť istý správnosťou nášho softvéru.

## Zhodnotenie prvých troch šprintov

Po troch šprintoch sa nám podarilo navrhnuť a spojazdniť webovú stránku tímu, kde pravidelne aktualizujeme dokumenty. Spoločne sme si ujasnili návrh a architektúru celého systému od volebného terminálu až po server. Úspešne sme navrhli používateľské rozhranie pre volebný terminál a pre gateway. Pre používateľské rozhranie sme vykonali aj používateľské testovanie na 7 používateľoch z rôznych vekových skupín.

Z hľadiska riadenia sme mali zo začiatku problém s odhadovaním času jednotlivých úloh, čo sa postupom času zlepšovalo a odhady sa viac približovali realite. Komunikácia medzi členmi tímu stále trochu viazne. Spoločné dohadovanie a hľadanie konsenzu počas stretnutí nám ale ide veľmi dobre. V členoch tímu sa objavuje aj sebareflexia, pretože keď jeden člen tímu v jednom šprinte preukáže slabší výkon, v ďalšom sa nadpriemerne snaží. Spoločná práca nás baví a robí šťastnými.

# Zhodnotenie Šprintu 1

**Začiatok:** 6.10.2021

**Koniec:** 19.10.2021

**Scrum master:** Marek

## Cieľ šprintu

V prvom šprinte bolo cieľom vytvoriť webovú stránku tímu, analyzovať a vybrať potrebný hardvér pre náš projekt, navrhnuť high-level architektúru, zvoliť si technológie, navrhnuť prvú verziu používateľského rozhrania aplikácie na volebnom termináli a zaregistrovať náš tím na TP Cup. Ciele sa nám podarilo naplniť.

## Retrospektíva

Ešte pred šprintom sme si spravili high-level analýzu problému a načrtli architektúru riešenia. Na prvom stretnutí sme si určili niekoľko user stories a planning pokerom sme ich ohodnotili storypointami. Ďalej po stretnutí sme vygenerovali niekoľko desiatok taskov reflektujúcich tieto user stories. Do šprintu sme potom vybrali 9, pričom sa jednalo najmä o analýzy riešenia a tvorbu tímovej webovej stránky.

Na stredošprintovej ceremónii sme preberali prvý progres v taskoch. Začali sme zhurta a už v tomto momente boli začaté práce na analýze sieťových možností aplikácie, analýze Gatewayu, tvorbe tímovej stránky a analýze hardvéru pre volebný terminál. Trochu sme sa poradili o našich taskoch a v poslednej časti stretnutia sme s vedúcim preberali skôr organizačné veci okolo riadenia tímu.

Z diskusie vysvitlo, že scrum master sa bude meniť pre každý jeden šprint. Toto nám nariadil vedúci. Čosi bolo spomenuté aj ohľadom rozdelenie rolí v tíme, no nebolo to nijako záväzné a preto pokračujeme bez rozdelenia úloh.

Na stretnutí na konci šprintu sme okrem jednotlivých taskov riešili administratívu okolo dokumentácie šprintov. Nastavili sme nejaké veci v Jire a snažili sme sa z nej exportovať rôzne údaje. A teraz k jednotlivým členom a ich práci počas šprintu:

### Timo

- Analýza hardvéru pre VT - 3 tasky: (EV-11, EV-13, EV-26)

### Lucka

- Dokončenie prihlášky na TP Cup
  - podarilo sa ušetriť čas, kvôli recyklácii motivačného listu
- Návrh používateľského rozhrania
  - ešte to nie je úplne dokončené
- Review Timovi na hardvér

## Adam

- Práca s Matúšom na analýze FE + BE
- Review Marekovi - analýza sieťových možností

## Matúš

- Práca s Adamom na analýze FE + BE
- Práca na analýze G
- Review Timovi na hardvér

## Libor

- Práca s Denisom na webovej stránke
- Review Matúšovi na analýze G

## Denis

- Práca na webovej stránke
- Review Lucke - návrh klientskej aplikácie (identifikácia a doladovanie)

## Marek

- Práca na analýze sieťových možností
- Review Timovi na hardvér

## Zhodnotenie práce

Počas šprintu boli splnené všetky úlohy. Avšak, na návrh používateľského rozhrania pre klientsku aplikáciu (EV-59) bude potrebné nadviazať aj v ďalšom šprinte v rámci iteratívneho vyladovania dizajnu. Analýzy zariadení je potrebné ešte v krátkej dobe spojiť a s pomocou vedúceho doručiť správnym ľuďom na fakulte, aby bolo možné tento hardvér zaobstarat'. Analýzy sieťových možností, architektúry, komunikácie, technológií a funkcionality G tvoria dobrý základ pre ďalší návrh systému.

Počet taskov	9
Dokončené tasky	9
Nedokončené tasky	0
Celkový alokovaný čas na tasky	98h 30m
Celkový spotrebovaný čas na taskoch	73h 55m
Nezalogovaný čas (vysvetlené v problémoch)	40h
Celkový spotrebovaný čas	113h 55m

Odhady taskov sme viac prestrelili než podstrelili, takže sme nemuseli preťahovať a robiť zadarmo. Avšak, boli aj výnimky ako EV-32 tvorba tímovej stránky, kde boli asi o tretinu nad odhadom.

## Na aké problémy sme narazili

Na inicializačnom stretnutí nám dosť dlho trvalo spísať user stories a následne ich ohodnotiť. Ešte v ten istý deň (utorok) sme poobede a večer finalizovali nahadzovanie taskov do backlogu a tiež samotný výber niektorých taskov do šprintu 1. Na výbere sme sa zhodli a odsúhlasili ho až v stredu, takže aj náš šprint začal až vtedy. Teda sme začali o deň neskôr a šprint mal dokopy 13 dní.

Logovanie nám v prvej polovici šprintu veľmi nešlo, až sme si o tom trochu pokekali a vyjasnili si to na stredošprintovej ceremónii. Tu sme teda stratili niekoľko odpracovaných hodín.

Ďalším problém bolo to, že čas na spoločných stretnutiach sme si nikam nelogovali. Do ďalšieho šprintu sme si už dali nejaké generické tasky pre inicializáciu, retrospektívu a projektový manažment. Počas prvého šprintu sme sa stretli dokopy asi 6 krát. Či žu s vedúcim alebo bez neho. Časť tohto času sme venovali konkrétnym taskom a zalogovali sme ho, no väčšinu nie.

Dovoľme si odhadnúť, že sme takto odrobili ešte minimálne ďalších 40 hodín:

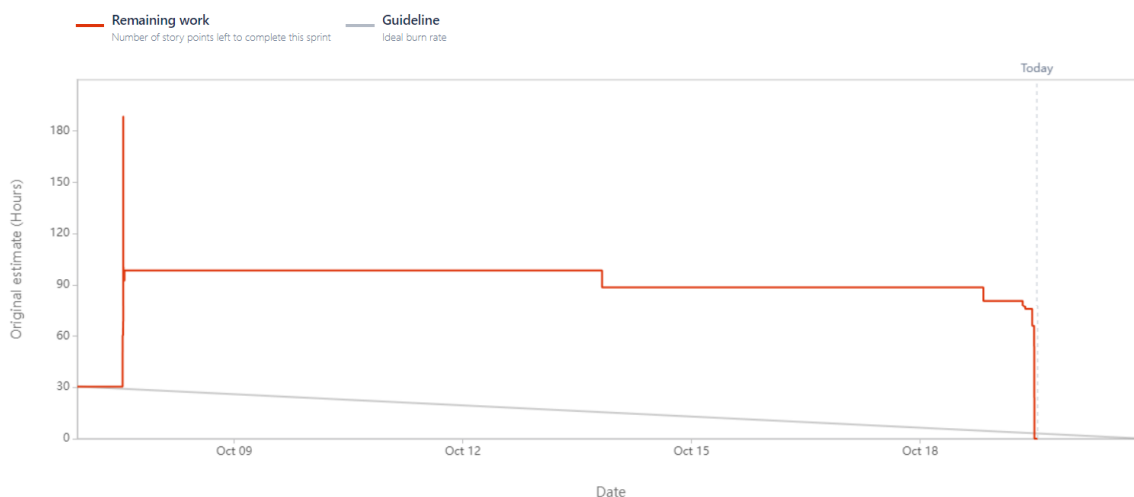
- Úvodná ceremónia =  $7 \times 3 = 21h$ .
- Poobedné vymýšľanie, hodnotenie a vyberanie taskov =  $7 \times 2 = 14h$
- Stretnutie niekedy v druhom týždni k výberu mena projektu =  $7 \times 1 = 7h$

Čo sa týka odhadov, ako už bolo spomenuté, väčšinou sme prestrelili (aspoň podľa worklogu). Práca sa natiahla v podstate iba pri tvorbe stránky. Pri tasku EV-12 sme najprv odhadli 6h, v strede šprintu sme to prehodnotili na 12h, no nakoniec to zabralo iba 7h. Boli sme ešte málo skúsení, čo sa týka odhadov, takže sa to dá pochopiť.

## Burndown chart

Date - October 6, 2021 - October 20, 2021

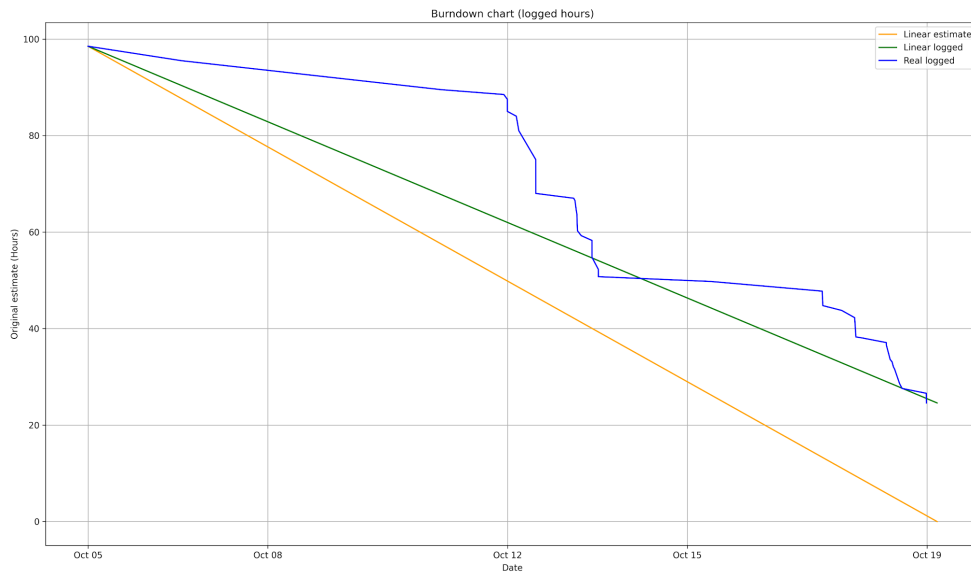
Sprint goal - Analýza hardvéru pre voľbu a vytvorenie stránky pre tímový projekt



*Burndown chart podľa ukončených úloh*

Tento graf je trochu rozbitý. Nepodarilo sa nám prísť na príčinu, ale vyzerá to tak, že na nejaký moment boli do šprintu zaradené ďalšie úlohy, ktoré ale boli následne aj naspäť odobraté. V grafe ale aj tak zostáva taký výrazný skok na začiatku. Ďalej tam v momente spustenia šprintu bolo menej taskov než by sme si prijali. Z toho dôvodu guideline začína na nesprávnej úrovni.

Ďalej vidno, že všetky úlohy boli označené ako DONE až v posledných hodinách pred ukončením šprintu, nakoľko sme si vytvárali väčšie úlohy, na ktorých sme síce mohli robiť aj priebežne, no tento graf zohľadňuje iba kompletne ukončenie úlohy..



*Burndown chart podľa zaznamenaného času*

Nakódili sme si vlastný skript, ktorý nám na základe worklogu z Jiry vygeneruje burndown chart podľa logovaných a teda odrobených hodín. Z takého grafu už potom máme informáciu o tom, ako veľmi priebežne sme počas šprintu pracovali, nejde iba o absolútne ukončenie tasku v posledný deň.

## Prehľad úloh

Task	Pridelená	Odhadovaný čas	Skutočný čas
EV-32 Vytvorenie webovej stránky	Denis	3d 6h	4d 5h 25m
EV-59 Návrh používateľského rozhrania pre klientsku aplikáciu	Lucia	3d	7h 30m
EV-12 Analýza sieťových možností	Marek	1d 4h	7h
EV-26 Analýza hardvéru pre prenosný VT	Timo	6h	2h 30m
EV-58 Analýza technológií pre tvorbu volebnej aplikácie	Matúš	4h	6h
EV-11 Analýza hardwaru pre VT	Timo	1d	7h
EV-40 Vybrať optimálne zariadenie pre Gateway	Matúš	1h 30m	1h 30m
EV-13 Analýza vhodnej tlačiarne	Timo	3h	2h
EV-34 Registrácia na TP cup	Lucia	1d 2h	3h
<b>Spolu</b>		12d 2h 30m	9d 1h 55m

## Podiel práce členov tímu

Marek	15%
Libor	15%
Lucia	15%
Denis	20%
Timo	15%
Adam	10%
Matúš	10%

**[EV-11] Analýza hardwaru pre VT**

Created: 05/Oct/21 5:56 PM - Updated: 19/Oct/21 10:12 AM - Resolved: 18/Oct/21 8:09 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Marek Ceľuch	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US03_Volič_úspešne_odvolí		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	1 hour		
<b>Time Spent:</b>	7 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
Nájdenie vhodného zariadenia pre zobrazenie - tablet, rásberičko k nemu. Čítačka NFC tagov. Mechazmus na zobratie tokenu .Musí to byť spolu kompatibilné

Work Log	User	Time Spent
12/Oct/21	Timotej Králik	2 hrs
12/Oct/21	Timotej Králik	0.5 hrs
13/Oct/21	Timotej Králik	1.5 hrs
18/Oct/21	Marek Ceľuch	1.5 hrs
18/Oct/21	Timotej Králik	1 hrs
19/Oct/21	Timotej Králik	0.5 hrs
<b>TOTAL</b>		<b>7 hrs</b>

<b>Comments</b>
<i>Timotej Králik added a comment - 12/Oct/21 10:28 AM</i>
Link na dokument s draftom analýzy → <a href="https://docs.google.com/document/d/1jN9cOhaVxncye3F-wAXc8ienQWCIW2XjDQIJ2xLN9f4/edit?usp=sharing">https://docs.google.com/document/d/1jN9cOhaVxncye3F-wAXc8ienQWCIW2XjDQIJ2xLN9f4/edit?usp=sharing</a>
<i>Timotej Králik added a comment - 13/Oct/21 4:36 PM</i>
Link na finalny dokument → <a href="https://docs.google.com/document/d/1SjzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing">https://docs.google.com/document/d/1SjzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing</a>
<i>Marek Ceľuch added a comment - 18/Oct/21 3:23 PM</i>
Pozrel som to a popísal som pripomienky ako komentáre do docu. Neber to ako osobný hate a opytovacie vety sú naozaj otázky, nie rečnicke otázky. Napr.: "Nevieme postaviť valstný case?" neznamená: "toto je zbytočné, postavme vlastný case", ale znamená: "Existuje ešte alternatíva, že postavíme vlastný case? prečo áno a prečo nie? bol by to príliš veľký problém?"



**[EV-12] Analýza sieťových možností**

Created: 05/Oct/21 5:57 PM - Updated: 19/Oct/21 11:54 AM - Resolved: 19/Oct/21 11:54 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Marek Ceľuch
<b>Reporter:</b>	Adam Slatinský	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US02_Člen_komisie_overí_voliča_a_dovolí_mu_odvolit'		
<b>Original Estimate:</b>	1 day, 4 hours		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	7 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
Ako bude vyzerat' systémová architektúra, ako budú pospájané zariadenia, ako budú komunikovat', čo bude čo čomu posielat'.

Work Log	User	Time Spent
06/Oct/21 Spísal som nejakú základnú architektúru zariadení a zdôvodnil som použitie RSA s fyzickou distribúciou kľúčov prostredníctvom zodpovedných osôb.	Marek Ceľuch	3 hrs
18/Oct/21 Pridal som pekné obrázky a ďalší obkek ku komunikácii a architektúre.	Marek Ceľuch	3 hrs
18/Oct/21 Gateway architecture	Marek Ceľuch	1 hrs
<b>TOTAL</b>		<b>7 hrs</b>

<b>Comments</b>
<i>Marek Celuch added a comment - 12/Oct/21 11:36 AM</i>
<a href="https://docs.google.com/document/d/1mbBp9S_tbYtsFgZrsbXCVbGv4GrB9nuQ4fCG7UOf7Pk/edit">https://docs.google.com/document/d/1mbBp9S_tbYtsFgZrsbXCVbGv4GrB9nuQ4fCG7UOf7Pk/edit</a>
<i>Marek Celuch added a comment - 19/Oct/21 12:29 AM</i>
Niečo som dal dokopy. Treba pozriet', či to dáva zmysel, či to tak môže byt', či sú tie reči o bezpečnosti pravdivé a či je to dobre navrhnuté.

