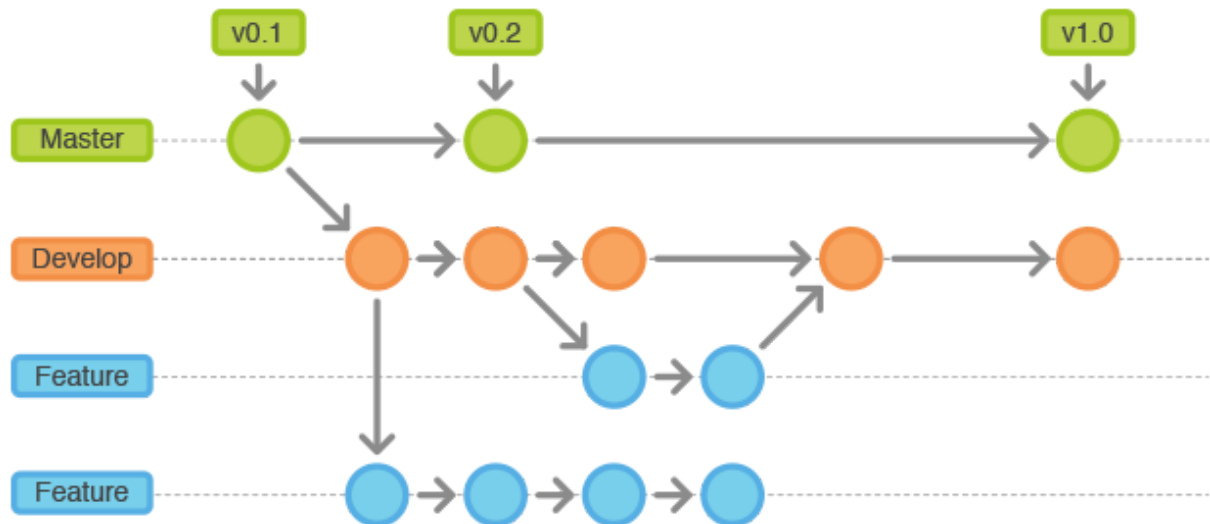


# Metodika práce s gitom

## Spôsob práce s gitom



V projekte sa budú udržiavať dve hlavné vetvy:

- **Master** - Hlavná vetva, ktorá odráža otestovaný funkčný kód
- **Develop** - Vetva, do ktorej sa priebežne “pushujú” zmeny z ďalších vetiev
- **Feature vetvy** - Vedľajšie vetvy, každá predstavuje samostatnú funkcionality.

Po skončení práce na konkrétnej feature sa merge daná vetva s developom. Po dôkladnom otestovaní novej funkcionality sa develop môže merge-núť s master vetvou, alebo v prípade nájdených problémov sa vytvorí nová vetva za účelom opravenia chýb.

Vždy sa najprv mergeje develop vetva do feature vetvy. Potom sa feature vetva otestuje či všetko fičí tak ako má. Pokiaľ je všetko v poriadku tak sa merge daná vetva do developu. To isté platí pri mergovaní do mastera.

# Vývojový cyklus

Pri začatí práce na novej funkcionalite postupujeme podľa nasledujúcich krokov:

1. Dostal som zadanú novú úlohu
2. Prepnem sa do vetvy, z ktorej budem vychádzať `git checkout nazov_vetvy`
3. Aktualizujem túto vetvu(*master, develop*), pre prípad, že niekto medzičasom vykonal nejaké zmeny `git pull origin nazov_vetvy`
4. Vytvorím novú vetvu `git checkout -b nova_vetva`
5. Vykonám ľubovoľné zmeny v zdrojových kódach a iných súboroch
6. Stagenem vykonané zmeny `git add -A`
7. Commitnem zmeny `git commit -m "popis_commitu"`
8. Zmeny 'pushnem' do gitu `git push origin nazov_vetvy`

