

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt

Webable

Prvý kontrolný bod - dokumentácia k riadeniu

Tím: stableFamily (tím č. 10)

Vedúci tímu: Ing. Jakub Šimko, PhD.

Členovia tímu: Bc. Michal Fabiš, Bc. Katarína Rafčíková, Bc. Daniela Sitárová, Bc. Maroš Vašš, Bc. Andrej Zaťko, Bc. Martin Žák

Ak. rok: 2018/2019

Úvod	5
Role členov tímu a podiel práce	6
Aplikácie manažmentov	8
Manažment úloh	8
Vytváranie úloh	8
Prioritizovanie úloh (Backlog grooming)	9
Vyberanie úloh do šprintu	9
Vykonávanie úloh v šprinte	9
Manažment komunikácie	9
Manažment verzií a prehliadok kódu	10
Manažment dokumentácie	10
Sumarizácie šprintov	11
Prvý šprint - Alveola	11
Druhý šprint - Biceps	13
Tretí šprint - Clavicula	14
Globálna retrospektíva	16
Motivačný dokument	17
Metodiky	22
Definition of Ready	22
Definition of Done	23
Komunikácia	25
Pravidlá pre verzionovanie	26

Úvod

V rámci predmetu Tímový projekt sa tím stableFamily zaoberá tvorbou webového prehliadača pre nevidiacich s názvom Webable. V tomto dokumente sa nachádza podrobný opis spôsobu riadenia tímu v tomto projekte. V nasledujúcich častiach sa uvádza, aké zodpovednosti majú jednotliví členovia tímu, popis jednotlivých manažmentov, ktoré sú využívané, podrobný záznam z prvých troch šprintov, ktoré zatiaľ boli uskutočnené, globálna retrospektíva, motivačný dokument a jednotlivé metodiky, ktorými sa tím riadi.

Webový prehliadač pre nevidiacich Webable je projekt, ktorý vznikol počas minulého akademického roka, ako aplikácia určená do súťaže Imagine Cup. Preto je náš tím v odlišnej situácii oproti iným tímom, nakoľko pracujeme na softvéri, ktorý už bol vyvíjaný a preto sa stretávame s aj s inými problémami ako ostatné tímy. Táto situácia spoločne s nedostatkom skúseností s agilným prístupom spôsobila v prvých týždňoch, že sme pracovali neefektívne a darilo sa nám plniť iba malé množstvo úloh.

Role členov tímu a podiel práce

V rámci tímu vystupujú všetci členovia ako vlastníci produktu (product owner). Tiež sa všetci podieľajú na vývoji a okrem toho má každý ešte ďalšie špecifické úlohy, ktoré sa uvádzajú nižšie. Podiel práce jednotlivých členov tímu na častiach dokumentácie je uvedený v tabuľke 1.

Meno	Časti dokumentácie, na ktorých sa daný člen podieľal
Michal Fabiš	Sumarizácia šprintov
Katarína Rafčíková	Definition of Done
Daniela Sitárová	Úvod, Aplikácie manažmentov, Komunikácia
Andrej Slaninka	Definition of Ready
Maroš Vašš	Globálna retrospektíva
Andrej Zaťko	Pravidlá pre verzionovanie
Martin Žák	Motivačný dokument

Michal Fabiš

Michal je zodpovedný za administráciu nástroja JIRA, ktorý je používaný na podporu správy úloh pri agilnom vývoji. Správa nástroja JIRA obnáša počiatočnú konfiguráciu, to znamená vytvorenie kont pre jednotlivých členov tímu, prispôbenie rozhrania, aby vyhovovalo potrebám tímu a zadanie prvých úloh, a v priebehu práce na projekte vytváranie jednotlivých úloh a zapisovanie.

Katarína Rafčíková

Kataríninou úlohou je správa verzií kódu a vývoj pre Mac platformu. Je zodpovedná za úložisko kódu na Githube, s čím bola spojená počiatočná inicializácia a vytvorenie spôsobu manažovania jednotlivých verzií - vytváranie vetiev, nastavenie integrácie jednotlivých modulov a vykonávanie prehliadok kódu.

Daniela Sitárová

Daniela vykonáva funkciu zapisovateľa na jednotlivých tímových stretnutiach. Je zodpovedná za celkovú dokumentáciu - za vytváranie jednotlivých metodík, zaznamenávanie priebehu šprintov a retrospektív, dokumentáciu riadenia v tíme a dokumentáciu inžinierskeho diela. Je tiež jednou

z členov tímu, ktorí sa venovali vývoju webového prehliadača pre nevidiacich v rámci svojich bakalárskych prác.

Andrej Slaninka

Andrejovou úlohou v tíme je správa webového sídla. Navrhol a vytvoril webovú stránku a stará sa o to, aby bola vždy aktuálna. Tiež sa vývoju prehliadača venoval už v predošlom akademickom roku, preto sa tiež zaoberá analýzou riešení z pohľadu prístupnosti a je zodpovedný za navrhovanie nových funkcionalít.

Maroš Vašš

Maroš je zodpovedný za úlohy súvisiace s obchodnými stránkami nášho produktu. Zaoberá sa možnosťami propagácie, predaja, šírenia nášho webového prehliadača a komunikáciou s mentormi a potenciálnymi partnermi. Tiež sa podieľa na analýzach potrebných na vytváranie novej funkcionality v rámci softvéru.

Andrej Zaťko

Andrejovou úlohou je dohliadať na vytváranie testov a spôsob testovania. Táto úloha obnáša nakonfigurovanie testovacieho prostredia, vytvorenie šablón a zaškolenie ostatných členov tímu.

Martin Žák

Martin vystupuje v tíme v úlohe SCRUM mastera. Vede jednotlivé stretnutia a určuje ich priebeh. Počas šprintov kontroluje, či si ostatní členovia tímu plnia svoje povinnosti a ako si ich plnia. Zároveň tiež pracoval na projekte v rámci svojej bakalárskej práce, preto pozná celkovú architektúru systému a dokáže poradiť ostatným, menej skúseným členom tímu.

Aplikácie manažmentov

Pri riadení projektu je kľúčové, akým spôsobom prebiehajú jednotlivé procesy. Je dôležité dodržiavať určité pravidlá, ktoré si tím stanoví. V nasledujúcej časti sa podrobne zameriavame na tie, ktoré zatiaľ aplikujeme pri riadení nášho projektu.

