

8.10.2014

1. Stretnutie otvára Robo a oznamuje nám, že budúci týždeň bude musieť byť stretnutie v iný deň, pretože bude odcestovaný. Skúsi, či si ešte pamätá naše mená.
2. Robo chce vidieť náš plagát, Martin Svrček mu ho ukáže na notebooku a Robo je spokojný, hovorí, že môže byť a aj názov tímu Focus sa mu páči.
3. Začínajú sa riešiť organizačné veci, Robo navrhuje nech máme nejaké centrálné úložisko, kde si budeme ukladať dokumenty. Šimon odpovedá, že už sme si spravili Google drive. Robo povie, že môže byť a nech ho tam pridáme.
4. Robo sa pýta ako okrem emailu medzi sebou komunikujeme, pretože to nestačí. Vie, že máme aj Facebook-skupinu ale radšej by bol keby to bolo niečo ako instant messaging, či niečo také nepoznáme z práce? Šimon navrhuje heapchat.
5. Robo nám začína vysvetľovať jednotlivé funkcie rolí:  
Náš tím by mal mať obsadene nasledujúce role: Manažér komunikácie, Manažér dokumentácie, Vedúci tímu, Zástupca tímu, Manažér kvality: 2 ľudia môžu byť, jeden bude dozerať nato ako sa budú robiť testy – pomocou tfs-ka, druhý bude mať na starosti code-review, to znamená, že každý kód budú vidieť dvaja ľudia a až potom pôjde na schválenie k Robovi. Tento proces by sa mal zachytiť v nejakej metodike manažérom kvality.  
Manažér pre code versioning – napríklad v Gite, napísať aj postup ako to bude fungovať.  
Tieto metodiky môžeme využiť potom aj v manažmente.  
Manažér plánovania: v podstate sa plánovanie robí spoločne, najprv sa stanovia featury, tie sa rozbijú na user stories a tie na tasky. Plánovanie sa robí pomocou metódy pokru, kedy spoločne prideme nato koľko story pointov obsahuje task.  
Manažér rizík a rozsahu – bude robiť aj manažér plánovania.
6. Martin Svrček sa opýtal, kto sú naši zákazníci? Robo odpovedal, že máme možnosti:  
- možnosť, že to budú členovia fakulty - napríklad profesorka Bieliková, Michal Barla. Preto sa Robo pýta, či máme záujem o stretnutie s nimi, kde by sme sa dohodli o našich a ich predstavách, aby sme nezačali robiť niečo, čo oni nechceli.  
- plus sú to ľudia, ktorí používajú ux-lab, nejakí diplomanti a mohli by používať náš produkt  
- Robova funkcia je product owner
7. Rozdelenie rolí:  
Vedúci tímu – Petet Kiss  
Zástupca – Matej Kucek  
Manažér komunikácie – Peter Kiss  
Manažér kvality a integrácie – kvalita testovania – Peter Kušnir  
- integrácia – Matej Kucek  
- Git- Šimon Váliček  
- code review – Marek Šulek  
Manažér plánovania a rozsahu- Martin Svrček  
Manažér dokumentácie - Viktória Lovasová
8. Dohodnutie sa na stretnutí so zákazníkmi, či máme záujem:  
- súhlasíme  
- Peter Kiss napíše prvý mail
9. Robo sa pýta, či chceme ísť na TP cup?  
- dve strany treba napísať a poslať do 28.10.2014  
- prezentácia bude na iit src – tam nejaké 4 strany po anglicky napísať  
- hodnotenie bude na základe prezentácie a aj na základe výstupnej správy

- porota je z praxe a môžeme dostať spätnú väzbu
- Záverom je, že sa ešte rozhodneme.

10. Prichádza odborný praktikant Mišo a začína nám vysvetľovať projekt, na ktorý my nadväzujeme. Má tri komponenty: informačný systém, chrom-addon, firefox, dve zariadenia na sledovanie pohľadu. Server – obsahuje dve databázy: nosql ravenDB v c# písaná na framework .NET, podporuje JSONy tie sa posielajú, druhá je MSSQL: obsahuje dáta o používateľoch. Eyetracker posiela súradnice očí na server a takisto identifikuje element, ďalej aj focused application, kde sa nachádza kurzor, pracuje sa aj na tom kde sú kliky, takisto aj timestamps. Komunikácia je zabezpečená cez REST operácie. Zvonku je nedostupný gaze data service. Najvyššia úroveň:

1. Prepojenie cez api s aplikáciou ALEF

- alef bude komunikovať s našou aplikáciou, bude môcť robiť nejaké akcie: reagovať na základe pohľadu a myši a odporúčať. Toto je najdôležitejšie, lebo to je použiteľné, nahradíme ALEF dáme tam niečo iné a bude to fungovať.

2. Štatistiky a vizualizácia

3. Podpora priebehu samotného testovania

- zadať pokyny, má nižšiu prioritu ako ostatné dve.

### **Prepojenie cez api s aplikáciou ALEF**

Prebieha diskusia o features. Autentifikácia – posielanie v requeste apikey. ALEF si musí uložiť apikey, aby ho mohol poselať. Potrebne získať aké sú zadané objects of interests. Get pre aoi koľkokrát sa tam používateľ pozrel, ako dlho sa tam pozeral, nie len kde sa pozerá, momentálne, xpath tiež môže vracaať kam sa pozerá. Šimon to celé zrušil lebo to je zlé pre biznis model. Preto sa bude poselať len Get fix count, fix time -> rel, absolut. Uložiť v alefe apikey treba spraviť, na jeho strane nič nie je. V requeste posielam apikye, project\_id, session\_id, user\_id.

