

# Metodika pre code review

Verzia 11.11.2015

*Tabuľka 1. Autori*

Autor	Rola
Pavol Beťák	

Tabuľka 2. História zmien

Verzia	Dátum	Autor	Popis
1.0	11.11.2015	Pavol Beťák	Vytvorenie dokumentu

## Obsah

1	Úvod .....	1
1.1	Roly vývojárov .....	1
2	Pravidlá code review .....	2
2.1	Základné pravidlá .....	2
	Posudzovateľ .....	2
	Časová následnosť .....	2
2.2	Sumarizácia pravidiel .....	2
3	Priebeh code review .....	3
3.1	Postup vykonávania code review .....	3

# 1 Úvod

Cieľom tejto metodiky je zosumarizovanie pravidiel a vytvorenie postupu pre vykonávanie code review. Vzhľadom na to, že kód aj po odtestovaní môže byť problematický, alebo je ho možné ešte upraviť do podoby, ktorá by viacej vyhovovala podmienkam projektu, je potrebné vykonávať kontrolovanie kódu.

## 1.1 Roly vývojárov

Na vytváraní zdrojového kódu sa podieľajú vývojári, ktorých úlohy môžeme rozdeliť do dvoch základných rolí a to:

- Autor kódu – je osoba, ktorá vytvorila daný zdrojový kód a tým sa aktívne podieľa na vývoji systému.
- Posudzovateľ – je osoba, ktorá zodpovedá za kontrolovanie zdrojového kódu, ktorý vytvoril „autor kódu“. Kontrolovanie môže prebiehať rôznymi formami, najčastejšie je to formou poznámok, ktoré sú následne predložené autorovi kódu.

## 2 Pravidlá code review

Vzhľadom na to, že existuje veľké množstvo spôsobov a pravidiel na vykonávanie code review, je potrebné si stanoviť pravidlá, ktoré zachovajú konzistentnosť a tým zvýšia efektívnosť. Je dôležité aby pravidlá boli jednoduché a zrozumiteľné.

### 2.1 Základné pravidlá

#### Posudzovateľ

Posudzovateľ je zodpovedný za kontrolovanie kódu a informovanie o chybách autora kódu. Posudzovateľ by mal byť osoba, ktorá sa doteraz aktívne nepodieľa na vytváraní kódu. Je to z toho dôvodu, aby nebola ovplyvnená doterajším vývojom a tým mohla priniesť do problematiky vlastný názor. Code review by taktiež nemala vykonávať osoba, ktorá nemá o danej problematike žiadne vedomosti.

#### Časová následnosť

Posudzovať kontroluje a hodnotí kód až keď je vytvorený a otestovaný. Je to z toho dôvodu, aby posudzovateľ nestrácal zbytočne čas z kontrolovaním kódu, ktorý nefunguje. Úlohou posudzovateľa nie je opraviť chyby kódu, ale ich detekovať, prípadne navrhnúť riešenie.

### 2.2 Sumarizácia pravidiel

- Posudzovateľ sa aktívne nepodieľa na vytváraní kódu
- Posudzovateľ posudzuje kód až po jeho vytvorení a otestovaní
- Posudzovateľ neopravuje chyby, ale ich detekuje
- Posudzovateľ by mal mať minimálne vedomosti o danej problematike

## 3 Priebeh code review

Na to aby code review mohlo úspešne prebiehať a aby si zachovala konzistentnosť je potrebné si zdefinovať základný postup jeho vykonávania.

### 3.1 Postup vykonávania code review.

#### 1. Vytvorenie task-u

Pri vytváraní task-ov, osoba, ktorá zodpovedá za vyriešenie danej úlohy vytvorí s názvom codereview, ktorý následne priradí inému členovi tímu. Týmto označí, ktorý člen bude vykonávať rolu posudzovateľa.

#### 2. Nastavenie statusu

Po vytvorení zdrojového kódu a následnom otestovaní osoba zodpovedná za vyriešenie úlohy zmení status úlohy z „active“ na „resolved“ a tým oznámi, že úloha je hotová a pripravená na kontrolovanie.

#### 3. Vytvorenie pull requestu

Po označení úlohy za resolved, autor kódu push-ne do repozitára a vytvorí pull request, pričom nastaví hodnotu „solver“ na meno osoby, ktorá bude mať na starosti code review.

#### 4. Získanie riešenej úlohy

Posudzovateľ si stiahne aktuálnu verziu riešenia úlohy nad ktorou bude vykonávať posudzovanie.

#### 5. Preverenie kvality

Posudzovateľ vyhodnotí kvality riešenia, pričom sa zameria na splnenie základných úloh, kvalitu kódu a pod..

#### 6. Vyhodnotenie code review

- V prípade, že posudzovateľovi sa zdá byť riešenie úlohy korektné pokračuj krokom 10.
- V prípade, že používateľ má výhrady k riešeniu úlohy pokračuj krokom 7.

#### 7. Poskytnutie spätnej väzby

Posudzovateľ pridá komentáre na miesta, ku ktorým má pripomienky, prípadne navrhne riešenia daných problémov. Je možné problémy konzultovať aj inou formou avšak pre transparentnosť sa odporúča táto forma poskytnutia spätnej väzby alebo využitie tímového emailu, prípade konzultácia na stretnutí.



### **8. Opätovné nastavenie statusu**

Autor kódu zmení status s „resolved“ na „active“.

### **9. Úprava kódu**

Autor kódu upraví zdrojový kód aby vyhovoval podmienkam posudzovateľa. A pokračuje bodom 2.

### **10. Uzavretie pull requestu**

Posudzovateľ zmení hodnotu pull requestu na „resolved“.

### **11. Merge do projektu**

Vytvorené a skontrolované riešenie sa mergne do projektu.