**[EV-13] Analýza vhodnej tlačiarne**

Created: 05/Oct/21 5:57 PM - Updated: 19/Oct/21 8:32 AM - Resolved: 19/Oct/21 8:32 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Lucia Janíková	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US03_Volič_úspešne_odvolí		
<b>Original Estimate:</b>	3 hours		
<b>Remaining Estimate:</b>	1 hour		
<b>Time Spent:</b>	2 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
Najst tlačiaren ktora sa bude napajat cez USB, bezat s linuxom a bude tlačit male papiery

Work Log	User	Time Spent
12/Oct/21	Timotej Králik	1 hrs
13/Oct/21	Timotej Králik	1 hrs
<b>TOTAL</b>		<b>2 hrs</b>

<b>Comments</b>
<i>Timotej Králik added a comment - 12/Oct/21 10:28 AM</i>
Link na dokument s draftom analýzy → <a href="https://docs.google.com/document/d/1jN9cOhaVxncye3F-wAXc8ienQWCIW2XjDQIJ2xLN9f4/edit#">https://docs.google.com/document/d/1jN9cOhaVxncye3F-wAXc8ienQWCIW2XjDQIJ2xLN9f4/edit#</a>
<i>Timotej Králik added a comment - 13/Oct/21 4:37 PM</i>
Link na finalny dokument → <a href="https://docs.google.com/document/d/1SJzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing">https://docs.google.com/document/d/1SJzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing</a>

**[EV-26] Analýza hardvéru pre prenosný VT**

Created: 05/Oct/21 6:25 PM - Updated: 19/Oct/21 11:18 AM - Resolved: 19/Oct/21 11:18 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Matúš Staš	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US04_Volič_úspešne_odvolí_doma		
<b>Original Estimate:</b>	6 hours		
<b>Remaining Estimate:</b>	3 hours, 30 minutes		
<b>Time Spent:</b>	2 hours, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

Description
<ul style="list-style-type: none"> <li>• Výber vhodnej batérie s veľkou výdržou</li> <li>• Výber pamäte pre prenosný VT <ul style="list-style-type: none"> <li>• Treba externú pamäť pre ukladanie token, hlasov alebo nie?</li> </ul> </li> </ul>

Work Log	User	Time Spent
13/Oct/21	Timotej Králik	2 hrs
18/Oct/21	Timotej Králik	0.5 hrs
<b>TOTAL</b>		<b>2.5 hrs</b>

Comments
<p><i>Timotej Králik added a comment - 13/Oct/21 4:37 PM</i></p> <p>Link na finalny dokument → <a href="https://docs.google.com/document/d/1SjzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing">https://docs.google.com/document/d/1SjzWajr5JtCL5nTQ4ci33q0ihuZQsdENxWpbZ4PRCwQ/edit?usp=sharing</a></p>
<p><i>Marek Ceľuch added a comment - 18/Oct/21 8:05 PM</i></p> <p>Citujem z description tasku: "Treba externú pamäť pre ukladanie token, hlasov alebo nie?" V dokumente som sa nedozvedel odpoveď 😊</p>

**[EV-32] Vytvorenie webovej stránky**

Created: 06/Oct/21 11:53 AM - Updated: 19/Oct/21 12:08 PM - Resolved: 19/Oct/21 11:57 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Denis Klenovič
<b>Reporter:</b>	Marek Ceľuch	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	3 days, 6 hours		
<b>Remaining Estimate:</b>	2 days, 3 hours, 15 minutes		
<b>Time Spent:</b>	1 day, 2 hours, 45 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

Sub-Tasks	Type	Status	Assignee
[EV-56] Design vo figme	Vedľajšia úloha	Done	Denis Klenovič
[EV-57] Nakódenie HTML, SCSS, PHP	Vedľajšia úloha	Done	Adam Slatinský

Work Log	User	Time Spent
11/Oct/21	Libor Duda	6 hrs
13/Oct/21	Kontrola/Uprava textov pre web Lucia Janíková	0.75 hrs
18/Oct/21	Libor Duda	4 hrs
<b>TOTAL</b>		<b>10.75 hrs</b>

<b>Comments</b>
<i>Timotej Králik added a comment - 12/Oct/21 2:03 PM</i>
Link na záložku → <a href="https://drive.google.com/drive/u/2/folders/1SyLloRdDxVJLpyeF-00flmXW9Sle7DRK">https://drive.google.com/drive/u/2/folders/1SyLloRdDxVJLpyeF-00flmXW9Sle7DRK</a>

**[EV-34] Registrácia na TP cup**

Created: 06/Oct/21 11:53 AM - Updated: 17/Oct/21 4:46 PM - Resolved: 13/Oct/21 7:58 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Lucia Janíková
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	1
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day, 2 hours		
<b>Remaining Estimate:</b>	7 hours		
<b>Time Spent:</b>	3 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
Napísať a odovzdať prihlášku na TP Cup, ktorá prejde revíziou celého tímu.

<b>Links</b>
<b>Web Link</b>
<i>links to</i> Google Doc <a href="https://docs.google.com/document/d/1MOjkgC2MmiicX81gzY85PTO2q4XzLntmb6oyrpUIT8">https://docs.google.com/document/d/1MOjkgC2MmiicX81gzY85PTO2q4XzLntmb6oyrpUIT8</a>

Work Log	User	Time Spent
17/Oct/21 Vytvorenie prihlasky, upravy, odovzdanie	Lucia Janíková	3 hrs
<b>TOTAL</b>		<b>3 hrs</b>

<b>Comments</b>
<i>Marek Ceľuch added a comment - 12/Oct/21 8:20 PM</i>
Pridal som jednu čiarku a asi na dvoch miestach som dvoma slovami upravil text. Trochu som upravil formátovanie (medzery medzi odsekmi, zarovnanie na celú šírku v poslednom, zmazanie nejakých dvoch prázdnych riadkov a úprava medzier okolo nadpisov odsekov na prvej strane tak, aby sa ten posledný na prvej strane nerozdelil aj na druhú).
Inak to vyzerá fajn, za mňa už takto OK. Ešte môžeš <a href="#">Lucia Janíková</a> pozrieť, či ty si ok s mojimi úpravami.
<i>Lucia Janíková added a comment - 12/Oct/21 9:16 PM</i>
Super, vďaka <a href="#">Marek Ceľuch</a> ! Je pravda, že formátovanie som ešte neriešila, chcela som v prvom rade s vami prediskutovať text.

**[EV-40] Vybrať optimálne zariadenie pre Gateway**

Created: 06/Oct/21 12:00 PM - Updated: 19/Oct/21 9:06 AM - Resolved: 19/Oct/21 9:06 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Matúš Staš
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US02_Člen_komisie_overí_voliča_a_dovolí_mu_odvoliť		
<b>Original Estimate:</b>	1 hour, 30 minutes		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	1 hour, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
Zamysliť sa, či potrebuje mať nejaké vstupno-výstupné zariadenie, zväžiť zabezpečenie a zamysliť sa, čo všetko potrebuje G robiť.

Work Log	User	Time Spent
12/Oct/21	Matúš Staš	1 hrs
18/Oct/21	Matúš Staš	0.5 hrs
<b>TOTAL</b>		<b>1.5 hrs</b>

<b>Comments</b>
<i>Matúš Staš added a comment - 12/Oct/21 11:16 AM</i>
Link na dokument: <a href="https://docs.google.com/document/d/1lxc2JQX_VpZ3c2X5aGMKUJF3fxdyXIPVHF90ezW8E5U/edit?usp=sharing">https://docs.google.com/document/d/1lxc2JQX_VpZ3c2X5aGMKUJF3fxdyXIPVHF90ezW8E5U/edit?usp=sharing</a>

**[EV-58] Analýza technológií pre tvorbu volebnej aplikácie**

Created: 07/Oct/21 12:57 PM - Updated: 19/Oct/21 11:18 AM - Resolved: 19/Oct/21 11:18 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Matúš Staš
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US03_Volič_úspešne_odvolí		
<b>Original Estimate:</b>	4 hours		
<b>Remaining Estimate:</b>	1 hour		
<b>Time Spent:</b>	6 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 1

<b>Description</b>
FE a BE volebnej aplikácie pre VT

Work Log	User	Time Spent
12/Oct/21	Matúš Staš	1 hrs
13/Oct/21	Adam Slatinský	2 hrs
13/Oct/21	Matúš Staš	2 hrs
18/Oct/21	Adam Slatinský	0.5 hrs
19/Oct/21	Timotej Králik	0.5 hrs
<b>TOTAL</b>		<b>6 hrs</b>

<b>Comments</b>
<i>Matúš Staš added a comment - 12/Oct/21 2:27 PM</i>
Link na document <a href="https://docs.google.com/document/d/1e-99QH_mWjDnWEYGOER9Zh6qcvSGGiiI559d-ZKDyN8/edit?usp=sharing">https://docs.google.com/document/d/1e-99QH_mWjDnWEYGOER9Zh6qcvSGGiiI559d-ZKDyN8/edit?usp=sharing</a>

**[EV-59] Návrh používateľského rozhrania pre klientsku aplikáciu**

Created: 07/Oct/21 12:58 PM - Updated: 19/Oct/21 11:59 AM - Resolved: 19/Oct/21 11:59 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Lucia Janíková
<b>Reporter:</b>	Denis Klenovič	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	3 days		
<b>Remaining Estimate:</b>	2 days, 30 minutes		
<b>Time Spent:</b>	7 hours, 30 minutes		

Agile

**Sprint:** EV Sprint 1**Description**<https://www.figma.com/file/jaYSDFGuIUWnhrtEF7iX5H/Klient?node-id=0%3A1>

Work Log		User	Time Spent
13/Oct/21	Zaciatok tvorby navrhov vo figme	Lucia Janíková	1.5 hrs
15/Oct/21	Tvorba navrhov	Lucia Janíková	1 hrs
17/Oct/21	Tvorba navrhov	Lucia Janíková	2 hrs
17/Oct/21	Finalizacia prvotneho navrhov	Lucia Janíková	1 hrs
18/Oct/21	Varianty dizajnu	Lucia Janíková	1 hrs
19/Oct/21	Varianty dizajnu	Lucia Janíková	1 hrs
<b>TOTAL</b>			<b>7.5 hrs</b>

**Comments***Lucia Janíková added a comment - 18/Oct/21 12:32 AM*[Denis Klenovič](#)

Mas prosim nejaky navrh na upratanie/upravenie bieleho nadpisu hore? Uplne s tym nie som spokojna.

Premyslám ci k obrazovkam s priebehom volieb nedat nejaky nenapadny stepper, z ktoreho by bola jasnejsie v akej faze volby sa pouzivatel nachadza.

*Marek Ceľuch added a comment - 19/Oct/21 11:59 AM*

Ide iba o prvotný návrh. Jeto ok.



# Zhodnotenie Šprintu 2

**Začiatok:** 19.10.2021

**Koniec:** 02.11.2021

**Scrum master:** Libor

## Cieľ šprintu

Hlavnou náplňou práce počas Šprintu 2 je vytvoriť návrh dizajnu volebnej aplikácie na volebnom termináli. Rovnako vytvoriť návrh GUI pre gateway. Vytvoriť tiež metodiky na code style, definition of done, logovanie času, testovanie. Pokračovať a priebežne zapracovávať pripomienky k dizajnu webovej stránky tímu.

## Retrospektíva

### Libor

- Úprava a nahranie dokumentov na stránku
- Úprava webovej stránky - s Denisom
- Metodika testovania - s Denisom

### Denis

- Metodika testovania - s Liborom, ani polovica času nebola minutá
- Úprava webovej stránky - s Liborom
- Analýza zložitosti RSA a AES

### Marek

- Dizajn na stránke, návrhy na patterny do pozadia
- Metodika k Pythonu - nedokončené
- Manažment kódu - nedokončené
- Migrácia všetkého kódu z Bitbucketu na Github

### Timo

- Metodika logovania
- Návrh dizajnu volebnej aplikácie - s Luckou

### Lucka

- Návrh dizajnu volebnej aplikácie - s Timom

### Adam

- Návrh dizajnu potvrdenia voľby (vytlačeneho hlasovacieho lístka)

- Metodika code style JS

## Matúš

- Návrh používateľského rozhrania na gateway-i.

## Zhodnotenie práce

Počas šprintu sa splnili všetky úlohy s výnimkou dvoch metodík, ktoré sme sa dohodli spísať do konca týždňa. Najväčšiu časť práce v tomto šprinte zobralo prototypovanie vo Figma na čo boli pridelení dvaja ľudia. Do budúcnosti potrebujeme vedieť lepšie odhadovať úlohy, aby sa minimalizoval rozdiel akolkovaného a reálneho času.

Počet taskov	14
Dokončené tasky	12
Nedokončené tasky	2
Celkový alokovaný čas	142h
Celkový spotrebovaný čas	96h

Častejšie sme odhad taskov prestrelili ako podstrelili, čo je podľa nás ten lepší prípad.

## Čo by sme mali zlepšiť

- pracovať priebežne a nenechávať si úlohy na poslednú chvíľu
- zlepšiť časové odhady
- prídanie priebežnej vzájomnej kontroly progresu, motivovanie sa k práci na taskoch

## Na aké problémy sme narazili

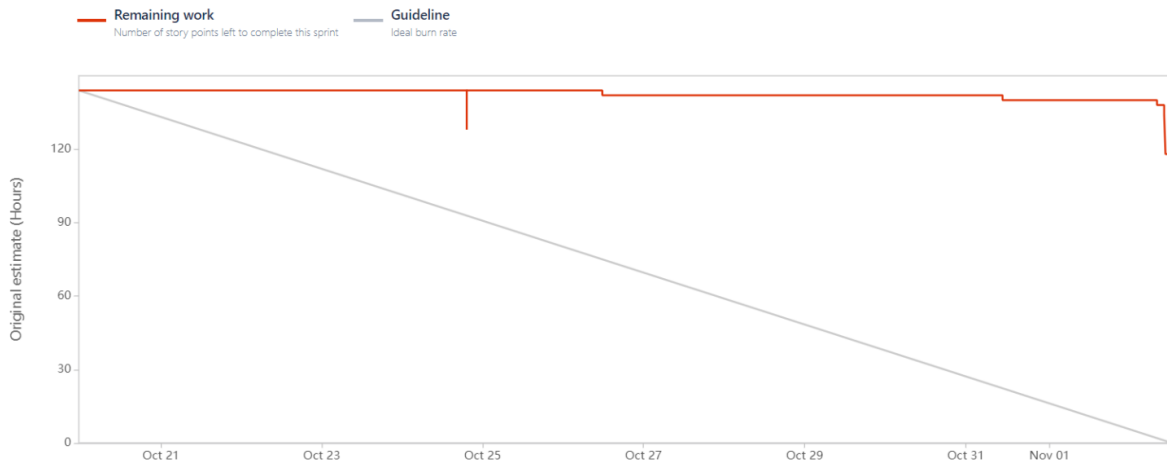
V tomto šprinte sme sa prvýkrát stretli s prípadom keď task nebol dokončený a potrebovali sme ho presunúť do ďalšieho šprintu. Nestihli sme dokončiť code style metodiku pre jazyk Python a metodiku manažmentu kódu. Tieto tasky v nasledujúcom šprinte čo najskôr dokončíme. Jeden task (Analýza zložitosti RSA v AES na raspberry) bol odstránený po zistení jeho nepotrebnosti.

Príliš veľa analýz a málo reálnych výsledkov práce na nás pôsobia ako keby sme nič nespravili, ale vieme že mať dobrú analýzu je základom.

Mali sme nerovnosti v spôsobe logovania času. Každý pristupoval k času na spoločných stretnutiach individuálne, preto sme to v metodike logovania času definovali a dohodli sa ako budeme riešiť spoločné stretnutia.

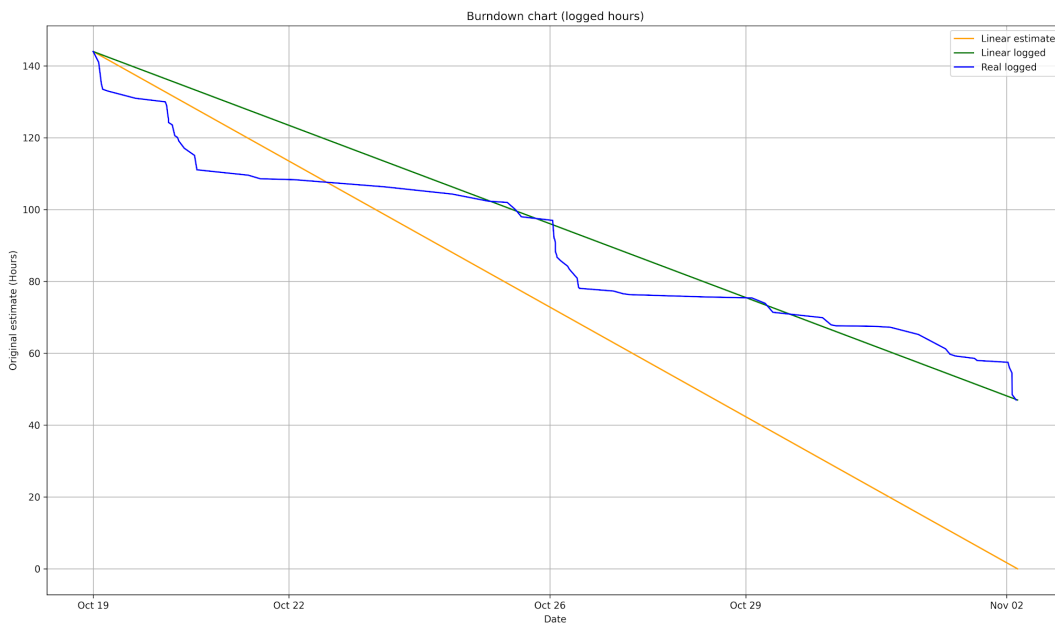
# Burndown chart

Date - October 19, 2021 - November 2, 2021



*Burndown chart podľa ukončených úloh*

Z diagramu vidno, že všetky úlohy boli označené ako DONE v posledných dňoch pred ukončením šprintu, nakoľko sme si vytvárali väčšie úlohy.



*Burndown chart podľa zaznamenaného času*

## Prehľad úloh

Task	Pridelená	Odhadovaný čas	Skutočný čas
EV-74 Inicializácia a retrospektíva	-	2d 5h	2d 3h 30m

EV-72 Finalizácia nákupu zariadení	Timo	2h	30m
EV-72 Metodika - Manažment kódu	Marek	1d	-
EV-70 Metodika Testovanie	Denis	3d	1d 2h 20m
EV-69 Metodika - Code style JavaScript	Adam	4h	3h 30m
EV-68 Metodika - Code style Python	Marek	4h	2h
EV-67 Metodika - Definition of done	Lucka	2h	50m
EV-66 Metodika - Logovanie času	Timo	2h	2h 50m
EV-65 Analýza zložitosti RSA na Raspberry	Denis	3h	1h 30m
EV-64 Projektový manažment	-	3d 6h	1d 5h 5m
EV-63 Úprava vizuálu webovej stránky	Libor	2d	7h 35m
EV-62 Dokončenie používateľského rozhrania volebnej aplikácie	Lucka	2d	3d 2h 25m
EV-61 Návrh používateľského rozhrania na G	Matúš	1d	1d 4h
EV-60 Analyzovanie a navrhnutie vytlačenej lístka	Adam	4h	1h 25m
<b>Spolu</b>		17d 6h	12d

## Podiel práce členov tímu

Marek	12%
Libor	13%
Lucia	18%
Denis	14%
Timo	15%
Adam	14%
Matúš	14%

**[EV-60] Analyzovanie a navrhnutie vytlačeného lístka**

Created: 19/Oct/21 12:02 PM - Updated: 02/Nov/21 12:20 PM - Resolved: 02/Nov/21 12:20 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Adam Slatinský
<b>Reporter:</b>	Matúš Staš	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US03_Volič_úspešne_odvolí		
<b>Original Estimate:</b>	4 hours		
<b>Remaining Estimate:</b>	2 hours, 35 minutes		
<b>Time Spent:</b>	1 hour, 25 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Vytvorenie návrhu papiera ktorý sa vytlačí a vhodí do urny. Čo bude na papieri všetko napísané, ako to bude vyzerat'. Fotky v messengeri.
<a href="https://www.figma.com/file/nZQvOkmYpGrOGFcbSynenw">https://www.figma.com/file/nZQvOkmYpGrOGFcbSynenw</a>

Work Log	User	Time Spent
27/Oct/21 Prvotný návrh vo figme pre 3 typy volieb + čo má obsahovať qr kód	Adam Slatinský	0.75 hrs
01/Nov/21 Skompaknenie dizajnu, vysvetlenie obsahu a generovania QR	Adam Slatinský	0.67 hrs
<b>TOTAL</b>		<b>1.42 hrs</b>

<b>Comments</b>
<i>Marek Celuch added a comment - 01/Nov/21 1:57 PM</i>
<ol style="list-style-type: none"> <li>Vieme nejako "zhustiť" tie lístky? Mám pocit, že 50% lístka je prázdne miesto. Čo ty na to, <a href="#">Adam Slatinský</a> ?</li> <li>Čím to budeme podpisovať? Private key toho konkrétneho VT, nejakým PK vytvoreným presne pre tento účel pre celú volebnú miestnosť alebo azda PK konkrétneho G?</li> </ol>
<i>Adam Slatinský added a comment - 01/Nov/21 10:42 PM</i>
<a href="#">Marek Celuch</a> Zhustil som dizajn lístkov a upresnil som, ako budú QR kódy podpísané. Keďže sa kľúč nachádza na tom istom gateway zariadení, tak nemá podľa mňa zmysel riešiť dva rôzne P kľúče. PK musí byť pre miestnosť unikátny - ak by PK unikol, nebude možné zrazu zmanipulovať hlasy na celom Slovensku

**[EV-61] Návrh používateľského rozhrania pre člena komisie na G**

Created: 19/Oct/21 12:43 PM - Updated: 02/Nov/21 11:43 AM

<b>Status:</b>	In Progress
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Matúš Staš
<b>Reporter:</b>	Lucia Janíková	<b>Votes:</b>	0
<b>Resolution:</b>	Unresolved		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	1 day, 1 hour		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Treba sa zamyslieť, čo všetko má toto rozhranie robiť. A treba rovno aj navrhnuť rozhranie.

