LEGENDRONE

Metodika pre code review a verziovanie

Tím:	#1, LEGENDRONE
Vedúci tímu:	Ing. Viktor Šulák
Členovia tímu:	Tomáš Čičman, Dušan Drábik, Dušan Hatváni, Patrik Kadlečík, Mário Keszeli, Martin Mocko, Lukáš Visokai
Akademický rok:	2016/2017
Autor:	Martin Mocko
Verzia číslo:	2
Dátum poslednej zmeny:	26.11.2016

STU FIIT

LEGENDRONE

Obsah

STU FIIT

LEGENDRONE

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.

Code review je pre fungovanie tímu veľmi dôležité, pretože prináša do tímu väčšiu istotu ohľadom implemenácie nejakej úlohy alebo príslušného riešenia. Je dôležité nielen z pohľadu kódera, ktorý kód vytvoril, aby dostal spätnú väzbu k jeho kódu, ale slúži taktiež na to, aby sme mali minimálne jedného ďalšieho člena tímu, ktorý je oboznámený s príslušným riešením danej úlohy. Takýmto spôsobom sa zdieľajú poznatky a pokiaľ by prišlo k výpadku člena tímu, nie je to pre tím kritické.

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 podieľala 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

1.2. Slovník pojmov

Pojem	Význam
Branch (vetva)	Verzia/strom objektov, ktoré máme v záujme sledovať. Je robená na to aby sa na viacerých objektoch (súboroch) dalo pracovať paralelne.
Push	Skopírovať informácie z lokálnej vetvy do vzdialenej (od nás do nejakej inej).
Pull	Natiahnutie informácií zo vzdialenej vetvy do lokálnej vetvy (z nejakej inej ku nám).
Pull request	Slúži na formalizovanie (overenie) dokončenej úlohy. Úloha musí prejsť reviewom kódu a je potrebné vyriešiť konflikty.
Code review	Prehliadka kódu, ktorá určí, či je kód pripravený na mergnutie do rodičovskej vetvy.
Commit	Zachytenie nejakého "stavu" 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 efektivitu. Je dôležité aby pravidlá boli jednoduché a zrozumiteľné.

2.1. Pravidlá

Posudzovateľ

Posudzovateľ je zodpovedný za kontrolovanie kódu a informovanie o chybách autora kódu. Posudzovateľ by mal byť osoba, ktorá sa doteraz aktívne nepodieľala 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 odtestovaný. Je to z toho dôvodu, aby posudzovateľ nestrácal zbytočne čas kontrolovaním kódu, ktorý nefunguje. Úlohou posudzovateľa nie je opraviť chyby kódu, ale ich detekovať, prípadne navrhnúť riešenie.

2.2. Sumarizácia

- Posudzovateľ sa aktívne nepodieľa na vytváraní kódu
- Posudzovateľ posudzuje kód až po jeho vytvorení a odtestovaní
- Posudzovateľ neopravuje chyby, ale ich detekuje (výnimka v prípade, že identifikované chyby sú triviálne a oprava nezaberie viac ako 5 minút)
- Posudzovateľ by mal mať minimálne vedomosti o danej problematike

3.Ako požiadať o code review

Postup notifikácie člena tímu o žiadosti o code review. Nasledujúce kroky robí autor kódu:

1. Keď je úloha hotová, pushnúť commity na server cez Sync – Push outgoing commits
2. Po tom, ako splnil úlohu si autor kódu musí pullnúť najnovšiu verziu „master“ vetvy do svojej lokálnej „master“ vetvy
3. Túto lokálnu „master“ vetvu autor kódu mergne do svojej branche, na ktorej pracoval
4. Ak je to potrebné, autor kódu vyrieši merge konflikty
5. Po úspešnom vyriešení merge konfliktov môže autor kódu vytvoriť nový pull request cez Pull requests – New pull request (pre lepšiu predstavu pozrieť kapitolu 5.2 Vytvorenie pull requestu)
6. Označiť člena(ov) tímu, ktorý bude zodpovedný za review pull requestu
7. Potvrdiť požiadanie o pull request (tým pádom žiadame aj o code review)

