

Slovenská technická univerzita
Fakulta informatiky a informačných technológií

Dokumentácia riadenia

TP2 - Tím 17

Jakub Hromada
Filip Petrán
Sofia Shatokhina

Adam Novák
Jakub Povinec
Alica Urbanová

FIIT STU, 2022/2023

e-mail: team17timak@gmail.com

Obsah

1 Úvod	1
1.1 Zoznam príloh dokumentu	1
1.2 Členovia tímu a naše povinnosti a úlohy	1
1.3 Rozvrh ZS	2
1.4 Rozvrh LS	3
2 Tímový web	4
2.1 Obsah nášho webu	4
2.2 Projekt	4
2.3 Časová os ZS	5
3 Metodiky	6
3.1 Metodiky scrumu	6
3.1.1 Riadenie v GitLabe	6
3.1.2 Scrum	8
3.2 Metodiky kontroly hotovej práce	10
3.2.1 Review a code review	10
3.3 Metodiky komunikácie	11
3.3.1 Komunikačné prostriedky	11
3.4 Metodiky verziovania (git)	13
3.5 Metodiky kvality	14
3.5.1 GQM	14
3.6 Metodiky kódu	24
3.6.1 Frontend	24
4 Záznamy zo šprintov	26
4.1 Burndown grafy zo šprintov	26
4.1.1 Šprint 1	26
4.1.2 Šprint 2	27
4.1.3 Šprint 3	28
4.1.4 Šprint 4	29
4.1.5 Šprint 5	30
4.2 Prehľad story pointov (Velocity)	31
4.3 Letný semester	31
5 Harmonogram LS	32
Príloha A - Zápisnice	33
Príloha B - Retrospektívy	67
Príloha C - Otázky na externistov	73
Príloha D - Úvodné pokyny	74

1 Úvod

Tento dokument je vytvorený za účelom evidovania všetkého, čo sa netýka samotného projektu, ale nášho tímu, našich zodpovedností, či využívaných metodík. Taktiež bude dokumentácia obsahovať aj informácie o webovom sídle nášho tímu. V samostatnej kapitole bude aj záznam z jednotlivých šprintov a to konkrétne naše burndown grafy a záznam o sory pointoch za všetky šprinty.

Nakoniec vytvoríme aj približný harmonogram práce na letný semester.

1.1 Zoznam príloh dokumentu

- Príloha A - Zápisnice; obsahuje zápisnice zo všetkých stretnutí, ktoré sa konali s vedúcim tímu.
- Príloha B - Retrospektívy; obsahuje retrospektívy za všetky šprinty.
- Príloha C - Otázky na externistov; obsahuje otázky či nejasnosti, ktoré sme potrebovali prebrať so zadávateľmi projektu (teda so zákazníkmi).
- Príloha D - Úvodné pokyny; obsahuje zápis/poznámky zo stretnutia, ktoré sa konalo ešte pred započatím prvého šprintu.

1.2 Členovia tímu a naše povinnosti a úlohy

Aby sme prerozdělili zodpovednosti, pridělili sme každému členovi tímu nejakú rolu už počas prvého šprintu. Každý člen tímu sa ujal svojej zodpovednosti. Taktiež uvediem použité technológie, v ktorých sa daný člen tímu ako tak vyzná a mal za používanie technológie zodpovednosť. Náš zákazník bolo CVTI (ďalej v dokumente označovaný ako zadávateľ projektu či zákazník).

Juraj Vincúr - Vedúci tímu, konzultant, product owner, komunikácia so zákazníkom

Adam Novák - Manažér webu, Front-end, Dev-Ops

Sofia Shatokhina - Scrum master, Front-end, Dev-Ops

Alica Urbanová - Manažérka dokumentácie, Back-end

Filip Petrán - Manažér evidencie úloh, Databáza, Dev-Ops

Jakub Povinec - Manažér testovania, Dev-Ops

Jakub Hromada - Manažér kvality kódu, Back-end, Dev-Ops

1.3 Rozvrh ZS

[illegible]

Obrázok 1.3-1: Rozvrh nášho tímu a voľné časové bloky.

V tabuľke sú uvedené časy voľných okien pre konzultácie s vedúcim a pre spoločnú prácu. Naše pravidelné stretnutia nakoniec prebiehali nasledovne:

- Pondelok 14:00-16:00: spoločné stretnutie tímu
- Utorok 8:00-11:00: stretnutie tímu s vedúcim
- Streda 13:00-15:00: spoločné stretnutie tímu

Každý šprint pozostával z 2 týždňov, a keďže sme začali s miernym oneskorením mali sme 5 šprintov a jeden 1 týždňový šprint. Každý šprint začínal v stredu a skončil v utorok večer o presne dva týždne neskôr.

1.4 Rozvrh LS

		8.00 - 8.50	9.00 - 9.50	10.00 - 10.50	11.00 - 11.50	12.00 - 12.50	13.00 - 13.50	14.00 - 14.50	15.00 - 15.50	16.00 - 16.50	17.00 - 17.50	18.00 - 18.50	19.00 - 19.50
Pondelok	Adam Novák												
	Sofia Shatkhina												
	Alica Urbanová												
	Filip Petrán		OZNAL									OZNAL	
	Jakub Povinec												
	Jakub Hromada												
Utorok	Adam Novák												
	Sofia Shatkhina		OZNAL		GA								
	Alica Urbanová				VD							VD	
	Filip Petrán							NSIETE		TP2			
	Jakub Povinec		PV		OZNAL								
	Jakub Hromada		OZNAL		VD							VD	
Streda	Adam Novák							PMNIS		PMNIS			
	Sofia Shatkhina												
	Alica Urbanová												
	Filip Petrán												
	Jakub Povinec		SMVE		OZNAL		SMVE	PV					
	Jakub Hromada												
Štvrtok	Adam Novák		GA										
	Sofia Shatkhina												
	Alica Urbanová		SMVE										
	Filip Petrán												
	Jakub Povinec								NSIETE		NSIETE		
	Jakub Hromada												
Piatok	Adam Novák												
	Sofia Shatkhina												
	Alica Urbanová												
	Filip Petrán												
	Jakub Povinec												
	Jakub Hromada												
	Legenda:	Prednášky											
		Cvičenia											

Obrázok 1.4-1: Rozvrh nášho tímu a voľné časové bloky.

V tabuľke sú uvedené časy pre konzultácie s vedúcim a pre spoločnú prácu v letnom semestri.

2 Tímový web

Web nášho tímu je prístupný na: <https://team17-22.github.io/homepage/>



Obrázok 2-1: Úvodná stránka nášho tímu.

2.1 Obsah nášho webu

Na našom webe tímu 17 nájdete všetky dokumenty týkajúce sa nášho projektu ako aj samotný projekt, vrátane našich retrospektív či zápisníc alebo využitých metodík.

Dokumenty delíme na dokumentáciu projektu, dokumentáciu riadenia, zápisnice a retrospektívy. Avšak zápisnice a retrospektívy budú taktiež časťou tohto dokumentu (dokumentu riadenia) a to konkrétne ako prílohy A a B a budú taktiež na web-stránke jednotlivo ako samostatné dokumenty. Budú tu taktiež prílohy C (otázky na externistov) a D (úvodné pokyny).

2.2 Projekt

Cieľom projektu je vytvorenie informačno-komunikačného systému, ktorý umožní evidenciu, monitorovanie a správu vedecko-výskumných projektov. Je uskutočňovaný v spolupráci s Centrom vedecko-výskumných informácií.

Daný portal obsahuje zoznam projektov, výskumníkov pracujúcich na daných projektoch a organizácií ku ktorým patria výskumníci. Výskumníci vedú manažovať svoje projekty, zadávajú projekty na schválenie, ich potenciálne výstupy, a priradujú výskumníkov na tieto projekty.

Súlád s CERIFom

Zosúládene s európskym štandardom vedecko-výskumnej domény (CERIF) umožňuje jednoduchú integráciu s inými službami

Cloudové riešenie

Webstránková implementácia je nasadená v Dockeri, čo slúži pre jednoduché nasadenie stránky

Dôraz na UX

Ohľad na používateľa pri robení stránky zabezpečuje jednoduchosť navigácie a estetický vzhľad

2.3 Časová os ZS

Sprint 1 – Kabuki – 11. okt 2022 - 25. okt 2022

V prvom šprinte sme začali analýzu požiadavok očakávaného produktu. Naštudovali sme dokument špecifikácie, rozobrali sme CERIF model a oboznámili sme sa s diagramami z Enterprise Architectu.

Sprint 2 – Watson – 26. okt 2022 - 8. nov 2022

Začali sme analýzu technológií pomocou ktorých budeme implementovať webstránku a ich kompatibilitu. Podľa uskutočnenej analýzy sme prešli k hierarchii obrazoviek a prípadom použitia aplikácie.

Sprint 3 – Arroyo – 9. nov 2022 - 22. nov 2022

Hierarchiu obrazoviek sme pretvorili do samotných mid-fidelity návrhov z hlavných prípadov použitia aplikácie. Taktiež sme vytvorili základnú aplikáciu, ktorú sme dockerizovali.

Sprint 4 – Glen – 23. nov 2022 - 6. dec 2022

Stretli sme sa so zadávateľom projektu a zjednotili sme si predstavu o aplikácii s ním a spolupracujúcim tímom, ako aj spoločnú autentifikáciu. Začali sme vytvárať front-end aplikácie.

Sprint 5 – Downtown – 7. dec 2022 - 13. dec 2022

Zjednotili sme front-end, back-end a databázu pre fungujúce vytvorenie projektu.

3 Metodiky

V rámci projektu sme sa riadili predovšetkým Agile metodikou scrum, no využili sme aj istú metodiku kvality, ktorú si opíšeme bližšie neskôr v samostatnej podkapitole. Stanovením metodík sme docieli zvýšenie kvality nášho projektového manažmentu a teda v konečnom dôsledku aj samotného produktu, ktorý vyvíjame, keďže kvalita vývoja je v lineárnom vzťahu aj s kvalitou konečného vyvíjaného produktu. Kvalitnejší produkt znamená napríklad nižšiu chybovosť, či lepšiu použiteľnosť a práve toho by sme sa mali pokúšať docieľiť.

V nasledujúcich podkapitolách si povieme, ako sme si naše metodiky zadefinovali.

3.1 Metodiky scrumu

Náš projektový manažment sme si riadili v nástroji GitLab. Nástroj je o niečo pokročilejší ako napríklad Trello a jemu podobné. Podobným nástrojom s podobnými funkcionalitami ako GitLab je napríklad ešte Jira. Pre GitLab sme sa rozhodli preto, lebo nástroj sme mohli použiť zdarma, keďže náš vedúci tímu nám ho mohol sprostredkovať a taktiež sme nenašli lepší nástroj. Či už v Jire alebo aj v GitLabe, ktorý sme použili, je nutné vyriešiť niektoré drobné problémy, respektíve si náš nástroj prispôbiť. GitLab napríklad nepodporuje zadávanie story pointov ku epicom (ktoré sme boli nútení používať namiesto user stories, ktoré tu chýbali) a teda sme toto museli robiť manuálne tak, že sme story pointy zaznačili do komentáru k danému epicu.

V tejto kapitole si povieme všetko o tom, ako sme nastavovali GitLab ale taktiež o tom ako sme dodržiavali Agile metodiku scrum, či aký priebeh mali naše priebežné stretnutia.

3.1.1 Riadenie v GitLabe

Riadenie v GitLabe si môžeme rozdeliť na 3 základné časti - user stories, tasky a milestones (používané v našom prípade pre jednotlivé šprinty). Pre obe (user stories a tasky) sme si vytvorili samostatný board. V našom nástroji je podpora pre vytváranie taskov, no user stories sme vytvoriť nemohli a teda sme board pre user stories nahradili boardom pre epicy a tvárili sa, že ide o user stories. Taktiež nejde prepojiť epicy (v našom prípade user stories) s milestones (v našom prípade so šprintami) a teda aj tu sme sa museli vynásť - a to tak, že sme prepojili iba dané tasky ku jednotlivým šprintom/milestones a tasky boli prepojené ku epicom (user stories). Taktiež súčet story pointov všetkých user story (velocity) ku jednotlivým šprintom sme museli spísať manuálne k opisu daného šprintu a story pointy user stories sú spísané v opise daných user stories, preto tu treba odignorovať hodnotu "weight" pri jednotlivých user stories, lebo tá nemá so story pointami nič spoločné, ide iba o súčet original estimatov jednotlivých taskov priradených k danému epicu.

Od tohto momentu budeme v dokumente volať GitLab epicy ako user stories (lebo v GitLab user stories neexistujú a takto sme si ich nahradili) a milestones budeme volať šprinty (lebo aj tie v GitLab chýbajú).

User stories

Pri user stories sme si vytvorili 1 board s tromi stĺpcami - backlog, WIP (work in progress) a closed. Do týchto stĺpcov následne spolu s vedúcim na začiatku každého šprintu plníme Backlog s user stories, ktoré by sa niekedy mali spraviť. Do WIP stĺpca dávame user stories k danému šprintu, ktorý sa bude začínať (nasledujúci deň). Do stĺpca closed pôjdu tie user stories, ktoré sme na konci šprintu spolu s vedúcim uzavreli.

Pre každý jeden user story sme si vytvorili Label, ktorý slúži na rozlíšenie daných user stories. Tieto Labels ďalej priradujeme ku všetkým taskom pre dané user story, keďže chceme vidieť, pod ktorý user story daný task zapadá. Taktiež tu priradujeme labels s menami osôb, ktoré sú poverené, aby spravili k danému user story review, keďže sa tu nedala táto osoba priradiť. K jednotlivým user stories je potom možné priradiť zodpovedajúce tasky a takto vieme prepojiť aj user stories so šprintami, keďže tasky sa ku šprintom priradiť dajú ale user stories nie.

Do opisu user stories dávame informáciu o počte story pointov, ktorú sme si dohodli v rámci planning pokeru a dávame tu taktiež vždy nejaký opis daného user story, DOD (definition of done) a taktiež link na príslušnú dokumentáciu, kde je daný user story zdokumentovaný.

Po rozkliknutí jednotlivých user stories vidíme, aké tasky doň spadajú. A pri boarde s taskami vieme vďaka spomínaným labels, do ktorého user story spadajú.

Tasky

Pri taskoch sme si spravili 1 board, ale je možné tu použiť filter a vyfiltrovať si pre ktorý šprint si chceme zobrazíť tasky. V boarde máme 4 stĺpce - open, WIP, to be reviewed a closed. Stĺpec open slúži na to aby sme tu mali tasky, ktoré ešte neboli priradené k žiadnej osobe a sú naplánované pre daný šprint. v stĺpci WIP sú tasky, ktoré si už niekto zobral na starosť a robí na nich, v stĺpci to be reviewed sú tasky, ktoré čakajú na review poverenej osoby a nakoniec v stĺpci closed sú uzavreté tasky, ktoré prešli kontrolou od človeka zodpovedného za review k danému user story, pod ktorý daný task spadá. K jednému user story vždy prislúcha viacero taskov.

Jednotlivé tasky sú označené labelom podľa toho, ku ktorému user story prislúchajú a taktiež je táto informácia vidieť aj po ich rozkliknutí ako aj šprint, ku ktorému prislúchajú. Po rozkliknutí daných taskov je v description vidieť aj nejakú informáciu k danému tasku.

Taktiež sme v GitLabe k taskom stanovili weight hodnotu, ktorá označuje náš original estimate, čiže čas v hodinách, na ktorom sme sa všetci zhodli, že práca na danom tasku zaberie. ďalej sme tu nastavili vždy aj spent a estimate, to robil človek, ktorý na danom tasku pracoval (popríklad viacerí ľudia). Hodnota estimate značí to, koľko ten človek odhaduje, že daná úloha ešte zaberie a spend je súčet všetkých hodín, ktoré už boli využité na robenie danej úlohy. Obe časy udávame v hodinách.

Jednotlivé tasky si volíme sami, bez vedúceho každú stredu na začiatku šprintu (podľa toho ako to uznáme za vhodné).

Šprinty

Pri jednotlivých šprintoch máme nastavený v GitLabe čas trvania. Každý šprint začína v stredu a končí sa o dva týždne v utorok večer. Ku šprintu sú priradené všetky prislúchajúce tasky a so user story je tento šprint prepojený cez dané tasky.

V opise šprintu je zaznačený súčet všetkých storypointov za daný šprint (velocity) a v každom šprinte máme aj burndown graf, ktorý znázorňuje našu prácu, teda to ako plníme/uzatvárame dané tasky.

Pravidlá

- User story uzatvárame iba spolu s vedúcim, tasky uzatvárame sami, ale jedine v prípade, že prebehlo review.
- V priebehu šprintu nemôžu pribúdať nové tasky, tie si určujeme iba na začiatku.
- Ku každému user story bude priradený minimálne jeden človek na review na prvom stretnutí šprintu.
- Každý bude nastavovať svoje vlastné estimates a spends k jednotlivým taskom, na ktorých pracujú.
- V prípade, že story pointy daného user story presiahnu hodnotu 21 (40 a vyššie), tak delíme daný user story na dve.
- Hodnotu original estimate si dohadujeme spolu.
- Spendy a estimates sa zaokrúhľujú na konci na hodiny.