Work Log	User	Time Spent
20/Oct/21	Matúš Staš	1 hrs
20/Oct/21	Matúš Staš	1 hrs
20/Oct/21 pomoc Matúšovi, návrh jednej obrazovky	Libor Duda	0.5 hrs
20/Oct/21	Matúš Staš	1 hrs
26/Oct/21 vyber návrhu a diskusia	Libor Duda	0.5 hrs
26/Oct/21	Matúš Staš	0.5 hrs
26/Oct/21 Konzultacia návrhov	Lucia Janíková	0.5 hrs
26/Oct/21 Pomahanie a hlasovanie s vizualom vo figme	Timotej Králik	0.5 hrs
26/Oct/21 menu, voľby, zápisnica, úvodná obrazovka	Matúš Staš	1.5 hrs
01/Nov/21 Pridanie ďalších funkcionalít + čiastočné spísanie	Matúš Staš	2 hrs
<b>TOTAL</b>		<b>9 hrs</b>

<b>Comments</b>
Matúš Staš added a comment - 20/Oct/21 3:32 PM
Figma: <a href="https://www.figma.com/file/6WI2kXw7wWTddVpFe7oEjq/Untitled?node-id=0%3A1">https://www.figma.com/file/6WI2kXw7wWTddVpFe7oEjq/Untitled?node-id=0%3A1</a>

**[EV-62] Dokončenie používateľského rozhrania volebnej aplikácie**

Created: 19/Oct/21 1:04 PM - Updated: 02/Nov/21 12:10 PM - Resolved: 02/Nov/21 12:10 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Lucia Janíková
<b>Reporter:</b>	Denis Klenovič	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	US03_Volič_úspešne_odvolí		
<b>Original Estimate:</b>	2 days		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	3 days, 2 hours, 25 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
<b>Link na hlavný dizajn:</b> <a href="https://www.figma.com/file/jaYSDFGuIUWnhrtEF7iX5H/Klient">https://www.figma.com/file/jaYSDFGuIUWnhrtEF7iX5H/Klient</a>
Link na zvsne varianty: <a href="https://www.figma.com/file/jAOZX8yeiQaoydhuwNj0Ka/Varianty-komponentov?node-id=0%3A1">https://www.figma.com/file/jAOZX8yeiQaoydhuwNj0Ka/Varianty-komponentov?node-id=0%3A1</a>
Link na IDSK dizajn: <a href="https://www.figma.com/file/2TOrqUYZuOyO4RliPpQg7T/IDSK-Dizajn-syst%C3%A9m?node-id=1%3A639">https://www.figma.com/file/2TOrqUYZuOyO4RliPpQg7T/IDSK-Dizajn-syst%C3%A9m?node-id=1%3A639</a>
Klikateny prototyp v Axure.
Spolupraca s <a href="#">Timotej Králik</a>
Libor 20.10. 1h zapísane

Work Log	User	Time Spent
19/Oct/21 Dorabanie pagingu a dokončovanie potvrdzovacích obrazoviek	Timotej Králik	2 hrs
20/Oct/21 Stretnutie, vyber dizajnu	Lucia Janíková	2 hrs
20/Oct/21 Podieľal som sa na konzultácii dizajnu rozhrania.	Marek Ceľuch	1 hrs
20/Oct/21 hlasovanie, vyjadrovanie názorov, malé úpravy	Denis Klenovič	0.67 hrs
20/Oct/21 Úprava dizajnu prezidentských volieb	Lucia Janíková	1.5 hrs
20/Oct/21 pomoc Lucke	Libor Duda	1 hrs
20/Oct/21 Dokončenie prezidentských volieb, Parlamentne voľby, Interakcie	Lucia Janíková	4 hrs
26/Oct/21 diskusia k návrhom	Libor Duda	0.5 hrs
26/Oct/21 pomoc s dizajnovými rozhodnutiami	Denis Klenovič	0.25 hrs
26/Oct/21 výber, úpravy a konzultácie dizajnov	Adam Slatinský	0.5 hrs
26/Oct/21 Konzultácia návrhu	Lucia Janíková	0.5 hrs
26/Oct/21 Zase sme spolu riešili veci ohľadom počtu strán, väčšieho miesta, atď.	Marek Ceľuch	0.5 hrs
26/Oct/21 Prerábanie dizajnu na pripomienky z cvičenia - cize strankovanie, posuvanie a menenie veľkosti.	Timotej Králik	2.5 hrs

Work Log		User	Time Spent
26/Oct/21	Upravy paging, rozloženie obrazoviek	Lucia Janíková	2.5 hrs
27/Oct/21	Prerobenie komunalnych volieb do noveho dizajnu (pridavanie fotiek, ...). Dokoncenie prezintnych volieb bez prechodov.	Timotej Králik	0.75 hrs
29/Oct/21	Figma + Axure plugin Prechod z Axure RP9 na Axure RP10 Prekopirovanie Figma frames do Axure	Lucia Janíková	2.5 hrs
30/Oct/21	Finalne upravy Figma dizajnu, odstranenie duplicitnych obrazoviek	Lucia Janíková	1.5 hrs
30/Oct/21	Uprava vizualu v axure aby tam neboli bugy a vyzeralo to ako vo figme a oprava chybiek vo figme	Timotej Králik	2 hrs
30/Oct/21	Finalizacia	Lucia Janíková	0.25 hrs
<b>TOTAL</b>			<b>26.42 hrs</b>

### Comments

*Timotej Králik added a comment - 20/Oct/21 12:02 PM*

Link → <https://www.figma.com/file/jaYSDFGulUWnhrtEF7iX5H/Klient?node-id=337%3A143>

*Lucia Janíková added a comment - 26/Oct/21 12:43 PM*

Pridať zoznam už zvolených poslancov + zväžiť search nad poslancami.



**[EV-63] Úprava vizuálu webovej stránky**

Created: 19/Oct/21 1:05 PM - Updated: 02/Nov/21 10:38 AM - Resolved: 02/Nov/21 10:38 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Libor Duda
<b>Reporter:</b>	Marek Ceľuch	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days		
<b>Remaining Estimate:</b>	1 day, 25 minutes		
<b>Time Spent:</b>	7 hours, 35 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
+ <a href="#">Denis Klenovič</a>

Work Log	User	Time Spent
20/Oct/21 Skúšanie úprav hero imagu a hľadanie svgčok	Denis Klenovič	1 hrs
20/Oct/21 drobne opravy na stranke, background images	Libor Duda	0.5 hrs
20/Oct/21 Riešili sme grafiky na stránku a nejaké ďalšie vizuálne drobnosti.	Marek Ceľuch	1 hrs
20/Oct/21 prerábanie obsahu, detailov a ilustrácií, úprava mobilného vzhľadu	Denis Klenovič	2 hrs
21/Oct/21 rovnaka vyska kachliciek	Libor Duda	1.5 hrs
21/Oct/21 Kontrola, riešenie problémov s rovnakou výškou, debugovanie	Denis Klenovič	1 hrs
26/Oct/21 uprava webpacku	Libor Duda	0.33 hrs
27/Oct/21 oprava pretekania na cardoch s gutterom	Libor Duda	0.25 hrs
<b>TOTAL</b>		<b>7.58 hrs</b>

<b>Comments</b>
<i>Marek Ceľuch added a comment - 31/Oct/21 9:45 PM</i>
Podľa mňa good enough for now. Už nemám žiadne nápady, čo by sme vylepšili. Vyzerá to fajn to aj na všelijakom zariadení.
Ale chceli sme vlastne 3 easter eggy. Tak na tomto môžeme ešte popracovať

**[EV-64] Projektový manažment**

Created: 19/Oct/21 1:06 PM - Updated: 02/Nov/21 12:32 PM - Resolved: 02/Nov/21 12:32 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Libor Duda
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	3 days, 6 hours		
<b>Remaining Estimate:</b>	2 days, 4 hours, 55 minutes		
<b>Time Spent:</b>	1 day, 2 hours, 5 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Spoločné stretnúť, kedy si nevieme čas zaradiť do iného tasku.

Work Log	User	Time Spent
23/Oct/21 Zistil som, že Bitbucket asi nikdy nebude fungovať. Vytvoril som github organization a premigroval som repozitár stránky a prepojil github s jirou. Nastavil som tam veci, pridal ľudí a hlavne rozbehal self-hosted runnera na našom serveri a nastavil pipu pre deployovanie stránky.	Marek Ceľuch	2 hrs
26/Oct/21 Utorková ceremónia uprostred šprintu. Povedli sme si, kto, čo robil.	Marek Ceľuch	1 hrs
26/Oct/21 Tímové stretnutie	Adam Slatinský	1 hrs
26/Oct/21 Tímové stretnutie	Matúš Staš	1 hrs
26/Oct/21 tímové stretnutie	Libor Duda	1 hrs
26/Oct/21 Tímové stretnutie	Lucia Janíková	1 hrs
26/Oct/21 Utorkové posedenie s veducim, prezentovanie doterajšej práce na projekte	Timotej Králik	1 hrs
26/Oct/21 standup progressu po týždni	Denis Klenovič	1 hrs
26/Oct/21 Úprava zápisnice č. 5	Adam Slatinský	0.83 hrs
31/Oct/21 zjednotenie vzhladu + stiahnutie a nahranie metodik	Libor Duda	0.25 hrs
<b>TOTAL</b>		<b>10.08 hrs</b>

**[EV-65] Analýza zložitosti RSA na Raspberry**

Created: 19/Oct/21 1:07 PM - Updated: 02/Nov/21 12:26 PM

<b>Status:</b>	Rejected
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Denis Klenovič
<b>Reporter:</b>	Marek Ceľuch	<b>Votes:</b>	0
<b>Resolution:</b>	Unresolved		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	3 hours		
<b>Remaining Estimate:</b>	1 hour, 30 minutes		
<b>Time Spent:</b>	1 hour, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Analyzovať AES vs RSA. Či by vôbec AES bolo bezpečné. Ale meníme kľúče poštou, takže možno. Treba sa zamyslieť a rozhodnúť. Zvážit' veľkosti RSA 2048 a 4096.

<b>Work Log</b>	<b>User</b>	<b>Time Spent</b>
29/Oct/21      vyskúšanie si DES, AES, RSA a zváženie časovej zložitosti	Denis Klenovič	1.5 hrs
<b>TOTAL</b>		<b>1.5 hrs</b>

**[EV-66] Metodika - Logovanie času**

Created: 19/Oct/21 1:09 PM - Updated: 02/Nov/21 8:06 AM - Resolved: 02/Nov/21 8:06 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Lucia Janíková	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 hours		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	2 hours, 50 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Ako logovať, kedy.
<a href="https://docs.google.com/document/d/1VL DYJfSfGheFMZ3WZCmsdqzEo4wclfHUakZRqIIJ57c/edit">https://docs.google.com/document/d/1VL DYJfSfGheFMZ3WZCmsdqzEo4wclfHUakZRqIIJ57c/edit</a>

Work Log	User	Time Spent
22/Oct/21    Review	Lucia Janíková	0.25 hrs
24/Oct/21    Hladal som dlho ako spravit ten report pre ukazanie casov v JIRE. zvyssnu hodinu som spisoval veci do dokumentu	Timotej Králik	2 hrs
25/Oct/21    Oprava komentorov od Lucky a formalizovanie docku	Timotej Králik	0.33 hrs
29/Oct/21    Review	Lucia Janíková	0.25 hrs
<b>TOTAL</b>		<b>2.83 hrs</b>

<b>Comments</b>
<i>Lucia Janíková added a comment - 25/Oct/21 12:50 PM</i>
<a href="#">Timotej Králik</a> , dala som ti nejake komentare k tej metodike. Obsahovo podla mna ta posledna cast dobre, ale cele mi to pride prilis neformalne napisane. Ja si to predstavujem ako napr. cast pracovnej zmluvy, ktoru ti daju v robote podpisat a mas tam napisane pravidla.

**[EV-67] Metodika - Definition of done**

Created: 19/Oct/21 1:09 PM - Updated: 31/Oct/21 10:02 AM - Resolved: 31/Oct/21 9:59 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Lucia Janíková
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 hours		
<b>Remaining Estimate:</b>	1 hour, 10 minutes		
<b>Time Spent:</b>	50 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
<a href="https://docs.google.com/document/d/19H5g7QsaGNsRJ7xXfe5G32jS09wU_zydA_aCsPTUU9c/edit#">https://docs.google.com/document/d/19H5g7QsaGNsRJ7xXfe5G32jS09wU_zydA_aCsPTUU9c/edit#</a>

<b>Work Log</b>	<b>User</b>	<b>Time Spent</b>
28/Oct/21    Formulacia definicie	Lucia Janíková	0.67 hrs
31/Oct/21    review	Libor Duda	0.17 hrs
<b>TOTAL</b>		<b>0.83 hrs</b>

**[EV-68] Metodika - Code style Python**

Created: 19/Oct/21 1:09 PM - Updated: 02/Nov/21 11:06 AM

<b>Status:</b>	In Progress
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Marek Ceľuch
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Unresolved		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	4 hours		
<b>Remaining Estimate:</b>	2 hours		
<b>Time Spent:</b>	2 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

Work Log		User	Time Spent
01/Nov/21	Študoval som PEP 8. Pozrel som niekoľko videí na YT a prečítal som niekoľko článkov o peknom a efektívnom Python kóde.	Marek Ceľuch	1.5 hrs
02/Nov/21	Niečo som dolepil do docu, ale je to bieda. Povedal som, akú verziu Pythonu použijeme a prečo a prilepil som link na PEP8.	Marek Ceľuch	0.5 hrs
<b>TOTAL</b>			<b>2 hrs</b>

<b>Comments</b>
<i>Marek Ceľuch added a comment - 02/Nov/21 11:06 AM</i>
Je to bieda. Pozeral som rôzne veci o peknom a efektívnom kóde, ale do docu som dal iba verziu pythonu a prečo a dohodil som link na PEP8. Bude treba ešte doplniť príklady formátu kódu a príklady na efektívny a pekný kód.

**[EV-69] Metodika - Code style JavaScript**

Created: 19/Oct/21 1:11 PM - Updated: 02/Nov/21 10:14 AM - Resolved: 02/Nov/21 10:14 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Adam Slatinsky
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	4 hours		
<b>Remaining Estimate:</b>	30 minutes		
<b>Time Spent:</b>	3 hours, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
<a href="https://docs.google.com/document/d/1rsGiR5ATPTcY9Yrcb9hblyYfMEKnod8fnXEs0PdpoEI/edit?usp=sharing">https://docs.google.com/document/d/1rsGiR5ATPTcY9Yrcb9hblyYfMEKnod8fnXEs0PdpoEI/edit?usp=sharing</a>

<b>Work Log</b>	<b>User</b>	<b>Time Spent</b>
20/Oct/21      Nájdenie a naštudovanie možných dostupných riešení	Adam Slatinsky	1 hrs
25/Oct/21      Spracované lintery a formátovače kódu, spracované highlights z mozilla guidelines.	Adam Slatinsky	2 hrs
01/Nov/21      review a drobna uprava	Libor Duda	0.5 hrs
<b>TOTAL</b>		<b>3.5 hrs</b>

**[EV-70] Metodika - Testovanie**

Created: 19/Oct/21 1:11 PM - Updated: 02/Nov/21 11:33 AM - Resolved: 02/Nov/21 11:33 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Denis Klenovič
<b>Reporter:</b>	Adam Slatinský	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	3 days		
<b>Remaining Estimate:</b>	1 day, 6 hours, 10 minutes		
<b>Time Spent:</b>	1 day, 2 hours, 20 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
+ <a href="#">Matúš Staš</a> <a href="#">Libor Duda</a> <a href="#">Adam Slatinský</a>
<a href="https://docs.google.com/document/d/1G9cK-TUklcGUAs7BsOkIRVyDDMHLjrdZmABepel-L6g/edit?usp=sharing">https://docs.google.com/document/d/1G9cK-TUklcGUAs7BsOkIRVyDDMHLjrdZmABepel-L6g/edit?usp=sharing</a>

Work Log		User	Time Spent
20/Oct/21	čítanie článku o unit testoch a spísanie hajlajtov	Denis Klenovič	0.75 hrs
25/Oct/21	pomoc Denisovi	Libor Duda	2 hrs
25/Oct/21	písanie metodiky testovanie, zameranie na feature testy a doplnanie unit testov	Denis Klenovič	2 hrs
26/Oct/21	pouzivatelske testovanie	Libor Duda	0.5 hrs
26/Oct/21	doplnenie regression testov a akceptacnych testov	Denis Klenovič	0.5 hrs
31/Oct/21	dokončovanie, kontrola pravopisu	Libor Duda	2 hrs
01/Nov/21	dopisovanie metodiky testovania - príklady na django a drawbacks of unit testing	Denis Klenovič	2 hrs
01/Nov/21	Doplnené informácie špecifické pre testovanie Svelte web aplikácie.	Adam Slatinský	0.58 hrs
<b>TOTAL</b>			<b>10.33 hrs</b>



**[EV-72] Metodika - Manažment kódu**

Created: 19/Oct/21 1:13 PM - Updated: 20/Oct/21 4:41 PM

**Status:** In Progress  
**Project:** e-volby

**Type:** Task  
**Reporter:** Matúš Staš  
**Resolution:** Unresolved  
**Labels:** None  
**Original Estimate:** 1 day  
**Remaining Estimate:** 1 day  
**Time Spent:** Not Specified

**Assignee:** Marek Ceľuch  
**Votes:** 0

*Agile***Sprint:** EV Sprint 2**Description**

GIT

Idem niečo robiť - ako začnem? ako vytvorím repositár? kde? ako ho nazvem? ako nazvem branche? ako budem commitovať? ako robiť PR? atď. (CI/CD, pipelines, scripts)

**[EV-73] Finalizácia nákupu zariadení**

Created: 19/Oct/21 1:14 PM - Updated: 26/Oct/21 11:43 AM - Resolved: 26/Oct/21 11:43 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 hours		
<b>Remaining Estimate:</b>	1 hour, 30 minutes		
<b>Time Spent:</b>	30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

<b>Description</b>
Komunikácia s Erdelyim ohľadom chcených zariadení a vyriešenie nákupu.

