

Tím č. 27

Kill Bills
Metodika prehliadok

1 Zahájenie prehliadky kódu

Požiadavka na prehliadku kódu vzniká tým, že sa vytvorí v používanom nástroji na verziovanie kódu požiadavka na zapojenie zmien z vetvy so zmenami do development vetvy - vytvorí sa *merge request*. Tento merge request slúži na oboznámenie toho aké zmeny nastali a na komunikáciu medzi autorom a prehliadačom jednotlivých kódov, preto by mal mať primeraný názov a popis (podľa metodiky pre git). Prehliadka kódu môže začať iba vtedy ak je úloha pre ktorú má byť vykonaná v nástroji na manažment úloh v stĺpci **Ready for review**.

2 Určenie osoby, ktorá prehliadku vykoná

Človek, ktorý má prehliadku vykonať sa prideluje až po vytvorení požiadavky na takúto prehliadku. Spravidla ide o človeka, ktorý sa do danej problémovej oblasti vyzná. Avšak každý člen tímu je schopný vykonať prehliadku kódu. Na začiatku prehliadky kódu treba presunúť relevantnú úlohu do stĺpca **In review** aby boli členovia tímu informovaný, že danú úlohu už niekto prezerá - tým sa určuje kto prehliadku vykonáva.

3 Prehliadka kódu

Pri prehliadke ide o overenie, či dané zmeny sú relevantné pre danú úlohu a teda prehliadač kódu si všíma nasledujúce veci:

- Dodržanie konvencií daného programovacieho jazyka
- Dodržanie predpísaných pravidiel z metodík pre konkrétne jazyky
- Či kód neobsahuje nejaké logické chyby, alebo vedľajšie efekty, ktoré by boli nechcené
- Dostatočné a relevantné pokrytie testami
- Efektívnosť kódu - či sa zbytočne neopakujú veci, či je jednoducho pochopiteľné čo sa deje, či sú vhodne nazvané premenné, či sú komentáre na horšie pochopiteľných miestach
- Korektnosť kódu - či daná funkcionálna je naozaj funkčná a či zbehnú všetky testy

Pri prehliadke kódu je často potreba komentovať určité zmeny, ktoré boli vykonané a tieto komentáre musia tiež spĺňať niektoré podmienky:

- Komentáre musia byť výhradne v nástroji na verziovanie kódu
- Komentáre musia byť k veci - aby každý pochopil čo sa rieši
- Komentáre by mali obsahovať aj nápad na zmenu - obmedziť sa na komentáre typu "to je zle"
- Komentáre musia byť dostatočne výstižné a opisné aby každý pochopil o čo sa jedná

Komentáre majú iba dva stavy - riešený a uzatvorený. Do stavu uzatvorený môže previesť komentár iba prehliadkar zmien. Ak príde k nezhode medzi autorom zmien a ich prehliadkarom tak sa inicializuje konverzácia aj so zvyšnými členmi tímu prostredníctvom nástroja Slack.

4 Zapracovanie pripomienok

Návrhy na ďalšie zmeny, ktoré sú relevantné a schvália sa, musí autor zmien zapracovať pridaním ďalšieho *commitu*. Následne musia zmeny prejsť celým procesom prehliadky kódov od začiatku, avšak bez nutnosti vytvárať nový *merge request* keďže najnovšie zmeny sa do toho pôvodného automaticky pridávajú. Ak sú vytvorené pripomienky na ktoré sa má autor pozrieť tak sa to indikuje presunutím relevantnej úlohy do stĺpca **In progress** v nástroji na manažment úloh.

5 Ukončenie prehliadky

Ukončenie prehliadky prebieha spojením vetvy so zmenami do development branche. Toto ukončenie má nasledujúce pravidlá:

- Ukončiť prehliadku kódu môže iba ten čo prehliadku vykonáva - alebo autor s tým, že sa vetvy nespoja
- Ukončiť je možné iba vtedy ak sú všetky diskusie pri zmenách uzatvorené
- Všetky testy zbehli
- Po ukončení prehliadky treba pôvodnú vetvu zmazať
- Relevantnú úlohu treba presunúť do stĺpca **Ready for deploy** v nástroji na manažment úloh

Výnimky ohľadom prehliadok

Nasledujú prípady kedy prehliadka kódu prebieha iným spôsobom:

1. Prehliadka dokumentu - postupuje sa rovnako ako pri prehliadke kódu, avšak diskusia prebieha v príslušnom nástroji na dokumentovanie (Google Docs, Sharelatex). Zároveň pri inicializovaní prehliadky je potrebné zadať odkaz na príslušný dokument do nástroja na manažment úloh. Samozrejme pri dokumentoch neexistujú testy, ktoré by mohli zbehnúť. Taktiež dokumenty musia byť gramaticky aj sémanticky korektné
2. Párové programovanie - netreba prehliadku vôbec vykonávať.