3.1.2 Scrum

V rámci každého šprintu sme si teda riadili projekt v GitLabe a okrem toho sa vykonávala aj rada iných aktivít, napríklad sme realizovali planning poker vždy na začiatku (respektíve deň pred začiatkom) šprintu, mali sme pravidelné stretnutia ako tím ale aj s vedúcim. Nanešťastie sa nám nepodarilo dohodnúť si pravidelné stretnutia so zadávateľmi projektu. Náhradou za daily standupy bolo to, že sme si každý deň písali správy o tom, čo ešte treba na daných taskoch dorobiť, čo už je hotové a informovali sme sa takto aj o tom, či niektorí členovia tímu nepotrebujú s danou úlohou pomôcť. Taktiež sme takto mali prehľad o tom, či stíhame plniť dané tasky za daný šprint. Samozrejme sme písali aj retrospektívy v prípade, že nám niečo na danom šprinte prekážalo alebo sa nám niečo páčilo a nechýbalo ani robenie zápisníc zo stretnutí. Všetci členovia tímu boli zladený cez komunikačný kanál (discord).

Planning poker

Náš planning poker sme vykonávali pred začatím šprintu pod dozorom vedúceho. Ohodnocovali sme si tak dôležitosť našich user story. Využili sme kartičky na planning poker. Pri zvolenej hodnote 21 sme sa mali začínať zamýšľať nad tým, či daný user story nerozdeliť a na hodnote 40 ho už bolo treba rozdeliť. Priebeh bol taký, že každý z nás vyložil lícom nadol svoju kartičku z balíka a následne sme ju naraz otočili - vybrala sa priemerná hodnota či medián, podľa dohody. Ak niekto mal príliš odlišnú hodnotu ako ostatní a výsledok bol nerozhodný, tak sme debatovali, prečo zvolil takú hodnotu.

Pravidelné stretnutia (ako tím, s vedúcim alebo so zákazníkom)

Priebeh jedného šprintu bol nasledovný:

1. Prvé stretnutie s vedúcim (deň pred začiatkom šprintu) - vždy sme si najskôr prešli všetky splnené úlohy z minulého šprintu a uzavreli sme splnené user story. následne sme pridali do backlogu spolu s vedúcim nové úlohy a tie, čo sa mali riešiť daný nasledujúci šprint sme presunuli do stĺpca WIP. Učili sme si k nim DOD, dali nejaký popis, spravili preň planning poker a zistili story pointy, určili sme aj osobu na robenie review k danému user story. Vytvorili sme nový šprint v GitLab a dali sme do popisu súčet sstory pontov za dané user stories k danému šprintu (velocity). Prediskutovali sme prípadné otázky. Napísali sme retrospektívu pre predchádzajúci šprint. Počas stretnutia sme písali zápisnicu.
2. Prvý deň šprintu - rozdelili sme si user story na tasky a stanovili sme si original estimaty a začali s prácou na úlohách.
3. Počas šprintu sme mali ešte 3 ďalšie 2 hodinové stretnutia spolu ako tím a jedno 3 hodinové stretnutie s vedúcim, kedy sme diskutovali našu prácu, pýtali sa otázky či pracovali na projekte.
4. Mimo stretnutí sme pracovali na úlohách a pravidelne si písali správy a informovali sa o detailoch našej práce.
5. Stretnutia mimo naplánovaný rozvrh - konalo sa aj niekoľko našich stretnutí, mimo nami naplánovaný rozvrh, šlo o stretnutie so zákazníkom (/zadávateľom projektu) a s druhým tímom, ktorý rieši rovnaký projekt, len inú jeho časť.

Daily standupy

Daily stand upy sme, ako som už spomínala, nahradili pravidelným písaním správ, keďže bolo vzhľadom na náš rozvrh, nereálne stretávať sa každý deň. V prípade, ak bolo treba, aby niekto niečo urobil pridali sme to do nášho "to do" kanála na discorde.

Retrospektívy

Pre naše retrospektívy sme mali definovanú presnú šablónu. Napísali sme, čo sa nám na šprinte páčilo, čo by šlo zlepšiť a napísali sme aj nejaké zhodnotenie. Snažili sme sa do nej nepísať veci ohľadom projektu ale skôr ohľadom spôsobu práce tímu, teda nejaké pozitívne alebo negatívne postrehy. Tento dokument sme písali všetci spolu.

Zápisnice

Aj pre zápisnice sme si spravili šablónu - robili sme si zoznam vecí, ktoré máme spraviť (zaznelo to na stretnutí), ďalej sme spísali body stretnutia a napokon samotné poznámky zo stretnutia. Taktiež sme spísali, kto bol prítomný na stretnutí a kedy sa stretnutie konalo. Na písanie zápisníc bol poverený 1 človek z nášho tímu. Ak tento človek ochorel, podujali sa na tom iní (podľa dohody).

Otázky na externistov

Kedže sme nemohli organizovať pravidelné stretnutia so zadávateľmi projektu, spísali sme aspoň naše otázky či nejasnosti, ktoré im mali byť prostredníctvom vedúceho práce sprostredkované. Tento písaný dokument s otázkami bude v prílohách.

3.2 Metodiky kontroly hotovej práce

3.2.1 Review a code review

Tento dokument opisuje to, akým spôsobom poverený človek kontroloval dokončené úlohy. Tento človek je zvolený vždy na začiatku šprintu, spoločným dohovorom, pre dané konkrétne user story. V prípade, že tento človek pracoval na danej úlohe, tak musí poveriť ešte sám niekoho ďalšieho, kto spraví review za neho.

Tento človek nekontroluje iba kód ale aj dokumentáciu a všetky súvisiace artefakty. Je zvolený na základe toho, že je na kontrolu danej úlohy najkompetentnejší z celého tímu - napríklad má prehľad v danej technológii alebo z predchádzajúcich skúseností vieme, že je najlepšie práve jeho zvoliť na túto zodpovednosť.

Povinnosti reviewera

- Prezrie dokumentáciu a zhodnotí, či obsahuje všetko, čo prislúcha k danej úlohe/tasku (podľa dohodnutej definície danej úlohy, ktorá vyplynula zo spoločného dohovoru tímu).
- Prezrie kód (ak bol nejaký vyprodukovaný pre splnenie daného tasku). Pozrie, či je kód funkčný a či spĺňa všetko, čo prislúcha k danej úlohe/tasku (podľa dohodnutej definície danej úlohy, ktorá vyplynula zo spoločného dohovoru tímu). Taktiež zhodnotí, či je kód čitateľný, či obsahuje komentáre a či sa riadi štandardne zaužívanými postupmi písania kódu. Doposiaľ avšak nebolo veľa user stories, kde by trebalo aj niečo implementovať.
- Zhodnotí aj ostatné vyprodukované artefakty súvisiace so splnením danej úlohy podľa rovnakých kritérií ako má písanie kódu a dokumentácie.
- V prípade nesúladu oboznámi človeka, ktorí sa podieľali na danej úlohe o tom, čo treba zlepšiť/prerobiť/dopísať/pridať.

Tento človek začne vykonávať review po tom ako pravidelnou kontrolou GitLabu zistí, že daná úloha, ktorou bol poverený bola zaradená v boarde taskov do stĺpca *to be reviewed*.

Pravidlá

- Review je nutné vykonať ešte aspoň 24 hodín pred ukončením šprintu.
- Ak nejaká úloha nie je 24 hodín pred ukončením šprintu zaradená do kategórie *to be reviewed*, je úlohou reviewera na to upozorniť osobu zodpovednú za danú úlohu.
- Len reviewer presúva daný task v GitLabe zo stĺpca *to be reviewed* do stĺpca *closed* a to v prípade, že spĺňa všetky spomínané požiadavky.
- Ak tento človek nie je spokojný s výsledkom danej úlohy, tak ju vráti do WIP stĺpca a informuje autora. Ak ju aktor dorobí a presunie do stĺpca *to be reviewed* znova, tak je nutné ju znova skontrolovať.

3.3 Metodiky komunikácie

Komunikovali sme spolu buď prostredníctvom nejakých komunikačných kanálov priamo alebo aj nepriamo (písanie si, volania, komunikácia v Mire prostredníctvom poznámok) alebo aj prezenčne, čiže sme sa spolu aj fyzicky stretávali.

Dali sme priestor každému vyjadriť svoj názor a na všetkom sme sa dohodli kolektívne ako tím.

3.3.1 Komunikačné prostriedky

Používali sme viaceré komunikačné prostriedky, či už na dohadovanie sa medzi sebou, s vedúcim, poprípade s druhým tímom, ktorí riešil inú časť projektu od tých istých zadávateľov.

Discord (interná komunikácia)

V rámci tohto četu sme viedli skupinové konverzácie v tíme, ale riešili aj veci pomimo, čiže išlo o viac "casual" komunikáciu a bavili sme sa tu aj o veciach, s ktorými sme nechceli otravovať vedúceho. No tento kanál poslúžil k našej vzájomnej spolupráci a k stužovaniu tímu.

Discord (komunikácia s druhým tímom a s vedúcim)

Na tento účel sme mali vytvorený samostatný server na discorde s viacerými kanálmi:

- **general** - Riešili sme tu primárne komunikáciu s našim vedúcim. Snažili sme sa byť viac vecní a riešiť tu podstatné otázky.
- **links** - Na tomto linku boli dôležité linky v rámci nášho projektu, napríklad link na Figma, v ktorej sú všetky naše návrhy obrazoviek.
- **resources** - V tomto kanáli sa nachádzajú dôležité zdroje, napríklad je tu link na špecifikáciu projektu od zadávateľov, či stránka s podobným projektom, aký riešime, na inšpiráciu (skcrisis.sk).
- **reports** - V tomto kanáli vždy pribudne zápisnica v podobe pdf za posledné stretnutie.
- **gitlab logs** - Tento kanál bol pôvodne "to-do list", lenže pribudol tu náš bot, ktorý sleduje dianie na GitLabe a hlási nám všetky zmeny, takto každú zmenu v GitLab hneď uvidíme v notifikácií z discordu.
- **to-do** - Na tomto kanáli riešime napríklad to, čo bude treba prebrať na stretnutiach.
- **komunikacia** - Ide o kanál určený pre komunikáciu s druhým tímom, keďže riešime ten istý projekt od zadávateľov a teda je nutné prediskutovať určité záležitosti.
- **Voice channel General** - Tento kanál sme využívali na volania si s tímom alebo s vedúcim.

Miro

Aj prostredníctvom Mira sme sa dohadovali na výsledných návrhoch obrazoviek a pridali sme tu ku každej obrazovke nejaké poznámky, ktoré sme chceli zapracovať.

Figma

Dostali sme taktiež link na Figma druhého tímu, aby sme s nimi vedeli zladiť dizajn a taktiež máme vlastnú Figma, kde sme všetci spolu riešili vzhľad obrazoviek.

Google meet (záloha)

Tento komunikačný kanál sme využili v prípade nefunkčnosti Discordu, keď sme si potrebovali zavolať.

E-mail

Cez náš tímový mail sme komunikovali napríklad s vedúcimi predmetu TP1. Náš tímový mail je: team17timak@gmail.com.

Otázky na externistov

Do 10. týždňa semestra sme sa so zadávateľmi nestretli, tak sme na nich aspoň takto spísali otázky. Tieto otázky im mal vedúci tímu sprostredkovať. Dokument s týmito otázkami je vložený ako príloha tohto dokumentu.

3.4 Metodiky verziovania (git)

Jazyk

- Celý git je v angličtine (branche, commity etc).
- URL odkazy budu písané v slovenčine bez diakritiky.

Branches

- Unit testy budú na tom istom branchi ako aj implementácia feature.
- Pomenovanie:
 - **feature**/pomenovanie-feature
 - **bugfix**/pomenovanie-bugfixu
 - **refactor**/pomentovanie-refactoru

Commits

- Nemusia mať zjednotený format (stačí simple komentár).

Merging to main

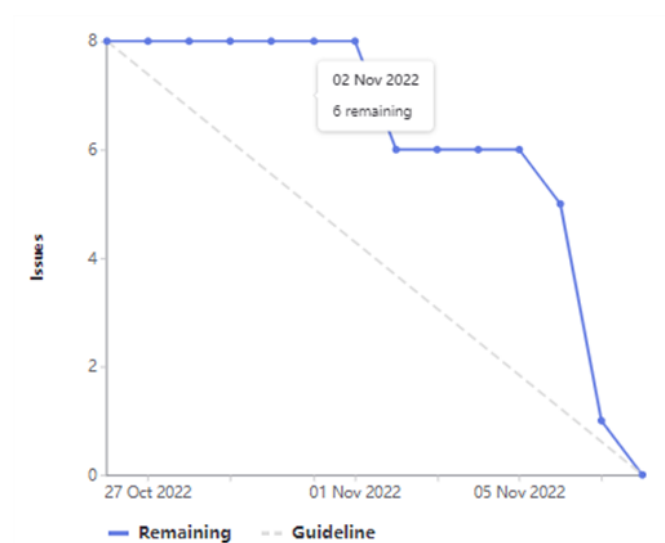
- *Pull main to your own branch*
- *Merge (+ resolve conflicts)*
- *Submit merge request*

3.5 Metodiky kvality

3.5.1 GQM

Autor: Alica Urbanová (Projekt bol vytváraný v rámci predmetu KPAIS_I)

V rámci predmetu KPAIS_I jedna z členov tímu aplikovala metodiku kvality GQM na náš tímový projekt, respektíve na náš projektový manažment. Konkrétne na problém, ktorý bol skôr záležitosťou prvých 2 šprintov, kedy sme ešte neboli úplne zabehnutí. Problém spočíval v tom, že celé prvé stretnutie šprintu sme strávili iba nastavovaním GitLabu a nevenovali sa produktívnej práci, čo mohlo zapríčiniť čiastočne aj to, že sme mali pomalý "rozjazd" šprintu a prvý týždeň sme nedosiahli žiadne pokroky.



Obrázok 3.5.1-1: „Burndown“ graf nášho druhého šprintu.

Prvé stretnutie prvých dvoch šprintov (stretnutie bez vedúceho) vyzeralo takto:



Obrázok 3.5.1-2: Priebeh prvého stretnutia šprintu (za 1-2 šprint).

Ideálne sme chceli dosiahnuť takýto stav:



Obrázok 3.5.1-3: Ideálny priebeh prvého stretnutia šprintu.

Aby sme tento stav docielili zaviedli sme spomínanú metodiku GQM. Ide o proces, ktorý by mal vyriešiť naše problémy tak, že si stanovíme ciele, k nim si položíme relevantné otázky a na tieto otázky sa budeme pokúšať odpovedať v podobe metrík. Následne zmeriame tieto stanovené metriky, budeme ich analyzovať a nakoniec vyvodíme nejaké zistenia a zavedieme opatrenia. Následne možno metriky zmerať znova a zistiť, či sa naše problémy vyriešili. Všetky merania metrík času budú uvádzané v hodinách.

Zavedenie cieľov

Problém	Prvé spoločné stretnutie šprintu je celé venované nastavovaniu nástroja na náš projektový manažment a nie produktívnej práci na úlohách. Takže 2 hodiny času strávime neproduktívnou prácou.
Účel	Pretransformovať/ minimalizovať
Problém	Čas strávený nastavovaním GitLabu
Objekt	V prvom stretnutí šprintu
Uhol' pohľadu	Z pohľadu celého nášho tímu (nie jednotlivca)
Cieľ 1	Minimalizovať čas strávený nastavovaním GitLabu v prvom stretnutí šprintu.
Upresnenie cieľa	Skrátiť čas nastavovania GitLabu na 1 štvrtinu pôvodného času. Cieľ je definovaný vzhľadom na spoločnú prácu tímu, nie vzhľadom na jednotlivcov.

Účel	Maximalizovať
Problém	Čas (ne)trávený produktívnou prácou na úlohách šprintu
Objekt	V prvom stretnutí šprintu
Uhol pohľadu	Z pohľadu celého nášho tímu (nie jednotlivca)
Cieľ 2	Maximalizovať čas strávený produktívnou prácou v prvom stretnutí šprintu.
Upresnenie cieľa	Zvýšiť čas produktívnej práce na 3 štvrtiny času v prvom stretnutí šprintu. Cieľ je definovaný vzhľadom na spoločnú prácu tímu, nie vzhľadom na jednotlivcov.

Tabuľka 3.5.1-1: Formálne definovanie cieľov.

Zavedenie otázok a metrík

Otázka 1	Koľko času strávime nastavovaním GitLabu?
Metriky	<p>M1: Počet hodín strávených nastavovaním GitLabu pred prvým stretnutím šprintu.</p> <p>M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.</p> <p>M3: Počet ľudí v tíme, ktorí sa podieľajú na nastavovaní GitLabu. (Okrem rozdeľovania „user stories“ na úlohy, lebo to robíme spolu. Sú na mysli aktivity, ktoré môže vykonávať aj jednotlivec bez prítomnosti iných členov tímu.)</p> <p>M4: Počet hodín potrebných na nastavenie GitLabu (všetci spolu, čiže to, čo by nezvládol jednotlivec).</p> <p>M5: Počet hodín potrebných na nastavenie GitLabu (jednotlivec, teda to, čo dokáže spraviť aj jeden človek v tíme samostatne).</p>
Otázka 2	Trávime behom prvého stretnutia v šprinte čas produktívnou prácou na úlohách pre daný šprint?