Work Log	User	Time Spent
19/Oct/21	Timotej Králik	0.5 hrs
<b>TOTAL</b>		<b>0.5 hrs</b>

**[EV-74] Inicializácia a retrospektíva**

Created: 19/Oct/21 1:29 PM - Updated: 02/Nov/21 12:33 PM - Resolved: 02/Nov/21 12:33 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Unassigned
<b>Reporter:</b>	Marek Ceľuch	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days, 5 hours		
<b>Remaining Estimate:</b>	1 day, 4 hours		
<b>Time Spent:</b>	2 days, 3 hours, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2

Work Log	User	Time Spent
19/Oct/21 Inicializacia sprintu 2	Timotej Králik	1.5 hrs
19/Oct/21 Inicializácia	Matúš Staš	1.5 hrs
19/Oct/21 Nahádzali sme tasky do Jiry a začali sme šprint na spoločnom stretku.	Marek Ceľuch	1.5 hrs
19/Oct/21 Sprint stretnutie	Lucia Janíková	1.5 hrs
19/Oct/21 Nahádzali sme tasky do Jiry a začali sme šprint na spoločnom stretku	Denis Klenovič	1.5 hrs
19/Oct/21 Inicializácia	Adam Slatinský	1.5 hrs
19/Oct/21 Inicializácia	Libor Duda	1.5 hrs
02/Nov/21 Retrospektíva	Denis Klenovič	1.5 hrs
02/Nov/21 Retrospektiva sprintu	Timotej Králik	1.5 hrs
02/Nov/21 Retrospektíva	Adam Slatinský	1.5 hrs
02/Nov/21 Retrospektíva	Marek Ceľuch	1.5 hrs
02/Nov/21 Retrospektíva	Matúš Staš	1.5 hrs
02/Nov/21 Ukoncenie sprintu, retrospektiva	Lucia Janíková	1.5 hrs
02/Nov/21 Retrospektíva	Libor Duda	1.5 hrs
<b>TOTAL</b>		<b>21 hrs</b>

# Zhodnotenie Šprintu 3

**Začiatok:** 02.11.2021

**Koniec:** 16.11.2021

**Scrum master:** Timo

## Cieľ šprintu

Hlavným cieľom šprintu 3 bolo vytvorenie plne funkčného prototypu na testovanie použiteľnosti našej volebnej aplikácie, kde môžu budúci voliči voliť. Nemenej podstatné množstvo práce sa vykonalo na low-level analýzach serveru a volebných gateway-ov. Tieto analýzy mali vytvoriť odrazový mostík pre prvé implementačné úlohy.

## Retrospektíva

### Timo

- vytvorenie prototypu v Axure
- na pomoc Denisovi nebol čas
- review Liborovi na EV-88
- review Matúšovi na use-caseoch pre gateway

### Marek

- dokončenie Metodika Code style (úloha z minula)
- dokončenie Metodika Manažment kódu (úloha z minula)
- analýza a dátový model pre G
- review Lucke - Príprava materiálov na testovanie
- review Timovi - Klikateľný prototyp na testovanie
- spracovanie burndown grafov pre zhodnotenia šprintov

### Denis

- návrh dátového modelu na serveri
- používateľské testovanie
- review Adamovi klientskú aplikáciu - bolo to v poriadku

### Matúš

- návrh GUI na G
- analýza návrhu use-casov na G (z minula)
- review Marekovi na Metodika manažment kódu

## Adam

- programovanie volebnej aplikácie vo Svelte
- pomoc Matúšovi

## Libor

- analýze zapisovania kľúčov na raspberry pred voľbami
- dve používateľské testovania
- zhodnotenia šprintu 2, šablóna, ktorá sa bude používať aj na ostatné zhodnotenia

## Lucka

- pomoc Matúšovi na GUI pre G
- práca na používateľskom testovaní - tvorba scenárov
- review Marekovi na dátový model pre G

## Zhodnotenie šprintu

Počas šprintu sme splnili všetky úlohy, ktoré sme si do šprintu zobrali a taktiež sme dohnali dvoj-úlohový dlh z predošlého šprintu. Najviac času sa spálilo na analýze use-casov na gateway a tvorbe obrazoviek pre jednotlivé use-casy. Veľká porcia času bola taktiež vynaložená na tvorbu klikateľného prototypu a úvodné používateľské testovanie. Sumárna časová kalkulácia vyzerá, že sme sa odhadom trafili presne do uskutočnenej reality, ale tento pohľad trochu klame, nakoľko boli úlohy, kde bol odhadovaný čas podstrelený o 8 hodín.

Počet taskov	11
Dokončené tasky	11
Nedokončené tasky	0
Celkový alokovaný čas	141h
Celkový spotrebovaný čas	142h

Celkové časové hodnotenie sa nám ale skoro rovná, pretože sme investovali viac času do projektového manažmentu ako sme plánovali.

## Čo by sme mali zlepšiť

- lepšie trackovať čas - stále sú jednotlivci ktorý počas ukončenia šprintu retrospektívne logujú čas
- viac komunikovať medzi sebou pomimo hlavných dvoch ceremónií počas šprintu
- robiť skôr reviews ako dva dni pred ukončením šprintu

## Na aké problémy sme narazili

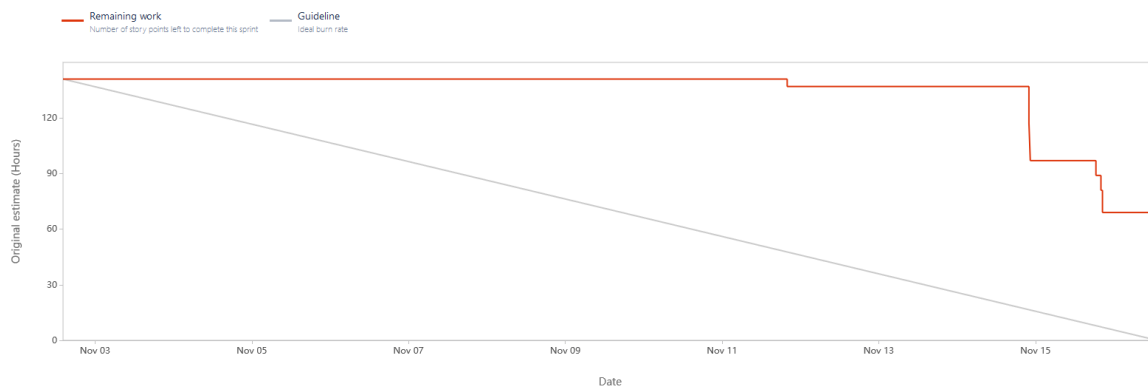
Pri prvotnom používateľskom testovaní sme narazili na problémy s naším návrhom aplikácie. Vzhľadom k tomu, že aplikáciu budú používať aj starí ľudia, ktorí dovtedy ešte nikdy tablet nepoužívali, musí byť volebný interface veľmi ľahko použiteľný. Ukázalo sa, že označovanie zvolených kandidátov a viditeľnosť search-baru sú nedostatočné.

Taktiež sme zistili, že sme doteraz úplne vynechali analýzu zápisnice, ako bude vyzeráť a čo bude obsahovať. Túto úlohu sme teda doplnili do backlogu.

Pri analýze dátových modelov sme vyvolali živú diskusiu na tému či použijeme NoSQL alebo obyčajnú relačnú databázu. Po celkovo hodinovej rozprave sme sa dohodli, že nakoniec použijeme NoSQL databázu.

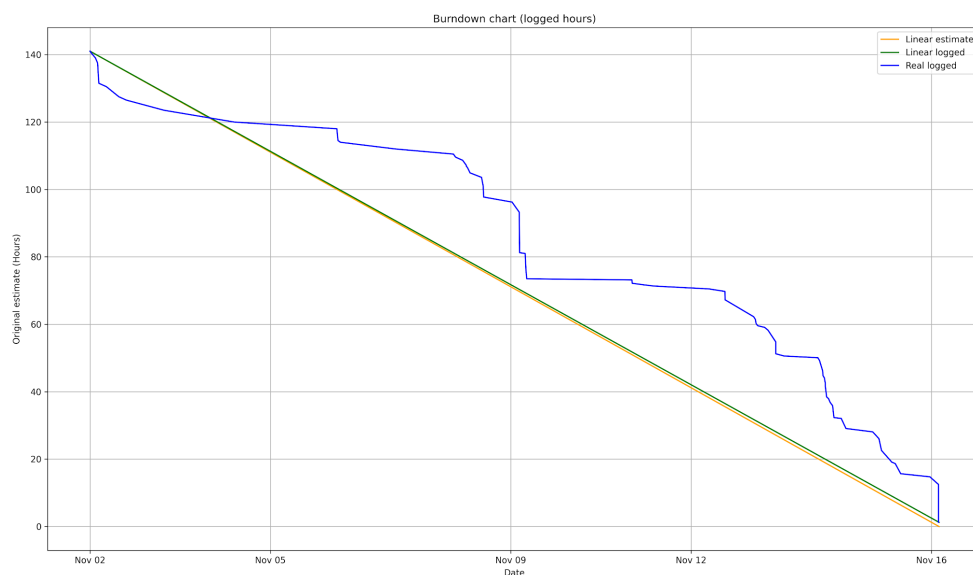
Z analýzy gateway-u vyplynulo, že použitie Django frameworku nemusí byť pre náš prípad efektívne, tak sme zvolili liberálnejšiu cestu pomocou frameworku FastAPI.

## Burndown chart



### *Burndown chart podľa ukončených úloh*

Z diagramu vidno, že všetky úlohy boli označené ako DONE v posledných dňoch pred ukončením šprintu, nakoľko sme si vytvárali väčšie úlohy.



*Burndown chart podľa zaznamenaného času*

Z diagramu vidno, že sme sa snažili na úlohách pracovať priebežne, ale stále vidno, že boli dni keď sa pracovalo viac a inokedy vôbec.

## Prehľad úloh

Task	Pridelená	Odhadovaný čas	Skutočný čas
EV-90 Inicializácia a Retrospektíva šprintu	Timotej Králik	2d 5h	3d 2h 15m
EV-89 Projektový manažment	Timotej Králik	2d	3d 2h 55m
EV-61 Návrh používateľského rozhrania pre člena komisie na G	Matúš Staš	1d	2d 3h
EV-78 Analýza návrhu funkcionality G	Matúš Staš	2d 4h	1d 5h 5m
EV-88 Zanalyzovanie spôsobu zabezpečenia komunikácie medzi G a S	Libor Duda	1d 4h	4h 30m

EV-68 Metodika - Code style Python	Marek Ceľuch	4h	4h 40m
EV-86 Analýza funkcionality S a vytvorenie dátového modelu	Denis Klenovič	2d	4h 30m
EV-81 Vytvorenie základu pre klientsku aplikáciu v Svelte	Adam Slatinský	1d	7h
EV-72 Metodika - Manažment kódu	Marek Ceľuch	1d	1d
EV-80 Klikateľný prototyp	Timotej Králik	2d 4h	2d 6h 20m
EV-85 Analýza a vytvorenie dátového modelu pre G	Marek Ceľuch	1d	7h
<b>Spolu</b>		17d 5h	17d 6h

## Podiel práce členov tímu

Marek	18%
Libor	12%
Lucia	16%
Denis	11%
Timo	15%
Adam	13%
Matúš	15%



**[EV-61] Návrh používateľského rozhrania pre člena komisie na G**

Created: 19/Oct/21 12:43 PM - Updated: 16/Nov/21 1:08 PM - Resolved: 16/Nov/21 1:08 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Matúš Staš
<b>Reporter:</b>	Lucia Janíková	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	2 days, 3 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2, EV Sprint 3

<b>Description</b>
Treba sa zamyslieť, čo všetko má toto rozhranie robiť. A treba rovno aj navrhnúť rozhranie.

Work Log	User	Time Spent
20/Oct/21	Matúš Staš	1 hrs
20/Oct/21	Matúš Staš	1 hrs
20/Oct/21 pomoc Matúšovi, návrh jednej obrazovky	Libor Duda	0.5 hrs
20/Oct/21	Matúš Staš	1 hrs
26/Oct/21 vyber návrhu a diskusia	Libor Duda	0.5 hrs
26/Oct/21	Matúš Staš	0.5 hrs
26/Oct/21 Konzultacia návrhov	Lucia Janíková	0.5 hrs
26/Oct/21 Pomahanie a hlasovanie s vizualom vo figme	Timotej Králik	0.5 hrs
26/Oct/21 menu, voľby, zápisnica, úvodná obrazovka	Matúš Staš	1.5 hrs
01/Nov/21 Pridanie ďalších funkcionalít + čiastočné spísanie	Matúš Staš	2 hrs
09/Nov/21 Finalizácia grafického rozhrania	Matúš Staš	1.5 hrs
09/Nov/21 Refaktorovanie	Matúš Staš	1.5 hrs
13/Nov/21 Finalizovanie návrhu	Matúš Staš	3.5 hrs
13/Nov/21 Finalizovanie návrhu, Review	Lucia Janíková	3.5 hrs
<b>TOTAL</b>		<b>19 hrs</b>

<b>Comments</b>
Matúš Staš added a comment - 20/Oct/21 3:32 PM
Figma: <a href="https://www.figma.com/file/6WI2kXw7wWTddVpFe7oEjq/Untitled?node-id=0%3A1">https://www.figma.com/file/6WI2kXw7wWTddVpFe7oEjq/Untitled?node-id=0%3A1</a>
Lucia Janíková added a comment - 08/Nov/21 12:18 PM
<a href="#">Matúš Staš</a> toto ako vyzerá? Spravíme najprv tu <input checked="" type="checkbox"/> <a href="#">EV-78</a> Done ?

**[EV-68] Metodika - Code style Python**

Created: 19/Oct/21 1:09 PM - Updated: 16/Nov/21 1:06 PM - Resolved: 11/Nov/21 7:57 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Marek Ceľuch
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	4 hours		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	4 hours, 40 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2, EV Sprint 3

<b>Description</b>
<a href="https://docs.google.com/document/d/1FbshShX7FzIBjDb86VSLIMPjBaMyNmcwOfZPyZjkVEU/edit#">https://docs.google.com/document/d/1FbshShX7FzIBjDb86VSLIMPjBaMyNmcwOfZPyZjkVEU/edit#</a>

Work Log		User	Time Spent
01/Nov/21	Študoval som PEP 8. Pozrel som niekoľko videí na YT a prečítal som niekoľko článkov o peknom a efektívnom Python kóde.	Marek Ceľuch	1.5 hrs
02/Nov/21	Niečo som dolepil do docu, ale je to bieda. Povedal som, akú verziu Pythonu použijeme a prečo a prilepil som link na PEP8.	Marek Ceľuch	0.5 hrs
08/Nov/21	Dal som dokopy výcucy z rôznych guidov. Ide o klasický PEP 8, typing, docstring a extensions do editora.	Marek Ceľuch	1.33 hrs
11/Nov/21	Zpracoval som nejaké Timove pripomienky, skrášľoval som formátovanie dokumentu a vyhodil som nie až tak podstatné bullet pointy.	Marek Ceľuch	0.5 hrs
11/Nov/21	Pridal som ešte príklady rozdelenia a odsadenia viacriadkových volaní.	Marek Ceľuch	0.33 hrs
13/Nov/21	review	Timotej Králik	0.5 hrs
<b>TOTAL</b>			<b>4.67 hrs</b>

<b>Comments</b>
<i>Marek Ceľuch added a comment - 02/Nov/21 11:06 AM</i>
Je to bieda. Pozeral som rôzne veci o peknom a efektívnom kóde, ale do docu som dal iba verziu pythonu a prečo a dohodil som link na PEP8. Bude treba ešte doplniť príklady formátu kódu a príklady na efektívny a pekný kód.
<i>Marek Ceľuch added a comment - 08/Nov/21 10:59 PM</i>
<a href="#">Timotej Králik</a> , môžeš

**[EV-72] Metodika - Manažment kódu**

Created: 19/Oct/21 1:13 PM - Updated: 16/Nov/21 9:52 AM - Resolved: 15/Nov/21 6:30 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Marek Ceľuch
<b>Reporter:</b>	Matúš Staš	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	1 day		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 2, EV Sprint 3

<b>Description</b>
GIT
Idem niečo robiť - ako začnem? ako vytvorím repositár? kde? ako ho nazvem? ako nazvem branche? ako budem commitovať? ako robiť PR? atď. (CI/CD, pipelines, scripts)

<b>Links</b>		
<b>Web Link</b>		
<i>links to</i>	Dokument metodiky manažmentu kódu	<a href="https://docs.google.com/document/d/1qTPMYQpXTLuRqPYvd1CnG7nzexiEnD3Z71LKq4NI1vE">https://docs.google.com/document/d/1qTPMYQpXTLuRqPYvd1CnG7nzexiEnD3Z71LKq4NI1vE</a>

Work Log		User	Time Spent
14/Nov/21	Spisoval som veci ohľadom vytvárania repositárov a popri tom som čerpal aj z diskusií a článkov na internete, ako je dobré deliť codebase.	Marek Ceľuch	2.5 hrs
14/Nov/21	Pridal som plno ďalších vecí. Ostáva ešte opísať PR a integráciu testov do GitHub actions.	Marek Ceľuch	1.17 hrs
14/Nov/21	Dopísal som ešte o pipelines, integrácii testov a štruktúre repositáru. Veľa času mi zabralo štúdium pipelines a testov.	Marek Ceľuch	3 hrs
15/Nov/21	Pridal som ešte veci ohľadom Microservices - každá služba má vlastný repositár a pri testovaní je potrebné v pipeline zahrnúť postavenie a spustenie imageu.	Marek Ceľuch	0.25 hrs
15/Nov/21	Report	Matúš Staš	0.17 hrs
16/Nov/21	Pozeral som na nete, aká je vhodná štruktúra codebase pri Microservices a ich deployovaní a testovaní. Pridal som tak ešte nejaké poznatky do dokumentu. Zhodou okolností tak logged work na tomto tasku vyšla presne na odhad.	Marek Ceľuch	0.92 hrs
<b>TOTAL</b>			<b>8 hrs</b>

**[EV-78] Analýza návrhu funkcionality G**

Created: 02/Nov/21 12:05 PM - Updated: 16/Nov/21 1:07 PM - Resolved: 14/Nov/21 10:24 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Matúš Staš
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days, 4 hours		
<b>Remaining Estimate:</b>	6 hours, 55 minutes		
<b>Time Spent:</b>	1 day, 5 hours, 5 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

<b>Description</b>
Čo všetko by mal robiť admin s G. Čo znamenajú jednotlivé UC v pozadí.
Spolupráca s <a href="#">Lucia Janíková</a>