Manažment úloh

Riadenie celého procesu vytvárania, hodnotenia a vykonávania úloh je najdôležitejšou a najobsiahlejšou časťou riadenia. Jednotliví členovia tímu predstavujú zároveň aj vlastníkov produktu, preto sa tiež podieľajú na vytváraní úloh.

Vytváranie úloh

Úlohy sú vytvárané priebežne, počas trvania jednotlivých šprintov. Všetky novovytvorené úlohy sa zhromažďujú v tzv. **backlogu**. Do backlogu môže pridať úlohy ktorýkoľvek člen tímu a aj vedúci, ktorý predstavuje tiež vlastníka produktu a reprezentuje ho na tímových stretnutiach.

Pre potreby nášho projektu sme definovali viaceré typy úloh:

- *Používateľská úloha* (User story) - úlohy tohto typu sú tie, ktoré sa bežne využívajú pri agilnom prístupe vývoja softvéru, jedná sa o scenár, ktorý by mal po vykonaní úlohy prehliadač splniť.
- *Analytická úloha* (Analytic story) - úlohy, v ktorých je potrebné vykonať analýzu nejakého problému alebo oblasti, výsledkom je vytvorenie nových používateľských príbehov
- *Úloha slúžiaca na tímový manažment* (Team management story) - úlohy súvisiace s riadením procesov v tíme, napr. vytváranie metodík a podobne.
- *Chybová úloha* (Error story) - úlohy slúžiace na opravovanie chýb, ktoré sa v kóde nachádzali pred začiatkom predmetu Tímový projekt

Jednotlivé úlohy sú zároveň združované do väčších celkov, tzv. **epics**. Tie sú definované nasledovne:

- *UX vylepšenia*
- *Výpočtové vylepšenia*
- *Nová funkcionálnosť*
- *Inteligentný webový prehliadač*
- *Infraštruktúra a refaktoring*

Každá úloha by mala spĺňať pravidlá, ktoré tím vytvoril vo vlastnej definícii pripravenosti úlohy (Definition of Ready), viď časť Metodiky - Definition of Ready. Pre jednotlivé typy úloh sú vytvorené odlišné definície.

Prioritizovanie úloh (Backlog grooming)

Počas jednotlivých tímových stretnutí je potrebné úlohy usporiadať do vhodného poradia. Poradie úloh určuje ich priorita, ktorá sa môže v priebehu jednotlivých šprintov meniť. Úlohy s najvyššou prioritou by sa mali nachádzať vo vrchnej časti backlogu.

Vyberanie úloh do šprintu

Do šprintu sa vyberajú úlohy s najvyššou prioritou, to znamená tie, ktoré sú navrchu backlogu. Náročnosť jednotlivých úloh je vyjadrovaná pomocou tzv. story pointov, ktoré určujú členovia tímu pomocou plánovacieho pokeru. Pri tejto metóde sa všetci musia zhodnúť na číslu, ktorým danú úlohu ohodnotia. Pokiaľ sa nezhodnú, diskutujú o svojich názoroch a potom opäť zvolia nejaké číslo. Tento proces sa opakuje, kým sa nezhodnú. Na základe počtu story pointov vie tím odhadnúť, koľko úloh môže zaradiť do ďalšieho šprintu. Počet úspešne splnených story pointov za jeden šprint reprezentuje výkonnosť tímu.

Vykonávanie úloh v šprinte

Po výbere úloh do šprintu je potrebné rozdeliť ich medzi jednotlivých členov tímu. Ak je to potrebné, vytvoria sa k úlohám podúlohy. Každá úloha je priradená jednému členovi tímu, ktorý je zodpovedný za jej vykonanie. K úlohe môže byť priradený aj spolupracovník, ktorý sa na nej bude podieľať a tiež sú k nej priradení členovia tímu, ktorí budú zodpovední za prehliadku kódu. Počas šprintu sa môže úloha nachádzať v týchto stavoch:

- *Urobiť* (To do)
- *Prebieha práca* (In progress)
- *Prípravené na prehliadku* (Ready to review)
- *Vykonáva sa prehliadka* (In review)
- *Hotovo* (Done)

Na to, aby mohla byť úloha presunutá do stavu *Hotovo*, musí spĺňať pravidlá definície hotovej úlohy (Definition of Done), viď časť Metodiky - Definition of Done.

Manažment komunikácie

Komunikácia v tíme je zabezpečovaná viacerými spôsobmi. Najviac sa využíva komunikačný nástroj Slack, zápisy v nástroji JIRA, Github a osobné stretnutia.

Slack je rozdelený na viacero kanálov, kde každý slúži na iný účel. Kanály sú nasledovné:

- *general* - slúžiaci na komunikáciu s vedúcim a oficiálne správy, ktoré sa týkajú všetkých.
- *standups* - kanál, do ktorého sa píše on-line reporty raz za týždeň, aby jednotliví členovia vedeli, na čom robia ostatní, podrobné pravidlá sa nachádzajú v časti Metodiky - Komunikácia.
- *privatechat* - komunikácia medzi členmi tímu.
- *jira-notifications* - upozornenia z nástroja JIRA.

V nástroji JIRA je možné vidieť progres jednotlivých členov tímu a zároveň je možné k úlohám pridávať komentáre, ak je to potrebné. Na Githube komunikujú jednotliví členovia tímu pri vykonávaní prehliadok kódu.

Manažment verzí a prehliadok kódu

Pri agilnom spôsobe vývoja softvéru je potrebné dôkladne spravovať jednotlivé verzie softvéru. Nakoľko na projekte pracuje viacero ľudí, všetci musia dodržiavať pravidlá, aby tím pracoval efektívne a správne. V našom tíme je používaný nástroj Github, ktorý slúži tiež na riadenie prehliadok kódu. Pravidlá na vytváranie verzí a spôsob vykonávania prehliadok kódu (Code reviews) sú popísané v časti Metodiky - Pravidlá pre verzionovanie.