Metriky	<p>M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.</p> <p>M3: Počet ľudí v tíme, ktorí sa podieľajú na nastavovaní GitLabu. (Okrem rozdeľovania „user stories“ na úlohy, lebo to robíme spolu.)</p> <p>M6: Počet hodín strávených produktívnou prácou na úlohách behom prvého stretnutia šprintu.</p> <p>M7: Počet ľudí, ktorí sa behom prvého stretnutia šprintu venujú produktívnej práci na úlohách.</p> <p>M8: Počet hodín strávených „ničnerobením“ behom prvého stretnutia šprintu. (Vypočítame podľa vzorca individuálne pre každého jedinca to, koľko času trávil inak ako produktívnou prácou na úlohách či nastavovaním GitLabu a následne tieto hodnoty spočítame. Údaj je braný len v rámci 2 hodinového časového bloku nášho stretnutia, nadčasy sa pripočítavať nebudú.)</p> <p>M9: Počet voľných hodín prvého stretnutia šprintu, ktoré je možné vyplniť produktívnou prácou. (prípád, kedy stretnutie ukončíme skôr alebo ho dočasne prerušíme.)</p>
Otázka 3	Z čoho pozostáva naše prvé stretnutie v šprinte?
Metriky	<p>M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.</p> <p>M6: Počet hodín strávených produktívnou prácou na úlohách behom prvého stretnutia šprintu.</p> <p>M9: Počet voľných hodín prvého stretnutia šprintu, ktoré je možné vyplniť produktívnou prácou. (prípád, kedy stretnutie ukončíme skôr alebo ho dočasne prerušíme.)</p> <p>M10: Počet hodín strávených inou aktivitou než nastavovaním GitLabu a produktívnou prácou na úlohách behom prvého stretnutia šprintu. (ako tím, nie ako jednotlivec)</p> <p>M11: Reálny počet hodín nášho stretnutia.</p> <p>M12: počet hodín strávených čakaním na započatie prvého stretnutia šprintu. („delay“ od plánovaného termínu začiatku stretnutia.)</p>
Otázka 4	Prípravujeme sa na prvé stretnutie šprintu? (Prípravou mám na mysli len GitLab veci.)

Metriky	<p>M1: Počet hodín strávených nastavovaním GitLabu pred prvým stretnutím šprintu.</p> <p>M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.</p> <p>M5: Počet hodín potrebných na nastavenie GitLabu (jednotlivec, teda to, čo dokáže spraviť aj jeden človek v tíme samostatne).</p>
Otázka 5	Aké úsilie venujeme behom prvého stretnutia šprintu práci na projekte?
Metriky	<p>M13: Počet ľudí tímu, ktorí behom prvého stretnutia v šprinte nerobia nič v rámci projektu. (Čiže nevenujú sa aktívnej práci na úlohách alebo nastavovaniu gitlabu. Nepočítam aktivity, ktoré musíme behom tohto stretnutia robiť všetci spoločne v rámci GitLabu.)</p> <p>M3: Počet ľudí v tíme, ktorí sa podieľajú na nastavovaní GitLabu. (Okrem rozdeľovania „user stories“ na úlohy, lebo to robíme spolu.)</p> <p>M7: Počet ľudí, ktorí sa behom prvého stretnutia šprintu venujú produktívnej práci na úlohách.</p> <p>Ďalšie súvisiace metriky: M2, M6, M8, M9, M10, M12</p>

Tabuľka 3.5.1-2: Zavedenie otázok a metrík.

Namerané metriky

Metrika	1. šprint	2. šprint	Priemer
M1: Počet hodín strávených nastavovaním GitLabu pred prvým stretnutím šprintu.	0.0	0.0	0.0
M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.	2.666	2.0	2.333
M3: Počet ľudí v tíme, ktorí sa podieľajú na nastavovaní GitLabu. (Okrem rozdeľovania „user stories“ na úlohy, lebo to robíme spolu. Sú na mysli aktivity, ktoré môže vykonávať aj	4	2	3

jednotlivec bez prítomnosti iných členov tímu.)			
M4: Počet hodín potrebných na nastavenie GitLabu (všetci spolu, čiže to, čo by nezvládol jednotlivec, majú sa na mysli spoločné aktivity pri nastavovaní GitLabu).	0.8333	0.5	0.666
M5: Počet hodín potrebných na nastavenie GitLabu (jednotlivec , teda to, čo dokáže spraviť aj jeden človek v tíme samostatne).	1.8333	1.5	1.666
M6: Počet hodín strávených produktívnou prácou na úlohách behom prvého stretnutia šprintu.	0.0	0.0	0.0
M7: Počet ľudí, ktorí sa behom prvého stretnutia šprintu venujú produktívnej práci na úlohách.	0	0	0
M8: Počet hodín strávených „ničnerobením“ behom prvého stretnutia šprintu. (Vypočítame podľa vzorca individuálne pre každého jedinca to, koľko času trávil inak ako produktívnou prácou na úlohách či nastavovaním GitLabu a následne tieto hodnoty spočítame. Čas je nameraný len v rámci nášho 2 hodinového bloku pre stretnutie, neráta sa nadčas. Nie sú tu taktiež zarátané časové bloky, ktoré sme strávili čakaním.)	4.0 (2*2.0)	8.0 (4*2.0)	6.0
M9: Počet voľných hodín prvého stretnutia šprintu, ktoré je možné vyplniť produktívnou prácou. (napríklad prípad, kedy stretnutie ukončíme skôr alebo ho dočasne prerušíme.)	0.0	0.0	0.0
M10: Počet hodín strávených inou aktivitou než nastavovaním GitLabu a produktívnou prácou na úlohách behom prvého stretnutia šprintu. (ako tím, nie ako jednotlivec. Poznámka:	0.0	0.0	0.0

neráta sa čas, kedy sme stretnutie ukončili skôr.)			
M11: Reálny počet hodín nášho stretnutia.	2.666	2.0	2.333
M12: počet hodín strávených čakaním na započatie prvého stretnutia šprintu. („delay“ od plánovaného termínu začiatku stretnutia.)	0.0833	0.5	0.2916
M13: Počet ľudí tímu, ktorí behom prvého stretnutia v šprinte nerobia nič v rámci projektu. (Čiže nevenujú sa aktívnej práci na úlohách alebo nastavovaní gitlabu. Nepočítam aktivity, ktoré musíme behom tohto stretnutia robiť všetci spoločne v rámci GitLabu.)	2	4	3

Tabuľka 3.5.1-3: Namerané metriky.

Navrhnuté opatrenia

Opatrenie 1	<p>Opis nedostatku: Jedným z nedostatkov bolo, že sme netrávili žiaden čas prípravou na prvé stretnutie šprintu, kedy by sme mohli prednastaviť GitLab, aby sme na stretnutí už nestrácali čas tým, čo môže spraviť aj jednotliviec. Konkrétne mám na mysli veci, ktoré nemusíme robiť všetci spolu (ako rozdeľovanie „user stories“ na tasky, ktoré realizujeme všetci).</p> <p>Opatrenie: Ako opatrenie preto navrhujem, aby aspoň jeden jednotliviec strávil čas týmto nastavením GitLabu ešte pred prvým stretnutím a aby sa na samotnom stretnutí riešili ohľadom GitLabu len tie veci, ktoré by sa mali nevyhnutne riešiť ako tím.</p> <p>Očakávaný dopad zavedenia opatrenia: Takto by sme mali obmedziť čas strávený nastavovaním GitLabu na prvých stretnutiach šprintov na minimum a tým nám vznikne viac voľného priestoru na to, aby sme sa aktívne venovali práci na úlohách.</p>
-------------	---

Opatrenie 2	<p>Opis nedostatku: Problematické, hlavne pri prvom stretnutí druhého šprintu, bolo že len 2 ľudia trávili čas nastavovaním GitLabu a zvyšok riešil veci netýkajúce sa či už nastavovania GitLab alebo produktívnej práce na úlohách. Behom prvého stretnutia v prvom šprinte zas trávili až 4 nastavovaním GitLabu a zvyšok nerobil nič týkajúce sa práce na úlohách, či GitLabu.</p> <p>Opatrenie: Preto ako druhé opatrenie navrhujem, aby GitLab nastavovalo, čo možno najmenej ľudí tímu, ak ide o veci, ktoré môže spraviť aj jednotlivec samostatne, a zvyšok by sa mohol venovať produktívnej práci na úlohách.</p> <p>Očakávaný dopad zavedenia opatrenia: Takto by sme mali doceliť to, že viac ľudí bude mať na prvom stretnutí šprintu voľný čas, ktorý môže venovať práve produktívnej práci na úlohách. Taktiež navrhujem, aby tento čas netrávili inými aktivitami, nesúvisiacimi s projektom.</p>
Opatrenie 3	<p>Opis nedostatku: Problém bol aj ten, že nie všetci prichádzali na stretnutie včas, lebo mali predtým iné aktivity. To spôsobilo, že stretnutie sa predĺžilo a my ostatní sme museli čakať, kým budeme môcť riešiť nastavovanie niektorých vecí v GitLabe, ktoré realizujeme ako tím.</p> <p>Opatrenie: Preto ako ďalšie opatrenie navrhujem, aby sa stretnutia posunuli o tých pár minút, poprípade, aby sme sa predom dohodli pred daným stretnutím, že kedy presne sa stretneme.</p> <p>Očakávaný dopad zavedenia opatrenia: Takto by sme mali aspoň trochu minimalizovať dobu čakania na zvyšok tímu.</p>

Tabuľka 3.5.1-4: Navrhnuté opatrenia.

Namerané metriky po nasadení opatrení

Metrika	Priemer za 1. a 2. šprint	3. šprint
M1: Počet hodín strávených nastavovaním GitLabu pred prvým stretnutím šprintu.	0.0	0.75

M2: Počet hodín strávených nastavovaním GitLabu behom prvého stretnutia šprintu.	2.333	0.583
M3: Počet ľudí v tíme, ktorí sa podieľajú na nastavovaní GitLabu. (Okrem rozdeľovania „user stories“ na úlohy, lebo to robíme spolu. Sú na mysli aktivity, ktoré môže vykonávať aj jednotlivec bez prítomnosti iných členov tímu.)	3	1
M4: Počet hodín potrebných na nastavenie GitLabu (všetci spolu, čiže to, čo by nezvládol jednotlivec, majú sa na mysli spoločné aktivity pri nastavovaní GitLabu).	0.666	0.583
M5: Počet hodín potrebných na nastavenie GitLabu (jednotlivec , teda to, čo dokáže spraviť aj jeden človek v tíme samostatne).	1.666	0.75
M6: Počet hodín strávených produktívnou prácou na úlohách behom prvého stretnutia šprintu.	0.0	0.1666
M7: Počet ľudí, ktorí sa behom prvého stretnutia šprintu venujú produktívnej práci na úlohách.	0	2
M8: Počet hodín strávených „ničnerobením“ behom prvého stretnutia šprintu. (Vypočítame podľa vzorca individuálne pre každého jedinca to, koľko času trávil inak ako produktívnou prácou na úlohách či nastavovaním GitLabu a následne tieto hodnoty spočítame. Čas je nameraný len v rámci nášho 2 hodinového bloku pre stretnutie, neráta sa nadčas. Nie su tú taktiež zarátané časové bloky, ktoré sme strávili čakaním.)	6.0	11.666 (4*2.0+2*1.833)

M9: Počet voľných hodín prvého stretnutia šprintu, ktoré je možné vyplniť produktívnou prácou. (napríklad prípad, kedy stretnutie ukončíme skôr alebo ho dočasne prerušíme.)	0.0	1.416
M10: Počet hodín strávených inou aktivitou než nastavovaním GitLabu a produktívnou prácou na úlohách behom prvého stretnutia šprintu. (ako tím, nie ako jednotliviec. Poznámka: neráta sa čas, kedy sme stretnutie ukončili skôr.))	0.0	0.0
M11: Reálny počet hodín nášho stretnutia.	2.333	0.583
M12: počet hodín strávených čakaním na započatie prvého stretnutia šprintu. („delay“ od plánovaného termínu začiatku stretnutia.)	0.2916	0.033
M13: Počet ľudí tímu, ktorí behom prvého stretnutia v šprinte nerobia nič v rámci projektu. (Čiže nevenujú sa aktívnej práci na úlohách alebo nastavovaniu gitlabu. Nepočítam aktivity, ktoré musíme behom tohto stretnutia robiť všetci spoločne v rámci GitLabu.)	3	4

Tabuľka 3.5.1-5: Namerané metriky po zavedení opatrení.

Zhrnutie

Z meraní vidíme pokrok, ktorý zavedenie GQM prinieslo. Splnil sa nám náš 1 stanovený cieľ a to, aby sme skrátili nastavovanie GitLab na prvom stretnutí šprintu o 3 štvrtiny. Avšak druhý cieľ sme nesplnili, keďže sme tento uvoľnený čas nevyužili produktívnou prácou - keďže boli sme v tom, že prácu naplánovanú na dané stretnutie máme hotovú. Tento cieľ sme avšak naplnili už v ďalších šprintoch, tak môžeme to zhodnotiť tak, že sme takto vyriešili jeden z našich problémov a optimalizovali našu prácu.

3.6 Metodiky kódu

Metodiky kódu obsahujú súbor pravidiel na zjednotenie kódu a zjednodušenie práce pomedzi všetkých členov tímu, ktorí sa zaoberajú implementáciou prototypu.

Táto príručka platí pre implementáciu kódu bez použitia Bootstrapu (čiže v súčasnosti je už zväčša neaktuálna). Po prejení na implementáciu pomocou Bootstrapu základnou metodikou kódu pre frontend je zhrnutie najlepších praktík¹ pre tento framework tvorcami Bootstrapu.

3.6.1 Frontend

Štruktúra priečinkov komponentov

Každý komponent je tvorený jedným priečinkom, ktorý obsahuje **.js** súbor so samotnou implementáciou komponentu a **.css** súbor so štylovaním komponentu.

Dôvodom je jednoduchšia navigácia medzi komponentami a ich štýlmi (v prípade, že chceme upraviť štýl komponentu, nemusíme potrebný súbor hľadať v úplne inom priečinku) ako aj údržba komponentov (ak je všetko pokope, úprava štruktúry komponentov či vymazanie nepotrebných komponentov a patriacich im štýlom je omnoho jednoduchšia, než pri rozhádzaných súboroch).

Tieto komponentové priečinky sú usporiadané do nasledovnej štruktúry:

- **buttons** – tlačidlá, ako napr. basic *Button* alebo navigačné buttons v pipeline vytvorenia projektu (*ProjectCreationPipeline*)
- **tables** – tabuľky
- **userInput** – komponenty, ktoré poskytnú aplikácii používateľský vstup (napr. *TextArea*, zadanie kľúčových slov alebo *Dropdown* menu)
- **misc** – ostatné komponenty, ktoré sa nehodia do vyššie spomenutých priečinkov

Štruktúra priečinkov štýlov

Priečinok štýlov (`/frontend/styles/`) obsahuje štýly, ktoré sa neviažu na komponenty.

Nachádzajú sa tam štýly jednotlivých stránok aplikácie (v priečinku

`/frontend/styles/pages/`) a globálne štýly (ako napríklad fonty písma).

Štylovanie komponentov

Štylovanie komponentov bude mať oddelenú logiku samotného štýlu a layoutu komponentov. Vznikla ako výsledok analýzy návrhu spôsobu oddelenia layoutu a štýlu (podrobnejšie popísaná v kapitole 13 dokumentácie projektu).

Aj štýl, aj layout komponentov sa budú nachádzať v rovnakom súbore a pod jednou triedou, avšak budú mať svoje podtriedy **.style** a **.layout** pre každý komponent. V layoutovej zložke sa bude popisovať rozloženie komponentu a jeho vnútorných častí. Štýlová zložka sa bude

¹ <https://getbootstrap.com/docs/4.0/getting-started/best-practices/>

zaoberať estetickou stránkou – farbami, fontami, atď. Príklad takéhoto komponentu je vidieť na obrázku nižšie.

```
.button, .layout {
  display: flex;
  align-items: center;
  justify-content: space-around;
  cursor: pointer;
}

.button, .style {
  background-color: #d9d9d9;
  color: black;
  font-weight: 700;
  border: 1px solid black;
  text-decoration: none;
}

.button:hover {
  background-color: #b4b1b1;
}
```

Obrázok 3.6.1-1: Kód štýlu *Button.module.css*.

Písanie komponentov

Jednotlivé komponenty majú mať zjednotený systém importovania súborov pre jednoduchšiu navigáciu.

