Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2 , 842 16 Bratislava

Extrakcia dát z webu

[WebExtraction]

Metodika pre manažment verzií

Tím: č. 16, WebX
Vedúci tímu: Ing. Ivan Srba, PhD.
Členovia tímu: Ján Brechtl, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý, Michal Kren, Martin Lacek, Andrej Vaculčiak
Akademický rok: 2016/2017
Autor: Michal Kren
Verzia číslo: 1.4
Dátum poslednej zmeny: 4.4.2017

1 Úvod

Cieľom tejto metodiky je definovať základné postupy odovzdávania kódu a udržiavania verzií. Metodika opisuje prácu s nástrojom Git.

Touto metodikou sa riadi tím č. 16 WebX v rámci predmetu tímový projekt v akademickom roku 2016/2017. Metodikou sa riadia všetci členovia tímu.

2 Skratky a značky

Odovzdanie	(angl. commit) je množina zmien v súboroch projektu v určitom čase podpísaná autorom
Vetva	(angl. branch) je postupnosť odovzdaní. Vetvenie umožňuje spravovať súbory v jednej vetvy repozitáru nezávisle od iných vetiev
Feature	Funkcionalita, ktorá je podporená používateľským príbehom
Git	Distribuovaný systém na správu revízií zdrojového kódu

3 Postupy

3.1 Vetvy

- master - produkčné prostredie (spustenie v letnom semestri)
- staging
- development - spoločná vetva pre vývoj
- ďalšie
    - názov: feature_<nova funkcionalita>, fix_<chyba>
    - napr: feature_registration

Novy kód sa odovzdáva len cez *pull request* (ďalej PR) z vlastnej vetvy. Iný člen tímu mergne PR do development vetvy po tom ako prebehne *code-review*.

Do development vetvy sa smie priamo *pushnut* len v prípade opravy jedného riadku - musí schváliť aspoň jeden ďalší člen tímu.

Prísny zákaz *force push* do inej ako vlastnej vetvy - v prípade porušenia kupuje vinník pivo celému tímu.

Postup nasadzovania zmien

feature_branch --- PR ---> development --- CI ---> staging (neskôr master)

Odovzdávanie kódu

- commit message
    - v angličtine
    - začína sa slovesom s veľkým začiatočným písmenom
    - jedna správa obsahuje najviac dve slovesá oddelené čiarkou
    - napr.: Fix email confirmation, add error messages to login form
- sprava nekončí bodkou
- každý PR sa mergne až po code-review

3.2 Pomôcka k základným príkazom

Vytvorenie vlastnej vetvy:

(aktuálna vetva je development)
git pull
git checkout -b feature_brach

Commitnutie zmien

git commit -am “Add new feature”

Zobrazenie zmien oproti poslednému commitu

git diff

Prvý push vlastnej vetvy

git push -u origin feature_branch

Udržiavanie aktuálneho kódu z development vetvy vo vlastnej vetve

1. **merge**

   git checkout development
   git pull origin development
   git checkout feature_branch
   git merge development

   (pozn.) git merge --squash

   - umožní vyhnúť sa zbytočným merge commitom
   - merge commit sa pripojí k ďalšiemu “obyčajnému” commitu
   -

2. **rebase**

   git checkout feature_branch
   git rebase development / git rebase -i development (interactive)

   podľa pokynov vyriešiť konflikty a následne
   git rebase –continue

   1. proces možno treba opakovať niekoľko krát, podľa počtu konfliktov

   git rebase --abort

   2. zruší zmeny spôsobené rebasom a vráti sa na pôvodny stav

Vrátenie sa na predošlý commit (opatrne)

git reset --hard <označenie commitu>

- označenie commitu získame cez git log
- stratia sa všetky zmeny od commitu, na ktorý sa vrátime