Work Log		User	Time Spent
08/Nov/21	Vytvorenie prvého návrhu funkcionality G	Adam Slatinský	2.67 hrs
08/Nov/21	Vytvorenie prvého návrhu funkcionality G	Matúš Staš	2.67 hrs
08/Nov/21	Pokračovanie v analýze - stav volebných terminálov	Matúš Staš	0.5 hrs
13/Nov/21	Komunikácia o zmenách v GUI, analýza [ <a href="https://tp17-2021.atlassian.net/browse/EV-12">https://tp17-2021.atlassian.net/browse/EV-12</a> ] <a href="https://tp17-2021.atlassian.net/browse/EV-12">[smart-link]</a> pre zamedzenie duplicitnej práce	Adam Slatinský	0.83 hrs
13/Nov/21	Komunikácia o zmenách v GUI, analýza [ <a href="https://tp17-2021.atlassian.net/browse/EV-12">https://tp17-2021.atlassian.net/browse/EV-12</a> ] <a href="https://tp17-2021.atlassian.net/browse/EV-12">[smart-link]</a> pre zamedzenie duplicitnej práce	Matúš Staš	0.83 hrs
13/Nov/21	Analyzovanie UC+ rozmyšľanie nad pridaním diagramov ku každému UC	Matúš Staš	0.67 hrs
14/Nov/21	Spisovanie jednotlivých use case-ov	Matúš Staš	3.5 hrs
14/Nov/21	review	Timotej Králik	0.33 hrs
14/Nov/21	Rozpísanie UC	Adam Slatinský	1 hrs
14/Nov/21	Zpracovanie na zmenách od [ <a href="#">~accountid:615306da78e5e4007008fdb</a> ]	Matúš Staš	0.08 hrs
<b>TOTAL</b>			<b>13.08 hrs</b>

<b>Comments</b>
<i>Matúš Staš added a comment - 03/Nov/21 12:59 PM</i>
Link na dokument: <a href="https://docs.google.com/document/d/1v_bylyrrKeuE3KwIAxUo4FS4B6PTq5yWPcO8-vuXfCc/edit#heading=h.hgxoz0h60bpu">https://docs.google.com/document/d/1v_bylyrrKeuE3KwIAxUo4FS4B6PTq5yWPcO8-vuXfCc/edit#heading=h.hgxoz0h60bpu</a>

**[EV-80] Klikateľný prototyp**

Created: 02/Nov/21 1:09 PM - Updated: 15/Nov/21 8:29 PM - Resolved: 14/Nov/21 9:59 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days, 4 hours		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	2 days, 6 hours, 20 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

<b>Description</b>
taktiež analyzovanie ako nasadiť prototyp na testovanie pre tablety. A tiež scenáre.
Spolupráca s <a href="#">Lucia Janíková</a>

Sub-Tasks	Type	Status	Assignee
[EV-91] Príprava materiálov na používateľské testovanie	Vedľajšia úloha	Done	Lucia Janíková
[EV-92] Aktualizovať návrh dizajnu vo Figma	Vedľajšia úloha	Done	Lucia Janíková

Work Log	User	Time Spent
02/Nov/21 Príprava klikateľnosti prototypu	Lucia Janíková	1 hrs
03/Nov/21	Timotej Králik	3 hrs
04/Nov/21 Dokončenie klikateľnosti poslaneckej časti, taktiež hýbajúce sa číslo v pagingu.	Timotej Králik	3 hrs
06/Nov/21 Dokončenie klikateľnosti scenárov 3-4-5. a Zacatie prvých dvoch	Timotej Králik	2 hrs
07/Nov/21 Dokončenie celého prototypu, klikateľny searchbar. Spojazdnenie scenára v NR SR so spätnou opravou.	Timotej Králik	2 hrs
08/Nov/21 Prechod cez prototypy, kontrola so scenarmi, potrebné malé opravy	Lucia Janíková	1.5 hrs
08/Nov/21 Malé opravy pri zvolení 0 kandidátov, kontrola s Figma + vytvorenie EV-92	Lucia Janíková	0.33 hrs
13/Nov/21 Upravovanie bugov vo axure	Timotej Králik	5 hrs
14/Nov/21 Vykonanie používateľského testovania na 4 respondentoch	Denis Klenovič	3 hrs
15/Nov/21 používateľské testovania 2 ľudia + spisanie poznámok	Libor Duda	1.5 hrs
<b>TOTAL</b>		<b>22.33 hrs</b>

**[EV-81] Vytvorenie základu pre klientsku aplikáciu v Svelte**

Created: 02/Nov/21 1:11 PM - Updated: 16/Nov/21 10:59 AM - Resolved: 16/Nov/21 10:59 AM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Adam Slatinský
<b>Reporter:</b>	Denis Klenovič	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	1 hour		
<b>Time Spent:</b>	7 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

<b>Description</b>
<a href="https://github.com/tp17-2021/vt_frontend_aplikacia">https://github.com/tp17-2021/vt_frontend_aplikacia</a>
link na skid github s komponentami <a href="https://github.com/id-sk/id-sk-frontend">https://github.com/id-sk/id-sk-frontend</a>

<b>Work Log</b>		<b>User</b>	<b>Time Spent</b>
02/Nov/21	Extend svelte typescript šablóny, github prepojenie, tvorba časti stránok	Adam Slatinský	3 hrs
02/Nov/21	Možnosti dockerizácie build procesu	Adam Slatinský	1 hrs
15/Nov/21	pridaný modal komponent, router, api volanie, stores, bootstrap 5 je buildený z scss	Adam Slatinský	3 hrs
<b>TOTAL</b>			<b>7 hrs</b>

**[EV-85] Analýza a vytvorenie dátového modelu pre G**

Created: 02/Nov/21 1:28 PM - Updated: 15/Nov/21 8:02 PM - Resolved: 15/Nov/21 8:02 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Marek Ceľuch
<b>Reporter:</b>	Lucia Janíková	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day		
<b>Remaining Estimate:</b>	1 hour		
<b>Time Spent:</b>	7 hours		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

<b>Description</b>
Analýza a vytvorenie dátového modelu, dátových tokov pre G.

<b>Links</b>		
<b>Web Link</b>		
<i>links to</i>	Dokument Analýza a vytvorenie dátového modelu pre G	<a href="https://docs.google.com/document/d/1yjSIakf5VOGga-uhgr2zoGRBjzeuHr73GHS8mAZrZrM">https://docs.google.com/document/d/1yjSIakf5VOGga-uhgr2zoGRBjzeuHr73GHS8mAZrZrM</a>
<i>links to</i>	Priečinok na Drive	<a href="https://drive.google.com/drive/u/1/folders/1LM7aE6TLte8HgALZ6MXglmyb7uHprsuU">https://drive.google.com/drive/u/1/folders/1LM7aE6TLte8HgALZ6MXglmyb7uHprsuU</a>

Work Log	User	Time Spent
15/Nov/21 Dal som dokopy návrh architektúry a spravil osm rozhodnutie, že pôjdeme cestou Microservices. Podľa toho som vytvoril fancy obrázkov a do dokumentu som začal opisovať komponenty.	Marek Ceľuch	3.5 hrs
15/Nov/21 Dopísal som veci ku "knižniciam" a databázam. Ešte je potrebné popísať služby a možno aj špecifikovať rozhrania.	Marek Ceľuch	2 hrs
15/Nov/21 Dodal som popisy k jednotlivým službám, aby bolo možné vyzrozumieť, ktorá služba je za čo zodpovedná.	Marek Ceľuch	1 hrs
15/Nov/21 Review	Lucia Janíková	0.5 hrs
<b>TOTAL</b>		<b>7 hrs</b>

<b>Comments</b>
<i>Lucia Janíková added a comment - 14/Nov/21 4:14 PM</i> Ako to vyzera s tymto <a href="#">Marek Ceľuch</a> ?
<i>Marek Ceľuch added a comment - 14/Nov/21 4:17 PM</i> Čakal som, kedy sa ozveš teraz robím ešte iný task, večer po F1 idem už na toto. A zajtra je ešte celý deň, takže pohoda
<i>Lucia Janíková added a comment - 14/Nov/21 4:21 PM</i>

Super, som rada ze stihame!

*Marek Ceľuch added a comment - 15/Nov/21 12:25 AM*

Večer po F1 som ešte dokončoval <https://tp17-2021.atlassian.net/browse/EV-72>. Takže toto až zajtra - vlastne dnes.

*Marek Ceľuch added a comment - 15/Nov/21 6:25 PM*

[Lucia Janíková](#) , niečo som dal dokopy. Neviem, či názov tohto tasku úplne vystihuje to, čo som spravil. Možno išlo viac o návrh architektúry s tým, že presné rozhrania si špecifikujú už ľudia, čo na tých daných službách budú robiť.

Budem čakať na feedback

*Lucia Janíková added a comment - 15/Nov/21 7:56 PM*

[Marek Ceľuch](#) Super, veľmi pekne si to spracoval! Ocenujem ze si nespravil iba samotny navrh datovych modelov, ale riesil si rovno aj celu architekturu. Za mna DONE (prehod ty), je to pekny navrh a ako hovoris, upresni sa to uz ked sa zacne implementovat, ale mame teda slusny zaklad, z coho vychadzat!

Prosim ta, ak vznikli z tohto tasku nejake dalsie tasky (co treba doriesit, ako ten Portable VT), mohli by sme to asi nahadzat do Backlogu, nech na to myslime pri dalsom planovani.

*Marek Ceľuch added a comment - 15/Nov/21 8:02 PM*

Okay, vďaka. Vytvoril som na to task <https://tp17-2021.atlassian.net/browse/EV-93> . Ďalšie mi nenapadajú. Už potom v ďalšom šprinte azda pôjdeme kódovať, takže zajtra na ceremónii si povytvárame a porozdelujeme už tie kódovacie tasky.



**[EV-86] Analýza funkcionality S a vytvorenie dátového modelu**

Created: 02/Nov/21 1:30 PM - Updated: 16/Nov/21 12:03 PM - Resolved: 16/Nov/21 12:03 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Denis Klenovič
<b>Reporter:</b>	Libor Duda	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days		
<b>Remaining Estimate:</b>	1 day, 5 hours		
<b>Time Spent:</b>	4 hours, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

Work Log	User	Time Spent
08/Nov/21 analýza funkcií servera, kreslenie náhľadu, tvorenie fyzického modelu	Denis Klenovič	1.5 hrs
08/Nov/21 diskusia s Denisom a pomoc s návrhom dátového modelu	Libor Duda	1.5 hrs
15/Nov/21 zamýšľanie sa o použití MonogDB vs PostgreSQL	Denis Klenovič	1 hrs
15/Nov/21 pomoc Denisovi	Libor Duda	0.5 hrs
<b>TOTAL</b>		<b>4.5 hrs</b>

**[EV-88] Zanalyzovanie spôsobu zabezpečenia komunikácie medzi G a S**

Created: 02/Nov/21 1:55 PM - Updated: 16/Nov/21 1:07 PM - Resolved: 15/Nov/21 8:30 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Libor Duda
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	1 day, 4 hours		
<b>Remaining Estimate:</b>	7 hours, 30 minutes		
<b>Time Spent:</b>	4 hours, 30 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

**Description**

Zanalyzovať ako bude komunikovať S s G , napr. či sa kľúčové rozošlú fyzicky, mailom alebo úplne inak. Prípadne sa popýtať skúsených ľudí.

[https://docs.google.com/document/d/1HmBrdztbeDOKITDCIv4SvsngHaXo1ZTPHcBi2oT\\_Nsg/](https://docs.google.com/document/d/1HmBrdztbeDOKITDCIv4SvsngHaXo1ZTPHcBi2oT_Nsg/)

<b>Work Log</b>		<b>User</b>	<b>Time Spent</b>
06/Nov/21	spisanie zakladnych poznatkov	Libor Duda	0.5 hrs
08/Nov/21	debatovanie o možnostiach	Denis Klenovič	0.25 hrs
08/Nov/21	diskusia s Denisom k navrhom	Libor Duda	0.5 hrs
09/Nov/21	spisanie zaverov zo stretnutia	Libor Duda	0.25 hrs
13/Nov/21	zapracovavanie pripomienok	Libor Duda	0.5 hrs
14/Nov/21	review	Timotej Králik	0.5 hrs
14/Nov/21	Riešil som s Liborom nahrávanie konfigurácie pre G - že to bude v nejakej krajskej centrále automatizovane cez ethernetový kábel.	Marek Ceľuch	0.83 hrs
14/Nov/21	zapracovavanie pripomienok, podrobnejšie rozpisanie	Libor Duda	1 hrs
14/Nov/21	Konzultoval som s Liborom, aké obrázky by sa hodili. Neskôr som si ich pozrel, Marek Ceľuch dal feedback a potom zase pozrel narýchlo dokument.	Marek Ceľuch	0.17 hrs
<b>TOTAL</b>			<b>4.5 hrs</b>

**Comments**

*Marek Ceľuch added a comment - 11/Nov/21 11:24 AM*

Nie som reviewer, ale pozrel som si to a pridal som nejaké postrehy, ktoré by sme možno chceli ešte podrobnejšie ujasniť.

**[EV-89] Projektový manažment**

Created: 02/Nov/21 1:56 PM - Updated: 16/Nov/21 1:08 PM - Resolved: 16/Nov/21 1:08 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days		
<b>Remaining Estimate:</b>	0 minutes		
<b>Time Spent:</b>	3 days, 2 hours, 55 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

Work Log	User	Time Spent
09/Nov/21 Ceremonia v strede sprintu	Timotej Králik	3 hrs
09/Nov/21 Ceremónia v strede šprintu 3	Matúš Staš	3 hrs
09/Nov/21 Sprint call	Lucia Janíková	3 hrs
09/Nov/21 Utorková stredošprintová ceremónia.	Adam Slatinský	3 hrs
09/Nov/21 Ceremónia v strede šprintu 3	Denis Klenovič	3 hrs
09/Nov/21 spoločne stretnutie	Libor Duda	3 hrs
09/Nov/21 Bola utorková stredošprintová ceremónia.	Marek Ceľuch	3 hrs
12/Nov/21 spisovanie a zapracovanie pripomienok k zhodnoteniu sprintu 2	Libor Duda	0.83 hrs
12/Nov/21 Zbúchal som nejaký BASH + Python na vytvorenie burndown grafu podľa odpracovaných hodín z Jiry. A tiež som to použil na vygenerovanie grafu do zhrnutia šprintu 2.	Marek Ceľuch	2.5 hrs
13/Nov/21 Generoval som BD grafy a sumár zalogovaných hodín pre zhodnotenia šprintov. Vyžadovalo si to ešte nejaké zmeny kódu v mojom skripte.	Marek Ceľuch	0.83 hrs
14/Nov/21 pomoc so zhodnotením šprintu 2	Denis Klenovič	0.75 hrs
14/Nov/21 finalizácia zhodnotenia sprintu 2, nova tabulka taskov s casmi	Libor Duda	1 hrs
<b>TOTAL</b>		<b>26.92 hrs</b>

**[EV-90] Inicializácia a Retrospektíva šprintu**

Created: 02/Nov/21 1:56 PM - Updated: 16/Nov/21 1:19 PM - Resolved: 16/Nov/21 1:19 PM

<b>Status:</b>	Done
<b>Project:</b>	e-volby

<b>Type:</b>	Task	<b>Assignee:</b>	Timotej Králik
<b>Reporter:</b>	Timotej Králik	<b>Votes:</b>	0
<b>Resolution:</b>	Done		
<b>Labels:</b>	None		
<b>Original Estimate:</b>	2 days, 5 hours		
<b>Remaining Estimate:</b>	1 day, 5 hours, 30 minutes		
<b>Time Spent:</b>	3 days, 2 hours, 15 minutes		

<i>Agile</i>	
<b>Sprint:</b>	EV Sprint 3

Work Log	User	Time Spent	
01/Nov/21	Timotej Králik	1.5 hrs	
02/Nov/21	Inicializácia	Libor Duda	1.5 hrs
02/Nov/21	Inicializácia šprintu.	Denis Klenovič	1.5 hrs
02/Nov/21	Inicializacia	Lucia Janíková	1.5 hrs
02/Nov/21	Inicializácia šprintu.	Marek Ceľuch	1.5 hrs
02/Nov/21	init	Adam Slatinský	1.5 hrs
02/Nov/21	Inicialiacia	Matúš Staš	1.5 hrs
16/Nov/21	Retrospektiva	Matúš Staš	2.25 hrs
16/Nov/21	Retrospektiva	Timotej Králik	2.25 hrs
16/Nov/21	Retrospektíva	Denis Klenovič	2.25 hrs
16/Nov/21	Retrospektiva	Lucia Janíková	2.25 hrs
16/Nov/21	Retrospektíva	Adam Slatinský	2.25 hrs
16/Nov/21	Retrospektíva	Marek Ceľuch	2.25 hrs
16/Nov/21	Retrospektíva	Libor Duda	2.25 hrs
<b>TOTAL</b>		<b>26.25 hrs</b>	

# Metodika - Code style JavaScript

## ESLint - TypeScript/ JavaScript / svelte linter

ESLint vyhledá najčastejšie chyby v Javascripte automaticky a vyznačí vo vývojovom prostredí. Tiež poskytuje pre niektoré problémy automatické opravy.

<https://github.com/eslint/eslint>

Analýzou kódu nájde chyby ako

- Vytvorená premenná, ktorá nikdy nie je použitá
- Zabudnutý console.log z debugovania
- Nepoužité camelCase pomenovanie premenných
- Implicitné globálne premenné (zabudnuté let / const)
- ...

ESLint - 20k hviezdíčiek na githube, 17 mil stiahnutí za týždeň na npm

Cez inicializáciu (eslint --init) sa dá nakonfigurovať. Napríklad:

- aká verzia syntaxu sa používa (ES6)
- ES6 moduly alebo commonJS (node),
- spaces vs tabs
- typ ukončenia riadku (\n vs \r\n)
- Používať bodkočiarky na konci riadka alebo nie?

Ale **ESLint** plugin pre **Svelte** framework **nemá podporu TypeScriptu**

<https://github.com/sveltejs/eslint-plugin-svelte3>

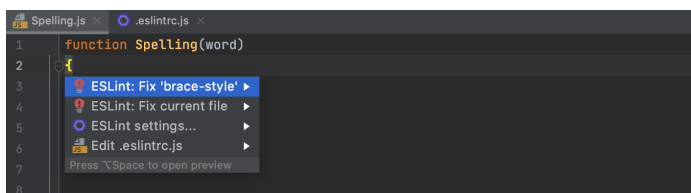
Typescript ESLint:

<https://github.com/typescript-eslint/typescript-eslint>

ESLint podpora pre IntelliJ

[https://www.jetbrains.com/help/webstorm/eslint.html#ws\\_js\\_linters\\_eslint\\_before\\_you\\_start](https://www.jetbrains.com/help/webstorm/eslint.html#ws_js_linters_eslint_before_you_start)

Je možné nastaviť, aby opravil problémy automaticky po uložení.



Prettier - Alternatíva k formátovaniu kódu cez ESLint

Deaktivovaním formátovania cez ESLint a nech to ide cez Prettier

<https://github.com/prettier/eslint-plugin-prettier>

# JS - Mozilla guide

Mozilla má dobrý guide:

[https://developer.mozilla.org/en-US/docs/MDN/Guidelines/Code\\_guidelines/JavaScript](https://developer.mozilla.org/en-US/docs/MDN/Guidelines/Code_guidelines/JavaScript)

## Highlights z Mozilla guideline

Na jeden riadok písať vždy len jeden príkaz, nepoužívať “jednoriadkovce”

### Správne

```
function myFunc() {  
  console.log('Hello!');  
};
```

### Nesprávne

```
function myFunc() { console.log('Hello!'); };
```

Používať **medzery medzi operátormi a parametrami**

### Správne

```
if(dayOfWeek === 7 && weather === 'sunny') {  
  goOnTrip('beach', 'car', ['ice cream', 'bucket and spade',  
  'beach towel']);  
}
```

### Nesprávne

```
if(dayOfWeek === 7 && weather === 'sunny') {  
  goOnTrip('beach', 'car', ['ice cream', 'bucket and spade',  
  'beach towel']);  
}
```

Nepoužívať `var`, ale **let** / **const**.

