

Metodika riadenia verzii

Last updated by | Bc. Viera Maronova | Thu, 13 Dec 2018 13:53:15 GMT

Git repozitár pozostáva z 3 typov vetiev:

- **master vetva** - produkčná verzia, na 100% funkčná
- **develop vetva** - funkčná verzia najnovšie pridávanej funkcionality, práve testovaná, neskôr bude spojená s master vetvou
- **ostatné vetvy** - obsahujúce rôznu novú funkcionality, na ktorej sa pracuje v rámci šprintu (šprintov), bugfixy, samostatné user stories...

Workflow

1. Zosynchronizovanie všetkých vetiev s repozitárom pomocou príkazu
`git fetch --prune`
2. Vytvorenie novej vetvy z develop vetvy. Postupne zmeny v rámci vetvy.
3. Keď je úloha hotová, vytvorí sa merge request podľa *code review* metodiky.
4. Po prehliadke kódu, zapracovaní pripomienok a následnom schválení, autor merge requestu vykoná spojenie vetvy s develop vetvou.

Pomenovanie vetiev

Pre riešenie novej funkcionality programu je potrebné vytvoriť vetvu podľa nasledovných pravidiel

- Názov vetvy nech obsahuje id a krátky popis user story
- Celý názov branche je napísaný v angličtine a malým písmom
- Oddeľovanie slov pomocou pomlčky (znak -)

Príklad: `feature/us-<id-user-story>-<popis-user-story>` alebo `hotfix/us-<id-user-story>-<popis-user-story>`.

Príklad spolu s príkazom na vytvorenie novej vetvy: `git checkout -b feature/us-8806-created-repository`

Commit správy

- Správy prisluchajúce commit-om sú písané v anglickom jazyku.
- Slovesá sú v správach uvádzané v tvare imperatívu.
- Správa by sa mala začínať s identifikátorom tasku z TFS v hranatých zátvorkách.

Príklad spolu s príkazom: `git commit -m "[US-8806] Update introduction to Unity and LUA"`.