Dohodli sme sa na nasledovných Features:

**Features:** Ukladanie Apikey v ALEFe

Vytvorenie projektu

Vytvorenie sedenia

Pridanie používateľov k projektu

Získavanie dat

- aoi agregacie-pohľad,mys

Upozornenie/Prispôsobenie obsahu(odporúčanie)

Štatistika/Vizualizácia

Robo navrhuje aj aby sme to dotiahli až na odporúčania v ALEFe.

### **Štatistiky a vizualizácia**

**Features:**

**Poskytnutie prehľadu** – koľko tam je toho nazbieraného, project, session, user,  
- počet, čas

**Počítanie agregácií** – nejaké predrátavanie by tam malo byť

(Vysvetlenie): ak sa niekam pozeral 1 s tak mi vytvor

anotáciu(značku) viem si nastaviť aj že 5 sekúnd napríklad dá sa to nastaviť cez rozhranie.

Mám objekty A1,A2,A3,..... kam sa pozeral a koľko. Takže ak už robím agregácie tak z týchto anotácií. Takže aj pri prehľade sa dá z anotácií robiť aj agregácie. Priebežne aktualizovať anotácie, zvyšovať počet pozretí aby som mala predpripravené keď príde request. Teda mať tam nejaké countre .

**Počet fixácií**

**Zmena v čase**

**Trvanie -> relativne  
->abolutne**

**Zachytávanie obrazovky-**

(Vysvetlenie): robiť cez screenshot? Alebo video? Alebo stiahnuť stránku obraz(Mirror)

**Heatmapy**

**Graf- poradie kam používateľ šiel**

**Zoom**

**Analýza pozerania na video**

Šimon navrhuje personalizovanú reklamu – Robo hovorí, že to je super point na prezentáciu do TP cupu, nie len na výučbový systém ale aj na reklamu a bol by tam aj biznis model lebo že aj minulý rok sa sťažovali, že neboli stavané tak, že majú víziu posunúť to ďalej. Minimálne prezentovať aj keď nie realizovať.

11. Ďalej nasledovalo plánovanie úloh do najbližšieho šprintu.

Nabudúce si povieme v akom stave sú úlohy, na konci je retrospektíva, čo by sme mali zmeniť, čo bolo dobre. Potom je plánovanie nového šprintu.

Hotovo je keď:

Ten, kto implementuje hneď spraví aj testy, najprv by mal byť test a až potom implementácia, keď všetky testy prešli, kód spĺňa špecifikáciu, potom to preje k druhej osobe, ktorá bude zodpovedná za code review, dá mu feedback, ak je to ok tak to ide na kontrolu Robovi.

Ako sa presne bude robiť code review si povieme nabudúce, podobne ako s gitom to je je tam strom kde viem pridávať komentáre potom do kódu.

Peter Kiss sa pýta, či bude code review robiť stále ten istý alebo môžu byť aj viacerí?

Robo hovorí, že to môže byť jeden, ktorý bude potom implementovať menej a je dobrý v .NETE alebo budú dvojčky a jeden implementovať a druhý code review robiť alebo pri každej úlohe sa určí kto robí review.

Robo hovorí nech je úloh v šprinte menej ale aby sa to stihlo spraviť. Už po prvom týždni to má ísť ku code reviewerovi. K Robovi to môže ísť najneskôr na dokončení šprintu.

**Úlohy:**

1. ALEF- vytvoriť funkciu v rámci gaze api, ktorý si alef vypýta apikey a tá funkcia v api vráti len obyčajné ok, nejaká jednoduchá komunikácia so slovom ok, a v alefe sa zobrazí modálne okno ok.

User story: ALEF chce komunikovať cez api .

Rozdelenie úloh - tasky:

ALEF: Forknúť si git repozitár nad našou vlastnou verziou alefu

Rozbehať alef

Volanie cez api

Zobrazenie výsledkov

Testy

api: Vytvoriť api kľúč pre ALEF - cez rozhranie spraviť

Vytvoriť volanie

Support: rozbehať a nastaviť con. integration cez tfs

stránka tímu- jednoduchá, statická

- vždy tam treba dať zápisnicu a dokumenty a plán

Zápisnica – všetky veci majú byť v tfs, teda aj zápisnica a exportovať sa vždy bude zapisnica z tfs

Budúce stretnutie bude viesť Viktória Lovasová, pretože písala túto

dokumentáciu, povieme si čo sme mali spraviť.

Štúdium – ako sa pracuje s gitom, .net, prostredie (VS), Ruby on Rails, pozrieť dokumentáciu z min roku ako je zdokumentované api a databáza. Tieto úlohy spraví každý. Každý, za čo je zodpovedný začať nízko úrovňovo písať metodiku. Testovanie, integrácia a git je dôležité.

Šimon sa pýta kedy bude budúce stretnutie. Bude vo štvrtok. Presný čas nám ešte Robo dá vedieť. Dohadovanie, kto bude čo robiť:

stránka – do tfska nahodiť kto je zodpovedný, za task je zodpovedný vždy jeden, stránku robí Martin Svrček a Peter Kušnir.

Apikey spraviť pre ALEF- Šimon,

Api vytvoriť funkciu – Marek Šulek, codeReview: Peter Kiss

Nastavovanie v tfs – Matej Kucek

ALEF – Viki a Robo , code review Martin Svrček

12. Ukončenie stretnutia.