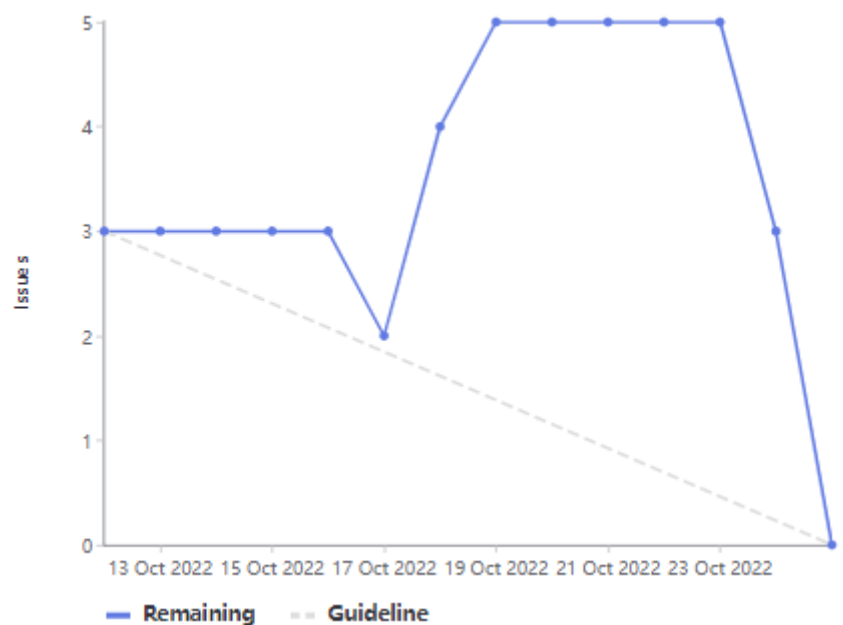
- Každý import by mal obsahovať celú cestu k importovanému súboru (napr. `/components/misc/Header`), **nie** odkazovať sa na priečinky cez bodky (`/../../Header`)
- Poradie importov:
 - Externé importy
 - next, react komponenty
 - ostatné (napr. *date picker*)
 - Interné importy (nami vytvorené súbory)
 - komponenty (.js)
 - štýly (.css)
- Ak je importov viac ako 5, treba pridať prázdny riadok medzi skupiny importov pre lepšiu čitateľnosť
- Dva prázdne riadky po importoch

4 Záznamy zo šprintov

Na burndown grafoch z našich šprintov je vidieť, aký pokrok sme robili v plnení našich stanovených taskov v priebehu daných šprintov. Čiara na grafe stúpane vždy, ak pridáme v priebehu šprintu nejaký nový task a klesne, len až ho uzatvoríme. Tieto grafy teda hovoria o priebehu našej práce počas jednotlivých šprintov, i keď samozrejme neprezeradia detaily.

4.1 Burndown grafy zo šprintov

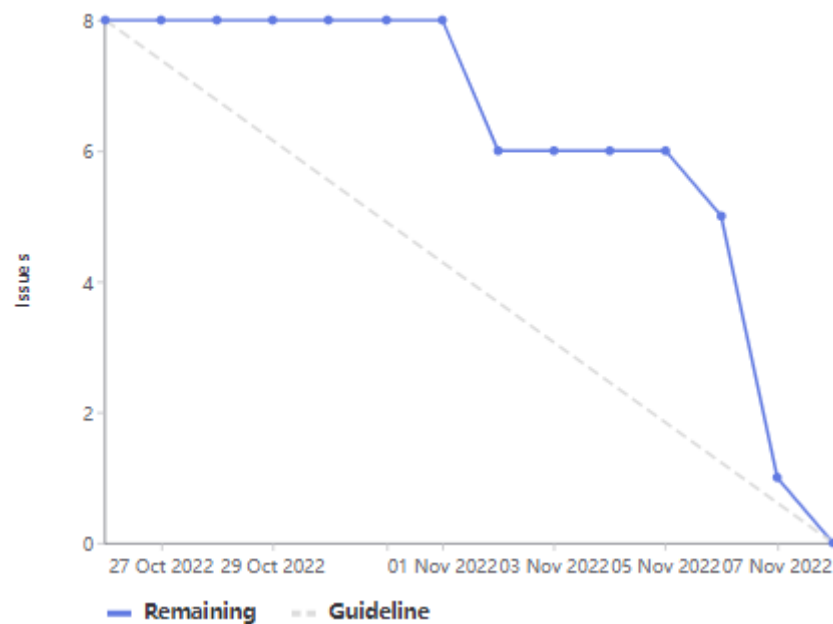
4.1.1 Šprint 1



Obrázok 4.1.1-1: Burndown graf zo šprintu 1.

Počas prvého šprintu sme neboli veľmi zabehnutí a teda to skončilo tak, že sme si tasky pridávali aj počas šprintu a dopadlo to tak ako to dopadlo, čiže náš burndown graf o dosť vyskočil, ale poučili sme sa. Dalo by sa povedať, že výsledok je očakávaný, keďže išlo iba o prvý šprint.

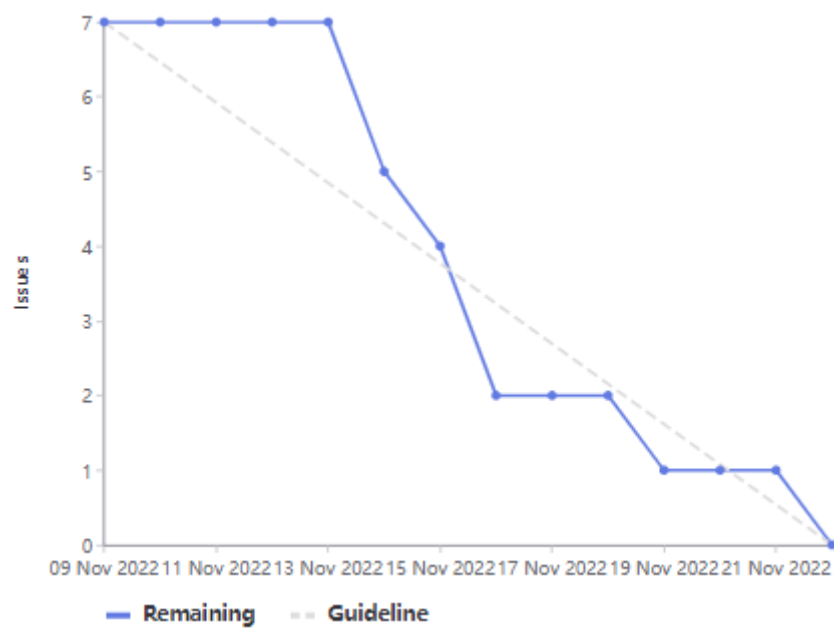
4.1.2 Šprint 2



Obrázok 4.1.2-1: Burndown graf zo šprintu 2.

V druhom šprinte je na našom burndown grafe vidieť už pokrok oproti prvému šprintu, keďže sme sa poučili a už sme si nepridávali tasky behom šprintu, ale iba na začiatku. No je vidieť, že sme prvý týždeň moc nepokročili a druhý týždeň sme dokončili tasky nárazovo. Avšak situácia nebola až taká zlá, len sme mali rozrobených viac taskov naraz a išlo to mierne pomalšie.

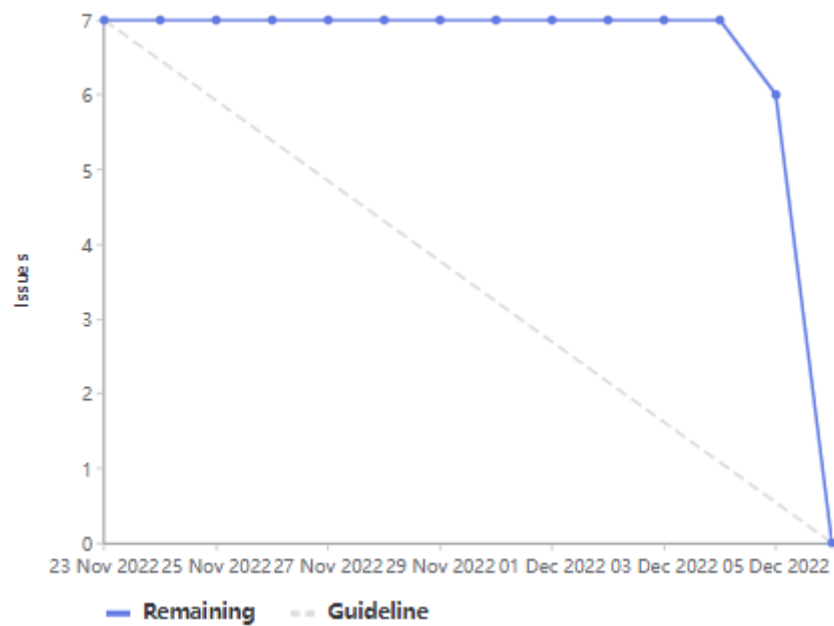
4.1.3 Šprint 3



Obrázok 4.1.3-1: Burndown graf zo šprintu 3.

Oproti predošlým dvom šprintom, tento prebiehal ideálne. Zrejme to bolo tým, že už sme boli viac zabehnutí do našej práce a mali sme voľnejší týždeň, čo sa týka ostatných predmetov.

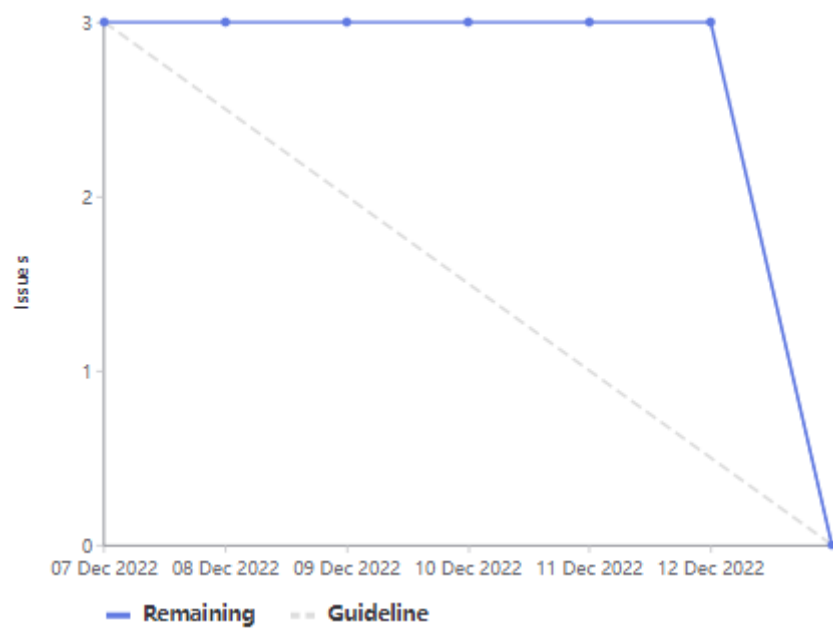
4.1.4 Šprint 4



Obrázok 4.1.4-1: Burndown graf zo šprintu 4.

Burndown krivka tohto šprintu nebola priblížená očakávanej trajektórii oproti predošlým šprintom. No môže za to zrejme fakt, že viacerí sme boli v tejto dobe chorí a komunikácia bola teda o niečo horšia. Viacero taskov bolo rozrobených naraz a uzavretých až na konci. No i napriek tomu sme dokončili, čo sme chceli dokončiť daný šprint. Taktiež sme toho mali veľa na iných predmetoch.

4.1.5 Šprint 5

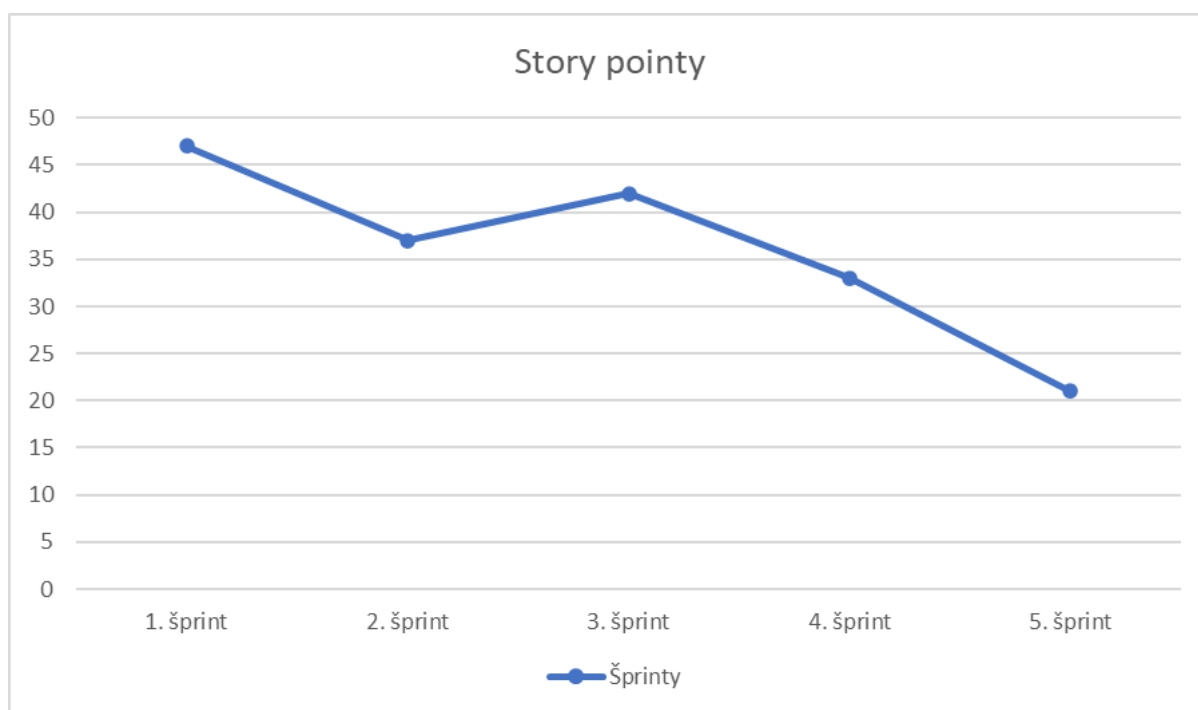


Obrázok 4.1.5-1: Burndown graf zo šprintu 5.

Piaty šprint bol kratší o 1 týždeň a práve preto burndown graf nevyzerá veľmi dobre. Taktiež sme v tento šprint mali veľa práce na iných predmetoch.

4.2 Prehľad story pointov (Velocity)

Na grafe nižšie je znázornený počet súčtu všetkých story pointov všetkých user story za daný šprint, čiže ide o údaj "velocity". Treba tu ale podotknúť, že do 4. šprintu sme boli nútení presunúť nejakú časť tasku z 3. šprintu a story pointy za dorobenie tohto tasku nie sú započítané k údaju o story pointoch za 4. šprint, takže táto hodnota by tu bola o malý kúsok vyššia. A čo sa týka 5. šprintu, tak išlo iba o 1 týždňový skrátený šprint a teda je pochopiteľné, že tu bude menej story pointov. Variabilita počtu story pointov od šprintu k šprintu vznikla aj kvôli tomu, že nám potrvало zopár šprintov na zosúladenie významov počtu story pointov.



Obrázok 4.2-1: Prehľad velocity hodnôt za jednotlivé šprinty.

4.3 Letný semester

V letnom semestri sme mali priveľa požiadaviek na systém a nestíhali implementovať a teda sme v neskoršom štádiu semestra od scrumu (na pokyn vedúceho tímu) mierne upustili. Problémom bolo, že toho bolo priveľa a taktiež sme sa museli synchronizovať s tímom 25. Preto nemá zmysel ukazovať naše burndown grafy, keďže nemajú veľmi výpovednú hodnotu.

5 Harmonogram LS

Keďže začiatkom tohto (zimného) semestra sme zistili, že presne nevieme, kedy skončíme a nevedeli sme ani povedať, čo všetko bude náplňou práce počas zimného semestra, rozhodli sme sa si budúci harmonogram trochu viac zorganizovať, aby sme tak predišli chaosu aspoň v letnom semestri.

Šprint #	Plán práce
1	Finalizácia implementácie vytvorenia projektu (s doplnením všetkých povinných údajov).
2	Dokončenie implementácie schvaľovania projektu (frontend, backend, databáza).
3	Finalizácia implementácie schvaľovania projektu.
4	Dokončenie implementácie zvyšných obrazoviek.
5	Dorobenie dizajnu stránok (front-end).
6	Finálne polishovanie a testovanie.

Tabuľka 5-1: Náš plánovaný harmonogram na letný semester 2023.

Príloha A - Zápisnice

Súčasťou tejto prílohy budú všetky naše zápisnice zo stretnutí.

Pre zápisnice sme si spravili šablónu - robili sme si zoznam vecí, ktoré máme spraviť (zaznelo to na stretnutí), ďalej sme spísali body stretnutia a napokon samotné poznámky zo stretnutia. Taktiež sme spísali, kto bol prítomný na stretnutí a kedy sa stretnutie konalo. Na písanie zápisníc bol poverený 1 človek z nášho tímu. Ak tento človek ochorel, podujali sa na tom iní (podľa dohody).