Názov **premených** a **funkcií** štýlom **camelCase**. Názvy **tried** štýlom **PascalCase**

Zrozumiteľne nazývať premenné, nie “a”, “b”, ...

```
let playerScore = 0;
```

```
function sayHello() {
```

```
class Person {
```

Používať lepšie pochopiteľnejšie **porovnanie s porovnaním typov**

```
if(name === 'Chris')  
if(age !== 25)
```

Na vkladanie premenných do reťazcov používať **string literals**

```
console.log(`Hi! I'm ${myName}!`);
```

**Používať moderný JS (ES6) syntax** (arrow funkcie, literals pre tvorbu objektov a polí)

```
let sum = array1.reduce((a, b) =>
```

```
let myObject = { };
```

```
let myArray = [];
```

# Metodika - Code style Python

Užitočné linky:

- [Skrátený výcuc z používaného Python codestylu](#)
- [Typovanie v Pythone](#)
- [Docstring v Pythone](#)

## Verzia Pythonu

Projekt používa Python3. Konkrétne **3.10**. Nie je to ešte úplne stable verzia - napríklad sa stáva v Python 3.10 shelli, že nefunguje história, ale to nie je problém, keďže v tomto projekte nie je potrebné riešiť interaktívne veci. 3.10 so sebou ale prináša [ďalšie cool featurky](#), hlavne:

- Výpovednejšie Error Message
- Nové featurky pre Type Hinting
- Structural Pattern Matching

## Code style - čoho sa držať

Oficiálny code style Pythonu je [PEP 8](#). Problém je, že je príliš dlhý a čítanie by človeku zabralo aj 30 minút, no aj tak by si z toho asi veľa nezapamätal. Preto z neho [existuje výcuc](#) a z toho [ďalší výcuc](#).

A z toho je potrebné si zobrať hlavne tieto body, ktorých nedodržanie môže mať za následok odseknutie končatín samotným scrum masterom:

## Základ

```
# Good:
result = some_function_that_takes_arguments(
    'argument one',
    'argument two',
    'argument three'
)

long_foo_dict_with_many_elements = {
    'foo': 'cat',
    'bar': 'dog'
}

with open('/path/to/some/file/you/want/to/read') as file_1, \
    open('/path/to/some/file/being/written', 'w') as file_2:
    file_2.write(file_1.read())

income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)

# Bad:
result = some_function_that_takes_arguments(
    'argument one,
    'argument two', 'argument three')
result2 = some_function_that_takes_arguments('argument one', 'argument two', 'argument three')
```

1. Always use 4 spaces for indentation (don't use tabs)
  - a. Pozor! Toto neznamená, že bude potrebné ťukať vkuse do medzerníka. V editore si vie človek nastaviť, že používa na odsadzovanie 4 medzery a pritom normálne používa taby, keď píše kód, iba editor mu to rovno mení na medzery.



2. Max line-length: 80 characters (especially in comments)
3. Always indent wrapped code for readability

## Importy

```
# Good:
import os # STD lib imports first
import sys # alphabetical

from mypkg.sibling import example # 3rd party stuff next
from subprocess import Popen, PIPE # Acceptable

from .sibling import example # local stuff last

# Bad:
import os, sys # multiple packages
import sibling # local module without "."
from mypkg import * # wildcards
```

4. Don't use wildcards (from xxx import \*)
5. Don't import multiple packages per line
6. Import standard libs first, 3rd-party libs next, and local stuff last
7. Try to import in a alphabetical order
8. When using relative imports, be explicit (with .)

## Medzery a riadky

```
# Good:
spam(ham[1], {eggs: 2})

if x == 4:
    print x, y
    x, y = y, x
dict['key'] = list[index]

y = 2
long_variable = 3
hypot2 = x*x + y*y
c = (a+b) * (a-b)

def complex(real, imag=0.0):
    return magic(r=real, i=imag)

do_one()
do_two()

# Bad
spam ( ham[ 1 ], { eggs: 2 } ) # spaces inside brackets
if x == 4 : print x , y ; x , y = y , x # inline statements, space before commas
dict ['key'] = list [index] # space before dictionary key
y = 2 # Using spaces to line up assignment operators
long_variable = 3
hypot2 = x * x + y * y # Too much space around operators
c = ( a + b ) * ( a - b ) # Too much space around operators

def complex(real, imag = 0.0):
    return magic(r = real, i = imag) # Spaces in default values
```

9. 2 blank lines before top-level function and class definitions
10. 1 blank line before class method definitions
11. Use blank lines in functions sparingly
12. Don't use whitespace to line up assignment operators (=, :)
13. Spaces around = for assignment
14. No spaces around = for default parameter values
15. Spaces around mathematical operators, but group them sensibly
16. Multiple statements on the same line are discouraged

## Komentáre

```
def my_function():
    """ A one-line docstring """

def my_other_function(parameter=False):
    """
    A multiline docstring.

    Keyword arguments:
    parameter -- an example parameter (default False)
    """
```

17. Keep comments up to date - incorrect comments are worse than no comments
18. Try to write in plain and easy-to-follow English
19. Use inline comments sparingly & avoid obvious comments
20. Each line of block comments should start with #
21. Docstring one-liners can be all on the same line
22. In docstrings, list each argument on a separate line
23. Docstrings should have a blank line before the final """

## Názvy

```
A_CONSTANT = 'ugh.'

class MyClass:
    """ A purely illustrative class """
    __property = None

    def __init__(self, property_value):
        self.__property = property_value

    def get_property(self):
        """ A simple getter for "property" """

        return self.__property

    @classmethod
    def default(cls):
        instance = MyClass("default value")
        return instance
```

24. Class names in CapWords
25. Method, function and variables names in lowercase\_with\_underscores
26. Private methods and properties start with \_\_double\_underscore
27. "Protected" methods and properties start with \_single\_underscore
28. Use all-uppercase FIXED\_TERM for constant variables
29. Always use self for the first argument to instance methods

## Ešte niečo k deleniu kódu na viacero riadkov

Na obrázku je vyjavený príklad polovične pekného kódu. Nie je to 300-znakový riadok, ale je tam pekne rozdelený dictionary do 3 riadkov. Avšak, prvý riadok dictionary má viac ako 80 znakov. Netreba sa teda báť rozdeliť napríklad list comprehension.

```
# BAD
index[term] = {
    'collection_frequency': sum([song['frequency'] for song in posting_lists[term]]),
    'document_frequency': len(posting_lists[term]),
    'posting_list': copy.deepcopy(posting_lists[term])
}
```

Tuto je rovnaký kód už korektne naformátovaný:

```
# GOOD
index[term] = {
    'collection_frequency': sum([
        song['frequency'] for song in posting_lists[term]
    ]),
    'document_frequency': len(posting_lists[term]),
    'posting_list': copy.deepcopy(posting_lists[term])
}
```

Na tomto obrázku je možné vidieť príklad toho, že aj dictionary, ktorého niektoré key:value páry by sa zmestili do jedného riadka, je lepšie rozdeliť do osobitných riadkov. K tomuto je dobré si zapamätať ešte 2 veci:

1. otváracia zátvorka sa nachádza na konci predošlého riadka, zatváracia zátvorka je zarovnaná na úroveň riadka s otváracou a telo objektu je odsadené o jednu úroveň
2. v takomto formáte je vhodné písať čiarku aj za posledný prvok. Nijako to neovplyvňuje funkcionálnosť, ale uľahčuje to ďalšie potenciálne doplnenie alebo zmenu poradia v danom objekte

```
labels: dict[str, str] = {
    'chorus': 'chorus',
    'pre-chorus': 'chorus',
    'verse': 'verse',
    'ver': 'verse',
    'pre-verse': 'verse',
    'bridge': 'bridge',
}
```

Na záver máme na dvoch obrázkoch znázornené nesprávne a správne volanie funkcie s dlhým a komplexným zoznamom parametrov.

```
# BAD
join_datasets(outdir=DATA, indirs=[TEST_JSON_EN_DIR, TRAIN_JSON_DIR], pipe=[lambda x: get_fields(x, ('Artist', 'Song', 'Genre', 'Lyrics')),
remove_trash, lambda x: split_song_to_verses(x, 70), syllable_song,
lemmatize_song, split_song_to_parts])

join_datasets(outdir=DATA, indirs=[TEST_JSON_EN_DIR, TRAIN_JSON_DIR],
pipe=[lambda x: get_fields(x, ('Artist', 'Song', 'Genre', 'Lyrics')),
remove_trash, lambda x: split_song_to_verses(x, 70), syllable_song,
lemmatize_song, split_song_to_parts])

join_datasets(
    outdir=DATA,
    indirs=[
        TEST_JSON_EN_DIR,
        TRAIN_JSON_DIR,
    ],
    pipe=[
        lambda x: get_fields(
            x, (
                'Artist',
                'Song',
                'Genre',
                'Lyrics',
            )
        ),
        remove_trash,
        lambda x: split_song_to_verses(x, 70),
        syllable_song,
        lemmatize_song,
        split_song_to_parts,
    ]
)
```

```

# GOOD
join_datasets(
    outdir=DATA,
    indirs=[
        TEST_JSON_EN_DIR,
        TRAIN_JSON_DIR,
    ],
    pipe=[
        lambda x: get_fields(
            x, (
                'Artist',
                'Song',
                'Genre',
                'Lyrics',
            )
        ),
        remove_trash,
        lambda x: split_song_to_verses(x, 70),
        syllable_song,
        lemmatize_song,
        split_song_to_parts,
    ]
)

```

## Docstring

[Predpísaný formát](#), ktorým sa v Pythone vysvetľujú objekty (funkcie, classy, všetko), ich parametre, atribúty, návratové hodnoty a podobne. Nie je to vôbec dlhý článok, je fajn si to rovno na tej stránke.

Mal by existovať v každej jednej funkcii na začiatku, teda aspoň jednoriadkový opis, čo robí tá funkcia.

## Typovanie

Áno, Python v podstate nepozná typy premenných... ale je možné mu to aj tak povedať. Kód je potom robustnejší, lebo programátor má explicitne špecifikované, aký typ chce kam poslať. Malo by byť použité určite v hlavičkách funkcií. Optimálne aj vo zvyšku kódu.

Niečo od autorov PEP 484:

*"This PEP aims to provide a standard syntax for type annotations, opening up Python code to easier static analysis and refactoring, potential runtime type checking, and (perhaps, in some contexts) code generation utilizing type information."*

*Of these goals, static analysis is the most important. This includes support for off-line type checkers such as mypy, as well as providing a standard notation that can be used by IDEs for code completion and refactoring.”*

Viaže sa k tomu PEP 484 a PEP 526. PEPy sú extrémne dlhé, preto existuje napr. tento [cheatsheet](#). Vyzerá to v zásade nejako takto:

```
# This is how you annotate a function definition  
def stringify(num: int) -> str:  
    return str(num)
```

```
# And here's how you specify multiple arguments  
def plus(num1: int, num2: int) -> int:  
    return num1 + num2
```

```
# Use Union when something could be one of a few types  
x: List[Union[int, str]] = [3, 5, "test", "fun"]
```

```
# Use Any if you don't know the type of something or it's too  
# dynamic to write a type for  
x: Any = mystery_function()
```

```
# If you initialize a variable with an empty container or "None"  
# you may have to help mypy a bit by providing a type annotation  
x: List[str] = []  
x: Optional[str] = None
```

```
# This makes each positional arg and each keyword arg a "str"  
def call(self, *args: str, **kwargs: str) -> str:  
    request = make_request(*args, **kwargs)  
    return self.do_api_query(request)
```

## Odporúčané rozšírenia pre IDE

K typovaniu sú vhodné tieto dve rozšírenia:

- [PyRight](#)
- [Python Type Hint](#)

Vo VS Code je ich možné nájsť normálne v extensions. Pre JetBrains určite existujú tiež.

Okrem toho je vhodné mať nainštalovaný nejaký klasický Python syntax highlighter.

# Metodika - Manažment kódu

Na *GitHube* existuje tímová organizácia [tp17-2021](#). Tam sa nachádzajú všetky kódy potrebné pre samotné voľby ale aj pre stránku tímu a zvyšný projektový manažment. Všetci členovia tímu tam majú správcovský prístup a vedia teda upravovať a vytvárať repozitáre.

*Pozor! Všetky repozitáre sú public, preto je potrebné dbať na to, aby sa v nich nenachádzali žiadne citlivé údaje ako napríklad API kľúče alebo prihlasovacie údaje.*

*Pozor! Niektoré existujúce repozitáre majú slovenské názvy, ale všetko nové je potrebné vytvárať a písať v angličtine.*

## Kedy je potrebný nový repozitár

Nezávislé komponenty aplikácie alebo služby je potrebné oddeliť do rôznych repozitárov. Nový repozitár je preto potrebné vytvoriť v prípade, že vyvíjaný komponent je nezávislý na ostatných. Pod nezávislým komponentom si je možné predstaviť časť aplikácie, ktorá je ohraničená špecifikovaným rozhraním (rozhranie v zmysle architektúry softvéru) a je možné ju považovať za jeden logický celok.

Pri *Microservice* architektúre by mala mať každá služba vlastný repozitár, pričom služby z rovnakého systému by mali mať rovnaký prefix v názve repozitáru. Napríklad *gateway-modules* a *gateway-token-manager*. Ďalej by pri *Microservices* mal existovať jeden repozitár napr. *gateway*, ktorý by obsahoval rôzne alternatívy *Docker compose* súborov pre rôzne situácie. Napr. *deploy*, *load\_test*, *e2e\_test*, *staging* a podobne.

*[Tuto](#) je demo repozitár MS projektu. Dá sa ním inšpirovať, ale nesedia k tejto metodike napríklad názvy repozitárov. Ale je možné si aspoň vytvoriť predstavu o štruktúre.*

*Pri delení kódu na komponenty je potrebné riadiť sa architektúrou aplikácie, prípadne sa poradiť so zvyškom tímu alebo architektom, ak je určený.*

## Ako sa vytvára repozitár

Na [stránke](#) organizácie na *GitHube* je potrebné v časti *Repositories* kliknúť na *New*.

- V kolonke *Owner* je potrebné vybrať organizáciu *tp17-2021*, nie osobný účet alebo inú svoju organizáciu.
- Do kolonky *Repository name* je potrebné zadať rozumný samoopisný názov nového repozitáru v angličtine oddelený pomlčkami, ktorý by mal hlavne zodpovedať danému komponentu alebo inému kódu, ktorý sa bude nachádzať v repozitári. Pri viacerých repozitároch týkajúcich sa nejakej spoločnej časti je vhodné zadať rovnaký prefix do názvu repozitáru. Napríklad *gateway-modules* a *gateway-token-manager*.
- Do *Description* je potrebné zadať v angličtine niekoľko slovný popis toho, čo sa bude nachádzať v repozitári.
- Viditeľnosť repozitáru je potrebné nastaviť na *Public*.
- V poslednom rade je potrebné, aby repozitár obsahoval *README.md* a *.gitignore* súbory. *README.md* je možné nastaviť už v tomto kroku zaškrtnutím checkboxu, ak súbory ešte žiadne *README.md* neexistuje.

Na druhej strane `.gitignore` je lepšie vytvoriť až manuálne počas vývoja, lebo ponúkaná možnosť vytvorenia na tejto obrazovke je tak trochu pofidérna.

- Nakoniec je potrebné kliknúť na *Create repository*.

Výsledkom tohto je vytvorený nový repozitár.

## Ako si naklonovať repozitár

Na *Githube* je potrebné otvoriť daný repozitár. Vpravo hore sa nachádza zelené tlačidlo *Code*. Po jeho stlačení sa vyroluje menu pre klonovanie. Je odporúčané používať SSH možnosť. Je tam zobrazená adresa repozitára - text v tvare `git@github.com:...` Toto je potrebné si skopírovať. V konzole na svojom počítači je potrebné prejsť do adresára, kam bude repozitár naklonovaný. Repozitár naklonujeme príkazom `git clone git@github.com:...`

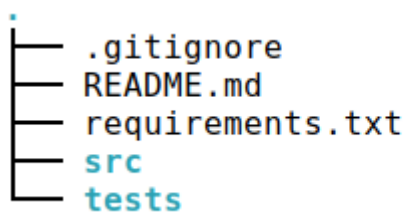
*Poznámka: git clone vytvorí v aktuálnom adresári nový adresár s názvom podľa názvu repozitára. Často sa stáva, že človek si vytvorí adresár s takým názvom a očakáva, že git clone rovno do aktuálneho adresára stiahne už obsah repozitára. Nie, vytvorí ešte podadresár.*

## Ako si naklonovať repozitár, ak už lokálne prebehol git init

Rovnako ako v predošlom postupe, je potrebné si skopírovať si adresu repozitáru. V lokálnom repozitári je potrebné spustiť `git remote add origin git@github.com:...` Potom je potrebné commitnúť spraviť pull a vyriešiť prípadné konflikty.

Ideálnym riešením problémov tohto spôsobu je nepoužívať ho a najprv vytvoriť repozitár a až tak začať niečo kódovať lokálne.

## Štruktúra repozitáru



Je odporúčané, aby repozitár napríklad Python API komponentu mal aspoň takú štruktúru ako na obrázku. V `requirements.txt` by mal byť zoznam potrebných knižníc. V `src/` by mal byť zdrojový kód a v `tests/` by mali byť testy.

Každá *Microservice* by mala mať aj *Dockerfile*.

# Git účty v PC

Ak človek používa vo svojom PC rôzne *GitHub* účty, je potrebné si dať pozor na to, aký účet používa v konkrétnom repozitári. Globálne a lokálne nastavenia (globálne sú default pre celý PC, lokálne sú pre aktuálny repozitár) je možné si zobraziť príkazmi `git config -l` a `git config --local -l`. Pre úprave nezrovnalostí je odporúčané použiť editovací mód: `git config --local -e`.

## Git a SSH

Niekedy sa stáva, že pri každom pulle a pushi musí vývojár zadávať svoje prihlasovacie údaje na *GitHub*. To je pomerne nepohodlné. Preto je odporúčané používať SSH kľúče. V nastavení svojho účtu na *github.com* v záložke *SSH and GPG keys* treba použiť *New SSH key*. Do *Title* sa zadáva vlastný názov kľúča a do *Key* je potrebné nakopírovať svoj public key. Na *Linuxe* a *Mac OS* je toto triviálna vec. Na *Windows* sa často vyskytujú problémy a chce to viac námahy na sfunkčnenie. Používateľom *Windowsu* ostáva individuálne si vygoogliť, ako to rozbehať alebo najlepšie použiť [tento návod](#).

## README.md

Každý repozitár by mal obsahovať *README.md* v angličtine s vysvetlením, čo sa nachádza v repozitári, prípadne, ako to spustiť a ako to funguje.

