

# Metodika na prácu s Gitom a verziovanie

**Autor:** Dávid Varga

**Dátum vytvorenia:** 5. 10. 2020

**Autor poslednej zmeny:** Filip Láštík

**Dátum poslednej zmeny:** 9.10.2020

## Verziovanie

V našom modeli vetvenia budeme používať tri typy vetiev:

1. **master** – hlavná vetva, obsahuje funkčnú nasadenú aplikáciu
2. **develop** – na rozdiel od vetvy master, obsahuje aj nové doposiaľ nenasadené prvky funkcionality
3. **pracovné vetvy** – pracovných vetiev môže existovať viac, obsahuje funkcionality, ktorú opisuje vždy jednu úlohu\chybu (prípadne User Story)

## Pravidlá git vetiev

Vetvy majú nasledovnú štruktúru:

1. **master** - hlavná vetva, ktorá odráža aktuálny stav projektu
  - Je zamknutá na priame pushovanie do vetvy, povolené len cez pull requesty pull request do tejto vetvy môže byť len z vetiev develop a hotfix
2. **develop** - vývojová vetva, ktorá odráža aktuálny stav vývoju
  - Je zamknutá na priame pushovanie do vetvy, povolené len cez pull requesty
3. **pracovné vetvy**
  - a. **feature/#ID\_ULOHY-POPIS\_ULOHY** - pridanie novej funkcionality, následne sa zlučuje s vetvou develop pomocou pull requestu
  - b. **hotfix/#ID\_CHYBY-POPIS\_CHYBY** - oprava chýb, následne sa zlučuje s vetvami master a develop pomocou pull requestu

## Proces pridávania nových zmien

1. Vytvorenie novej vetvy s určitou menovou konvenciou: **#ID\_ULOHY\CHYBY-POPIS\_ULOHY\CHYBY** (napr. **TP-01-user-registration**)
  - Nezabudnite napísať ID úlohy vo formáte TP-XX
2. Komitnutie a pushnutie zmien do novo vytvorenej vetvy
3. Vytvorenie Merge requestu do nadradenej vetvy
4. Po schválení minimálne 1 iným členom je možné zmeny mergeovať nadradenou vetvou

## Prehliadka kódu

Zachovanie, prípadné zlepšenie čistoty kódu - kód by mal byť schválený iba ak nezhoršil kvalitu aktuálneho kódu

## Pravidlá správneho komitovania

- Jednotlivé zmeny by mali byť logicky rozdelené do jednotlivých komitov

## Štruktúra komitu

- Správa komitu
- Podrobnejší popis - je možné ho napísať, ak nie je jasné z kódu čo sa zmenil

## Správa komitu

Stanovený formát budú mať aj správy komitov v jednotlivých vetvách. V správe (message) sa bude nachádzať [ID úlohy\chyby], na ktorom sa práve pracovalo a jednoduchý komentár k pridanej alebo zmenenej funkcionalite

**Ukážka správy commitu:** [TP-01] Vytvorenie formulára

## Vydanie nových zmien

Pri vydaní nových zmien postupujeme nasledovne:

1. Mergnutie *develop* vetvy do *mastra*
2. Oddelenie *develop* vetvy od *mastra*
3. Vytvorenie tagu (pozri vytváranie tagov)
4. Vydanie nových zmien

## Vytváranie tagov

- Tag má nasledujúcu štruktúru: **vČÍSLO\_VERZIE.ČÍSLO\_ZMENY.ČÍSLO\_PATCHU** (napr. v2.158.7)
  - **ČÍSLO\_VERZIE** sa zvýši, ak sa vytvárajú nekompatibilné zmeny
  - **ČÍSLO\_ZMENY** sa zvýši, ak pridávame kus funkcionality, avšak zachováme spätnú kompatibilitu
  - **ČÍSLO\_PATCHU** sa zvýši, ak vytvárame malé zmeny, opravy chýb, oprava kompatibility