

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tím č.11

Metodika verziovania

Vedúci tímu: Tibor Vincze, MSc.
Vypracoval/a: Bc. Filip Agh

1 Úvod

Tento dokument definuje pravidlá a normy, ktoré určujú prácu tímu 11 z roku 2021 na predmete "Tímový projekt" s repositármi.

Zadefinujeme si dva hlavné typy repositárov a to:

- repositár web stránky projektu
- repositáre implementácie nášho projektu

Zadefinujeme si roly účastníkov počas interakcie s repositárom, ako aj formát textu pre commit a pull request.

V poslednej časti si predstavíme odporúčania a návody na konkrétne časti práce s repositárom.

2 Repozitár

V tejto časti si opíšeme teoretické a praktické informácie o repositároch.

2.1 Technické riešenie repositárov

- Version control systém pre náš projekt: [GIT](#)
- Manažér pre version control system: [Azure DevOps](#)

2.2 Typ repositáru

Repozitár pre web stránku projektu, s názvom: Tim11

Tento repositár bude mať výnimku z nutnosti organizácie vetvenia repositára, keďže do daného repositára budú pribúdať dokumenty a linky a teda žiadne veľké zmeny. Za repositár je zodpovedný (git master) Patrik Vdoviak.

Repozitáre, v ktorých sú implementácie nášho riešenia

Názov repositára má opisovať projekt, ktorý zastrešuje. Pre repositár platia všetky princípy a pravidlá, ktoré si opíšeme v ďalšej časti dokumentu.

3 Vetvenie

Git funguje na základe vetvenia, a teda si definujeme nasledujúce vetvy.

3.1 master

Vetva master reprezentuje produkčnú vetvu a obsahuje commity s tagom pre jednotlivé publikované verzie. Do mastera sa pushuje len hotfix tasky alebo pri vydaní novej major/minor verzie sa master posunie na aktuálny dev, patch verzie, čiže hotfixy sa aplikujú následne aj na dev vetvu pomocou rebase nad masterom.

3.2 dev

Vetva dev je aktuálna dev verzia repositára, do ktorého sa pushujú nové featury, testy, alebo sa robí refaktor.

3.3 task

Keď začne developer riešiť task, musí si spraviť vlastnú vetvu. V prípade hotfixu je base master, inak dev vetva. Názov vetvy musí byť podľa formátu: číslo tasku[_názov tasku] ()

4 Roly

V tejto časti si opíšeme, aké roly existujú v procese práce s repositárom.

4.1 Git master

Git master je jedna osoba, ktorá je zodpovedná za master a dev vetvu repozitára, a teda mergeje do týchto vetiev, je zodpovedná za riešenie konfliktov medzi vetvou master a dev, dohliada na dodržiavanie metodiky vo svojom repozitári.

4.2 Autor pull requestu

Autor pull requestu je člen tímu, ktorý dokončil svoj task a chce ho mergenúť do príslušnej hlavnej vetvy.

4.3 Reviewer

Je osoba, ktorá má skontrolovať zmyslupnosť a správnosť zmien v commitoch. Reviewer je iná osoba ako autor pull requestu.

5 Udalosti v procesoch verziovania

5.1 Vydanie verzie

Vydanie verzie je proces, kedy git master určí konkrétny commit v masteri a pridá mu tag s verziou {major}.{minor}.{patch}, do prvého oficiálneho vydania je major verzia "0". Major verzia sa mení, ak sa posúva master na dev a nastávajú zmeny, ktoré nie sú kompatibilné s predošlou minor verziou. Ak zmeny sú kompatibilné, tak sa mení minor verzia. Patch sa mení v prípade pridania nových hotfixov.

5.2 Commit

Developer, ktorý pracuje na tasku, ktorý má zmeny v repozitári, pravidelne a granularne commituje do svojej task vetvy.

Formát commitu:

```
Task #{id tasku} {feature, fix, test, refactor, format}:{description}
```

[krátky opis zmien]

(max 72 znakov na 1. riadok)

5.3 Pull request

Developer, ktorý dokončil a úspešne otestoval prácu na tasku, ktorý má zmeny v repozitári, si vytvorí pull request na svoju base vetvu. Ak sa vyskytnú konflikty, musí ich vyriešiť, prípadne sa poradí s git masterom daného repozitára. Po vytvorení pull requestu sa uistí, že je zvolený aspoň 1 reviewer. V prípade vyžiadaných zmien autor pull requestu musí urobiť úpravy a znova vyžiadať review.

Formát pull requestu:

```
#{id tasku}
```

[krátky opis zmien v pull requeste]

5.4 Merge pull requestu

Git master má za úlohu finálne zvalidovať každý pull request a keď je schválený osobou, ktorá robila review, tak ak nie sú konflikty, merge pull request.

6 Pravidlá verziovania

- mergovanie je povolené len pre pull requesty do master/dev, inak sa použije rebase
- v prípade chyby alebo nutnosti zmeny commitu sa nerobí revert commit, ale posunie sa head na posledný validný commit a vytvoria sa lokálne nové commity, ktoré prepíšu chybnú pomocou force push.
- ak niekto urobí force push, upovedomí o tom všetkých developerov
- do repozitára sa neukladajú secrety, ak sa tak stane, developer musí secret force pushom odstrániť

7 Odporúčania

Na prácu s gitom odporúčam git gui rozhranie, hlavne pre git masterov, konkrétne pracovný nástroj Idea a plugin pre integráciu [Azure DevOps](#)

V prípade otázok, nejasností alebo problému kontaktujte git guru a autora metodiky verziovania:

Filip Agh
xaghf@stuba.sk