## Branche

Po novom (niekoľko mesiacov) sa hlavná brancha na *GitHub*e volá *main*, nie *master* ako v minulosti. Hlavnej branchi je potrebné nastaviť zákaz priameho pushovania a yvžadovanie akceptovaného pull requestu pred mergovaním. To je možné nastaviť v nastaveniach repozitára -> *Branches* -> *Add rule*. Do *Branch name pattern* je potrebné zadať názov branchy, teda *main*. V *Protect matching branches*, je potrebné vybrať *Require a pull request before merging* a v tom *Require approvals*. Samozrejme, podľa potreby je možné nastavovať ďalšie obmedzenia na ďalších branchiach.

Do *main* by sa malo mergovať iba ak je mergovaný kód spustiteľný a bez chýb (nemusí byť ešte úplný). Napríklad, sú špecifikované funkčné požiadavky komponentu, takže po každej plne a korektné naimplementovanej funkcionalite je možné mergnúť vývojové branchy do *mainu*.

Pre vývoj je potrebné používať vývojové branchy. Každý repozitár by mal mať jednu *development* branchu a potom niekoľko ďalších pre individuálne potreby. Napríklad, jedna brancha pre každú funkcionalitu. To už je na racionálnom zvážení vývojára v danom repozitári.

Čo sa týka názvu branchy, je potrebné dodržiavať spojovníkový formát (*foo-bar-aha*). Okrem samotného identifikátora Jira tasku, ktorý má v sebe pomlčku.



Ak sa brancha týka konkrétneho tasku v Jire, je vhodné zahrnúť identifikátor tasku v názve branche. Najlepšie ako prefix. Napríklad, *EV-123-million-dollar-problem-sol*.

## Commit message

### Asociácia s Jira taskami

Ak sa daný commit týka nejakého tasku, je nutné v commit message uviesť identifikátor tohto tasku. Každú prácu je možné namapovať na nejaký task, takže je očakávané, že každý jeden commit bude asociovaný s nejakým taskom pomocou identifikátora v commit message. Príklady commit messages: *EV-72 Test new github pipeline*, *EV-89 Update webpage assets - add python code style guide*.

### Informatívne popisy

V commit message je potrebné rozumne opísať, čo daný commit prináša, čo mení a podobne. Všetko je potrebné písať v angličtine.

Commit message by nemala presiahnuť dĺžku 72 znakov.

Od štvrtákov na FIIT STU je očakávané, že vedia písať rozumné commit message bez toho, aby dostali papekom po hlave.

## Pull requesty

Pull request (PR) je v podstate požiadavka na merge nejakej branche do nejakej inej. Teda, ak je kód pripravený na mergnutie do nadradenej branche, vývojár vytvorí PR, lebo chce mergnúť tento nový kód do nadradenej branche.

### Vytvorenie

Na *GitHub*e nájde tlačidlo *New pull request* a stlačí ho. Alternatívne sa nachádza na branchi, ktoré chce mergnúť, a *GitHub* mu už zobrazuje možnosť vytvorenia PR.

PR je potrebné rozumne nazvať. Defaultne tam *GitHub* dá commit message posledného commitu mergovanej brenche. Avšak, výpovednejšie je pri viacerých commitoch zhrnúť v názve podstatu PR. Netreba ale presahovať cca 5 slov. Ak sa PR týka konkrétneho tasku, je vhodné ho referencovať identifikátorom v názve PR.

Ďalej je potrebné pridať komentár/popis k PR. Tento môže byť ľubovoľne dlhý. Tu je vhodné zhrnúť, čo má vývojár na srdci a prečo si myslí, že by mal byť tento PR одобrený a mergnutý do nadradenej branche. Zoznam commitov tu nie je potrebné písať, pretože to pri PR vidno aj tak.

Ak je repozitár nastavený podľa tejto metodiky, pre prijatie PR je potrebná aspoň jedna review. Pri vytváraní PR je možné v pravej časti obrazovky pridať reviewerov. Ak je dôvod (napr. je to reporter toho tasku alebo je to človek, ktorý má na starosti túto časť, alebo existuje hocikaký iný dôvod, prečo by tam mal byť práve on) na to, aby tam bol niekto konkrétny, je potrebné ho pridať.

Okrem toho je možné pridať PR nejaké labely. Defaultne tam je niekoľko fajných. Avšak, oplatí sa vždy si vytvoriť ešte jeden "bugfix" napríklad defaultne žltou farbou. Totiž, často sa stane (aj keď by sa nemalo), že až po mergnutí je objavená chyba a je potrebné ju rýchlo opraviť a znova mergnúť. Vtedy je PR označený napr. týmto labelom.

Následne je potrebné kliknúť *Create pull request* alebo *Create draft pull request*, ak je známe, že kód ešte nebude možné prijať, ale je potrebné vytvoriť PR, aby sa rozbehla diskusia a vyriešil sa v nej nejaký konkrétny problém.

## Diskusia

Pri PR je možné písať komentáre a viesť tak diskusiu ohľadom prinesených zmien, otázok, objavených chýb a podobne. Tu si väčšinou vývojári vyjasňujú veci okolo daného PR a navrhujú zmeny.

Ak chce vývojár povoliť PR alebo formálne požiadať o zmeny, je potrebné, aby sa najprv pridal medzi reviewerov daného PR -> vpravo hore -> *Reviewers* -> klikne na seba. Potom je potrebné obnoviť stránku. Hore na stránke sa zobrazuje v žltom boxe oznam o tom, že PR čaká na jeho review. Vývojár potrebuje stlačiť zelené tlačidlo *Add your review*.

Na nasledujúcej obrazovke vidí diffy zmenených súborov. Vpravo hore je tlačidlo *Review changes*. Po kliknutí na toto tlačidlo vie vývojár napísať nejaký komentár a vybrať si, či je to iba komentár, alebo *Approve* alebo formálne *Request changes*.

Vývojár sa môže zapojiť do diskusie normálne aj keď nie je formálne medzi reviewermi, ale vtedy nevie udeliť approve na merge.

## Uzavretie

Ak je už všetko ok, reviewer PR by mal po *Review changes* stlačiť *Approve*. Následne už vie ľubovoľný vývojár stlačiť *Merge pull request* a *Confirm merge*.

## CI/CD pipelines

Na *GitHub*e je možné pekne si automatizovať deployovanie a tiež testovanie. Volá sa to *Actions*. Tie je možné nastaviť tak, aby sa automaticky spúšťali pri určitých eventoch v codebase. Napríklad, spustiť automaticky deploy stránky po prijatí pull requestu alebo automaticky spustiť testy kódu po vytvorení pull requestu a teda pred potenciálnym mergnutím, aby bolo hneď jasné, či kód prejde testami.

Tieto *Actions* je možné spúšťať na *GitHub* serveroch ale aj na vlastných. Náš projekt používa self-hosted workera na *team17-21.studenti.fiit.stuba.sk* serveri.

Automatické deployovanie už zrejme inde ako na tímovej stránke nebude potrebné použiť. Ak by sa aj naskytla príležitosť, je možné to vyriešiť individuálne. Je ale veľmi žiadané použiť automatické testy v projekte.

# Testovanie

GitHub má na internete takýto celkom pekný [návod](#) k testovaniu Pythonu v actions. Ale v skratke:

```
# nejaký názov pipeline
name: Run tests on XY branch

# kedy sa má automaticky spúšťať
on:
  # pri každom push do branche development
  push: [ development ]

jobs:
  build:
    # kde sa to má spúšťať (pre tento projekt je nakonfigurovaná self-hosted group)
    runs-on: self-hosted

    # čo sa má vykonať
    steps:
      # naklonuje aktuálnu branchu do work directory: /home/ubuntu/actions-runner/_work/...
      - uses: actions/checkout@v2

      # názov kroku
      - name: echo hello world
        # ľubovoľný linux command na runs-on^ stroji v tom work directory
        run: echo "Hello world!"

      # nejak tak sa vraj testuje
      - name: Test with pytest
        run: |
          pytest
```

Toto je *GitHub* pipeline. V repozitári je to súbor: test/.github/workflows/názov.yml. Na obrázku je vysvetlená základná syntax a funkcionálnosť.

V prípade Microservice architektúry s viacerými repozitármi je pre komplexnejšie testovanie aplikácie potrebné v pipeline naklonovať viaceré potrebné repozitáre. Ak ešte používame aj Docker, v pipeline je potrebné postaviť image, potom spustiť kontajner a otestovať požadovaný interface.

## Dockerfile

Dockerfile je špeciálny súbor, ktorý predpisuje *Dockeru*, ako má vyskladať požadovaný image z danej aplikácie. Príklad Docker-filu pre vytvorenie *FastAPI* image:

```
# predpripravený image z dockerhubu
FROM python:3.10

# hlavný priečinok v kontajneri
WORKDIR /code

# nakopírovanie súboru requirements.txt do kontajnera
COPY ./requirements.txt ./code/requirements.txt

# inštalácia potrebných knižníc podľa requirements.txt
RUN pip install --no-cache-dir --upgrade -r ./code/requirements.txt

# nakopírovanie zdrojových kódov do kontajnera
COPY ./src /code/src

# spustenie FastAPI služby na porte 80
CMD [ "uvicorn", "src.main:app", "--host", "0.0.0.0", "--port", "80" ]
```

Jednotlivé kroky je možné vnímať ako vrstvy pridávané do image/kontajnera (kontajner je konkrétna inštancia image). Ak by bolo pri každom jednom spustení potrebné sťahovať a inštalovať potrebné súbory, build time by bol nepohodlne vysoký. Preto si *Docker* chachuje image po vrstvách s tým, že ak sa obsah vrstvy nezmenil, použije sa nacacheovaná verzia a pokračuje sa ďalším krokom. Od prvej zmenenej vrstvy je ale nutné odznova sťahovať, kopírovať a inštalovať aj všetky ďalšie vrstvy. Z tohto dôvodu je pre optimalizáciu build time vhodné zoradiť kroky v Dockerfile podľa frekvencie zmien v nich od najmenej.

Konkrétne v tomto príklade sú najprv kopírované requirements.txt a sú nainštalované a až tak sú kopírované zdrojové kódy aplikácie. Je očakávané, že zdrojové kódy sa budú meniť možno aj pri každom builde, no potrebné knižnice sa zmenia iba výnimočne. Preto ich stačí v skutočnosti nainštalovať iba raz, čo zberie netriviálny čas, a po zvyšok vývoja *Docker* používa nacacheovanú verziu týchto nainštalovaných knižníc, čo zaberá prakticky 0 času.

## Docker build

Pre vytvorenie image z Dockerfile je potrebné zavolať príkaz `docker build`. Príklad:

```
docker build -t image-name .
```

Toto vytvorí image podľa Dockerfile v aktuálnom priečinku a názve ho `image-name`. (tá bodka na konci značí aktuálny priečinok, nie je možné ju vynechať)

## Docker run

```
docker run -d --name container-name -p 8222:80 image-name
```

Toto vytvorí a spustí inštanciu image `image-name` v kontajneri `container-name`.

Prepínač `-d` zabezpečí spustenie v detached mode, čiže nezostane v termináli otvorené spojenie do vnútra kontajneru, ale spustí sa na pozadí.

Prepínač `-p 8222:80` nabinduje port 80 vo vnútri kontajnera na port 8222 lokálneho počítača.

## Docker-compose

V prípade potreby zhlukovania viacerých kontajnerov (ako napríklad pri Microservices) je vhodné použiť `docker-compose`. Ide o predpis toho, aké kontajnery je potrebné spustiť a aké parametre im nastaviť. Dobrým príkladom tohto je *gateway* repozitár, ktorý zhlukuje viacerá služby G a spája ich jedným `docker-compose.yml` súborom. Okrem toho je v tomto repozitári ilustrované použitie *GitHub submodules*, teda referencovanie rôznych repozitárov v jednom hlavnom.

Týmto príkazom je možné skomponovať a rovno spustiť kontajnery na pozadí:

```
docker-compose up -d
```

## Užitočné docker príkazy

```
docker ps -a
```

Zobrazí všetky existujúce kontajnery.

```
docker logs container-name
```

Vypíše logy daného kontajnera.

```
docker stop container-name
```

Zastaví kontajner (ten bude stále existovať, ale bude zastavený).

```
docker start container-name
```

Rozbehne kontajner, ak je zastavený.

```
docker rm container-name
```

Vymaže kontajner. Najprv musí byť zastavený.

# Metodika - Testovanie

## Unit testovanie

Jednotkové testovanie slúži na testovanie malých izolovaných funkcionalít. Jednotkový test má najvyššiu granularitu. Jednotkové testy sú rýchle, spoľahlivé, ciele a malé.

### Jednotkový test obsahuje len jeden assert

Treba granulárnejšie písať testy. V teste by sa mal nachádzať iba jeden assertion statement. Ak majú viac assertov, často bývajú testy veľké a náročné na setup a tým pádom sa ťažšie uplatňujú a setupujú. Ak sa v teste nachádza len jeden statement, znamená to že je testovaná tá jediná vec, ktorú chceme.

### Jednotkový test by nemal obsahovať if podmienky

Test by nemal obsahovať if statement lebo to indikuje možnosť viacnásobného správania na základe situácie. Takéto správanie je neželané. Jednotkový test by mal v každom prípade vykonať to isté ak má byť vyhodnotený ako úspešný. Testy musia byť statické, nie s dynamickým správaním.

### Jednotkový test testuje jednu vytvorenú triedu

Jednotkový test by mal byť rozsahom limitovaný na testovanie jednej maximálne jednej metódy alebo triedy. V teste by sa malo vytvorenie triedy, ktorú testujete vyskytovať raz. Vytváranie viacerých tried predstavuje komplikáciu a test sa stáva neprehľadnejším a malé špecifickým. V prípade, že test zlyhá je potrebné vedieť okamžite určiť, v ktorej triede alebo metóde sa nachádza chyba, čo pri kompilácii viacerých tried nie je okamžite jasné.

Závislosti testovacej triedy by sa nemali vytvárať. Nahrádzajú sa použitím tzv. stubs (náhrada, atrapa). Mali by byť imitované pomocou imitovacích knižníc ako Moq alebo NSubstitute.

Taktiež by sa mali imitovať triedy, ktoré priamo neovládame ako napríklad file system. Takáto imitácia sa robí pomocou pomocnej wrapper triedy, ktorá povie ako sa má imitovaná trieda správať.

#### Príklad:

```
public void
should_find_only_text_files_in_the_specified_directory() {
    File file = mock(File.class);
    when(file.list()).thenReturn(new String[] { "readme.txt",
"foobar" });

    assertThat(store.list(file)).contains("readme.txt");
}
```

}

Takáto trieda zabezpečí, že sa nespoliehame na to, čo sa reálne v danom priečinku nachádza.

## Jednotkové testy neobsahujú natvrdo napísané hodnoty pokiaľ nie sú relevantné

V jednotkových testoch by nemali figurovať natvrdo napísané hodnoty pre vstupné jednotky. Mali by sa namiesto nich používať takzvané fixtures.

Fixture je nemenný set objektov pre testovanie, ktoré slúžia ako vstupy pre testy. Zmysel toho je že test pracuje v dobre poznanom prostredí vstupných hodnôt aby mohli byť výsledky testu sú opakovateľne dosiahnuté.

## Jednotkové testy sú bez poznania stavu aplikácie

Nezáleží na poradí vykonávania testov. Testy nepredpokladajú výsledky iných testov. Testy pracujú iba nad dátami, ktoré boli v rámci predprípravy vykonania testy pridané do testovacej databázy.

## Ako písať unit testy?

- Názvy testov by mali vystihovať čo test robí. Mali by začínať slovesom ako should alebo can (it should).
- Názvy testov by mali byť písané snake\_case.
- Nebáť sa primeranej duplicity kódu. Je lepšie mať duplicitné riadky kódu, lebo to zlepšuje čitateľnosť testu.
- Redukovať before a after dekorátory, pretože môžu narušiť príbeh pri čítaní testov.
- Nebáť sa viac riadkov kódu pri testovacích funkciách. Pre lepšiu čitateľnosť je lepšie mať kód pri sebe ako ho mať skrytý.
- Lokálne premenné - v niektorých prípadoch je lepšie vytvoriť lokálne premenné s cieľom lepšej čitateľnosti (napríklad výpočet) a niekedy je výhodnejšie ponechať natvrdo nastavenú premennú.

```
// not so clear
isAllowedToDrink(currentYear - yearOfBirth);

// this version is easier to understand
int age = currentYear - yearOfBirth;
isAllowedToDrink(age);
```

- Niekedy si vieme uľahčiť prehľadnosť premenných v assertoch.

```
assertThat(createUser("name", "email", "site")).isEqualTo(new  
User("name", "email", "site"));
```

```
assertThat(toLowerCase("Uppercase Characters")).isEqualTo("uppercase  
characters");
```

“Uppercase Characters” týmto spôsobom prehľadnejšie vystihuje podstatu assertu.

- Netreba písať veľké description pri assertoch lebo pri dobrej knižnici, ktorá disponuje veľa metódami na assertovanie, ktoré vedú chybovú správu dobre reprodukovať.
- V unit teste sa nepoužívajú logy, lebo chyby by mali byť jasne indikované assertami a nie je potrebné ich pomocou logov identifikovať. Ak danú chybu nie je možné identifikovať, je potrebné daný test refaktorovať.
- V niektorých prípadoch je potrebné pomocné triedy vytvárať využitím viac ako jedného riadku. Taktiež je niekedy potrebné poslať rôznych počet parametrov, čo sa má odzrkadliť na správaní danej triedy. Takúto triedu je v tom prípade možné simulovať vlastným builderom (triedou ktorá danú triedu vytvorí a vráti) v rámci súboru testu aby sa eliminovala redundancia ale zachovala sa čitateľnosť.

## Ako veľmi testovať unit testami

Niekedy sa tím ocitne v situácii, kedy netestuje reálne správanie testovaného softvéru, ale len píše testy, aby splnil požiadavku pokrytia kódu testami. Tieto testy nesprávne testujú a overujú funkcionálnosť a nedávajú tímu istotu že ich kód beží správne. Namiesto zamerania sa na kvantitatívne písanie unit testov pre každú metódu triedy je dobré sa zamyslieť nad testovaním správania v unit testoch. Pre každý komponent je vhodné otestovať jeho reálnu funkcionálnosť a nie len overovanie návratových hodnôt, ktoré nemusia reprezentovať jeho reálne používanie. Naplnenie pokrytia kódu niekedy vedie k potrebe písania veľmi triviálnych testov, ktoré zvyčajne spotrebujú náklady a čas.



# Integračné Testovanie

Integračné testovanie zahŕňa spoluprácu viacerých modulov, ktoré sa predtým otestovali samostatne. Pri väčších projektoch je typické, že softvér pozostáva z viacerých modulov, na ktorý pracovali viacerí programátori. Cieľom je odhaliť chyby, ktoré môžu nastať pri vzájomnej interakcii modulov.

## Stubs and Drivers

Sú jednoduché programy, ktoré simulujú správanie a komunikáciu modulov, ktoré ešte nie sú implementované.

stub (náhrada) - je volaný modulom počas testovania

driver (ovládač) - volá modul, ktorý má byť testovaný

## Typy integračného testovania

### Bottom-Up

Je stratégia, pri ktorej sa najskôr otestujú najmenšie moduly a postupne sa prechádza na vyššie úrovne pokiaľ nie sú zapojené všetky moduly softvéru.