4. Od vytvorenia tasku k zlúčeniu (merge) do projektu

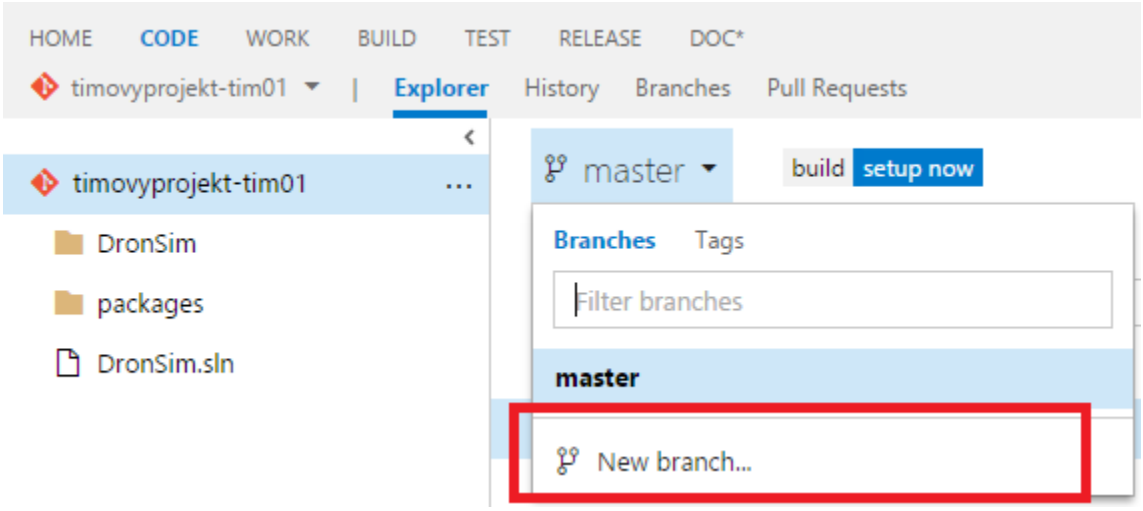
Pre efektívne fungovanie tímu musí taktiež byť v tíme jasne zadefinovaný postup od vytvorenia tasku až po mergnutie zdrojového kódu do projektu.

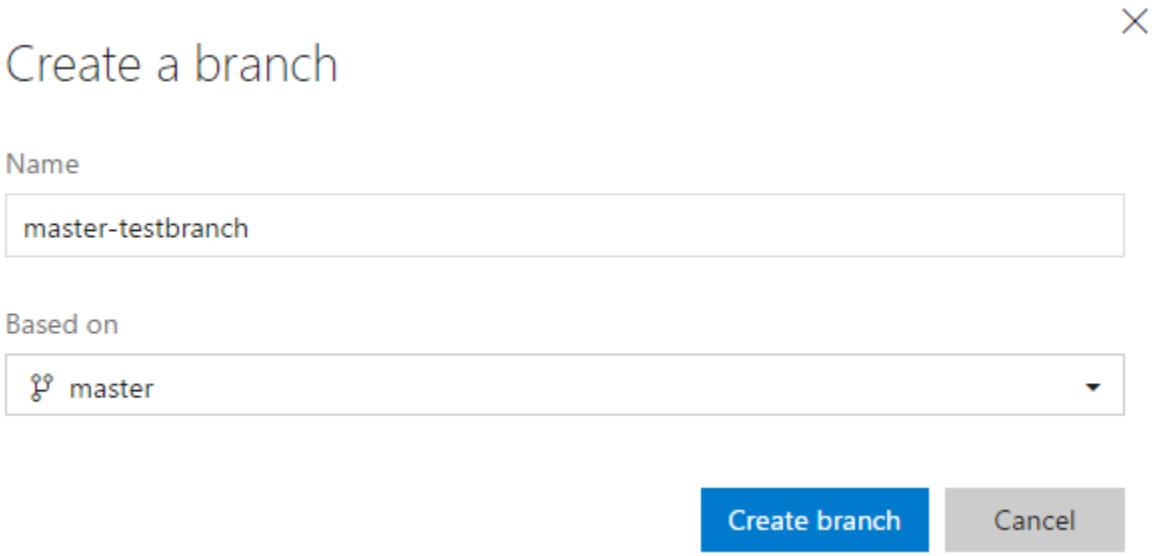
Tento postup bude vyzerať nasledovne:

1. Vytvorí sa úloha, priradí sa konkrétnemu členovi tímu
2. Autor kódu začne na tejto úlohe v systéme TFS pracovať – nastaví úlohu do stavu „In Progress“
3. Autor kódu si vytvorí novú branch z branche “master” a nazve ju spôsobom “<číslo_úlohy>-<názov_úlohy>”. Ak autor kódu implementuje viacej úloh, vytvorí branch spôsobom “<číslo_úlohy_1>-<číslo_úlohy_2>-..<číslo_úlohy_n>-<názov_úlohy>”. Čiže názov pre branch pre dve úlohy by vyzeral napr. **“5132-5135-Implementacia_modelu_UAV”.**
4. Autor kódu pracuje na úlohe, keď ju naimplementuje, môže spraviť commit, resp. check-in
5. Commit uložiť aj s komentárom, ktorý stručne vysvetlí spravené zmeny v kóde
6. Pokračuje sa pravidlami popísanými v 3.kapitole Ako požiadať o code review
7. Posudzovateľ si v nástroji TFS v sekcii “My Work” nájde nový review request
8. Posudzovateľ začne posudzovať review daných commitov, pokiaľ je kód v poriadku, akceptuje zmeny a pokračuje sa krokom 12, pokiaľ nie je kód v poriadku, pokračuje sa krokom 8
    a. Posudzovateľ sa pri posudzovaní sústredí na to, či daná implementácia spĺňa požiadavky úlohy, taktiež či je kvalita kódu dostatočná, či je kód riadne zdokumentovaný – viď metodiku dokumentácie
9. Kód nebol v poriadku, takže je potrebné poskytnúť dodatočnú spätnú väzbu autorovi kódu. Táto spätná väzba môže byť poskytnutá nasledujúcimi spôsobmi:
    a. Do komentára k danému code review requestu, ktorý posudzovateľ dostal
    b. Komentárom do autorovho kódu k častiami kódu, ktoré sú problematické
    c. Poskytnutím poznámok ku kódu iným spôsobom (napr. Napísať do Slacku)
10. Posudzovateľ vráti autorovi úlohu na prerobenie
11. Autor kódu upraví kód podľa pripomienok posudzovateľa. Pokračuje sa bodom 8.
12. Autor kódu, prípadne iný zainteresovaný človek vyrieši prípadné konflikty, ktoré mohli nastať pri pull requeste
13. Posudzovateľ akceptuje pull request, implementácia úlohy spĺňa požiadavky
14. Vytvorené a skontrolované riešenie sa mergne do branche “master”

5.Práca s branchami v TFS

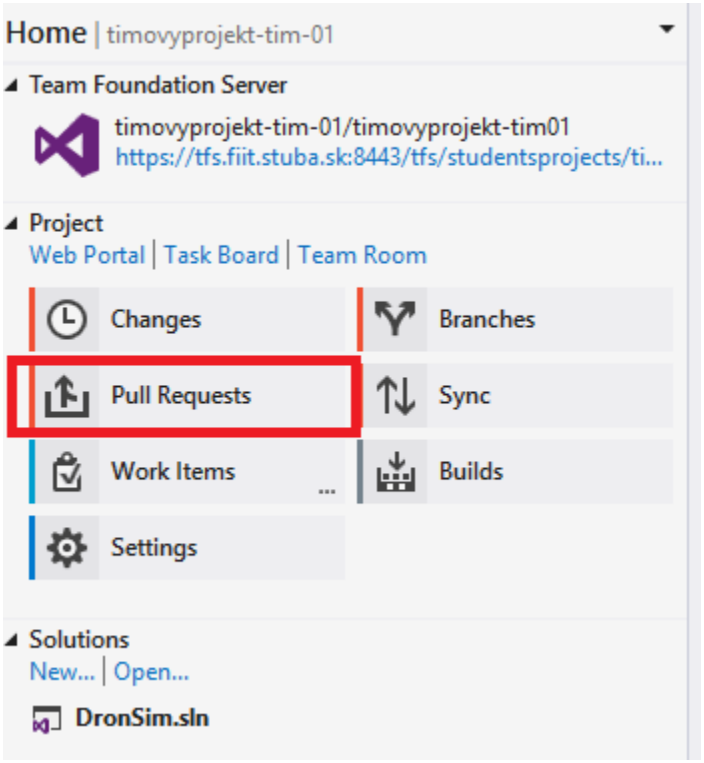
5.1. Vytvorenie novej branche

1. Ísť do CODE → Explorer → Zvoliť git repozitár projektu (napr. timovyprojekt-tim01) → Kliknúť na názov branche v ktorej sa aktuálne nachádzame (napr. „master“) → Kliknúť na New branch
2. 

