

Metodika verziovania

1. Úvod

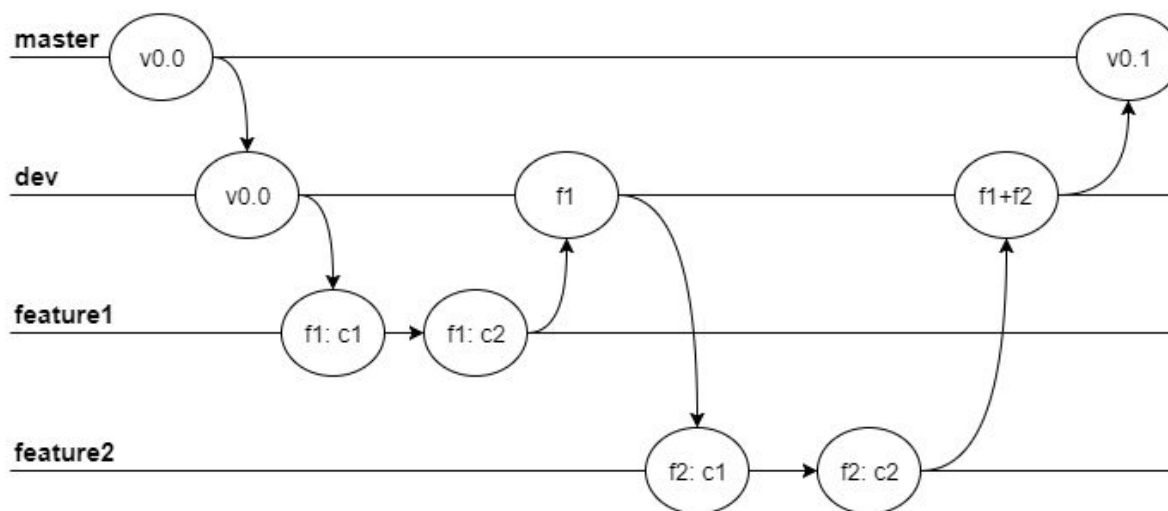
Cieľom metodiky je vyjasnenie si spôsobu práce so súbormi nášho projektu tak, aby každý programátor pracoval na správnom a najaktuálnejšom kóde. Ďalšou veľkou výhodou správneho verziovania je oddelenie práce rôznych autorov pracujúcich na rôznych úlohách (funkciách). Dobrým príkladom je, ak programátor vytvorí chybný kód, tak ostatní programátori pracujúci v iných vetvách nie sú daným chybným kódom zasiahnutí.

2. Slovník pojmov

Pojem	Význam
Branch (vetva)	Samostatná verzia projektu. Viaceré vetvy umožňujú paralelnú prácu
Push	Nahrание zmien súborov z lokálneho repozitára (lokálne commity) na vzdialený (remote repozitár)
Pull	Stiahnutie vzdialenej vetvy do lokálneho repozitára
Merge request	Akcia spájania jednej vetvy (napr. s novou funkciou) do inej vetvy
Commit	Zachytenie stavu súborov danej vetvy (Pridanie, odstránenie, zmena súboru)

3. Všeobecné vetvenie

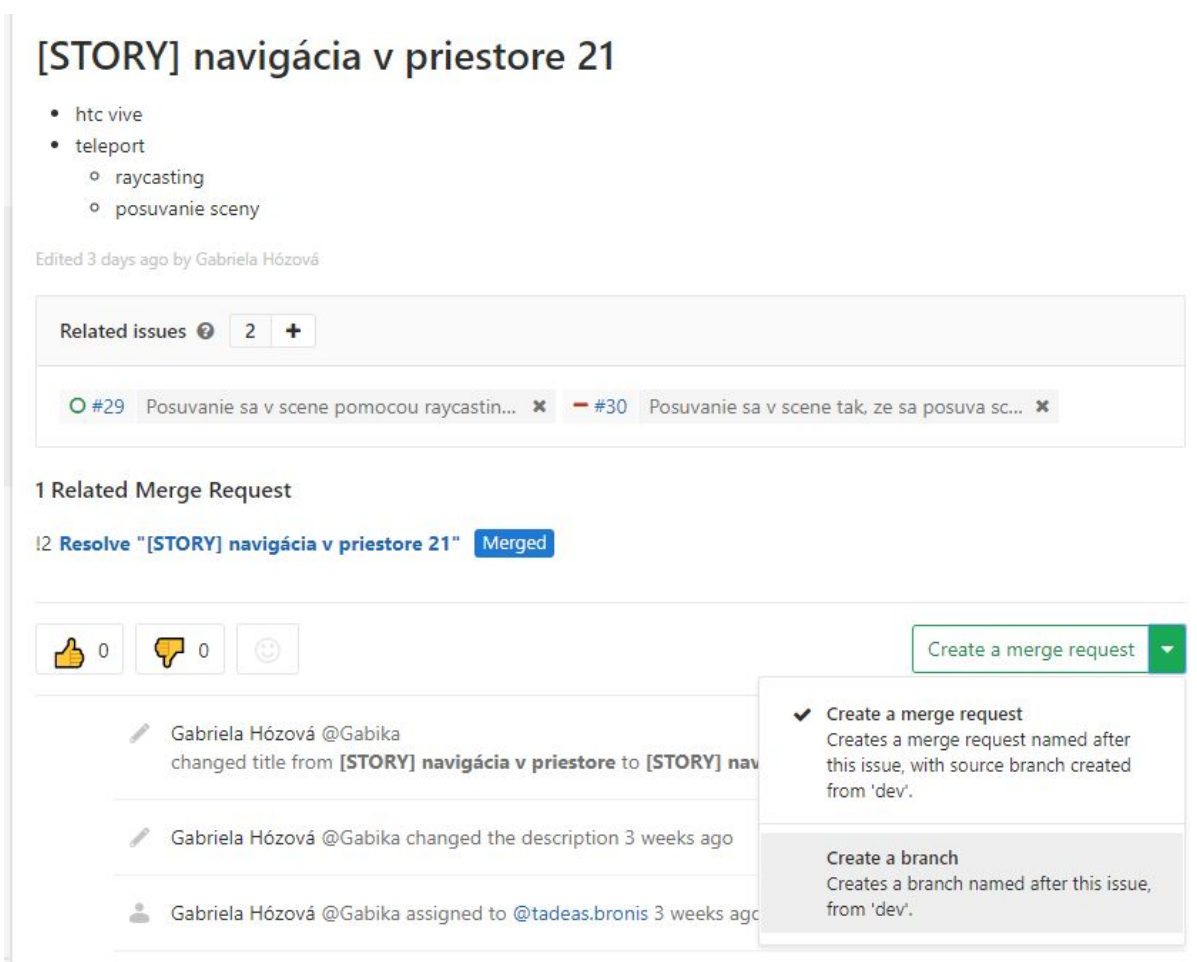
Všetky súbory projektu sú spravované nástrojom git s nadstavbou GitLab, ktorý ponúka prívetivé webové prostredie spájajúce úlohy projektu a správu vetiev projektu. Hlavný projekt je rozdelený na 2 stále vetvy dev a master. Vetva master predstavuje najstabilnejšiu verziu projektu, pričom vetva dev je určená na neustály vývoj. V určitých časových úsekoch sa vetva dev bude spájať s hlavnou vetvou master a vznikne tak nová stabilná verzia. Okrem týchto 2 vetiev sa pre každú väčšiu (u nás nazývanú story úlohu) vytvorí osobitná vetva pomenovaná podľa úlohy. Na obrázku č. 1 je možná ukážka priebehu vývoja.