Zápisnica č. 1

Tím 17 – EVP
11. októbra 2022

Prítomní: Všetci

Miesto stretnutia: 3.38, FIIT STU

TODO:

- ☐ Setup GitLabu
 - ☐ Naplánovať šprint
 - ☐ Backlog, do ktorého pôjdu user stories (in GitLab = issues)
 - ☐ Priradovať veci z backlogu do šprintov
 - ☐ Vytvárať pre story tasky
 - ☐ Priradovať story pointy pre user stories
 - ☐ Generovať burndown chart
- ☐ Analýza špecifikácie – vytvorenie tabuľky do dokumentácie
 - ☐ Zoznam rolí
 - ☐ Zoznam činností (2. stĺpec; priradiť k rolám)
 - ☐ Opisy (pokiaľ nájdeme; priradiť k činnostiam)
- ☐ Analýza CERIFu
 - ☐ Vytiahnuť všetky entity
 - ☐ Zistiť, čo znamenajú farby (či majú nejaký význam)
 - ☐ Hierarchia entít (čo sa nachádza pod jednou entitou, stromček) [nie najväčšia priorita]

Body stretnutia:

- Zvolenie si toolu pre project management
- Špecifikácia
- CERIF

Poznámky zo stretnutia:

- **Zvolenie si toolu pre project management**
 - Možnosti:
 - Trello – dajú sa tam robiť backlog, šprinty, storypoints, burndown chart (iné veci moc netreba)
 - <https://tfs.fiit.stuba.sk/> – nevie sa, či by to vydržalo
 - DevOps – ak si mergeneme dva profily (všetko na jednom mieste, ale bol by v tom chaos)
 - Finálny nástroj: *GitLab*

- **Špecifikácia**
 - Dokument nahraný na Drive
 - Modelované diagramy = vstup pre nás (my nič súvisiace s týmto nemodelujeme)
 - Prejsť cez diagram – spísať si základné činnosti do tabuľky
 - Poznámka k scenárom: ak nedostaneme ďalšie podklady, budeme sa riadiť špecifikáciou, CERIFom a improvizovať
- **CERIF**
 - Common European Research Information Format
 - <https://www.eurocris.org/Uploads/Web%20pages/CERIF-1.6/documentation/MImage.html>
 - Je potrebné pozrieť sa a zhrnúť v rámci dokumentu
 - V rámci analýzy všetko čo budeme robiť pôjde do dokumentu
 - Budeme sa opierať na tento dátový model – budeme CERIF compatible, využívať z toho classes (a vedieť aké fieldy musíme mať v rámci modelu)
 - CERIF je dobrý ak by sme potrebovali dať do nášho systému aj nejaké európske služby
 - Z dátového modelu treba vytiahnuť všetky entity, ktoré tam fungujú
 - Sú dané tou čiarou/textom (publication, patent, project, ocenenia si nebudeme riešiť – robíme demo, nerobíme produkt, person...)
 - Nevymýšľame nový model, ale pôjdeme podľa CERIFu
 - V rámci projektu nemusíme mať všetko
 - Nejaké triedy nemusíme použiť vôbec (načo ich mať v databáze, keď s tým nič nerobíme)
 - Bude potrebné naparovať entity sponínané v dokumente na CERIF model (ak sa to vôbec dá spraviť)
 - Ideálne prevziať celý model z CERIFu, naparovať na špecifikáciu a spraviť Control-View
 - Ako riešiť prípad, keby sú v špecifikácii výstupy, ktoré nebudu v CERIFe?
 - Asi si rozšírime model
- **Ďalšie poznámky**
 - Vyriešenie prístupu k učebni
 - Spravený abecedný zoznam členov v Exceli
 - Prístup: dáme isic – dajú kľúčik, v prípade obštrukcií sme v zozname

Prítomní: Všetci

Miesto stretnutia: 3.38, FIIT STU

TODO:

- ☐ Namapovať typy ktoré sú v špecifikácii na typy ktoré sú v CERIFe
- ☐ Stavový diagram projektu (stavy projektu)
- ☐ Identifikovať typy výstupov projektu
 - ☐ Activity diagram aj na veci ako zadavanie výstupov projektu
 - ☐ Vieme namapovať na CERIF?
 - ☐ Vložiť tabuľku (publikácie patenty priemyselné vzory etc) do internej analýzy – kde budeme riešiť typ výstupov

Body stretnutia:

- Namapovanie GitLabu na potreby TP
- Rozbor EA diagramov

Poznámky zo stretnutia:

- **Namapovanie GitLabu na potreby TP**
 - Hierarchia v rámci abstrakcie
 - Epics > User stories [US] > Tasks
 - Význam váhy
 - US = story points
 - Task = estimate
 - Oficiálny návod pre agile od GitLabu
 - <https://about.gitlab.com/blog/2018/03/05/gitlab-for-agile-software-development/>
 - Zápis US a taskov do GitLabu + stĺpce boardov
 - **US = epics** = backlog + in progress + closed
 - **Tasks = issues** = open + WIP + to be tested + testing + closed (máme dobre)
- **Weight US a taskov**
 - Trackovanie velocity – US storypoints sú abstraktné číslo
 - Velocity = koľko storypointov dokáže tím zvládnuť za jeden šprint
 - Velocity budeme trackovať sami
 - *Ked' vytvorím **epic**, tak zaznamenám, koľko storypointov sme mu dali (cez **description**)*

- V ďalších šprintoch budeme sledovať
 - Napr. “Keď som popsledných 10 šprintov vedel spraviť ~30 storypointov, prečo tento týždeň som zvládol iba ~10”
 - **Zaznamenávať v tabuľke do zápisnice cez template**
 - Kartičky sú **len** na user stories; ktoré user stories chceme v rámci tohto šprintu
- Všetky tasks majú mať na začiatku estimates (v hodinách) – tasky majú dohodnuté na začiatku (a schválené, budget hodín)
 - V Jire pre estimates (issues) boli original + remaining
 - **Weight** = original estimate, netreba upravovať keď presiahneme čas (treba to zachovať)
 - Budeme zaznamenávať v minútach (pretože sa dajú zadať celé čísla)
 - **Estimate** = to, čo si navrhne čo nám to bude trvať (prirátáš k tomu, koľko už máš spent. ak máš spent 1h a myslíš si, že nám to zaberie ešte pol hodiny, tak dáš /estimate 1.5h)
 - **Spent** = spent, koľko sme to už robili
- **Rozbor EA diagramov**
 - hlavný riešiteľ = zodpovedný, v podstate project user, ale má viac právomoci
 - projectDept – maintainer – nevytvorí projekt, ale vie to spravovať
 - admin projekt vytvorí, projectDept spravuje
 - štandardne hlavný riešiteľ – človek, ktorý žiadosť podáva
 - Ako sa mení loggedUser na hlavného riešiteľa (alebo teda ako vznikne prvá rola hlavného riešiteľa) -> si registered v nejakom portali, a si owner keď si vytvoríš projekt; hocikto môže zadať žiadosť o zavedenie projektu do systému
- **Ďalšie poznámky**
 - User stories
 - Nejaká funkcionálna z hľadiska product ownera
 - Napr. je to vytvorenie view – high-level vec, ktorú potom rozbiješ na tasky
 - Do jedného šprintu je fajn keď ide viacero user stories, nech je dobrá granularita – ale nebudeme ich mať viac než 3 per šprint
 - Stavový diagram projektu
 - *State (StP) Stav projektu.png*
 - Stavový diagram – sami to sketchnúť (namodelovať si stavy), nechať to odsúhlasiť
 - vieme sa na diagram odvolať
 - Keď na začiatku projekt v stave “created”, musí byť “approved” etc
 - Sedí životný cyklus projektu – nie sú to konkrétne stavy, znie to dosť high-level
 - Netrápi nás správca infraštruktúry
 - Nie všetkých aktérov riešime – potrebujeme spraviť jeden ucelený scenár

Prítomní: Všetci

Miesto stretnutia: 3.38, FIIT STU + online

TODO:

- ☐ Spísať do dokumentácie
 - ☐ Riadenie projektu (ako sú reprezentované US etc)
 - ☐ Inner review process (priradenie labels)
 - ☐ Analýza + zhodnotenie technológií
- ☐ Doplniť úlohy do backlogu

Body stretnutia:

- Retrospektíva
- Projektový manažment
- Analýza špecifikácie
- Analýza CERIFu
- Ďalšie úlohy do backlogu

Poznámky zo stretnutia:

- **Retrospektíva**
 - Zistenie, či treba niečo v rámci tímu zlepšiť, aby to lepšie fungovalo
 - *Čo bolo dobré* (v čom pokračovať)
 - *Čo by sa dalo zlepšiť* (čo by sme mohli začať robiť)
 - Zvážiť, či nespravť vlastný harmonogram (aj keď mal byť externý)
- **Projektový manažment**
 - Ako viem, ktoré tasky zasahujú do viacerých šprintov?
 - US môžeme zavrieť vtedy, keď sú všetky tasky done
 - Ak nezavrieme nejaký issue ku koncu sprintu, presunieme ho do ďalšieho sprintu
 - US nemáme zatvárať bez vedúceho
 - Tasky budú zaokrúhlené na full hodiny
 - Tiež dať do retrospektívy
 - Nie všetky US v backlogu sú priradené ku konkrétnemu šprintu
 - Burndown chart pre US netreba
 - Do backlogu dávať veci, ktoré majú prínos pre zákazníka – nie to, čo musíme robiť z pohľadu riadenia – ako sme nastavovali GitLab etc)

- **Inner review process** – aspon jeden človek skontroluje splnenie tasku
 - Bud' to niekomu priradiť, alebo sa budeme tváriť, že človek je zodpovedný (a že to dal správne)
 - Dá sa epic niekomu priradiť?
 - Robiť to cez label – každý člen tímu má svoj label
 - K jednotlivým epicom priradiť jedného (môže byť aj viacej) človeka, ktorý zaň bude zodpovedať – ten môže spraviť review
 - K taskom stavy – to be reviewed, review – stačí premenovať
- **Definition of done = DOD**
 - Podmienka splnenia = ku každej US by mal byť nejaký výstup (v komentári na GitLabe a potom aj v dokumentácii)
 - Pretože by mohol byť problém, že sa tasky zatvárajú, ale v podstate nič nie je done
 - Očakávajú sa 2 dokumenty alebo 1 dokument s 2 veciami (nezabudnúť to potom zlúčiť)
 - Analýza špecifikácie: výstupný dokument so spísaním rolí, činností a ich opisu; výstupná analýza diagramov uvedených špecifikácií
 - CERIF: analýza dokumentu, výstup analýzy z dokumentácie štandardu CERIFu
 - Ak je DOD splnené, mala by byť linka na dokument do GitLabu – že práve toto je náš výstup
- **US storypoints (planning poker)**
 - Na úrovni epicov potrebujeme iba sumu storypointov – žiaden graf, len potrebujem vedieť počet storypointov (váha daná kartičkami), ktorý dokážeme dať za jeden šprint
 - Dávať váhu v jednom mieste – keď sme to dali do GitLabu, netreba to dávať do zápisnice
 - Assignovanie storypointov niekedy na začiatku šprintu (ideálne od dnes)
 - Je to jedno, či priemer alebo median (hlavne to môžeme spomenúť v štruktúre)
 - Pokiaľ je medzi vyloženými číslami rozdiel iba jedného stupňa, tak pohoda; kebyže je tam rozpor – diskusia
 - Môžeme si vybrať priemer (nejaký stred), ale nie desatinné
 - Význam čísiel
 - 21 = koniec splniteľného (max, nič zložitejšie by to nemalo byť)
 - 40 = to sa nedá, treba to rozbiť.
- **Analýza špecifikácie**
 - Finančného manažéra zatiaľ neriešime, pozeráme hlavne z pohľadu schvaľovateľa a hlavného riešiteľa (vytvoríme projekt – schválime – posunieme ďalej)
- **Analýza CERIFu**
 - Dokument do referencií v rámci analýzy

- **Ďalšie úlohy do backlogu**

- *Analýza technológií*
 - Grafové vs relačné db – čo by bolo najlepšie
 - Frontend – napr. v čom vieme robiť
 - Backend – výhody, nevýhody, ako sa to vie kombinovať (možno nie podporu, ale aspoň do akej miery sa rozumie s graph db – budeme niečo robiť manuálne vs dá sa to automatizovať)
 - Tiež samostatná kapitola do dokumentácie – analýza + zhodnotenie
- *Spisanie rolí + ich činností*
 - Subtasky
- *Ako riešiť hierarchiu skupín v CERIFe*
 - Došpecifikovanie entít v rámci CERIFu
 - DOD = očakávame výstup čo sa týka namapovania
 - Vytváranie hierarchie v rámci CERIFu
 - Skúsiť nájsť dokument, ktorý bližšie opisuje CERIF entity
- *Návrh obrazoviek*
 - NEJDEME to riešiť teraz, ale môžeme to mať v backlogu (lebo v backlogu nie sú veci viazané iba na jeden šprint)
 - Hierarchia obrazoviek
 - Kto má aké práva na obrazovkách
 - Čo sa na nich dá robiť
 - Keď si to špecifikujeme, budeme asi musieť aj schváliť, kde to budeme robiť

- **Ďalšie poznámky**

- *Verziovanie*
 - CERIF to nerieši (ani to nie je v téme), avšak je v požiadavkách
 - Vymyslieť verziovanie v rámci relačnej db alebo transformovať na grafovú db – also namapovať CERIF
- Momentálne nie skončiť s analýzou (nevyhneme sa jej), ale mohli by sme ísť na nejaký návrh
- Nie je nejaký template pre dokumentáciu – podmienkou je iba číslovať kapitoly a mať obsah
- Otázky (na vedúceho/externistov) nejdu do dokumentácie – týka sa to iba nás (našej internej dokumentácie)

Zápisnica č. 4

Tím 17 – EVP
08. novembra 2022

Prítomní: všetci

Miesto stretnutia: 3.38, FIIT STU

TODO:

- ☐ Dokončiť retrospektívu
- ☐ Rozdeliť user stories na tasky
 - ☐ Ohodnotiť tasky, dať estimates
 - ☐ Priradiť tasky k šprintu
 - ☐ Rozdeliť si tasky

Body stretnutia:

- Prebratie si výsledkov šprintu (jednotlivých user stories [US])
 - Analýza technológií
 - Prípady použitia + hierarchie obrazoviek
 - CERIF
- Zadefinovanie nových user stories [new US]

Poznámky zo stretnutia:

- **[US] Analýza technológií**
 - Potrebný technickejší popis — minimálne snippets; ako sa s tým robí
 - Dopísať do dokumentu – všetky technológie čo máme v tabuľkách
 - Zatiaľ sa žiadne verziovanie riešiť nebude – vedúci to vidí na Postgres (s ktorým vraj robí aj druhý tím)
 - Vybrané technológie:
 - Django, React
 - Databáza – buď nejaká grafová alebo Postgres
- **[new US] Analýza možností využitia grafových databáz**
 - *Alternatívna grafová db v spojení s Djangom*
 - Iné možnosti čo sa týka grafových db
 - *Dokument, v rámci ktorého bude mapovanie hlavných entít modelu CERIF (vrátane snippetov)*
 - Zamerať sa na to, čo sa dá zadefinovať v rámci vzťahov
 - Menšia US – max dvaja ľudia
 - Dopísať benchmarks databáz – pozor na porovnania (či v príkladoch nie sú queries výhodnejšie pre konkrétne db)

- **[new US] Implementácia základného prototypu**
 - Nasadiť backend, spraviť základný frontend v Reacte, prepojiť to s databázou
 - Len nech to niečo robí a máme prvú verziu (klúčne aj hello world, najlepšie niečo potiahnuť z db)
 - S backendom a frontendom sa nič nebude meniť – jedine s db
 - Zatiaľ to klúčne môžeme cez Neo4j
 - Dá sa to prepojiť s analýzou – vyskúšať neo4j alebo ďalšie alternatívy
 - Neo4j **testing** nemusíme pushovať – ale treba to zdokumentovať
 - Testovane db dopísať do docu – netreba do GitLab
 - Testovacia vzorka pre db? Export zo SK CRIS alebo získať od nich
 - Zatiaľ **lokálne** zdockerizované, ale nech v **Gite** už máme projekt
 - Git → Docker → Azure/AWS student plan
- **[US] Prípady použitia + hierarchie obrazoviek**
 - Finančný riešiteľ ani sysadmin nás netrápi
 - Pokiaľ dokážeme prejsť hlavným activity diagramom (so schváľovaním žiadostí etc), tak je to úplne jedno, ako to dáme
- **[new US] Návrh obrazoviek z dokumentu**
 - Mid-fidelity
 - V čom spravíme návrh obrazoviek? Figma
 - Nech sa to dá preklikávať etc
 - Nech to zadávatelia schvália alebo povedia, čo zmeniť
 - Vyhľadávanie vložiť podľa [existujúceho projektu SK CRIS](#)
 - Pozrieť si výstupy + register projektov, keď budeme riešiť obrazovky
 - Bolo by dobré pozrieť si iný portal okrem SK CRIS (aby sme vedeli ako vyzerajú nejaké obrazovky – čo všetko tam dať)
 - PortalVS → Kega projects → Overview → Click na project
- **[US] CERIF**
 - Buď vytvoriť vlastné class id alebo nebudeme podporovať
 - Keď to nie je, teoreticky pracovať s tým, čo máme – zatiaľ pracovať priamo s nimi a potom new US do backlogu
- **[new US] Došpecifikovanie CERIF fieldov**
 - Vytvorenie vlastných fieldov
 - Dopísať k tomu dokumentáciu
- **Ďalšie poznámky**
 - Tasky su v našej réžii – vieme ich vytvoriť aj na stretnutí tímu
 - Retrospektíva
 - Netreba písať, že “stihli sme všetky tasky, stále pravidelné stretnutia”
 - Skôr sa zamerať na “Niečo sa udialo v tíme a malo to pozitívny dopad”
 - Nemusíme dávať nič do plusov (“čo bolo dobré na šprinte”)

