

Extrakcia dát z webu

[WebExtraction]

Metodika pre vývoj

Tím: č. 16, WebX
Vedúci tímu: Ing. Ivan Srba, PhD.
Členovia tímu: Ján BrechtI, Tomáš Juhaniak, Martin Kalužník, Rastislav Krchňavý,
Michal Kren, Martin Lacek, Andrej Vaculčiak
Akademický rok: 2016/2017
Autor: Michal Kren
Verzia číslo: 1.0
Dátum poslednej zmeny: 14.10.2016

1 Úvod

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

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	Distribučný systém na správu revízií zdrojového kódu

3 Postupy

Na vývoj sa využívajú tri základné vetvy:

- master - produkčné prostredie (spustenie v letnom semestri)
- staging - testovacie prostredie na virtuálnom serveri
- development - spoločná vetva pre vývoj

Okrem toho sa na vývoj novej funkcionality používajú tzv. „feature vetvy“. Jednej takejto vetve spravidla zodpovedá jeden používateľský príbeh (angl. user story). Názvy vetiev spĺňajú tvar *feature_<nova funkcionality>* alebo *fix_<chyba>* napr. *feature_registration*.

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

Do development vetvy sa smie priamo *pushnúť* 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

- odovzdávacia správa (angl. 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
 - správa nekončí bodkou
- každý PR sa merge až po code-review

Pomôcka k základným príkazom

Vytvorenie vlastnej vetvy:

- aktuálna vetva je development

```
git pull
```

```
git checkout -b feature_brach
```

Odovzdanie 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:

```
git checkout development
```

```
git pull origin development
```

```
git checkout feature_branch
```

```
git merge development
```

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 vrátíme