Manažment dokumentácie

Dobrá dokumentácia je základom celého projektu. Na každom stretnutí je vytváraný podrobný zápis o priebehu, ktorý má vopred definovanú štruktúru. Obsahuje informácie o tom, kedy a kde sa stretnutie konalo, kto sa ho zúčastnil a čo bolo jeho obsahom. Ak ide o stretnutie, na ktorom sa ukončuje šprint, súčasťou zápisu je retrospektíva skladajúca sa z troch častí:

- V čom pokračovať?
- S čím prestať?
- S čím začať?

Pokiaľ sa v priebehu šprintov objaví nový proces, ktorý potrebuje mať určité pravidlá, je vytvorená metodika. Ďalšou formou dokumentácie sú zápisy v nástroji JIRA.

Sumarizácie šprintov

V tejto kapitole sa nachádza prehľad úloh, ktoré boli zaradené do jednotlivých šprintov. Ak sa nepodarilo úlohu splniť, je pri nej uvedené, prečo sa to stalo.

Prvý šprint - Alveola

Prvý šprint slúžil tímu na to, aby sa zoznámil s agilným spôsobom vývoja softvéru. Počas tohto šprintu boli nastavované rozličné pravidlá a postupy, ako má tím fungovať tak, aby bol efektívny. Zároveň sa členovia tímu, ktorí nemali predošlú skúsenosť s vyvíjaným softvérom oboznamovali so štruktúrou, architektúrou a fungovaním nášho webového prehliadača. Celkový počet story pointov zaradených do šprintu bol 11, avšak nepodarilo sa splniť ani jeden. Dôvodom nesplnenia bola nedostatočná vedomosť členov o tom, ako funguje softvér a najmä príliš prísne stanovená definícia dokončenej úlohy. V definícii bolo jednou z podmienok, že nový kód musí byť otestovaný, avšak počas tohto šprintu sa nepodarilo nakonfigurovať fungujúce testovanie.

Úlohy

Názov úlohy: Skryť tutoriál

Popis: V prehliadači sa zobrazuje tutoriál pre používateľa. Je potrebné tento tutoriál odstrániť.

Počet story pointov: 1

Splnené: Nie

Dôvod nesplnenia: Príliš prísna definícia ukončenej úlohy a zlé rozloženie času, lebo nebola vykonaná prehliadka kódu.

Názov úlohy: Fix chyby: Pri znovu pripojení počítača k internetu sa nedá načítať predtým zadaná stránka

Popis: Postup k zreprodukovaniu:

1. Odpojiť počítač od internetu
2. V url bare napísať "sme.sk" a zadať enter
3. Pripojiť počítač k internetu
4. Prejsť na url bar a zadať enter

Akceptačné kritériá:

- Po kroku číslo 4. sa stránka v url bare načíta.

Počet story pointov: 2

Splnené: Nie

Dôvod nesplnenia: Nedostatočná orientácia v kóde.

Názov úlohy: Ako používateľ pri výpadku internetového pripojenia, chcem aby namiesto bielej stránky bola zobrazená primeraná chybová hláška, aby som bol informovaný

Popis: Postup k zreprodukovaniu:

1. Odpojiť počítač od internetu
2. V url bare napísať "sme.sk" a zadať enter

Akceptačné kritériá:

- Zobrazí sa error page, ktorá informuje používateľa o konkrétnom stave. (Buď problém u klienta alebo na serveri)
- Po zobrazení error page na určitej karte chcem, aby som užívateľ mohol navštíviť akúkoľvek webovú stránku
- Nech je error aktuálny po každom obnovení stránky

Počet story pointov: 3

Splnené: Nie

Dôvod nespĺnenia: Príliš prísna definícia ukončenej úlohy a zlé rozloženie času, lebo nebola vykonaná prehliadka kódu.

Názov úlohy: Ako používateľ po zadaní kľúčových slov, chcem byť presmerovaný na výsledky vyhľadávania googlom, aby som pohodlnejšie našiel to, čo hľadám

Popis: Postup k zreprodukovaniu:

1. V url bare napísať "test test" a zadať enter

Obsah error okna: ERROR LANGUAGES

"\ " Message: undefined Error: [object Object]"

Akceptačné kritériá:

- pokiaľ používateľ zadá kľúčové slová do URL baru v prehliadači, spustí sa vyhľadávanie cez search engine, ktorý má používateľ vybraný v nastaveniach prehliadača
- Automaticky vyhľadáva v predvolenom vyhľadávacom nástroji.

Počet story pointov: 1

Splnené: Nie

Dôvod nespĺnenia: Príliš prísna definícia ukončenej úlohy a zlé rozloženie času, lebo nebola vykonaná prehliadka kódu.

Názov úlohy: Fix chyby: Pri zatvorení sitemapy má TAB prehliadača stále názov "SITEMAP"

Popis: Postup k zreprodukovaniu:

1. Otvoriť sitemapu(klávesová skratka CTRL + M)

2. Zatvoriť sitemapu(klávesová skratka CTRL + M)

Akceptačné kritériá:

- Názov aktívneho TABU prislúchať otvorenej stránke.

Počet story pointov: 1

Splnené: Nie

Dôvod nespĺnenia: Príliš prísna definícia ukončenej úlohy a zlé rozloženie času, lebo nebola vykonaná prehliadka kódu.

Názov úlohy: Uskutočnenie pilotného testovania

Popis: Ak by to bolo možné, je potrebné uskutočniť pilotné testovanie podľa vopred navrhnutého testovacieho scenára.

Počet story pointov: 0

Splnené: Nie

Dôvod nespĺnenia: Neboli splnené základné podmienky umožňujúce vykonanie testovania.

Druhý šprint - Biceps

V tomto šprinte tím prispôbil svoje správanie podľa toho, aké chyby boli urobené v prvom šprinte. Prispôbili sa podmienky definície dokončenej úlohy tak, aby bolo možné úlohy označiť ako hotové, aj keď nebolo možné vykonať testovanie. Tiež boli aktualizované pravidlá pre vykonávanie prehliadok kódu.

Náplňou tohto šprintu bolo dokončenie úloh z predošlého šprintu a pokračovanie vo vylepšovaní použiteľnosti prehliadača. Všetky úlohy z predošlého šprintu okrem zobrazovania chybovej hlášky sa podarilo ukončiť. Úloha na vykonanie pilotného testovania nebola zaradená, pretože sa tím rozhodol, že ju zaradí až vtedy, keď bude reálne uskutočniteľná. Táto úloha nebola dokončená, pretože sa nepodarilo zmeny pridať do aktuálnej verzie. Celkový počet story pointov zaradených do tohto šprintu bol 21. Podarilo sa splniť 15.