Prítomní: všetci

Miesto stretnutia: 3.38 + online

TODO:

- ☐ Dokončenie prvotných návrhov obrazoviek

Body stretnutia:

- Prebratie progresu šprintu a vzniknutých otázok

Poznámky zo stretnutia:

- **Prebratie progresu šprintu**
 - Zrušenie taskov ohľadom grafových databáz (relačná db je ok)
- **Zadávanie výstupov projektu používateľa**
 - *Zadanie očekávaných výstupov (na začiatku pri vytvorení projektu)*
 - *Zadanie výstupov počas riešenia projektu*
 - Pri vytvorení projektu zadávanie očekávaných výstupov projektu
 - Je to súčasťou hlavného scenáru → riešime aj to
 - Stačia zatiaľ plánované výstupy — napr. "očekáva sa takýto patent"
 - Základná identifikácia článku (ako výsledky VaV na skcris)
 - Manuálne vyplňaný formulár (prípadne odniekadiaľ ťahať, ale nemáme odkiaľ)
 - Väčšinu údajov (možno akurát na typ publikácie) je možné zadať až keď to bude publikované
- **Ďalšie poznámky**
 - Potrebujeme jednotný výstup čo sa týka patentov + autentifikácia (to je niečo čo budeme robiť s ďalším tímom)
 - [obrazovky] Musíme riešiť personálne + finančné zdroje + infraštruktúru?
 - Ak sa cez activity diagram dá prejsť aj bez toho, tak netreba
 - Máme spravovať import údajov?
 - Nie, netreba to riešiť; importy a exporty nás momentálne nezaujímajú
 - Momentálne potrebujeme fungujúce demonštrovatelné demo
 - **Ďalší šprint:** začať robiť klikateľný prototyp pre základný scenár
 - Vyberiem si časť funkcionality – bude postupne narastať
 - Projekt je zatiaľ na Slovensku (nie medzinárodný) – nemusíme riešiť koruny

Zápisnica č. 6

Tím 17 – EVP
22. novembra 2022

Prítomní: všetci

Miesto stretnutia: 3.38

TODO:

- ☐ Retrospektíva

Body stretnutia:

- Ukončenie šprintu
- Rozbor editovania a stavov projektov
- Vytvorenie nových user scenárov

Poznámky zo stretnutia:

- **Rozbor obrazoviek**
 - Partneri
 - Uviesť meno aj typ – napr: “Meno (typ)” alebo tabuľka
 - Projekt môže mať viacero partnerov
 - Pracovný pomer – rieši sa skôr v rámci projektu
 - Nie pri zobrazení projektu, ale pri vytvorení
 - Typy – forma pracovneho pomeru [číselníky]
 - Na začiatku do db ručne nahodíme sample data userov, nebudeme mať UI na vytvorenie usera
 - Personálna matica = človekohodiny (napr. za týždeň), kapacita človeka
 - Activity [pu] – realne čerpanie hodín (aspoň zatiaľ) nechceme, to je vykazovanie, to robiť nechceme
 - Netreba riešiť hodiny na projekte (obrazovka o10)
 - Zmeny projektu zatiaľ nemusíme riešiť
 - **Bude potrebné spraviť obrazovky na detaily výstupu – formulár**
 - **Bude sa dať otvoriť samotný výstup ako detail – v detaile sú polia**
- **[new US] Návrh obrazoviek 2**
 - Pridať k US2 detaily výstupov a harmonogram
 - Ak niečo chýba k hlavnému scenáru a nie je optional, tak to treba dopísať
 - Hlavný scenár – potrebujem sa dostať cez activity diagram; ak je niečo optional, tak to nemusíme dať, ak nie, tak to musíme zadať (napr. vytvorenie personálnej matice nie je obligatorný)
- **Editovanie + stavy projektov**
 - Pokiaľ je project draft, malo by sa dať editovať všetko

- Aj žiadosť by mala byť prechádzať stavmi
 - V nejakom stave je editovateľná konkrétna sada filtrov, pokiaľ je žiadosť stále schválená, tak napr. nemôžem meniť jej názov
 - Schvaľovateľ schvaľuje celú žiadosť
 - Nie jednotlivé položky
 - Nevie upravovať žiadosť
 - Teoreticky by sa dalo pridať komentár – napríklad prečo nebol projekt schválený
- Pri implementácii by sme mali mať stavy – tým sa riadime pri rozhodovaní, či sa projekt dá editovať
 - **Konkrétne stavy sú v stavovom diagrame StP**
 - Pred schválením – edit všetkého
 - Keď nejaký iný stav – lock
 - Nemusíme teraz implementovať v prvej fáze, to je do budúcnosti
- Editovanie projektu môže byť neskôr
- **[new US]** Implementácia obrazoviek
 - DOD implementácie: vybrať si, ktoré obrazovky by sme chceli implementovať (aspoň 3)
- **[new US]** Implementácia prihlasovania
 - V rámci spolupráci s tím21
 - Zatiaľ je jedno, či to skončí pri implementácii alebo návrhu
- **Ďalšie poznámky**
 - Ak by sme posuvali task, mali by sme zmeniť priradenie ku šprintu, a do komentára dať poznámku, prečo sa to stalo
 - [tím21] Dať im vedieť, akým spôsobom došpecifikovávame typ patentu
 - Môže byť presmerovanie na stránku – stále by bol prihlásený user, ak sa prihlásil u nás (je to však robota navyše pre nás – polishing na konci, je to iba nice to have)

Prítomní: Filip, Jakub P., Alica, Adam (online), Jakub H.

Miesto stretnutia: 3.38

Body stretnutia:

- Preberanie progresu šprintu
 - Ukázali sme mu, ktorých 5 obrazoviek ideme implementovať
- Dodefinovanie DOD pre epic Implementácia obrazoviek 1
- Preberali sme hodnotenie TP
- Upresnenie potrebných metodík
- Riešili sme, kedy sa znova budeme kontaktovať počas prázdnin a znova riešiť projekt

Poznámky zo stretnutia:

- Radšej menej obrazoviek, ale nech je to funkčné.
 - Netreba ešte obrazovky implementovať tak, že by si načítavali dáta z databázy atď.
 - Ak by bolo treba môžeme dropnúť nejakú z tých stránok, ak budeme na krátko s časom.
 - Môžeme argumentovať s tým, že na tých veciach budeme pracovať aj počas prázdnin a cez skúškové.
- Treba ukázať obrazovky druhému tímu (dnes).
- Autentifikácia pôjde preč? Lebo oni to ešte nemajú (tie tasky s autentifikáciou možno pôjdu preč, ale dnes to ešte budeme preberať na tom meetingu s druhým tímom.) Nemá to až tak veľkú prioritu.
- Druhý tím nakoniec (pravdepodobne) nebude používať kubernetes.
- Metodiky treba:
 - kvalita (keď jeden člen tímu robil na iný predmet k timovému projektu nejaké hodnotenie kvality, tak to tam tiež môžeme dať),
 - komunikácia (opísať discord, jednotlivé channely - jeden pre komunikáciu s druhým tímom, automatizované issues - bot pre integráciu s gitlabom, reporty, resources, links...),
 - review, kód, scrum, opisovať veci, čo sme robili.
- Druhý semester nech už je webstránka. Nikto to nebude kontrolovať teraz. Kedysi to malo byť už prvé dva týždne.
- Jeden šprint by bol cez prázdniny.
 - Od 9. januára sa môžeme kontaktovať a pracovať na tom znova.
- Autentifikácia
 - Role string netreba mať v auth
 - Auto-refresh na dopytovanie zmien v profile

Poznámky zo stretnutia s druhým tímom:

- Ukázali sme im naše návrhy vo Figma.
- Mílnik je jeden konkrétny dátum, nie rozpätie dátumov.
- Evidovať zmeny v mílnikoch je potrebné až neskôr.
- Premenovať výstupy na očakávané výstupy pri vytváraní projektu.
- Taktiež by malo byť zrejmý "stav", teda to kde sa používateľ nachádza. Napríklad mu to chýbalo pri vytváraní projektu.
- Obávajú sa toho, že keď je toho veľa, či to všetko stihneme.
- Polia sa môžu potenciálne rozširovať, najmä pri vytváraní projektu
- Chýba možnosť priložiť prílohu, lebo čo keď v budúcnosti pribudnú nejaké nové dokumenty?
 - Dať to pod kľúčové slová (mali sme to v návrhu, ale nedali sme to do návrhov vo Figma)
- Mal by fungovať ten pingpong systém, keď nám oni pošlú dáta, aby sme mi vedeli na ne reagovať
- Časová os je dobrý nápad, ale nemala by byť náročná na implementáciu
- Mali by sme prioritizovať. To čo máme navrhnuté vo figme spĺňa len jednu časť projektu, zásadná je tá funkcionálna. Aby to aj robilo to, čo to má robiť.
- Podľa jvina je personálna matica optional
 - Personálna matica súvisí s úväzkami
- Dôležité piliere sú aj schvaľovanie a ?
- Aj do ukončeného projektu by malo byť možné pridávať výzvy?
- Možnosť pridávať veci aj počas behu projektu.
- V bežiacom projekte bude zas len detail projektu, ale budú iné možnosti editácie.
- Chcelo by to nakoniec aj to prepájanie obrazoviek/preklikávanie (ako v hi-fi)
- Ten druhý tím chce od nás vedieť, že čo my budeme ukladať do databáz, ale to by nám oni mali povedať
- Ich zaujímajú názov výstupu a typ výstupu, kľúčové slová + asi prílohy
- Zatiaľ by potrebovali len informácie o modeli
- **Do výstupu treba názov, na to sa zabudlo v tom formulári !!!**
- Taktiež budú potrebovať ľudí, konkrétne štruktúru za projektom tých ľudí
- Podľa Kapca by tam mali byť uvedení všetci ľudia
- Zoznam pracovníkov a ich príslušnosť by mala byť synchronizovaná s druhým tímom
- Organizácie by mali byť obmedzené len na tie, čo majú niečo spoločné s tým projektom
- Chcú doplniť informácie o organizáciách
- Párovanie na základe UUID
- Používateľ-projekt vzťahy si poriešime my na našej strane
- Hodíme im IDčka používateľov na projekte
- Údaje o zamestnancoch editovať cez mikroservis, čo máme spolu
- V tejto fáze len univerzitný level
- 1. verzia: ak má majoritný úväzok na SAVke tak môže byť len v SAVke
- Kapec to chce m x n mapovanie
- Treba aj u nás riešiť informácie o ľuďoch
- Je otázne kde ukladať organizácie

Zápisnica č. 8

Tím 17 – EVP
6. decembra 2022

Prítomní: Všetci

Miesto stretnutia: 3.38 + online

TODO:

- ☐ Scrum, planning poker pre nový šprint
- ☐ Retrospektíva

Body stretnutia:

- Rozbor návrhu obrazoviek
- Rozbor implementácie
- Plán ďalšieho šprintu

Poznámky zo stretnutia:

- **[Obrazovky] Schvaľovanie projektov**
 - Spomínalo sa, že možno práve druhý tím bude riešiť schvaľovanie. Zatiaľ to riešime my (asi ako aj externé schvaľovanie)
 - Projekty su schvalované viacerými ľuďmi (riaditeľ ústavu + dekan)
 - Ak som riaditeľ ústavu a dám schváliť projekt, zmizne?
 - Nie, je zobrazený v tabuľke “Schválené projekty”
 - Stav projektoov sú rozšírené – nie sú iba “Schválené”, ale sú “Schválené riaditeľom ústavu” a “Schválené dekanom”
 - Dve záložky v sidebare – *Projekty* (lebo nie sú nutne *riešené* projekty) a *Schvaľovanie projektov*
 - Schvaľovanie projektov vidí iba človek s povolením na to
 - Napríklad dekan môže byť zároveň zadávateľ projektu, ale zároveň môže byť aj schvaľovateľ
 - Zobrazíť typ výstupu a počet k očakávaným výstupom
 - Do tabuľky zapracovať aj počet splnených výstupov
 - V návrhu to byť môže, poprípade to neimplementujeme
- **[Obrazovky] Stav projektu**
 - Do zobrazenia stavu projektu dať aj názov
 - Napríklad do tooltipu / farebným textom ako druhý tím
- **[Obrazovky] Prikladanie dokumentov**
 - Momentálne nemusíme zabíjať čas s prikladaním súboru [pri vytváraní projektu], nie je to až tak podstatné

- [potenciálny ďalší feature] Pri nahraní dokumentov pridať tabuľku s info o dokumente:
 - Názov dokumentov
 - Verzia
 - Popis
 - Momentálne to nejdeme robiť, ale myslíme na to
 - Ak pri priložených dokumentoch budeme mať podrobnejší popis, dáme to na ďalšiu záložku
- **Implementácia**
 - Ako rozdeliť štýl + fonty + layout?
 - Treba sa dohodnúť s ďalším tímom
 - Hlavne nech to vyzerá nejako rovnako (farby, fonty)
- **Ďalší šprint**
 - Implementácia obrazovky aj s backendom, aby sa dal vytvoriť projekt
 - Ak dependencies s db – dummy data
 - Pri modeli bude treba spraviť viac (nebude tam info iba o projekte, ale aj o ľuďoch, inštitúciách; nahadzovať podľa CERIFu) – task kládne aj pre dvoch
- **Ďalšie poznámky**
 - Dať stránku tímu na github.io
 - Pre autentifikáciu by bolo potrebné spraviť mock usera
 - Jedna user story sa nemôže skladať iba z jedného tasku

Zápisnica č. 9

Tím 17 – EVP
13. decembra 2022

Prítomní: Adam, Alica, Filip, Jakub P., Sofia

Miesto stretnutia: online

TODO:

- ☐ Harmonogram šprintov (ako v záverečnej práci)

Body stretnutia:

- Ukladanie projektu
- Rozbor otázok k dokumentácii

Poznámky zo stretnutia:

- **Ukladanie projektu**
 - Možnosť uložiť nekompletný projekt
 - Ak to mám v rozrobenej verzii, tak by to ťahalo z databázy
 - **Save na každú obrazovku, odovzdať na poslednú, keď sa reloadnem, tak mi ten projekt načíta**
 - Možno keď dáme vytvoriť projekt, hodí nás to do edit módu
 - Edit = vytvorenie projektu (jednotné GUI)
- **[dokumentácia] Export šprintov**
 - Minimálne by mal byť uvedený burndown charts
 - Pokiaľ nejaké problémy/zdá sa nám burndown chart divný (moc sa líši od ideálneho priebehu), tak to popíšeme
 - Podrobný export netreba nikomu – nech sa iba vie ako sa pracovalo
 - Graf pre velocity – zoberieme si všetky storypoints pre šprint, dáme do grafu, nech vieme, koľko storypoints sme dali za šprint
- **[dokumentácia] Návrh obrazoviek**
 - Link na Figma, prehľad obrázkov
- **[dokumentácia] Implementácia**
 - Keď to budeme mať deploynuté, dať link na samotný web
 - “Implementované je to v rámci repozitára <link na repozitár>”
 - Môžeme tam dať linky na konkrétne commits kde sme implementovali GUI – poukázať že to bolo nejakto spravené, ak chcú prístup – riešiť neskôr
- **Ďalšie poznámky**
 - Ak zodpovedný za user story má nejaký task v rámci tej user story, tak si určí človeka, ktorý mu spraví ten code review

Prítomní: všetci

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ US1: Vizuál prototypu
- ☐ US2: Refaktoring komponentov
- ☐ US3: Implementácia screenu FE + BE + db

Body stretnutia:

- Naplánovanie úloh (user stories) na najbližši šprint, planning poker

Poznámky zo stretnutia:

- Úlohy: pokračovať v postupnom prepájaní FE a BE a vizuálu
 - Nebude sa riešiť nič iné okrem implementácie a vizuálu
- **Vizuál**
 - Chceme analýzu alebo to iba copypasteovať?
 - Submodul, oddelený layout od štylizácie?
 - Style by mal hovoriť o farbách, veľkostiach fontov etc, nie o layoute
 - Cieľ: spraviť návrh, že "takto sa to dá robiť", odprezentovať Kapcovi
 - Spoločné stretnutie sa možno nepodarí, kým nebudeme mať čo na ukazanie
 - Z pohľadu CVTI (používateľa) by to mal byť jednotný systém
 - Na tento šprint niekto kto by riešil prototyp
 - Nechce od nás priamu implementáciu do prototypu – toto dať do **[new US]**
 - Cieľ: analýza, návrh, prototyp (proof of concept)
 - Dokument: výsledný návrh riešenia, v rámci ktorého sa dosiahne oddelenia layoutu a štýlu
 - Prototyp: proof of concept implementácia návrhu
- **[new US2]** Refaktoring (kódu) vizuálnych komponentov
 - Napr. oddelenie layoutu od dizajnu
- **[new US3]** Pokračovanie implementácie
 - Obrazovka: o2 bez funkcionality vyhľadávania
 - Vybrať si ďalšiu časť
 - Spraviť generickú user story a meniť to číslo