Výhody: ľahšie nájdeme chybu

Nevýhody: celú (najdôležitejšiu časť) testujeme na koniec, nevieme testovať prototyp

### Top-Down

Najskôr sa testujú najvyššie moduly softvéru. Ak máme modul, ktorý nie je implementovaný používajú sa náhrady (stubs)

Výhody: ľahko nájdeme chybu, možné testovať prototyp, kritické miesta testujeme najskôr

Nevýhody: potrebujeme veľa náhrad (stubs), moduly na nižšej úrovni sú testované nedostatočne

## Príklady

Overenie platnosti tokenu - prijatie tokenu z VT, overenie či je token platný, vrátenie odpovede na VT

# Feature testovanie

Feature testovanie slúži na testovanie novo pridanej alebo pozemnej funkcionality systému. Jeho cieľom je odhaliť nefungujúce časti a bugy v kóde, pričom sa pomocou tohto testovania vyhodnocuje, či je daná funkcionality vhodná pre systém.

Ako spraviť feature testovanie:

- je potrebné ovládať danú feature (špecifikáciu, možné správanie, edge-casy) a poznať jej požiadavky.
- je dôležité zamerať sa na slabé články danej funkcionality, ktoré by ju mohli pokaziť a testy zamerať na ich testovanie
- vytvoriť vhodné testovanie scenáre, ktoré zahŕňajú aj pozitívne, negatívne, očakávané a neočakávané výsledky.
- feature test imituje ako reálny používateľ použije danú funkcionality

Feature testovanie sa vykonáva až po implementovaní celej funkcionality a otestovaní menšími testami (unit a integration).

Pri našom projekte elektronických volieb si ako integračný test vieme predstaviť overenie funkcionality volebného terminálu (aplikácie), pričom cieľom je od prvotného vloženia tagu sa dostať až po odoslanie hlasu na gateway.

Druhým takým feature testom by mohlo byť zobrazenie výsledkov volieb, kde sa volič dozvie rôzne štatistiky o priebehu a finálnom výsledku.

# Regresné testovanie

Regresný test slúži na odhalenie skrytých chýb, ktoré nevzniknú priamo v novo pridanej funkcionality, ale niekde inde v kóde ako následok jej pridania.

Pri regresnom teste je dôležité zvážiť aká veľká časť kódu sa pri regresnom teste testuje. Testovanie všetkých častí a spustenie všetkých menších testov (unit, integration a feature) býva pri veľkých projektoch nákladné a v takom prípade je potrebné určiť časti kódu, ktoré sú relevantné (a môžu byť ovplyvnené novo pridanou funkcionality)

Ako vybrať správne testy a časti kódu do regresného testovania:

- scenáre ktoré sú často chybné
- scenáre, ktoré sú pre používateľa viditeľnejšie
- hlavné časti systému
- časti, ktoré prešli najväčšími alebo nedávnymi zmenami
- všetky integračné testy
- všetky komplexné testy (feature)
- testovanie edgecasov
- testovanie úspešnej a neúspešnej funkcionality
- prioritizácia testov podľa dôležitosti

# Akceptačné testovanie

Akceptačné testovanie je vykonávané konečným používateľom alebo klientom v neskorých fázach projektu s cieľom validovať naplnenie požiadaviek na systém a business využitie.

Akceptačné testovanie sa môže robiť až keď jednotkové, integračné a regresné boli úspešné. Výsledkom akceptačného testovania je rozhodnutie klienta/koncového používateľa je akceptovanie alebo neakceptovanie daného produktu. Takéto akceptačné testovanie je potrebné hlavne v prípade, ak bol produkt vyvinutý podľa špecifikácie a nebol predmetom pravidelnej komunikácie a validácie produktu s klientom.

Predpoklady akceptačného testovania:

- jasná definícia biznis požiadaviek
- dokončený aplikačný kód
- vykonané unit, integration a feature testy
- iba drobné chyby sú akceptovateľné pred akceptačným testovaním
- neboli nájdené veľké chyby počas regresného testovania
- všetky testami odhalené nedostatky sú opravené

# Používateľské testovanie

Úlohou používateľského testovania je odhaliť možné chyby v použiteľnosti aplikácií. Malo by sa vykonávať s ľuďmi z rôznych - napr. vekových skupín, úrovne vzdelania a technickej zdatnosti. Pri testovaní pozorujeme chovanie ľudí čím môžeme odhaliť chyby, ktoré môžu byť počas vývoja skryté.

Najlepšie je testovanie vykonať na skupine 5 až 7 ľudí, ktorí sú dostatočnou vzorkou na overenie chýb v užívateľskom rozhraní aplikácie. Testovanie by malo byť dôkladne pripravené a malo by postupovať podľa striktného scenára, ktorý zabezpečí rovnaké testovacie scenáre pre účastníkov testovania. Pri testovaní by mal byť koordinátor, ktorý začne testovanie s participantom, číta mu otázku, odpovedá na prípadné nejasnosti a v kritickej situácii vie respondentovi pomôcť aby sa nenarušil priebeh testovania. Okrem koordinátora by mal testovanie sledovať zapisovateľ, ktorý zapisuje respondentove odpovede, reakcie a správanie, ktoré budú po teste vyhodnocované a spracovávané. Pri testovaní sa odporúča uchovávať audiovizuálny záznam aby sa nestratili žiadne informácie.

Po testovaní sa vykoná evaluácia v ktorej sa identifikujú problémy zistené počas testovania. Chyby sa kategorizujú podľa dôležitosti a dopadu a posunú sa klientovi na zapracovanie.

# Testovanie Django aplikácie

Praktiky:

- ak sa môže kód poukázať, treba ho testovať
- každý test by mal testovať len jednu funkciu
- testy by nemali byť zložité ani rozsiahle
- testy sa spúšťajú pri pull a push z repozitára a pri nasadení do stagingu

Knižnice:

- django-webtest
- coverage
- django-discover-runner
- factory\_boy, model\_mommy, mock (mimovacie knižnice a fixtures)

```
from django.test import TestCase
from whatever.models import Whatever
from django.utils import timezone
from django.core.urlresolvers import reverse
from whatever.forms import WhateverForm

# models test
class WhateverTest(TestCase):

    def create_whatever(self, title="only a test", body="yes, this is only a test"):
        return Whatever.objects.create(title=title, body=body, created_at=timezone.now())

    def test_whatever_creation(self):
        w = self.create_whatever()
        self.assertTrue(isinstance(w, Whatever))
        self.assertEqual(w.__unicode__(), w.title)
```

Na hore uvedenom príklade vidíme test vytvorenia modelu a overenia správneho vytvorenia nadpisu v danom objekte. Je to príklad unit testu.

## Testovanie frontend Views

Na testovanie view je v niektorých prípadoch potrebné použiť knižnicu Selenium na testovanie requestov a ich návratových kódov a taktiež automatizovane klikáť na elementy.

Jednoduchý príklad overenia návratovej hodnoty zo špecifickej url:

```
# views (uses reverse)

def test_whatever_list_view(self):
    w = self.create_whatever()
    url = reverse("whatever.views.whatever")
    resp = self.client.get(url)

    self.assertEqual(resp.status_code, 200)
    self.assertIn(w.title, resp.content)
```

Na nasledujúcom obrázku môžeme vidieť príklad testovania vyplnenia formuláru na URL /add. Selenium si stiahne obsah stránky a pomocou id nájde potrebné elementy a vyplní ich. Po vyplnení údajov vyvolá kliknutie nad submit tlačidlom a overí či bol presmerovaný na URL /.

```
# views (uses selenium)

import unittest
from selenium import webdriver

class TestSignup(unittest.TestCase):

    def setUp(self):
        self.driver = webdriver.Firefox()

    def test_signup_fire(self):
        self.driver.get("http://localhost:8000/add/")
        self.driver.find_element_by_id('id_title').send_keys("test title")
        self.driver.find_element_by_id('id_body').send_keys("test body")
        self.driver.find_element_by_id('submit').click()
        self.assertIn("http://localhost:8000/", self.driver.current_url)

    def tearDown(self):
        self.driver.quit

if __name__ == '__main__':
    unittest.main()
```

Testovanie formulárov spočíva vo vyplnení údajov formulára a následnom overení či je formulár validný.

```
def test_valid_form(self):
    w = Whatever.objects.create(title='Foo', body='Bar')
    data = {'title': w.title, 'body': w.body,}
    form = WhateverForm(data=data)
    self.assertTrue(form.is_valid())

def test_invalid_form(self):
    w = Whatever.objects.create(title='Foo', body='')
    data = {'title': w.title, 'body': w.body,}
    form = WhateverForm(data=data)
    self.assertFalse(form.is_valid())
```

Testovať sa môže aj API rozhranie, pri ktorom sledujeme či sa vráti správny formát odpovede. Podobným spôsobom môžeme testovať aj či odpoveď obsahuje správne dáta.

```
from tastypie.test import ResourceTestCase

class EntryResourceTest(ResourceTestCase):

    def test_get_api_json(self):
        resp = self.api_client.get('/api/whatever/', format='json')
        self.assertValidJSONResponse(resp)

    def test_get_api_xml(self):
        resp = self.api_client.get('/api/whatever/', format='xml')
        self.assertValidXMLResponse(resp)
```

Testy sa spúšťajú príkazom `manage.py test` (ktorý ako ďalší nepovinný parameter akceptuje skupinu alebo názov testu)

Výsledok všetkých testov sa zobrazí v prehľadnej tabuľke so stavom jednotlivých testov a celkovým časom trvania.

```
test_signup_fire (whatever.tests.TestSignup) ... ok
test_invalid_form (whatever.tests.WhateverTest) ... ok
test_valid_form (whatever.tests.WhateverTest) ... ok
test_whatever_creation (whatever.tests.WhateverTest) ... ok
test_whatever_list_view (whatever.tests.WhateverTest) ... ok
```

```
-----
Ran 5 tests in 12.753s
```

```
OK
```

## Alternatívy knižnice Selenium

Alternatívne k knižnici Selenium je možné použiť na testovanie frontendu testovaciu knižnicu Cypress. Selenium je všeobecný automatizér prehliadača, ktorý umožňuje aj testovanie web aplikácií. Cypress je vytvorený priamo na testovanie web aplikácií.

Priamo pre testovanie Svelte aplikácie existuje aj knižnica `testing-library/svelte-testing-library`.

## Rozloženie Svelte kódu vzhľadom na testovateľnosť aplikačnej logiky

Svelte jazyk odporúča logiku aplikácie rozdeliť na dve časti - vykresľovanie grafického rozhrania (.svelte súbory) a na .ts súbory obsahujúce hlavnú logiku aplikácie, ktoré sú importované do .svelte súborov. Toto rozloženie aplikácie umožní testovať hlavnú logiku aplikácie aj samostatne bez testera cez prostredie webového prehliadača.

## Zdroje

<https://medium.com/vx-company/the-5-unit-testing-guidelines-f21d39c33e0b>

<https://github.com/elefevre/elements-of-unit-testing-style>

<https://www.guru99.com/integration-testing.html>

<https://www.javatpoint.com/integration-testing>

<https://medium.com/@sameernyaupane/php-test-driven-development-part-5-integration-testing-51535ca56bf0>

<https://medium.com/swlh/laravel-end-to-end-testing-with-cypress-e574a73ce222>

<https://www.guru99.com/user-acceptance-testing.html>

<https://realpython.com/testing-in-django-part-1-best-practices-and-examples/>

<https://svelte.dev/faq>

# Metodika - Definition of done

Implementačné kritéria:

- Úspešné code review
  - Zodpovedajúca kvalita kódu, komentárov, úroveň testovania
- CI/CD ready
  - Vykonané potrebné úpravy konfigurácie, databázy,...
- Úspešné testy
  - Pokrytie všetkých potrebných typov testov vzhľadom na charakter funkcionality
  - Zodpovedajúce overenie pridanej funkcionality
  - Úspešné zbehnutie všetkých testov
- Vykonaný merge Pull request-u

Dokumentačné kritéria:

- Aktualizácia používateľskej dokumentácie
- Aktualizácia technickej dokumentácie
  - Popis rozhraní
  - Popis k zložitejším častiam kódu (ak je potrebný)
  - Popis práce s dátami
  - Popis prepojenia na aplikačnú architektúru

Ak nie sú splnené všetky požiadavky a úloha je vrátená z review:

- Prechod do TO DO

Finálna akceptácia business ownerom



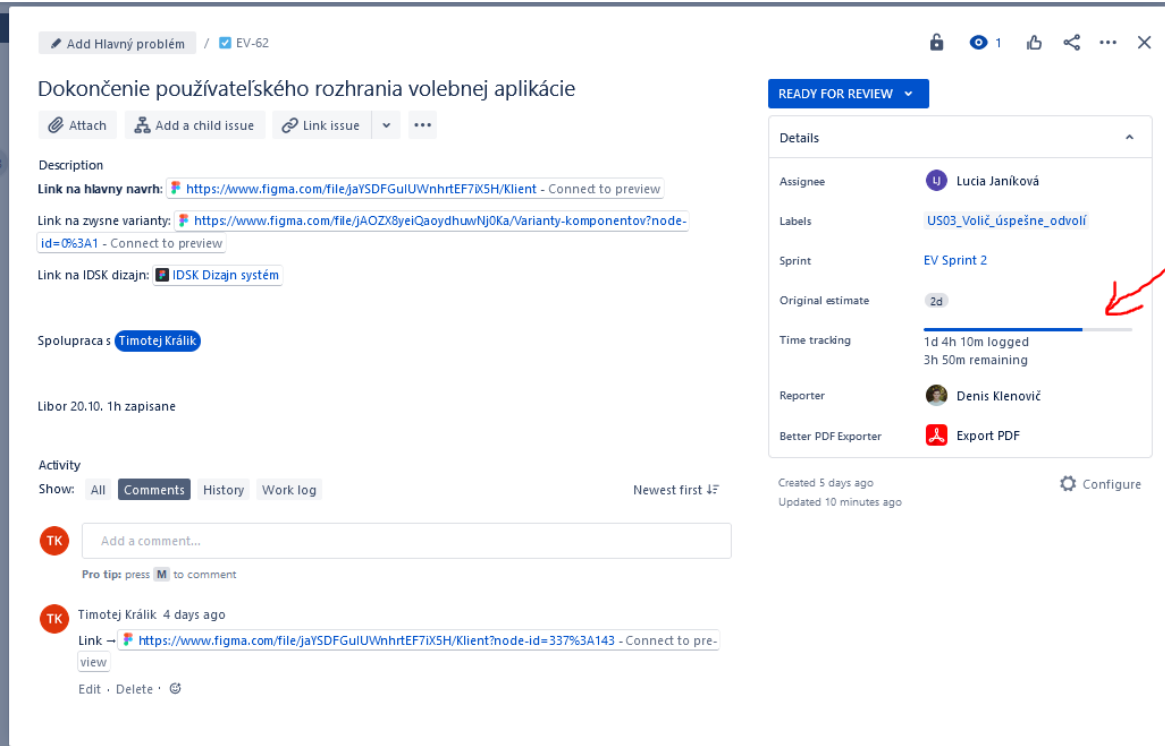
# Metodika - Logovanie času

## Usmernenia k logovaniu času

- Čas do JIRY sa loguje hneď po skončení práce na tasku
- Ak sa z časovej tiesne nepodarí logovať hneď, zalogovať čas je potrebné do konca pracovného dňa
- Všetky tasky **MUSIA** byť zalogované do ceremónie s cvičiacim, inak má scrum master právo použiť na ceremónii trest pre nezodpovedných členov tímu
- Čas sa loguje aj keď je prevýšený pôvodný časový odhad, žiadne skresľovanie aby sa task zmestil do budgetu - objektívne fakty sú dôležitejšie ako falošný pocit úspechu

## Manuál ku logovaniu stráveného času na tasku

Čas sa loguje vždy ku konkrétnemu tasku na ktorom sa pracovalo. Ku logovaniu práce na tasku sa dá dostať rozkliknutím tasku napríklad v sekcii Board. Po otvorení náhľadu je potrebné kliknúť na sekciu **Time tracking**.



The screenshot displays a Jira issue page for 'Dokončenie používateľského rozhrania volebnej aplikácie'. The issue is assigned to Lucia Janiková and is in the 'READY FOR REVIEW' state. The 'Time tracking' section shows 1d 4h 10m logged and 3h 50m remaining. A red arrow points to the 'Time tracking' section in the details panel.

Field	Value
Assignee	Lucia Janiková
Labels	US03_Volič_úspešne_odvolí
Sprint	EV Sprint 2
Original estimate	2d
Time tracking	1d 4h 10m logged 3h 50m remaining
Reporter	Denis Klenovič
Better PDF Exporter	Export PDF

Po kliknutí na sekciu sa otvorí nasledujúci panel.

The screenshot shows a 'Time tracking' dialog box. At the top, there is a progress bar. The left side of the bar is blue and labeled '1d 4h 10m logged'. The right side is grey and labeled '3h 50m remaining'. Below the bar, the text reads 'The original estimate for this issue was 2d'. There are two input fields: 'Time spent' with the value '2w 4d 6h 45m' and 'Time remaining' with the value '3h 50m'. At the bottom right, there are 'Save' and 'Cancel' buttons.

Do políčka **Time spent** sa zapisuje čas v metrikách - w,d,h,m je to čas, ktorý sa na riešení tasku strávil. Logovanie času v JIRE sa riedi jednotkovou sústavou.

- 1w == 5d
- 1d == 8h
- 1h == 60m

Po zadaní času sa okno zväčší a je možné doň napísať popis k vykonanej práci.

### Zadávanie správneho času pri retrospektívnom logovaní

V prípade, že sa čas loguje retrospektívne je potrebné taktiež správne zadať čas práce na danom tasku. JIRA automaticky pri logovaní času nastaví čas práce na tasku na aktuálny čas. Je potrebné súčasný čas zmeniť na ten, v ktorom sa na úlohe skutočne pracovalo.

#### Príklad

Na tasku sa pracovalo medzi 14-15 hodinou a čas sa loguje o 22:13. JIRA automaticky nastaví čas začiatku plnenia tasku na 21:13, čo je zlý čas, nakoľko sa skutočne na ňom pracovalo od 14. Je teda potrebné tento čas zmeniť na skutočnú pracovnú dobu.

### Šablóna k opisnému textu pri logovaní času

Pridávanie popisu k odpracovanému času je **POVINNÉ**. Opis práce, ktorá sa vykonala počas logovaného času, sa píše do textového okna **Work description**.

Time tracking

1d 7h 10m logged 50m remaining

The original estimate for this issue was 2d .

Time spent Time remaining

3h 50m

Date started \*

10/24/2021 4:32 PM

Work description

Aa v B I ... A v :≡ v + v

Type @ to mention a teammate and notify them about this issue.

Save Cancel

Je potrebné opísať krátko, maximálne 3-4 vetami čo sa vykonalo počas logovaného času. Nevypisujú sa tu zbytočne dlhé texty.

#### **Príklad správneho popisu k logovaniu času**

*Dokončil som obrazovky pre komunálne voľby. Spravil som všetky 3 druhy volieb plne klikateľnými prototypmi.*

#### **Príklad nesprávneho popisu k logovaniu času**

*Dokončenie tasku. 1 hodina.*