Úlohy

Okrem úloh prenesených z predošlého šprintu boli pridané aj ďalšie úlohy.

Názov úlohy: Zistiť ako funguje webpack a popísať jeho jednotlivé časti

Popis: Analytická úloha zameraná na lepšie pochopenie fungovania prehliadača.

Počet story pointov: 5

Splnené: Áno

Názov úlohy: Identifikovať a vyriešiť problém, ktorý nás brzdí vo vývoji prehliadača

Popis: Aktuálny stav:

- pri zmene vetvy po zadaní `npm install` sa nedotiahnutý dependencies. Musím vymazať priečinky `dist` a `node_modules` a nanovo spustiť `npm install`, aby mi veci fungovali (ale nefunguje to vždy)

Očakávaný stav

- po zmene vetvy po zadaní `npm install` sa nainštalujú všetky dependencies

Počet story pointov: 5

Splnené: Áno

Názov úlohy: Zistiť, ktoré balíčky nepoužívame/používame

Popis: Analytická úloha zameraná na lepšie pochopenie fungovania prehliadača.

Počet story pointov: 3

Splnené: Nie

Dôvod nesplnenia: Nedostatočná komunikácia medzi členmi tímu.

Tretí šprint - Clavicula

V tomto šprinte tím zapracoval do procesu riadenia poznatky získané v predošlom šprinte, týkajúce sa najmä prehliadok kódu. Vďaka úpravám môže tím pracovať efektívnejšie. Počas šprintu sa tím zaoberal najmä ďalším vylepšovaním použiteľnosti prehliadača, integráciou modulu na analýzu a úpravu kódu a analytickými úlohami. Celkový počet zaradených story pointov bol 17, podarilo sa splniť 9.

Úlohy

Do tohto šprintu boli prenesené aj dve úlohy z predchádzajúceho, ktoré sa podarilo splniť. Okrem nich boli pridané aj ďalšie úlohy, na ktorých tím pracoval.

Názov úlohy: Zistiť, ako nakonfigurovať vybraný framework na testovanie tak, aby sme mohli vytvárať unit testy

Popis: Zistiť, nakoľko treba upraviť konfiguráciu vo webpacku tak, aby fungovalo testovanie

Počet story pointov: 0

Splnené: Nie

Dôvod nesplnenia: Zložitý problém, na ktorý sa zatiaľ nepodarilo nájsť riešenie.

Názov úlohy: Fix chyby: Pri navštívení akejkoľvek stránky sa daná stránka načíta dvakrát

Popis: Opis reprodukovania:

- Stránka ktorú chcem načítať, sa reloadne po jej načítaní. Napríklad navštívim stránku www.sme.sk. Prehliadač načíta stránku <http://www.sme.sk> a potom načíta <https://www.sme.sk>

Počet story pointov: 3

Splnené: Áno

Názov úlohy: Modul analýza a úprava kódu: Nájdi lepší prístup textovej analýzy aký zatiaľ používame

Popis: Analytická úloha, slúžiaca na podporu vytvárania novej funkcionality

Počet story pointov: 0

Splnené: Áno

Názov úlohy: Ako používateľ, chcem aby prehliadač opravoval formuláre

Popis: Aktuálny stav:

- Existujúci modul zatiaľ funguje mimo nášho prehliadača

Akceptačné kritériá:

- Vymyslená infraštruktúra v prehliadači, aby sa do budúcnosti ľahšie pridávali moduly

Počet story pointov: 8

Splnené: Nie

Dôvod nesplnenia: Nedodržanie termínu na odovzdanie kódu na prehliadku.

Globálna retrospektíva

Po ukončení každého šprintu tím v retrospektíve zhodnotil, aký bol priebeh šprintu. V tejto časti dokumentu sú opísané poznatky, ktoré boli nadobudnuté počas prvých troch šprintov.

Úvod projektu sa niesol v duchu nastavovania pravidiel a vylepšovania jednotlivých procesov v rámci riadenia. Základom sú dobre definované kritériá, ktoré by mala spĺňať každá úloha, či už pri vytváraní alebo pri rozhodovaní, kedy je hotová. Na začiatku boli tieto kritériá stanovené príliš prísne, takže nebolo možné ich splniť. Ukázalo sa, že správne plánovanie je mimoriadne dôležité. Počas prvých šprintov sme zle odhadli náročnosť jednotlivých úloh, čo sa odrazilo na počte získaných story pointov za jeden šprint. Zároveň sme prišli na to, že je potrebné stanoviť naozaj podrobné pravidlá na jednotlivé procesy. Najviac sme sa zaoberali spôsobom vykonávania prehliadok kódu. Počas druhého šprintu totiž došlo k problémom spôsobeným práve nedostatočne definovaným pravidlami prehliadok kódu.

Po troch šprintoch sa nám podarilo vylepšiť jednotlivé procesy a vykonať úlohy tak, že je softvér v použiteľnom stave a môžeme začať pracovať na príprave používateľského testovania, ktoré je mimoriadne dôležité pre ďalší vývoj. V nasledujúcom období sa chceme tiež ďalej venovať refaktoringu a zlepšovaniu jednotlivých komponentov softvéru. Cieľom je, aby tím pracoval efektívne a mohol na konci zimného semestra prezentovať výsledky svojej práce v podobe prototypu.

Motivačný dokument

Tímový projekt - Webable (ImagineCup projekt)

Úvod

V dnešnej dobe je internet neoddeliteľnou súčasťou nášho každodenného života. Ľudia na ňom poväčšine trávajú čas na sociálnych sieťach, spravodajských či video portáloch, alebo nakupovaním. A to všetko z pohodlia domova, priamo cez internet, na zopár klikov. No nie pre všetkých je to také jednoduché. Na svete je veľké množstvo zrakovo postihnutých ľudí (slabozrakí alebo úplne nevidiaci), ktorí pri práci s počítačom používajú rôzne asistenčné technológie. Jednou z nich je napríklad **čítačka obrazovky**. Vďaka nej títo ľudia vedia, čo sa nachádza na obrazovke.