- **[new US4]** Analýza možnosti implementácie search engine
 - Ktorú 3rd party vec na vyhľadávanie by sme vedeli využiť (linky na návody do analýzy)
 - Alebo využiť čisté Django; indexovanie ale nie, custom riešenie by malo zahŕňať exact match
 - Napr. štrukturované dopyty: *autor:meno* (tak by sme nepotrebovali rozšírené vyhľadávanie)
 - Budeme riešiť tri typy vyhľadávania: autor, názov projektu, všetky polia
- Jedna US = jeden screen; keď chceme viac screenov, tak viac US
- ako [x] chcem urobiť [y], aby [z] ← popis k jednotlivým US

Zápisnica č. 11

Tím 17 – EVP
23. február 2023

Prítomní: všetci

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ **[US layout + style]** Dorobiť najlepšie do utorka

Body stretnutia:

- Poriešenie komponentov

Poznámky zo stretnutia:

- **Ako poriešime komponenty?**
 - My robíme niečo iné ako tím25
 - Vedúci sa bavili aj o komponentoch
 - Lepšie je mať dve classy (ako pre layout a pre štýly; tak ako to robíme momentálne)
 - V rámci prototypu spraviť naše riešenie – ľubovoľný page
 - Chce mať oddelené classes
- **[US layout + style]** Pozrieť si jss a zistiť ako sa dá riešiť styling (ako doceliť shareovanie)
 - Spomenuté shareovanie statickej časti – čo to je, ako sa pridáva?
- Ďalšie potenciálne stretnutie s tím25 – utorok 13:00

Prítomní: všetci

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ US1: Implementácia základného vyhľadávania
- ☐ US2: Analýza možností vyhľadávania v slovenčine
- ☐ US3: Validácia vstupných formulárov v rámci vytvárania projektu
- ☐ US4: Pokračovanie v implementácii
- ☐ Dat vedieť (na DC) na čom sme sa dohodli

Body stretnutia:

- Diskusia ohľadom komponentov
- Diskusia ohľadom implementácie vyhľadávania
- Zadefinovanie úloh pre ďalší šprint

Poznámky zo stretnutia:

- **Komponenty** a ich synchronizácia s tímom 25
 - Kým nemáme dokumentáciu k ich balíku, pokračujeme v implementácii na vlastný spôsob
- **Implementácia vyhľadávania**
 - Hľadanie NLP nástroja pre slovenčinu (stemming/lematizáciu)
 - Lemma-tag zo SAV by fungoval, otázne, či sa s nimi netreba dohodnúť
 - Problém by nebol ani vo fungovaní (vracia response), ale v počte requestov ktoré dokáže spracovať
 - Robia podľa slovníka, ale zjavne nevie fungovať bez diakritiky (exact match)
 - MorphoDiTa z ČUNI
 - Keď lemmatizer nezafunguje (slovo bez diakritiky), tak robí stemmer
 - Do SearchQuery napasujeme nejaký stemmer (prevzatý alebo aj vlastný)
- **[new US1] Basic search z frontendu**
 - Jednoduché demo, aby niečo vrátilo aj nad tými 4 projektami čo teraz máme
 - Nebudeme tu riešiť jazyk (kludne v angličtine)
 - Vyhľadávanie pomocou fulltext a modifikácia SearchQuery, aby to používalo slovenský stemmer
 - Treba vytvoriť konfigurák a či sa dá podhodiť vlastná funkcia (SK knižnica asi nebude)

- **[new US2] Analýza možností vyhľadávania v slovenčine**
 - Prípadne upraviť implementáciu vyhľadávania
 - Linky na dokumentáciu, popis nevyhnutných zmien, štruktúra súborov, príklady
 - “Je tu konfigurák, v tom musíme zmeniť X; tieto údaje predstavujú (napr.) cestu k súborom; aby to bolo funkčné pre slovenčinu, musíme spraviť Y...”
 - Či sa tam dajú podhadzovať súbory s priamym mapovaním slovo x slovo alebo niečo s vlastnou funkciou
 - Treba vedieť syntax v súbore, ako vytvoriť niečo vlastné
- **[new US3] Validácia vstupných formulárov v rámci vytvárania projektu**
 - Pridať do GitLabu poznamku pre seba, čo konkrétne ideme implementovať
- **[new US4] Pokračovanie v implementácii**
 - Je jedno, či chceme robiť nový screen alebo pokračovať v doladovaní už vytvorených obrazoviek (napr. spojať vytvorenie draftu)
 - Ak si zvolíme ďalšiu obrazovku, dopísať, ktorú (prípadne ktoré) ideme implementovať
 - Najprv spravíme planning poker pre prvé 3 US (bez implementácie), a podľa toho, koľko storypoints nám zostane, taký screen si zvolíme
- **Ďalšie poznamky**
 - Update návrhov vo Figma neriešime
 - Čo sa týka úprav projektu, musíme zohľadniť dve možnosti:
 - Permissions – či má user právo na úpravu
 - Stav projektu – či je *pred schválením*, *schválený* etc

Prítomní: všetci

Miesto stretnutia: 3.38 FIIT STU + online

TODO:

- ☐ Zistiť (spísať) výhody a nevýhody Bootstrap vs Tailwind

Body stretnutia:

- Implementácia slovenčiny
- Harmonogram projektu
- Komponenty a ich zdieľanie
- Responzivita

Poznámky zo stretnutia:

- **Diskusia knižnice pre vyhľadávanie v slovenčine**
 - Zistiť štruktúru, či je to funkčné, vyskúšať to
 - Django priamo podporuje Postgres search
 - `ts_query` = vyhľadávanie nad `ts_vectorom`
 - Nahodiť, nakonfigurovať do slovenčiny, generated column – migrácia + nastavenie slovenčiny
 - Dať pozor, ak najprv spustíme migráciu a potom pridáme slovenčinu, tak to bude treba este raz spustiť
- **Harmonogram projektu**
 - Otázky na zamyslenie:
 - Keď dám edit, kam ma to pošle?
 - Keď budem veci pridávať bez save-ovania, prekreslí mi to?
- **Komponenty a ich zdieľanie s tímom 25**
 - Hľadanie ktorú knižnicu využívajú – najpravdepodobnejšie `styled-components` and/or `stitches`
 - Prínos tejto knižnice je zatiaľ v rámci projektu nulový – zatiaľ nechávame tak
- **Responzivita**
 - Ak nemáme jednoduchý plán na riešenie responzivity, tak by sme mohli využiť knižnicu tímu25
 - Ak použijeme Bootstrap v rámci projektu, ako by to vyzeralo?
 - Najviac sa zmení `css` (`js` a `html` až tak nie)
 - Stiahneš si nejaké preddefinované štýly
 - Tailwind `css` znie tiež ako možné riešenie
 - Zatiaľ: **Bootstrap**

Prítomní: Adam, Filip, Jakub H., Alica, Sofia

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ [prenášaná US] Analýzu searchu v slovenčine
- ☐ [new US] Implementácia prototypu SK searchu
- ☐ [new US] Prerobenie na Bootstrap
- ☐ [new US] Implementácia ďalšej funkcionality – editácia

Body stretnutia:

- Prezentácia výsledkov šprintu
- Vyhľadávanie v slovenčine

Poznámky zo stretnutia:

- **Vyhľadávanie v slovenčine**
 - Čo sa deje vnútri tých súborov? (ich štruktúra)
 - Ak mi nebude fungovať slovo bez diakritiky, tak prečo?
 - Čo robí REP (čo sú suggestions, kde sa to vracia, či to rieši search)? Má to riešiť diakritiku?
 - Affix, dict?
 - Funguje to bez diakritiky? Vyzerá, že nie
 - word, asciword, ascihword, numwords, hword (všetky) – všade kde je english_stem (nastavuješ to z konfigurácie)
 - Zostane vyriešiť diakritiku
 - Možnosť 1 – fungovať built-in procesom (treba zistiť ako)
 - Možnosť 2 – prepísať slovníky tak, aby veci s/bez diakritiky mapovalo na rovnaké slovo (na to máme poznať štruktúru súbrov)
 - Možnosť 3 – odpáliť diakritiku; teoreticky by sa slovník prepísal – všetko by sa spravilo bez diakritiky, pri searchi by diakritika bola odstránená zo vstupu
- **Otázky na najbližšie stretnutie s tímom 25**
 - Ako si predstavujú komunikáciu s auth servisom? Ako navrhnu spôsob výmeny dát (sketch etc)?
 - Výzvy – zatiaľ neriešime
 - Spoločná virtuálka – tiež nič
 - Layout – skúsime Bootstrap, farby + fonty keby niečo udateneme

- **Ďalšie poznámky**
 - Server-side validácia pri vytváraní projektu – do backlogu
 - O týždeň by sme mohli vedieť zodpovedať na otázky fungovania SK searchu
- **[prenášaná US] Analýzu searchu v slovenčine [2 ľudia]**
 - analyza – struktura, preco to nejde etc
 - Pozrieť sa ako by sa to líšilo/napasvalo na cerif ten modeel, ktoorý nám poslali
 - Na konci bude fungovať už search (nie je to už len analýza)
- **[new US] Implementácia prototypu SK searchu**
 - Bude slovenské vyhľadávanie, ktoré nie je citlivé na diakritiku
- **[new US] Prerobenie na Bootstrap [2 ľudia]**
 - Popritom aj postupne hadzať do updated vzhľadu
 - Koľko stránok
 - odkaz na commit/pull request
 - Deployed verzia prototypu kde N stránok využíva Bootstrap na layoutovanie elementov
- **[new US] Implementácia ďalšej funkcionality – editácia**
 - Deployed verzia prototypu kde bude možné editovať dáta v databáze

Zápisnica č. 15

Tím 17 – EVP
23. marca 2023

Prítomní: Filip, Alica, Jakub P., Sofia, Adam

Miesto stretnutia: 3.38 FIIT STU + online

TODO:

- ☐ Sketch vyhladavania v sidebar layoute

Body stretnutia:

- Vzhľad stránky
- Search

Poznámky zo stretnutia:

- **Vzhľad stránky**
 - Header? Where?
 - Landing page robí tím25 (po ten splitpoint)
 - Header vecí dame do sidebaru
 - Ako spraviť sidebar (ktory bol header u nás) na mobile?
 - Riešenie:
 - sidebar, keď sa zmenší šírka stránky, tak bude schovaný pod hamburger menu
 - keď klikneme na menu, tak sa tie veci vysunu doprava a dole, ale nebude sa to skalovať
 - návrh vo figme
 - search – vyhladávanie, aby bolo nad výsledkami? (ako crzp)
 - project/info – úprava projektu (button) do pravého horného rohu (nech to nemáme v sidebar)
 - vyriešiť notifikácie -> tiež shortcut pre projekty na schválenie
 - vložíme žiadosť cez vstupný formulár -> uložíme -> editujeme -> ukážeme proces schvaľovania
 - tiež integrácia na autentifikačné služby
- **Search**
 - Ako funguje search suggestions (ako sa používajú a čo majú robiť)
 - Či sa pravidlá aplikujú aj na input, aj na indexovanie

Prítomní: Filip, Alica, Jakub P., Adam, Sofia

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ Prec lorem ipsum z readme
- ☐ [asap] **Gitlab CI**
- ☐ [asap] **Model organizacie** – user/id/projects
- ☐ Model patentu
- ☐ Integracia auth sluzby
- ☐ [lowest priority] File storage – django s3 adapter (download modify delete)
- ☐ [neskor] spísat navod ako to cele spustiť lokálne aby to fungovalo

Body stretnutia:

- Search
- Integracia s tim25

Poznámky zo stretnutia:

- **Search**
 - Postpone
- **Tim25**
 - My poriesime organizacie
 - Spravime dalsi endpoint, ktorý by vrátil `nazov`, `id`, `org_id`
 - Treba doriesiť aj pagináciu
 - Od nich treba prevziať reviewing process
 - K chce aby riešenie bežalo za jedným `docker compose` a jednou proxy
 - file storage - majú **s3 minio** (?)
 - majú poriesene files, môžu nám dať connection strings nech sa na to vieme napojiť
 - s3 adapter pre django
 - CI nám môžu poslať na React
 - dame timu25 deploy token a oni to deploynu
 - Nech nám pošlu model patentu, aby sme si to zosuladili
 - Patenty asi budeme ťahať iba zo žiadostí
- Nemáme dávať “closed” na tasky ktoré ešte nie sú mergeované do main
- **Re: patenty**
 - **spraviť entitno-relačný model** (keď sa niekto na to opýta, tak to iba ukážeme)

- merge odrazok s skcris a cerifom
- ako to vyzerá, aké sú fieldy, descriptions k tomu čo to robí
- mali by sme sa začať pozeráť na zjednodušený a modifikovaný model (to čo sme analyzovali pri cerife)
- ako **3.1-1 (10.1-1)** ale aj s description ako to ideme používať
 - 3.1-1 = prvý návrh
 - 10.1-1 = dávnejší export djanga
- **nemusíme modelovať ak to spravíme v djangu a exportneme**
- poprechádzať cez analýzy a dáť dokopy
- V dokumentácii sa spomína udelený patent, či to budeme niekde mať
- Majú explicitne napísané, že “patentová prihláška”
- Mali by sme mať patenty od druhého tímu
- **Co budeme od nich potrebovať v rámci patentu/patentovej prihlášky?**
- V dokumentácii si pozrieť čo sa týka patento
- Urcite budeme pridávať aj manuálne výstupy, ale patenty môžeme mať od tim25
- To čo máme vo Figma dať do db (+ skonsolidovať s CERIFom)
- Porozmýšľať nad možnosťou schovať reviewing proces organizácii do mikroslužby
 - Odošlem žiadosť a pred tým ako sa tá žiadosť niekomu pošle, tak musí prejsť cez reviewing proces (main activity diagram – schválenie internou inštitúciou - dekan etc), druhá úroveň je externý schvaľovateľ (súvisí s výzvou)
- tl;dr
 - 1: potrebujeme vytvoriť žiadosť a vedieť editovať (frontend, backend)
 - 2: potrebujeme sa integrovať s tim25
 - najbližšie spraviť deployment
 - porozhádzať si tie úlohy medzi sebou

Prítomní: vsetci

Miesto stretnutia: 3.38 FIIT STU + online

Body stretnutia:

- Upresnenie db modelu a implementovaných stránok

Poznámky zo stretnutia:

- **Draft vs submitted project**
 - nerobil by to ako draft tabulka (lebo napr. ak máme harmonogram, tak by sme potrebovali harmonogram aj harmonogram_draft tabulky)
 - skor by to malo prebiehať ako validácia na backende v djangu
- Nabuduce edit project – iba basic info
- Zmeny -> premenovať na zoznam zmien
 - teoreticky tam môžeme spraviť tabulku, že tam natvrdo sample data, nech iba vidia ako to bude vyzerat
 - používateľ X zmena poľa Y
- Basic info -> nech sú z db fetchované plánované výstupy

Prítomní: vsetci

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ Organizacne tabulky
- ☐ Validacia pre existenciu projektu pri jeho ulozeni
- ☐ Spravit pre jvina EA obrazok; ked su otazky k tomu, dat otazku

Poznámky zo stretnutia:

- Pridat do project detail organization
- Co sa tyka organizacii, musime spravit review process, ktory by tahali z endpointu
 - zadaju org id, a my z tabulky vratime review process
 - review process = tabulka s id a parent id (spajany zoznam) - akonahle clovek schvali, tak sa posunie dalej
 - api/project/id
 - ako mame participants role, tak aj org role (premenovat z roles na role)
 - neukladat v participantoch first last name a contact, ale iba id
 - participants: id, role
 - org: id, role
 - tretia tabulka v db:
 - org id, user id, parent id. type
 - user id = clovek kt. to ma schvalit
 - type = co sa schvaluje
- Typy schvalovania - [tim25] patenty, [my] projekty, [my] po zmene vystupu?
- Prvy lookup - ako bude vyzerat review pre org a typ
 - org_id type rp_id
- Kde sa bude modelovat typ usera? (ci je to externy schvalovatel etc)
 - z pohladu systemu – kde si v akej roli (mozem byt schvalovatel na jednom mieste, riesitel inde)
- Tabulky
 - org_reviewProcess:
 - org_id type rp_id
 - reviewProcess: (spajany zaznam)
 - id parent_id, user_id
- Neskôr ukázat na EA diagrame – zvýraznit vetvy ktore sme presli v hlavnom toku, nech nejaka z tych vetiev funguje – activity ZP
 - dorobit kontrolu pri zadavani nazvu