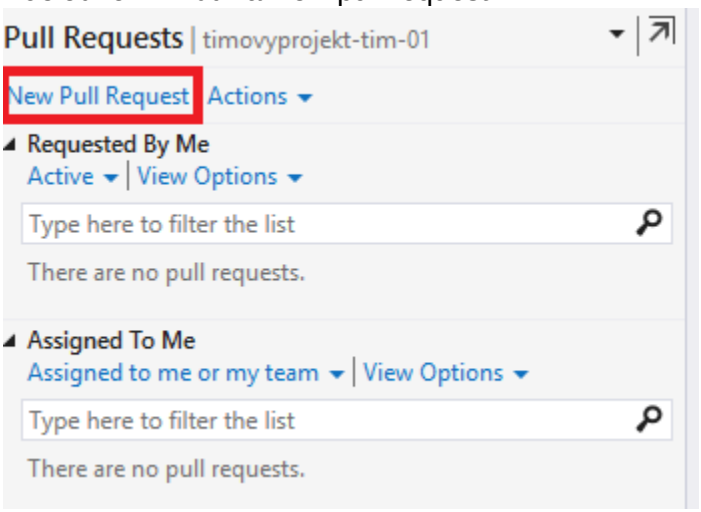
3. Vyplniť kolonku “Name” pre meno branche, zvoliť, z ktorej branche sa má nová brancha vytvoriť
4. 

5.2. Vytvorenie pull requestu

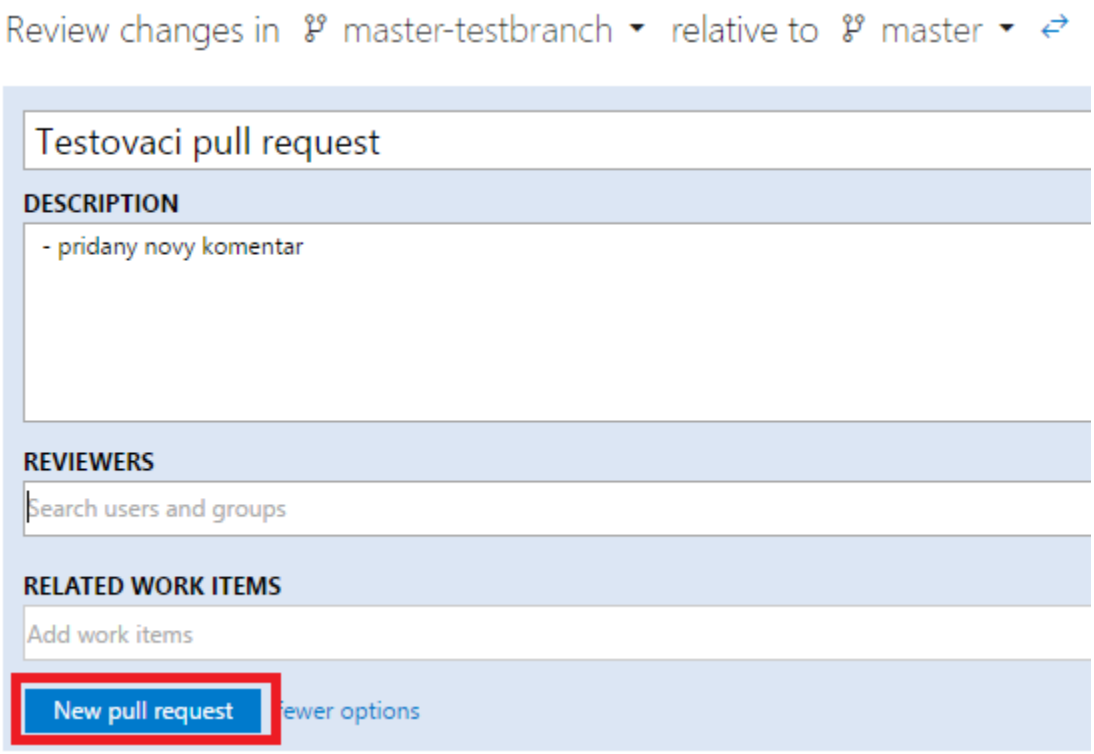
1. Nachádzame sa v Home, treba kliknúť na Pull requests



2.
3. Následne kliknúť na New pull request



4.
5. Otvorí sa webstránka, treba zadať názov pull requestu, kliknúť na more options, vyplniť description, pridať Reviewerov, pokiaľ niesú pridané súvisiace Work itemy, pridať aj tie, následne kliknúť na New pull request



6.
7. Pull request je týmto vytvorený a pripravený ku code review