Pre lepšie pochopenie, čítačka obrazovky je softvér, ktorý prevádza text na obrazovke do reči, tzv. „text-to-speech“ (TTS) softvér. V praxi to asi funguje tak, že keď sa takýto zrakovo postihnutý používateľ naviguje k nejakému elementu na webovej stránke, čítačka obrazovky prečíta jeho textovú reprezentáciu. Na pohyb po elementoch na webových stránkach, desktopových aplikáciach alebo používateľskom rozhraní rôznych operačných systémov títo ľudia nepoužívajú myš, na ktorú sú vidiaci ľudia zvyknutí. Namiesto nej používajú klávesové skratky na klávesnici, pomocou ktorých vedia niektoré činnosti vykonať veľmi rýchlo, ale niektoré aj veľmi pomaly. Napríklad sekvenčné prechádzanie nezaujímavých alebo nie dôležitých informácií je pre týchto ľudí veľmi unavujúce ak neexistuje možnosť, vďaka ktorej by mohli takýto obsah preskočiť.

Samotné čítanie obrazovky čelí **2 zásadným problémom**:

1. **Používateľ je preťažený informáciami** - Zrakovo postihnutý používateľ pomerne často musí sekvenčne prechádzať celý obsah na webovej stránke. Popri tom musí počúvať čítačku obrazovky a na základe nej sa rozhodovať, čo robiť ďalej. Často sa stáva, obzvlášť na neznámych stránkach, že užívateľ si musí zapamätať presný názov webového elementu, aj ak je nesprávne pomenovaný (napr. tlačidlo tu, sem), aby pri ďalšom navštívení tej istej stránky hneď vedel povedať, čo hľadá. Nehovoriac o tom, že ak sú na neznámej stránke zle popísané odkazy na iné stránky či podstránky, zrakovo postihnutému používateľovi veľmi dlho trvá, kým nájde to čo hľadá. Odkazy musí navštíviť, novo navštívené stránky zbežne prejsť a až tak zistí, čo sa na nich nachádza. Jednoducho vyhľadávanie špecifických informácií je pre zrakovo postihnutých ľudí častokrát ťažkopádne a náročné.
2. **Pri programovaní webových stránok web developeri ignorujú webové štandardy a čítačky obrazoviek tak nie sú schopné interpretovať obsah na stránke** - V súčasnosti sa webové stránky programujú tak, aby mali z nich vidiaci používatelia čo najlepší používateľský a estetický zážitok. To znamená, že web developeri sa zameriavajú skôr na to, ako stránka vyzerá a nie na korektnosť jej kódu.

Čítačky obrazoviek sú však závislé od toho, ako dobre je stránka naprogramovaná. Teda ak je kód stránky korektný, čítačka obrazovky vie umožniť zrakovo postihnutému používateľovi efektívnu navigáciu po stránke (pomocou klávesových skratiek sa ovláda *focus* čítačky, resp. to, kde sa práve nachádza) a získať z nej všetky potrebné informácie. V opačnom prípade je navigácia neefektívna, čítačka nie je schopná niektoré informácie spracovať alebo má problém vôbec sa k nim dostať.

Cieľ

Naším cieľom je vytvoriť nástroj, ktorý by vylepšil navigáciu a orientáciu na webe pre zrakovo postihnutých používateľov. Tým by sme zároveň zlepšili ich používateľský zážitok. Toto chceme docieľiť niekoľkými spôsobmi. Niektoré z nich sú nízko úrovňové a zameriavajú sa na automatizovanú analýzu a korekciu kódu, zlepšenie súčasnej a pridanie dodatočnej navigácie s apelom na ich prístupnosť (prístupnosť = aby zrakovo postihnutí používatelia mali k tomu prístup) a iné, vysoko úrovňové, zas na to, aby sa pri návšteve stránky dbalo na používateľský zámer, resp. to, s akým zámerom prišiel používateľ na webovú stránku.

Automatizovaná analýza a korekcia kódu

Máme na mysli pridávanie chýbajúcich popisov k elementom, bez ktorých sú neprístupné, resp. bez ktorých čítačka nie je schopná prečítať textovú reprezentáciu elementu. Nimi sú napríklad obrázkové tlačidlá, obrázky s textom, na ktoré sa dá aplikovať OCR, alebo textové polia vo formulároch, kde ich popisným elementom (element label) chýba atribút *for*. Toto chceme docieľiť analýzou kódu a vykonaním potrebných úprav.

Mapa webovej lokality

Ide o dodatočnú navigáciu, ktorú si bude môcť užívateľ zobraziť vyvolaním nejakej klávesovej skratky. Tento modul sa bude snažiť riešiť ten problém, kedy zrakovo postihnutí používatelia sú obzvlášť na neznámych stránkach dezorientovaní, keďže nepoznajú štruktúru webovej lokality (na aké stránky a podstránky sa člení webová lokalita) a častokrát nevedia, kde sa nachádzajú. Mapa webovej lokality s prístupnou efektívnou navigáciou a vyhľadávaním im umožní nadobudnúť lepšiu predstavu o hierarchickej štruktúre webovej lokality a rýchlejšie sa dostať na stránku, ktorú potrebujú.

Vytváranie zhlukov logicky súvisiacich elementov

Týmto chceme zlepšiť a tak urýchliť navigáciu po webe. Používateľ sa bude môcť pohybovať po blokoch a keď bude chcieť, môže sa do bloku vnoriť a v ňom sa ďalej navigovať. Predstavte si napríklad príspevok na facebooku. Ten má až 8 elementov. Vďaka zhlukovaniu sa bude môcť používateľ navigovať po jednotlivých príspevkoch a nie po jeho elementoch.

Používateľský zámer

Tuto funkcionality by sme v našom riešení chceli zakomponovať tak, že používateľ zadá do dialógového okna to, čo chce na danej stránke/webovej lokalite robiť a my mu vytvoríme zoznam odporúčaných stránok na navštívenej webovej lokalite s prípadným *focusom* na daný element, ktorý môže byť tlačidlom na vykonanie nejakej akcie, alebo iným elementom, ktorý hľadal. Tento odporúčaný zoznam by mohol preskočiť a navigovať sa po hlavnom obsahu stránky.