Obrázok č. 1: Príklad vytvárania vetiev

4. Vytváranie vetiev

Odporúča sa vytvárať nové vetvy od dev vetvy, zameranej na vývoj. Vytváranie vetiev úzko súvisí s úlohami pre daný šprint. Pre každú úlohu označenú ako story je potrebné vytvoriť novú vetvu pomocou nástroja GitLab podobne ako je na obrázku č. 2.



Obrázok č. 2: Vytvorenie vetvy

Užívatelia si môžu vytvárať aj svoje vlastné vetvy pre svoje menšie úlohy, ale daný postup sa neodporúča, aby sa zabránilo neprehľadnosti.

5. Pridávanie zmien do vetiev (Push)

Každý člen tímu sa musí držať týchto pravidiel pri odovzdávaní zmien na vzdialený git repozitár:

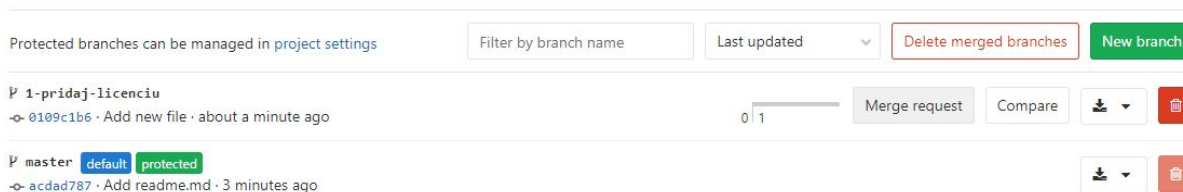
- Nevytvárať jeden obrovský commit, ale preferovať viacero commitov podrobne opísaných
- Zmeny treba vždy krátko ale výstižne opísať v commit správe
- Aby sme mohli paralelne pracovať treba často a pravidelne nahrávať všetky zmeny projektu na vzdialený git repozitár
- Všetky zmeny, ktoré riešia určitú úlohu, musia byť nahraté do správnej vetvy, teda vetvy danej úlohy
- Neodporúča sa vytvárať priame zmeny na dev vetve, ale vytvárať merge requesty (spájať) odvodené vývojové vetvy

6. Spájanie vetiev (Merge request)

Na obrázku č. 3 je znázornené vytvorenie merge request pomocou webového nástroja GitLab, pričom uzavretie tohto merge request pomocou spájania vetiev je na obrázku č. 4.

Základné pravidlá:

- Odvodená vetva (napr. vetva určitej úlohy) sa môže spojiť z rodičovskou vetvou až keď daná úloha prejde revíziou a testovaním (podrobnejšie opísané v metodike manažmentu úloh)
- Vytvoriť merge request je možné až keď pridelený programátor dokončil programovanie a jeho zmeny sú prípravné na revíziu a testovanie
- v merge requeste je potrebné označiť alebo prideliť ľudí, ktorí budú vykonávať revíziu a testovanie
- Po úspešnom otestovaní sa môže merge request ukončiť spojením vetiev
- Pri akýchkoľvek problémoch pri revízií alebo testovaní je potrebné oznámiť nájdené problémy pridelenému programátorovi a spoločne sa snažiť čo najrýchlejšie problémy odstrániť



Obrázok č. 3: Vytvorenie merge requestu

Open Opened 18 minutes ago by  **Michal Škuta**

Edit

Close merge request

Add new file

Closes #1

Request to merge 1-privadaj-licenciu  into master

Check out branch 

Merge Remove source branch

Closes #1

You can merge this merge request manually using the [command line](#)



Obrázok č. 4: Uzavretie merge requestu spojením vetiev