

Tímový projekt

MOB-UX

Metodika verziovania kódu a mergovania

Vedúci projektu: Ing. Eduard Kuric, PhD.
Názov tímu: MOB-UX
Členovia tímu: Bc. Tomáš Anda
Bc. Dávid Beňo
Bc. Matúš Buzássy
Bc. Martin Nagy
Bc. Patrik Pindéš
Bc. Ľuboš Štefunko
Bc. Igor Vereš
Vypracoval: Bc. Patrik Pindéš
Kontakt: team11fiitp@gmail.com
Akademický rok: 2017/2018, zimný semester

Obsah

1. Účel dokumentu	2
2. Workflow manažovania úloh ohľadom code review v ScrumDesk	2
2.1. Určenie reviewera	2
2.2. Zodpovednosť vývojára	2
2.3. Zodpovednosť code reviewera	3
2.4. Zodpovednosť Master code reviewera	3
3. Code review checklist	3
3.1. JavaScript	4
3.2. PHP	4

1. Účel dokumentu

Tento dokument sa zaoberá metodikou code review pre jazyk JavaScript a rozoberá presné postupy a zodpovednosti jednotlivých členov tohto procesu.

2. Workflow manažovania úloh ohľadom code review v ScrumDesk

Ak vývojár pracuje na User Story alebo na jeho časti (Task) a úlohu alebo jej časť dokončil, tak sa presvedčí, že z funkčného hľadiska funguje a presunie ju do stavu „To review“ v ScrumDesk. Žiadna z hotových úloh by sa nemala presúvať do stavu „To review“ neskôr, ako 1 deň pred koncom šprintu (t.j. najneskôr v stedu ráno o 8). Tento vývojár následne skontaktuje osobu, ktorá je zodpovedná za vykonanie code review . Code review sa vykoná **neformálnou** metódou „cez plece“ a to buď pri osobnom stretnutí alebo cez aplikáciu na zdieľanie pracovnej plochy (Skype, Discord). Táto udalosť by mala trvať najdlhšie 30 minút. Po dokončení code review, reviewer na daný task/user story pridá komentár do sekcie **Comments** vo forme **#review: Veci, ktoré treba upraviť**. Vývojár implementuje zmeny, a skontaktuje reviewera na následnú kontrolu. Rovnako sa zúčastnia spoločného stretnutia a po jeho skončení, ak je všetko v poriadku, reviewer pridá komentár **#review: Ok**. Toto druhé stretnutie samozrejme nie je potrebné, pokiaľ všetko bolo v poriadku hneď na začiatku, v tom prípade platí **#review: Ok**. Následne vývojár presunie svoju úlohu do stavu Done.

Pokiaľ nie je určené inak, code review budú vykonávať code review mastery. Toto platí hlavne pre počítačové šprinty. Neskôr, keď ostatní členovia tímu nadobudnú dostatočné skúsenosti s danými technológiami, budú môcť vykonávať code review aj oni.

2.1. Určenie reviewera

Špecifické určenie reviewera na task/story sa môže dohodnúť dopredu na začiatku šprintu po plánovaní pri pridovaní taskov/stories. Toto sa vykoná pridaním **#reviewer:@Meno_reviewera** do komentáru.

2.2. Zodpovednosť vývojára

- Úlohu posúva na review iba v prípade, že z funkčného hľadiska je v poriadku, čiže spĺňa akceptačné kritériá a na čas (nie neskôr ako 24 hod pre koncom sprintu)
- Presúvanie úlohy/story do stavu „To review“ a „Done“
- Skontaktovanie osoby zodpovednej za review
- Implementovanie zmien na základe poznámok reviewera
- Pripraví sa na review

- nájde súbory, v ktorých vykonal zmeny
- Opisuje kód reviewerovi zrozumiteľne a jasne
- Nezamlčí žiadne zmeny

2.3. Zodpovednosť code reviewera

- Vykona review podľa checklist-u.
- Pridá svoje komentáre k danej úlohe, aby bolo jasné, aké zmeny musí vývojár vykonať
- Pýta sa otázky
- Musí porozumieť kódu
- Objektívne hodnotí

2.4. Zodpovednosť Master code reviewera

- Vykona code review pre danú technológiu, jazyk v ktorom je zručnejší ako ostatný
- Môže skontrolovať dodržiavanie metodiky inými code reviewermi a poučiť ich
- Môže vykonať zmeny v metodike, na ktoré musí upozorniť všetkých členov tímu.
- Pokiaľ nie je určené inak, vykona code review

3. Code review checklist

Code review sa vykona neformálnou formou, review „cez plece“.

1. Skontroluj funkcionálnosť (pre istotu)
2. Skontroluj architektúru riešenia
 - a. Zodpovedá frameworku alebo dohode?
 - b. Dodržala sa hierarchia súborov?
3. Skontroluj výskyt zložitého kódu (prehľadnosť kódu)
 - a. Nedá sa zjednodušiť
 - b. Ak sa nedá zjednodušiť, sú tam komentáre na udržanie prehľadnosti?
 - c. Dá sa niečo prepoužiť?
 - d. Nie sú tam opakujúce sa časti kódu?
4. Skontroluj scope premenných
5. Je kód pokrytý Unit Testmi? (Ak majú zmysel)
6. Skontroluj dodržiavanie code style guide
 - a. Zmysluplné názvoslovie premenných a hlavne funkcií
 - b. Odsadzovanie (pokiaľ nie je automatizované)
 - c. Ostatné praktiky code style guide
7. Skontroluj závislosť externých knižníc

3.1. JavaScript

Pre vývoj v jazyku JavaScript využívame framework **Knockout JS**. Pri code review skontroluj, či kód zodpovedá pravidlám nášho dokumentu **Knockout Style Guide**.

3.2. PHP

Pre vývoj v jazyku PHP využívame framework **CakePHP3** . Pri code review skontroluj, či kód zodpovedá pravidlám nášho dokumentu **CakePHP3 Style Guide**.