Cieľová skupina

Najväčšie skupiny ľudí, ktorých sa týkajú tieto problémy sú:

- **nevidiaci:** ľudia, ktorí sú odkázaní používať na interakciu s počítačom iné zmysly ako zrak, najčastejšie sluch. Sú to ľudia, ktorí sa bez zraku narodili alebo ho stratili dôsledku choroby či nehody. Ich životný štýl je jemne odlišný ale potreby majú rovnaké ako bežne vidiaci človek. Pri interakcii s počítačom sú pre nich grafické elementy "odpad" / bariéru (napríklad obrázkové tlačidlá, obrázky obsahujúce textové informácie bez alternatívneho popisu, atď.)
- **prakticky nevidiaci:** ľudia, ktorí majú zachované zvyšky zraku, vďaka čomu vnímajú svetlo, obrysy a tvary predmetov, v niektorých prípadoch aj veľký text (pokiaľ je s podkladom vo vysokom kontraste). Grafické elementy na stránkach im často sťažujú extrakciu informácií. Preto používajú rôzne nástroje, ktoré dokážu odstrániť štýly a zobrazujú čisté HTML.
- **slabozrakí:** ľudia, ktorí majú väčšie alebo menšie problémy so zrakom. Vo všeobecnosti majú tiež problémy s vykonávaním zrakovej práce, ale disponujú užitočnými zvyškami zraku, ktoré vedia využívať. Grafické elementy na stránkach im často sťažujú extrakciu informácií. Preto používajú rôzne nástroje, ktoré dokážu odstrániť štýly a zobrazujú čisté HTML.
- (Zrakové problémy môžu byť u dvoch používateľov z tej istej kategórie úplne odlišné. Napríklad: farbosleposť a astigmatizmus)
- **starší ľudia:** ľudia, so zhoršujúcim sa zrakom, ktorí potrebujú čo najviac intuitívne prostredie s jednoduchým ovládaním. Pri tejto cieľovej skupine sú tiež časté menšie problémy so zrakom.

V súčasnosti máme nadviazanú intenzívnu spoluprácu s viacerými nevidiacimi a slabozrakými používateľmi, s Centrom podpory študentov so špecifickými potrebami Univerzity Komenského v Bratislave, so ZŠ internátna pre slabozrakých a nevidiacich v Bratislave a s Úniou nevidiacich a slabozrakých Slovenska.

Aktuálny stav

Projektom sme sa zaoberali počas bakalárskeho ročníka v rámci súťaže ImagineCup. Naším riešením je webový prehliadač, ktorý má implementované niektoré z modulov, ktoré boli uvedené v kapitole *Úvod*. Pre vlastný prehliadač sme sa rozhodli preto, lebo sme nechceli byť obmedzovaný API už existujúcich prehliadačov, keďže niektoré časti rôznych algoritmov si vyžadujú nemalé množstvo výpočtovej sily. Taktiež sme nechceli byť platformovo závislý od konkrétneho prehliadača.

Prototyp

Náš prehliadač s názvom Webable je spustiteľný a disponuje jednak základnou funkcionalitou, ktorú obsahujú všetky dnešné moderné webové prehliadače ale aj našou vlastnou. Jednými zo základných funkcionalít dnešných bežných webových prehliadačov, ktorou disponuje aj ten náš, sú napríklad tlačidlá potrebné pri navštevovaní stránok, URL input, do ktorého môže používateľ zadať URL adresu a následne bude presmerovaný na príslušný web. Taktiež náš prehliadač ponúka základné nastavenia, kde si môže používateľ nastaviť domovskú stránku, vyhľadávací engine a iné. Pri navštívení zaujímavej stránky je možné uložiť si ju do záložiek. Zobrazenie histórie prehliadaných webových stránok je tiež nezanedbateľnou súčasťou. Popri tom všetkom sme však dbali na prístupnosť každej jednej funkcionality. Z našich, vlastných externých modulov, máme plne funkčný modul generovania máp webových lokalít, ktorú si môže používateľ zobraziť vyvolaním klávesovej skratky Ctrl+M. Taktiež je umožnená navigácia po mape pomocou šípok hore, dolu, doľava, doprava a vyhľadávanie, ktoré vyhľadáva nad názvami stránok v mape nielen podľa zadaného slova, ale aj jeho synonym. Korekcia kódu a zhlukovanie elementov sú čiastočne implementované, ale ešte nie integrované. Implementácia používateľského zámeru je momentálne v teoretickej rovine ale počas tímového projektu by sme sa chceli ním viac zaoberať.

Súčasťou nášho riešenia je už existujúca čítačka obrazovky NVDA, ktorú pri inštalácii nášho prehliadača ponúkame používateľovi nainštalovať. A to iba v prípade, že ju na počítači ešte nemá.

Feedback

Samotnému vývoju nášho prehliadača a identifikovaniu jeho základných čŕt, ktoré by vedeli zrakovo postihnutým ľuďom pomôcť samozrejme predchádzala a celý vývoj aj sprevádzala komunikácia s týmito ľuďmi. Primárne sme komunikovali s nevidiacim rádio reportérom Ondrejom Rosíkom, ku ktorému sme sa dostali cez našu fakultu. Pomohol nám jasne definovať problémy, s ktorými sa zrakovo postihnutí ľudia najčastejšie stretávajú a od ktorých sme sa mohli odraziť. Postupom času keď naše riešenie začalo nadobúdať reálne kontúry, predstavili sme ho v organizácii Únia nevidiacich a slabozrakých Slovenska, kde sme dostali pozitívny feedback. Taktiež sa nám ho podarilo odprezentovať v jednej škole pre zrakovo postihnuté deti, kde nám projekt taktiež odobrili a prejavili oň záujem. Stretli sme sa aj s tým, že náš nápad sa začal svojvoľne šíriť a niekedy nám zavolali alebo napísali email pre nás neznámy zrakovo postihnutí ľudia, ktorí sa nás spýtali, kedy bude naše riešenie k dispozícii, lebo by mali oň záujem.

