

# **Metodika verziovania kódu**

**Číslo a názov tímu:** 9. QuestionHive

**Dátum:** 03.11.2019

**Vypracoval:** Jakub Bista

## 1. Účel metodiky

Táto metodika bola vytvorená za účelom stanovenia jednotnej formy verziovania zdrojového kódu projektu. Na zdieľanie kódu bude náš tím využívať git, konkrétne nástroj GitLab.

## 2. Zdieľané repozitáre

Zdrojový kód projektu je zdieľaný v dvoch repozitároch. Rozdelenie kódu do týchto repozitárov je logicky odvodené od základného členenia vyvíjanej aplikácie. V jednom repozitári sa samostatne nachádza frontend aplikácie. Druhý repozitár obsahuje backend aplikácie, ktorý je zahrnutý v štandardnom Laravel projekte.

Repozitár pre frontend sa nachádza na webovej adrese:

<https://gitlab.com/PSoltes/2019-fiit-team09-frontend>

Repozitár pre backend sa nachádza na webovej adrese:

<https://gitlab.com/xbencat/team-project>

## 3. Vetvy

Každý repozitár pozostáva z niekoľkých vetiev/branches. V oboch repozitároch sa nachádza jedna produkčná vetva/production branch a jedna vetva vývoja/develop branch. Okrem nich môžu vývojári vytvárať, upravovať a spravovať ďalšie vetvy.

Pravidlá pre prácu s vetvami sú nasledovné:

- 1) Každá vytvorená vetva prislúcha jednej User story z Product backlogu.
- 2) Názov vetvy pozostáva zo znaku „#“, čísla danej vetvy a slovného názvu vetvy.
- 3) Číslo vetvy je štvormiestne číslo. Prvá číslica čísla vetvy určuje o aký druh vetvy sa jedná:
  - 1 – vetvy backendu
  - 2 – vetvy frontendu
  - 3 – ostatné vetvy
- 4) Za číslom vetvy nasleduje názov vetvy. Názov vetvy začína znakom „\_“ a za ním nasleduje slovný názov vetvy, ktorý významovo zodpovedá názvu príslušnej User story. Jednotlivé slová v názve vetvy sú rozdelené znakom „\_“ (poprípade pomlčkou „-“).

Príklad názvu vetvy:

- #2001\_Show\_Test\_Screen

- Vetva frontendu prisluchajuca Issue ShowTestScreen z Product backlogu

5) Spravy pri commitoch musia mat' prislusny obsah uvedený v anglickom jazyku. Spravy obsahuju sloveso v imperatíve a slovný opis vykonaných zmien v zdrojovom kóde.

Príklad commit spravy:

- Add button for deleting event in TestCreation view

#### **4. Vetva develop**

Vetva develop v každom repozitári sluzi na spájanie príspevkov v kóde od jednotlivých vývojárov. Do vetvy develop sa spájajú všetky ostatné vetvy prisluchajúce k jednotlivým issues z product backlogu.

Postup spájania vetiev do vetvy develop:

- A. Vývojár, ktorý pracuje na vetve x vytvorí pre danú vetvu merge request
- B. K vetve je priradený code reviewer
- C. Code reviewer spraví review danej vetvy
- D. Pripomienky od code reviewera zapracuje vývojár do zdrojového kódu
- E. Merge request je odsúhlasený code reviewerom
- F. Vývojár vykoná samotný merge a vetva x je pripojená do vetvy develop

#### **5. Vetva production**

Vetva production obsahuje zdrojový kód, ktorý je pripravený do produkcie. Z vetvy develop sa do vetvy production dostane zdrojový kód po schválení product ownerom.