- keď zmeníš focus textfieldu, tak nech to skontroluje, či existuje projekt s takýmto názvom
- nech vyvolá ten názov
- zatiaľ keď dáme uložiť, tak nech tá validácia bude tam. zistí, že projekt už existuje až pri uložení
- drafty môžu mať rovnaké názvy, a kontroluje sa to iba pri schválených projektoch
- aj keď je to draft, aj keď poslem na schválenie, tak checknem against už schválené projekty
- podtyp – nevieme
- partneri – search select - začnem písať a vyhodí mi nejaké organizácie
 - zatiaľ vytiahnes všetky orgs a iba dáš search select (react-select)
 - do prepájacej tabuľky Proj_OrgUnit
 - projectId, organizationId a doplníš vzťah pre classId
 - budeme mať vlastný schemeId pre partnerov
 - contextId + classId (celý string schemeId + classId) + cfTerm (nejaký popis, aby sme vedeli v rámci db čo tam bolo) <- interná tabuľka
 - 1 - project_org
 - 2 - project detail
 - pri všetkých našich vymyslených veciach budeme mať jeden unikátny scheme id – každý prvok by mal svoj classId – uid generator
 - zaznačiť do docu, že toto schemeId je náš scheme id
- pracoviska – v rámci partnerov (UISI a UPAI sú dve pracoviska v rámci FIIT STU)
 - najst všetky pracoviska v rámci ktorých figurujú partneri
 - otázka: môže sa stať že chceme pracovisko ktoré tu nie je

Zápisnica č. 19

Tím 17 – EVP
20. apríla 2023

Prítomní: vsetci

Miesto stretnutia: 3.38 FIIT STU

TODO:

- ☐ ci/cd – updatenut druhy tím ohľadom progresu

Body stretnutia:

- check-in progresu
- modelovanie procesu schvalovania

Poznámky zo stretnutia:

- pokračovať v dorabani nedokoncených vecí
 - bootstrap
 - editovanie
- proces schvalovania

Prítomní: vsetci

Miesto stretnutia: FIIT STU 3.38

Body stretnutia:

- Dokumentacia (+ co je potrebne do nej dorobit)
- Schvalovací proces

Poznámky zo stretnutia:

- Dokončiť *rozrobene* veci + dokumentacia
- **Dokumentacia**
 - API, popis architektury (co je implementovane v jednotlivych triedach, "tato API robí toto a toto")
 - dokumentacia k jednotlivym funkciam
 - asi aj spomenúť knižnice (napr. django rest)
 - komponenty
 - k tomu class diagrams
 - ze tie veci mame v nejakom foldri, ked su to funkcie, tak class diagram zmysel nema
 - iba k jednotlivym komponentom - v ktorych suboroch to najdem, co tam je
 - ake komponenty, kde ich najdem, ake maju funkcie
 - endpoints
 - sample ako sa k tomu napojim
 - payload - ako to vyzerá, ako sa napojim
 - moze byt swagger
- **Organizacie** (review process)
 - relacny model ako obrazok, supnut do dokumentu a popisat
 - root by mal byt jeden (ale nemusi)
 - model databazy, zdokumentovat endpoint, swagger
- **Frontend**
 - todo: expiration token nech sa premaze z projektu (ze ked vytvoris projekt tak tie predosle data nie su tam ked vytvaras novy projekt)
 - pre bootstrap sme menili nejaké triedy, to opísať v dokumentácii

Príloha B - Retrospektívy

Súčasťou tejto prílohy budú všetky naše retrospektívy za všetky predošlé sprinty.

Pre naše retrospektívy sme mali definovanú presnú šablónu. Napísali sme, čo sa nám na šprinte páčilo, čo by šlo zlepšiť a napísali sme aj nejaké zhodnotenie. Snažili sme sa do nej nepísať veci ohľadom projektu ale skôr ohľadom spôsobu práce tímu, teda nejaké pozitívne alebo negatívne postrehy. Tieto dokumenty sme písali všetci spolu.

Retrospektíva č. 1

Tím 17 – EVP
26. októbra 2022

Začiatok šprintu: 12. októbra 2022

Koniec šprintu: 25. októbra 2022

Čo bolo dobré na šprinte?

- Každý člen tímu sa hneď ujal svojej pridelenej úlohy a postupne začal pracovať na jej splnení.
- Členovia tímu sa dokázali stretnúť viackrát do týždňa a aktívne pracovať na plánovaní činností v rámci tímu.

Čo by sa dalo zlepšiť na šprinte?

- Skôr zadávať tasky. Počas sprintu sme pridávali nové tasky a preto náš burndown chart nevyzeral dobre.
- V rámci predmetu TP sme zatiaľ nedostali harmonogram činností, takže pravdepodobne budeme musieť vytvoriť si vlastný.
- Dalo by sa zlepšiť zaznamenávanie času stráveného nad úlohami zo strany členov tímu.

Zhodnotenie:

- Počas prvého šprintu sme identifikovali menšie problémy s plánovaním činností v rámci projektu a teda existuje priestor na zlepšenie.
- Taktiež sme sa potrebovali oboznámiť s Gitlabom ako systémom na riadenie projektu, čo spôsobilo zmätok vo vytváraní taskov a celkovom plánovaní prvého šprintu.
- Napriek vzniknutým problémom sa nám podarilo dokončiť všetky vytvorené úlohy.

Retrospektíva č. 2

Tím 17 – EVP
08. novembra 2022

Začiatok šprintu: 26. októbra 2022

Koniec šprintu: 08. novembra 2022

Čo bolo dobré na šprinte?

- Zodpovednosť členov tímu – nemuseli sme mať centralnú figúru pre assignovanie všetkých taskov, keďže sa všetci niečoho ujali.

Čo by sa dalo zlepšiť na šprinte?

- Odpadol nám jeden deň (kvôli sviatku) – nedali sme si priebežné stretnutie
- Zvýšiť pracovné tempo – robiť tasky viac priebežne, nie nárazovo.
- Nastavovať si predom, ešte pred začatím šprintu, Gitlab (lebo potom to zaberá čas na stretnutí).
- Nezabíjať toľko času plánovaním (nemať prvé priebežné stretnutie čisto na plánovanie) – malo by to byť max. na pol hodiny, a už prvá práca by mala byť na prvom stretnutí.
- Chcelo by to viac času na prediskutovanie niektorých taskov, skôr ako ich uzavrieme (hlavne, ak na danej úlohe pracoval iba 1 človek a nie je si istý niektorými vecami).

Zhodnotenie:

- Keď odpadne meeting, mať priebežné stretnutie v iný deň.

Retrospektíva č. 3

Tím 17 – EVP
22. novembra 2022

Začiatok šprintu: 9. novembra 2022

Koniec šprintu: 22. novembra 2022

Čo bolo dobré na šprinte?

- Začali sme reálne robiť na úlohách počas tímových stretnutí namiesto toho, aby sme trávili toľko času plánovaním.

Čo by sa dalo zlepšiť na šprinte?

- Ešte stále sme nedostali odpovede od externistov na naše otázky a nejasnosti, čo nás mierne spomalovalo pri návrhu obrazoviek.
- Zlepšiť komunikáciu v prípade, ak člen tímu nemôže byť prítomný in person na stretnutí (ak napríklad ochorie).

Zhodnotenie:

- Napriek ťažkostiam v komunikácii, rozložili sme prácu efektívnejšie, čo viedlo k zatiaľ najlepším výsledkom jedného šprintu.

Retrospektíva č. 4

Tím 17 – EVP
06. decembra 2022

Začiatok šprintu: 23. november 2022

Koniec šprintu: 06. december 2022

Čo bolo dobré na šprinte?

- Fungujúci kávovar.

Čo by sa dalo zlepšiť na šprinte?

- Až koncom semestra sme dostali konkrétnejšie požiadavky od product ownera (ako aj stretnutie s ním).

Zhodnotenie:

- Kvôli požiadavkám product ownera na zosúladenie nášho tímu a tímu 25 (keďže pracujeme na rovnakej infraštruktúre) je spomalený progres práce na tomto projekte.

Retrospektíva č. 5

Tím 17 – EVP
13. decembra 2022

Začiatok šprintu: 06. december 2022

Koniec šprintu: 13. december 2022

Čo bolo dobré na šprinte?

- Nič neobvykle dobré sa neudialo.

Čo by sa dalo zlepšiť na šprinte?

- Neprišli sme počas tohto šprintu na nič, čo by sa dalo zlepšiť.

Zhodnotenie:

- Šprint trval iba týždeň a veľa sa toho neudialo.
- Dobrá komunikácia v tíme a včasné prerozdelenie si úloh sa osvedčilo na tento šprint.

Príloha C - Otázky na externistov

Táto príloha je venovaná otázkam, ktoré sme v priebehu šprintov spisovali na zadávateľov projektu (zákazníkov), keďže naše stretnutia nanešťastie neboli pravidelné a teda sme nemohli tieto otázky veľmi konzultovať osobne. Tieto otázky mali byť externistom sprostredkované prostredníctvom nášho vedúceho.

Otázky

1. Activity ZP
 - Kto a ako vytvára novú výzvu v systéme?
 - Keď projekt nemá výzvu, kto ho externe schvaľuje?
2. Máte namapovanie typov výstupov (z tabuľky v špecifikácii) na výstupy projektov v CERIFe?
3. Externý vs interný schvaľovateľ projektu – ako sa líšia ich povolenia, resp. sú zaznačené do grafu rovnakým aktérom?
4. Čo je SOFIA? Resp. do akej miery sa týmto systémom máme zaoberať?
5. Čo je myslené tým, že CVTI gateway mení stav?
6. Je hlavný riešiteľ (projectAdmin) zároveň riešiteľom projektu (projectUser)?
 - V diagrame používateľov nebol spojený s niektorými aktivitami riešiteľa projektu
7. Aké akcie môže s projektom robiť pracovník projektového oddelenia (projectDept, kto to je, môže napríklad vymazať projekt)?
 - Ako je projectDept priradený k projektom, čo znamená že všetky projekty sú "jeho". A čo znamená, že "spravuje projekty"?
 - admin projekt vytvorí, projectDept spravuje?
8. Môže loggedUser vytvoriť projekt a tým sa z neho stane projectAdmin? Teda či môže hociktorý registrovaný používateľ vytvoriť projekt.
9. Čo má spoločné schvaľovateľ projektu s úvodnými nastaveniami (diagram Activity UN - kto robí úvodné nastavenia?).
10. Ktoré údaje o projekte vidí neregistrovaný resp. registrovaný používateľ, ktorý nie je riešiteľom projektu.
11. V Use Case PP diagrame - čo všetko má v prehľade vidieť projectView (keď má mať "podobný prehľad"). Napríklad či vidí tiež základné údaje o financiách atď.
12. Chýbajúce údaje/súbory
 - Príloha č. 1 - polia systému ISEP (v súbore specifikacia.odt, str. 8)
 - Excel popisujúci stav infraštruktúry – konkrétne záložka "4. Zdroje" [A (IP)]
 - Príloha č.1b predmet podpory časť B - NITT II.doc
13. Pri opisoch UC ktore su vyjadrené activity diagramom chýbajú roly (ktory akter) - hlavne financny manazer
 - Vie hlavný riešiteľ robiť to iste ako financny manazer?
 - Include a extend nie su opisane
14. Personálne zdroje, finančné zdroje, infraštruktúra a výstupy projektu sa budú nastavovať zvlášť k celému projektu ale aj k jednotlivým míľnikom?
15. [A (SP)] – Čo je dopadové obdobie projektu?
16. [UC (START)] – Čo znamená, že LU "vie zadať svoje expertízy"

Príloha D - Úvodné pokyny

TODO:

- ☐ web sídlo
- ☐ rozchodiť project management system — všetci účty
- ☐ priradiť si roly

utorok 9:00 — štandardný čas stretnutí

- *1st milestone* — nejaky docu, prezentácia (riadenia — akou formou sme manažovali v rámci tímu, roly, produkcia, continuous integration, plnosť testovania, dodržiavanie scrumu, retrospektiva, planning poker etc)
- *2nd milestone* — final document, prezentácia
- prihlásenie sa do *TP Cup* — prezentácia

Web sídlo tímu

- bez ohľadu na to, či to od nás budú vyžadovať
- dostaneme virtualku
- idealne staticky page
 - dynamicky nema zmysel, pajcnuť free html sablonu
 - kontent ohľadom tímu
 - opis projektu
 - zoznam ľudí v tíme
 - stránka, kde budeme vesat zapisnice + priebežne vypisy
 - nahrávanie suborov staci vyriesit tym, ze raz za tyzden nieco pridáme do html
- html, js, nic ine

Zápisnice

- 1 scrum master — dohľad na to, aby v tíme všetko fungovalo (nosiť kavu :D, kontaktovaný veducim), nie je veduci
- template — kto bol prítomný, čo sa na stretnutí robilo, prípadne retrospec

Šprinty

- jeden sprint = 2 tyzdne
- 1. stretnutie — čo sa bude robiť
- pracovať
- na konci — ako sme splnili jednotlivé ulohy
- retrospec — čo sa pacilo, čo nie

Project management

- cloudovy **DevOps** cez studentsku licenciu
 - nie [možno fungujúci] stuba, ale ms cloud server
- vytvoriť accounts + sample project
- alternatívy:
 - jira? nie optimalna pre 3+4h sprinty
 - trello, gitops?

Role

- každý z nás za niečo zodpovedá
- manager of [...]
 - [documentation] — za všetku dokumentáciu
 - nerobí všetko v dokumentácii ten manažer, ale berie na seba zodpovednosť, keby sa niečo pokazilo

Backlog

- *externy* = bude zadávaný niekym externým mimo tímu
 - môže prejsť do nevhodnej granularity
- *interny* = prispôbenie externého pod náš projekt
 - budeme vidieť iba jeho v DevOps

User stories [US], tasks

- user story > tasks > subtasks
 - k US budeme mať pridelené niekoľko taskov (k jednotlivým taskom prípadne subtasks)
 - US — na úrovni storypointov, tasky — na úrovni hodín
- *user stories*
 - budeme to aj ohodnocovať cez storypoints
 - **planning poker**
 - nevzťahuje sa na hodiny, číslo nie je konkrétne
 - *kava* — vykaslime sa na to, dajme radšej pauzu, nikam to nevedie
 - ? — nepoužívať, radšej si niečo tipnime
 - 1 jednoducho, 21 strasne
 - ak dostaneme 21, tak to budeme musieť robiť
- v rámci sprintu sa rozhodnúť, ktoré US implementovať
 - do sprintu napchať niečo s najvyššou prioritou
- zodpovedný za US > rozbijeme na tasky > spravíme estimates > priradíme úlohy jednotlivým členom

Burndown chart [BC]

- nie na úrovni storypointov — ubudalo by to iba za US (ktorych máme iba 2)
- cize radšej ubudat z taskov
- úroveň estimateov a remainingov
- **treba sledovať, že sa BC mení naozaj**
 - v poriadku, ak sa na prvých sprintoch nepodarí; ale na konci vyhodnotenia riadenia sa *pozrú na burndown chart*
- keď BC stupňuje — zle naplánovanie, lebo nám stúpili tasky (a keď to dropne, tak je vidieť, že sme to spravili v jeden deň na konci)
- ako s bugmi a defektami?
 - mohlo by sa to dať aj cez git
 - v rámci Azure DevOps [pravdepodobne task] má typ bug — reportnes, naplánujeme kedy to budeš fixovať
 - neprihodíš to ako nejaký report
 - či to included priamo v project managemente, tak sa ešte dohodneme

Analytická časť

- od nás sa najviac očakáva návrh a implementácia
- grafová databáza
 - predbežne, ešte sa dohodne, či sa pôjde touto cestou
 - pretože je to vždy prepájané info
 - verziovanie
 - (nemusíme riešiť vizualizáciu)
 - **neo4j** = java graph db
 - backend môže byť v hocikom jazyku
- identifikovať typy projektov a fáz — ako vyzerá flow? (nie/len tento prípad sa bude riešiť)
 - napíše paper > pošle do systému 1 > reviewers napíšu review > review sa vráti chairovi > chair povie či prijať alebo nie ten paper > review sa vráti autorovi > spracovať, odovzdať odovzdať do ďalšieho systému > program konferencii je z prvého systému (stará verzia)
- môžu byť branchy dokumentov — spadlo by to do gitu, nemusíme asi
- management [spomínaný v zadani] — akoby 2. produkt, od nás sa to nečaká, ale aspoň niečo by sme mali touchnúť čo sa týka odrazok
- evidencia — upload sprievodných dokumentov ku konkrétnym dokumentom = prílohy
- web, kt. renderne pdf
 - nebudem pracovať so všetkými formátmi
 - formulár - web z db potiahne info, zobrazí v template
- eurocris — standard v rámci EU [zdroj zo zadania]
- typy projektov, fázy, proces výskumnej skupiny vzhľadom na fázu/typ projektu
- z toho všetkého sa spraví backlog a my si vyberieme, ktorú časť ideme implementovať

Návrh systému

- modelovanie — EA
 - use cases sa robia v komunikácii s nimi (radšej sa vyhnúť)
- diagram komponentov — pohľad na architekturu
- class diagram, sekvenčáky — nie

Implementácia

- rozchodiť environment
- docker — ak povieme, že bude len docker, tak bude len docker
- kubernetes
- na začiatku — test env, potom kvazi produkčný — stabilnejšia verzia, ktorú potom ukážeme
- pridať testy → *manazer testovania*
 - unit testy môžu byť všade — v kontexte toho celeho nemusia povedať veľa though
 - integration (pokryť viacero komponentov); selenium, simulácia používateľov (vykon automaticky)
- možno to bude bežať na rovnakom serveri ako súvisiaci projekt

Komunikácia

- dôležité z pohľadu riadenia (či bola integrácia)
 - msg do channelu
 - v rámci deploymentu
 - keď failnú testy
- continuous integration až keď spravíme env