

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Metodika revízie kódu

Tím sixPack

Bc. Jozef Blažíček

Bc. Ján Ďurica

Bc. Jakub Chalachán

Bc. Matúš Ivanoc

Bc. Maryna Kovalenko

Bc. Miloš Štefčák

Vedúci projektu: Ing. Ivan Kapustík

Predmet: Tímový projekt I

Ročník: 2016/2017

Obsah

1. Základné informácie.....	1
2. Priebeh revízie kódu.....	1

Táto metodika popisuje postup pri procese revízie kódu (code review) tímu **sixPack** počas práce na tímovom projekte v akad. roku 2016/2017. Revízia kódu spočíva na princípe komentovania kódu za účelom maximalizovania kvality, funkčnosti, prehľadnosti a zrozumiteľnosti kódu.

1. Základné informácie

Jeden z najdôležitejších cieľov každého vývojára je neustále zlepšovanie kvality svojej práce. Keďže každý vývojár ma svoje spôsoby písania kódu, rozhodli sme sa zaviesť revízie kódu aj do nášho projektu. Vzhľadom na to, že kód projektu je umiestnený na repozitári služby Bitbucket a po zisteniach od minuloročných projektov, sme sa rozhodli pokračovať v používaní rozhrania tohto prostredia.

Na základe skúsenosti jednotlivých členov tímu sme vytvorili dvojice, ktoré túto revíziu budú vykonávať:

- *Bc. Jakub Chalachán & Bc. Maryna Kovalenko*
- *Bc. Ján Ďurica & Bc. Miloš Štefčák*
- *Bc. Matúš Ivanoc & Bc. Jozef Blažíček*

Z toho vyplýva, že v každej dvojici bude aj riešiteľ, aj posudzovateľ.

2. Priebeh revízie kódu

Kontrola kódu bude prebiehať podľa nasledujúcich krokov:

1. Riešiteľ odovzdá kód s riešením úlohy do príslušnej vetvy repozitára.
2. Riešiteľ vytvorí „*pull request*“ prostredníctvom rozhrania nástroja Bitbucket, do poľa „*Reviewers*“ vloží meno svojej dvojice a do časti „*Description*“ napíše krátky opis riešenia z vyzdvihnutím nejakých hlavných častí + názov úlohy z JIRY.
3. Po vytvorení „*pull requestu*“ riešiteľ v JIRE pridá komentár s informáciou, že kód bol odoslaný na revíziu a zmení stav úlohy na „*resolved*“.
4. Posudzovateľ si stiahne verziu kódu s vyriešenou úlohou a preverí kvalitu a funkčnosť kódu. Ak boli vytvorené jednotkové testy, spustí ich a vyhodnotí.
5. V prípade, že je všetko v poriadku a kvalita a funkčnosť vyhovuje, posudzovateľ potvrdí úlohu stlačením „*Accept*“, napíše komentár do JIRY, že je všetko OK.
6. Po kontrole riešiteľ potvrdí spojenie vetiev výberom voľby „*Merge*“ a zmení stav úlohy na „*Closed*“.
7. V prípade nedostatkov posudzovateľ vyjadrí svoje pripomienky v komentároch a „*pull request*“ zamietne voľbou „*Decline*“.
8. Po zamietnutí „*pull requestu*“, posudzovateľ napíše komentár do JIRY s dôvodom zamietnutia a zmení stav úlohy na „*Reopened*“.
9. Riešiteľ následne opraví nedostatky a celý postup sa opakuje.

V prípade, že riešiteľ nemá svojho posudzovateľa z nejakého vážneho dôvodu ako napríklad ochorenie, posudzovateľ sa bude vyberať podľa nasledujúceho poradia:

Bc. Ján Ďurica
Bc. Jakub Chalachán
Bc. Matúš Ivanoc
Bc. Jozef Blažíček
Bc. Miloš Štefčák
Bc. Maryna Kovalenko

Poznámka: V takomto prípade nech sa riešiteľ dopredu dohodne s náhradným posudzovateľom, aby nevznikol zbytočný problém alebo nedorozumenie.

The image shows a GitHub interface for creating a pull request. At the top, there are two boxes representing the local and remote branches. The left box shows the local branch 'master' with a commit message 'Created 2013-10-09, updated 5 days ago'. An arrow points to the right box, which shows the remote branch 'TP_2015_TF_refactor'. Below this, the pull request form is shown. The title is 'Merged in TP_2015_TF_refactor (pull request #19)'. The description is 'Tp 2015 tf refactor'. There is a 'Reviewers' field and a 'Close branch' section with a checkbox for 'Close master after the pull request is merged'. A blue button labeled 'Create pull request' is at the bottom.

Obrázok 1: Ukážka obrazovky pre vytvorenie "pull requestu"

Vypracoval: Miloš Štefčák