

Metodika verziovania

V tomto dokumente budeme prezentovať základné princípy spolného vývoja viacerých osôb na spolnom projekte z hľadiska vytvárania programového kódu. V našom projekte budeme používať systém riadenia verzií Git (<https://git-scm.com/about>).

Pravidlá vytváranie zmien kódu

0. Informujem o plánovanej zmene ostatných členov tímu

Je veľmi vhodné, aby pred samotným vykonaním zmeny boli ostatní členovia, ktorých by sa daná zmena mohla týkať, informovaní. Predídete sa zbytným problémom a zároveň sa nestane, že budú dvaja členovia pracovať na tých istých astiach kódu súčasne.

1. Každá zmena musí byť vytváraná vo vlastnej vetve.

2. Meno vetvy musí byť v nasledovnom tvare:

EH-XX/[typ vetvy]/[meno konkrétnej zmeny]

príom typ vetvy môže byť:

- **feature** - nová funkcionálnosť, alebo vo všeobecnosti akékoľvek rozšírenie kódu. Napr. zrýchlenie algoritmu ukladania ...
- **bugfix** - oprava akýchkoľvek chýb, ktorá sa vyskytla mimo štandardného vývoja (mimo vetvy v, ktorej vyvíjame)
- **junk** - vetva, ktorá je určená na testovanie alebo akýchkoľvek iných o sa vymyká štandardným prípadom

Meno konkrétnej zmeny pomenúva samotnú zmenu, ktorú vykonávame. Napr. : ***bugfix/web_lang_error***, ***feature/picture_upload***, ***junk/partition_table_performance***. Názvy zmien sú v **anglickom jazyku**, **malými písmenami** a slová oddeľujeme **podtržníkom**.

Za zvyčajných okolností sa vetva vytvára z aktuálnej verzie hlavnej vetvy **main**:

Príklad vytvárania vetvy s použitím rozhrania git bash:

- Prepíname sa na vetvu **main**:

```
git checkout main
```

- Stiahnem si najnovšiu verziu **main**:

```
git pull origin main
```

- Vytvorím si novú vetvu z aktuálnej vetvy, v ktorej som (main):

```
git checkout -b feature/insert_log_endpoint
```

3. Iastkové zmeny v rámci vetvy by mali byť rozložené do komítov

Vždy keď sa uzavrie vývoj jednej malej asti v rámci vetvy a program je funkčný, je vhodné zmeny komitovať. V prípade, že sa následne vo vývoji dostaneme do takej fázy, že je kód v zlom stave a je jednoduchšie vrátiť sa do predchádzajúceho stavu, vieme sa pomerne jednoducho vrátiť do stavu z posledného komítu. Každý komit by mal obsahovať rýchlu správu, ktorá sumarizuje o sa v danom komite vykonalo. Dobrým pravidlom je, že sa píše **Preo** bolo zmena vykonaná a nie **o** sa zmenilo. Napr. : ("Deserializácia unix time na DateTime formát") je lepšie ako ("Pridaná funkcia na dešerializáciu")

Príklad komítovania s použitím rozhrania git bash:

- Najprv musíme Gitu poveda aké súbory majú by v komite. V tomto prípade pridáme všetky upravené súbory:

```
git add --all
```

Následne vykonáme samotný komit, pričom zadáme aj komit správu pomocou prepínaa **-m**.

```
git commit -m "testovanie spravania gitu"
```

4. Pred tým ako zmeny spojíme s main vetvou, musí ich schváli iný len tímu

V praxi to znamená, že každý merge request musí by schválený ešte alšou osobou, resp. nemôžem sám spoji svoju vetvu s main vetvou.

Merge request a následné spájanie sa zvyajne vykonáva cez interaktívne webové rozhranie. V rámci tohto kroku sa vykonáva aj "**Code review**".