

Tím č. 27

Kill Bills

Metodika pre verziovanie kódu

1 Dedikácia

Metodika určuje pravidlá a popis verziovania kódu v tíme Kill Bills. Keďže pre všetky moduly patriace do projektu sú uložené v jednom repozitári, je potrebné nastaviť pravidlá vytvárania nových verzií kódu. Je určená pre vývojárov, ktorí pracujú na projekte.

2 Štruktúra a pravidlá repozitára

Repozitár je privátny, teda si vyžaduje explicitné pridanie nových developerov do projektu. Používa systém GitLab a je umiestnený priamo na oficiálnej GitLab stránke <https://gitlab.com>.

2.1 Branch

Vetva (branch) reprezentuje jednu nezávislú líniu vývoja, ktorá predstavuje buď celý softvér alebo prírastok softvéru.

Repozitár je rozdelený do troch hlavných vetiev:

- **master** - vetva obsahujúca kód, ktorý práve beží na produkčnom serveri
- **staging** - kód, ktorý beží na staging serveri
- **development** - kód, ktorý sa práve vyvíja, zvyčajne je spúšťaný a testovaný lokálne developermi a CI

Všetky vetvy sú chránené pred priamym zápisom cez commit, jediný spôsob, ako do týchto vetiev vložiť kód je tzv. **Merge requestom**, iným názvom **Pull requestom**. V projekte je viacero typov requestov, popísané sú v sekcii Merge request nižšie v tomto dokumente.

Pre pridanie nového kódu do repozitára je potrebné si vytvoriť novú vetvu, ktorá musí byť nazvaná nasledovne:

`(meno_subprojektu)/(feature|fix|hotfix|update)/(nazov)[-cislo_story]` Okrúhle zátvorky znamenajú povinné argumenty.

- **meno_subprojektu** - musí byť jednoznačne identifikovateľný reťazec označujúci subprojekt, príklad správneho označenia: **android**, **parser**, **ocr**
- **(feature|fix|hotfix|update)** - označenie dôvodu otvorenia branche,
 - **feature** - pridanie novej funkcionality
 - **fix** - oprava bugu
 - **hotfix** - oprava bugu, ktorý sa vyskytol na branchi staging alebo master
 - **update** - aktualizácia závislostí
- **nazov** - jednoznačný názov, ktorý určuje účel vetvy, slová oddelené maximálne troma podtrhovníkmi; príklady správne napísaného názvu: **modify_pattern_logic**, **deploy_images**, **password_recovery**
- **[-cislo_story]** - číslo user story zo systému na riadenie úloh, tento argument je povinný vždy, keď sa pracuje na user story, môže tiež nastať prípad, kedy je potrebné pridať kód, ktorá nie súčasťou user story, v tomto prípade sa číslo vynechá

Príklady správneho označenia vetvy:

- **portal/feature/password_recovery-241225**

- `parser/fix/create_pattern_fail-241322`
- `android/update/version_number`
- `parser/hotfix/fix_dockerfile_and_ignores`

2.2 Merge request

Merge requesty slúžia na integráciu zmien do projektu z iných vetiev. Hlavným účelom merge requestu je ale ucelená prehliadka nového prírastku kódu. Pravidlá a postup pri prehliadkach kódu sú spísané v Metodike pre prehliadky kódu.

V projekte existuje viacero typov merge requestov:

- merge z *feature* vetvy do *development*
 - v tomto prípade je potrebné dať vetve adekvátny nadpis, ktorý popisuje, čo sa v merge požiadavke mení, príklad správneho názvu: `Fix Android application handling success message incorrectly`
 - na kód v tomto type merge requestu je potrebné vykonať prehliadku kódu ďalším členom tímu
- merge z *development* vetvy do *staging* vetvy
- merge z *staging* vetvy do *master* vetvy
 - v oboch prípadoch vyššie sa požiadavka na merge nazýva v nasledovnom tvare: `Merge to (staging|production) yyyy-mm-dd`, kde `yyyy-mm-dd` je dátum vytvorenia merge requestu. Tieto typy môže včleniť zakladateľ merge requestu bez prehliadky.

2.3 Commit

Commit správy slúžia na zrýchlenie procesu prehliadky kódu, pomoc pri tvorbe popisu nového vydania aplikácia a popis pre ostatných vývojárov kódu určiť, prečo bola pridaná zmena, preto správy musia dodržiavať nasledujúce pravidlá:

- do 50 znakov; ak je commit moc veľký a jeho opis sa nezmestí do 50 znakov, je potrebné ho rozdeliť
- opis všetkého, čo sa v kóde zmenilo
- je napísaný v angličtine a používa imperatív v prítomnom čase, príklad dobrej commit správy: `Add data id token to check items API`; príklad zlej commit správy: `Fixed some errors`
- ak je to potrebné, je možné pridať detailnejší popis od tretieho riadku commit správy, zalomený na menej než 80 znakov

3 Zoznam verzií

- 1.1.1 (11. 12. 2017)
 - oprava nepresných výrazov
- 1.1 (11. 12. 2017)
 - pridaná sekcia o merge requestoch
- 1.0 (10. 12. 2017)
 - pridané základné popisy k commitom a branchiam