Plány

Technické ciele

- **pripraviť MVP**
 - revidovať analýzu problému
 - zhodnotenie všetkých problémov, ktoré sme doteraz identifikovali + identifikácia nových
 - automatizovaná analýza a korekcia kódu formulárov na webových stránkach
 - integrácia externého modulu, ktorý opravuje chyby vo formulároch na stránke, vďaka čomu zrakovo postihnutý používateľ bude schopný samostatne vyplniť všetky potrebné údaje alebo mu to bude trvať oveľa kratšie ako dnes
 - overenie riešenia
 - **čas** - ako dlho trvá vyplnenie formuláru v inom prehliadači a potom v našom?
 - **pomoc inej osoby** - bola potrebná?
 - mapa webovej lokality
 - zlepšenie výkonnosti modulu, vďaka čomu bude menšia odozva pri interagovaní s mapou
 - zlepšenie prístupnosti generovanej štruktúrovanej mapy pre webové lokality
 - lepší on-boarding pre používateľov, ktorý sa majú naučiť pracovať s mapou webovej lokality (treba spraviť nejaký stručný návod, interaktívny tutoriál, alebo niečo iné)
 - overenie riešenia
 - **čas** - ako dlho trvá navigácia na konkrétnu stránku v rámci webovej lokality bez použitia mapy a s použitím
 - **pomoc inej osoby** - bola potrebná?

- vytváranie zhlukov logicky súvisiacich elementov
 - integrácia hotového modulu, ktorého úlohou je zrýchliť navigáciu na stránke a zjednodušiť pohyb medzi elementami
 - overenie riešenia
 - **čas** - ako dlho trvá kým sa používateľ dostane ku konkrétnej informácii v inom prehliadači a v našom?
 - **zredukovanie počtu prechodov** - koľko stlačení klávesových skratiek musí vykonať na inom prehliadači a na našom aby sa dostal ku konkrétnej informácii
 - **pomoc inej osoby** - bola potrebná?
- optimalizovanie správania nevidaceho používateľa pri práci s prehliadačom
 - UX záležitosť / zjednodušenie navigácie a práce s prehliadačom
 - overenie riešenia
 - **čas** - ako dlho trvá zmeniť nastavenie prehliadača?
 - **pomoc inej osoby** - bola potrebná?

Biznis ciele

- **analýza rôznych druhov biznis modelov a výber toho najvhodnejšieho vzhľadom na potenciál nášho projektu**
 - Komerčná sféra
 - analýza + zhodnotenie
 - Free
 - analýza + zhodnotenie
 - Open Source
 - analýza + zhodnotenie

Metodiky

V tejto časti sú spísané jednotlivé metodiky v takej podobe, ako sú používané v rámci práce na projekte. Využitie pri konkrétnych spôsoboch manažmentov je uvádzané v časti Aplikácie manažmentov.

Definition of Ready

User story

- musí byť nezávislá od iných stories
- musí sa dať vytvoriť test
- musí mať nejakú pridanú hodnotu pre koncového používateľa
- musia byť spísané akceptačné kritériá
- musí byť napísaná ako user story (AKO....,CHCEM...,ABY...)
- musí byť zrozumiteľná pre všetkých členov tímu (aj pre produktového vlastníka)
- je spísaný podrobný opis (používateľský scenár, eventuálne nakreslené obrazovky)
- všetci členovia tímu sa musia vedieť zhodnúť na konkrétnom počte story pointov
- každá story musí byť menšia/rovná 13 story points

Analytická story

- musí byť nezávislá od iných stories
- musí mať nejakú pridanú hodnotu
- musí byť zrozumiteľná pre všetkých členov tímu (v čom budem robiť prieskum?)
- musia byť jasne sformulované konkrétne otázky, na ktoré má dať analýza odpoveď
- všetci členovia tímu sa musia vedieť zhodnúť na konkrétnom počte story pointov

Error story

- chybovosť vyjadrená zlomkom (1/z akého počtu skúšani sa stalo)
- musí byť spísané ako sa dá error zreprodukovať
- vždy, keď sa dá, treba mať obrázok
- očakávaný stav - ako chcem, aby to fungovalo bez chyby
- musí byť nezávislá od iných stories
- každá story musí byť menšia/rovná 13 story points?????
- všetci členovia tímu sa musia vedieť zhodnúť na konkrétnom počte story pointov

Definition of Done

Tento dokument obsahuje DOD pre:

- Definitioin of Done pre user story
- Definition of Done pre analytickú story
- Definition of Done pre error story
- Definition of Done pre šprint
- Definition of Done pre release

Definition of Done pre user story

- Musí byť implementovaná
- Kód musí byť patrične zdokumentovaný (podľa našich pravidiel písania prehľadného kódu)
- Dokumentácia je zaktualizovaná podľa pridanej funkcionality
- Implementácia musí byť pokrytá testami
- Musia zbehnúť všetky testy bez chyby
- Lokálny build prejde bez chýb
- Implementované riešenie a testy su zrevidované aspoň jedným členom z tímu (pri pull requeste)
- Vetva, v ktorej sa vyvíjala nová funkcionality, musí byť zmergovaná do príslušnej vetvy šprintu
- Po zintegrování vetvy danej user story s vetvou sprint musí build prejsť bez chyby (remote)
- Musia byť splnené akceptačné kritériá

Definition of Done pre analytickú story

- Musí byť vytvorený výstupný dokument, ktorý bude obsahovať časti
 - Meno autora dokumentu
 - Zadanie témy, ktorá sa má analyzovať
 - stručný opis postupu riešenia
 - výsledok analýzy (odpovede na položené otázky)
 - V prípade, že je výsledkom analýzy odporúčenie zaviesť novú technológiu, potom musí byť táto technológia do podrobna charakterizovaná
- Z dokumentu musí byť jednoznačne jasné, či sa dá problém riešiť a ak áno, tak ako.
- Dokument musí byť uložený na zdieľanom úložisku dokumentov vo formáte markdown.
- Dokument musí byť zrevidovaný aspoň jedným členom z tímu
- Z výstupného dokumentu musí byť jasné, ako sa výsledky dajú uplatniť pri implementovaní ďalšej funkcionality

Definition of Done pre error story

- Chyba musí spĺňať oćakávaný stav ktorý bol zapísaný v Error story.
- Chyba musí byť opravená pre každý aktuálne podporovaný systém.
- Musia zbehnúť všetky testy bez chyby
- Musia byť napísané testy ktoré pokrývajú okrajové prípady
- Lokálny build musí zbehnúť bez chyby
- Implementované riešenie a testy su zrevidované aspoň jedným členom z tímu (pri pull requeste)
- Vetva, v ktorej sa opravovala chyba, musí byť zmergovaná do príslušnej vetvy šprintu
- Po zintegrování vetvy danej error story s vetvou sprint musí build prejsť bez chyby (remote)

Definition of Done pre šprint

- Keď sú všetky stories zo šprintu v stave done
- Vetva šprintu bude zmergovaná do vetvy dev

Definition of Done pre release

- Keď je projekt vybuildovaný bez chyby.
- Projekt spĺňa všetky jednotkové testy a funkčné testy sú zelené.
- Všetky akceptačné kritéria sú splnené.
- Product owner a tím odsúhlasili, že môže byť produkt vydaný.
- Používateľská príručka je doplnená o všetky zmeny.

Komunikácia

Tento dokument opisuje, ako medzi sebou komunikujeme v tíme a aké spôsoby používame.

Standups

Standups realizujeme v pondelky, keď spoločne pracujeme na projekte. Okrem toho sme zaviedli aj on-line standups prostredníctvom Slacku. Podstatou je, že každý člen tímu napíše do kanála určeného špeciálne na standups, čo urobil a čo plánuje robiť. Tento zápis by každý člen tímu mal urobiť v piatok poobede tak, aby bol celý standup kompletný v sobotu ráno.

Správa na Slack-u by mala mať nasledovnú štruktúru:

UROBIL SOM:

- prvá vec
- druhá vec
- ...

PLÁNUJEM:

- prvá vec
- druhá vec
- ...

MÁM PROBLÉM S: (nemusi byť, iba ak má daný člen tímu nejaký problém)

- prvý problém
- druhý problém
- ...

V príspevkoch je dobré označovať ľudí, s ktorými ste pracovali na daných úlohách, prípadne pri problémoch označovať konkrétnych ľudí, ktorí by mohli pomôcť problém vyriešiť.

Pravidlá pre verzionovanie

Commity

- Písané v jazyku angličtina
- Ak je vetva user story - US
 - commity sa budú vytvárať na vetvách user stories nasledovne T{Číslo tasku}: {Popis} (príklad: na vetve US53 bude nasledovný commit -> T35: Fix title on sitemap open/close...)
 - text commitu by mal obsahovať čo najviac kľúčových slov (dať do neho názvy všetkých modulov, ktorých sa týka zmena)
- Ak je vetva analytic story - AS
 - commity sa budú vytvárať na vetvách analytic stories nasledovne T{Číslo tasku}: {Popis} (príklad: na vetve AS53 bude nasledovný commit -> T35: Create list of packages, which are used in Webable and describe them...)
- Commit message by mala obsahovať čo najviac kľúčových slov (názvy všetkých modulov, ktorých sa týka zmena a pod.)

Vetvy

- master
 - hlavná vetva, ktorá je vždy funkčná
 - je možné ju meniť len cez pull requesty
 - kód v tejto vetve je nasadený v produkcii
 - build prebieha bez chyby
 - všetky testy sú úspešné
- dev
 - vetva z mastra
- sprint
 - vetva z devu
 - merge sa vždy na konci šprintu
- ID user story z Jiry (príklad názvu: US30 - UserStory ID 30)
 - vetva zo sprintu
 - developer, ktorý bude počas šprintu pracovať na konkrétnej user story, si vytvorí vetvu s názvom {ID user story}. V tejto vetve bude pracovať dovedy, pokiaľ nebude spĺňať body Definition of Done. Po dosiahnutí DOD môže vytvoriť Pull request na vetvu pre aktuálny šprint.

Pull requesty

Každý task v Jire musí mať prideleného človeka zodpovedného za review, ktorý mergeje pull request po splnení podmienok. Človek zodpovedný za review nemôže byť tá istá osoba ako tá, ktorá pracovala na tasku.

Kód pred podaním požiadavky na pull request musí byť:

- otestovaný so sprint vetvou (lokálne u seba sa spraví merge sprint vetvy do vetvy user story, ktorá je dokončená a tým sa zistí či user story bude fungovať so sprint vetvou)
- primerane komentovaný (pre účely dokumentácie).

Minimálne požiadavky pre metódy:

- krátke popísanie každej metódy
- opísanie parametrov
- návratu metód
- ďalšie vhodné informácie

Príklad:

```
/**  
 * Normalizes the url according to the conventions if the user does not specify all  
of its parts and also handles requests for special Webable url  
 * @param {string} path url we want to normalize and find out if it is not special  
webable one  
 * @returns {{ label: string, url: string }} label is what user sees and url the  
real address  
 * @memberof BrowserComponent  
 */
```

Podmienky pre pull requesty do vetiev, počet členov tímu, ktorý musí schváliť PR:

- sprint
 - minimálne 2 členovia tímu
- dev
 - minimálne 3 členovia tímu
- master
 - všetci členovia tímu

Code review

- Komentáre pri code review budú v angličtine so snahou písať ich čo najmenej ofenzívne s cieľom niečo sa dozvedieť alebo poradiť.
- Pri code review je povinný reviewer otestovať nové zmeny lokálne, vybuildovaním prehliadača z vetvy, ktorá sa bude mergovať. Nestačí len “pozrieť kód”.
- Po napísaní pripomienok, ktoré treba zapracovať, reviewer v jire presunie úlohu zo stavu *In Review* do stavu *To Do*

Štandardné fungovanie prehliadača

Predtým, ako je vytvorený pull request musí byť skontrolované, či nová funkcionlita neovplyvní štandardné fungovanie prehliadača. To znamená, že musia byť vyskúšané nasledovné scenáre:

- dá sa otvoriť viacero tabov bez toho, aby prehliadač začal sekať
- je možné ľubovoľne sa prepínať medzi tabmi a navštevovať na nich stránky
- taby sa dajú otvárať a zatvárať v náhodnom poradí
- po zadaní ľubovoľného textu do vyhľadávania sa zobrazia korektné výsledky
- po zadaní URL sa zobrazí požadovaná stránka
- po prepísaní URL sa zobrazí požadovaná stránka
- po stlačení tlačidla Reload sa daná stránka znova načíta