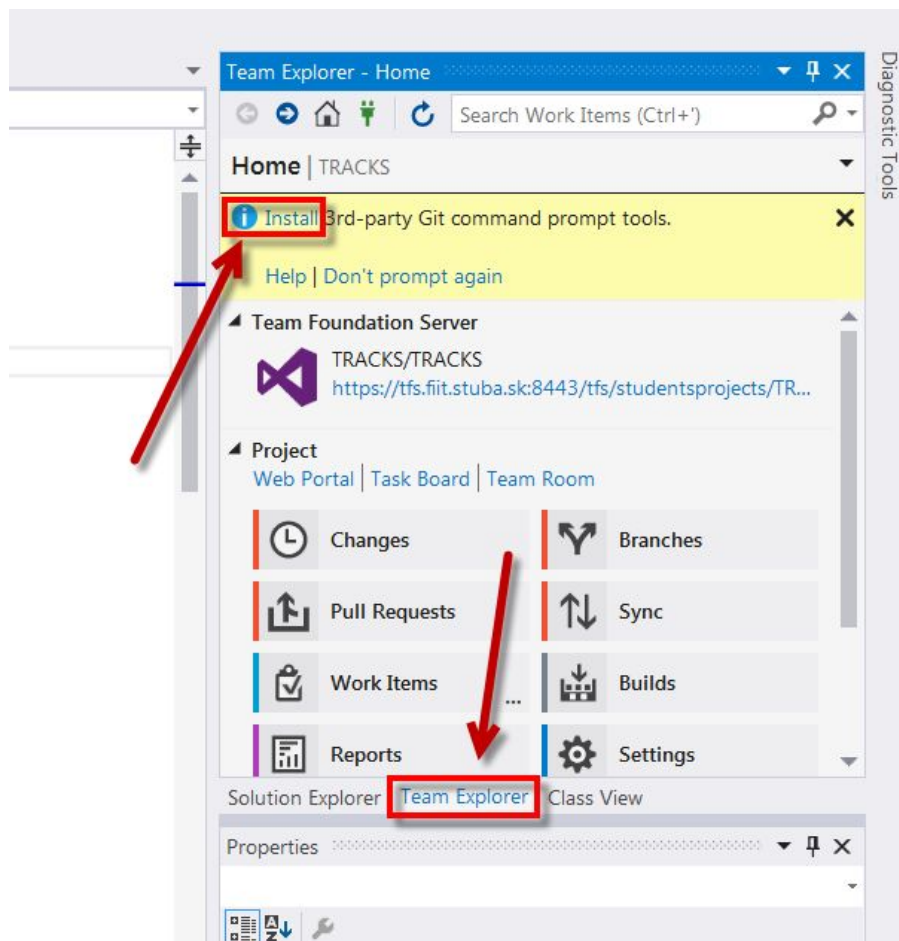
Pri pokračovaní práce na vetve, ktorá už v gite existuje postupujeme nasledovne:

1. Dostal som zadanú už existujúcu úlohu
2. Zistím, či požadovaná vetva existuje lokálne `git branch`
3. Prepnem sa do požadovanej vetvy `git checkout nazov_vetvy`
  - a. Pokiaľ vetva lokálne neexistuje, stiahnem si ju z gitu `git fetch origin remote_nazov_vetvy: local_nazov_vetvy`
4. Pokiaľ som novú vetvu práve nestiahol z gitu, aktualizujem ju `git pull origin nazov_vetvy`
5. Vykonám ľubovoľné zmeny v zdrojových kódach a iných súboroch
6. Stagenem vykonané zmeny `git add -A`
7. Commitnem zmeny `git commit -m "popis_commitu"`
8. Zmeny 'pushnem' do gitu `git push origin nazov_vetvy`

## Inštalácia gitu

- **Cez visual studio**

Pokiaľ git ešte nie je nainštalovaný, je možné ho stiahnuť cez visual studio(VS). Po otvorení VS otvoríme záložku 'Team Explorer' a v hore sa nám zobrazí možnosť na stiahnutie gitu. Všetko potvrdíme a po stiahnutí nainštalujeme stiahnutý inštalačný súbor gitu.

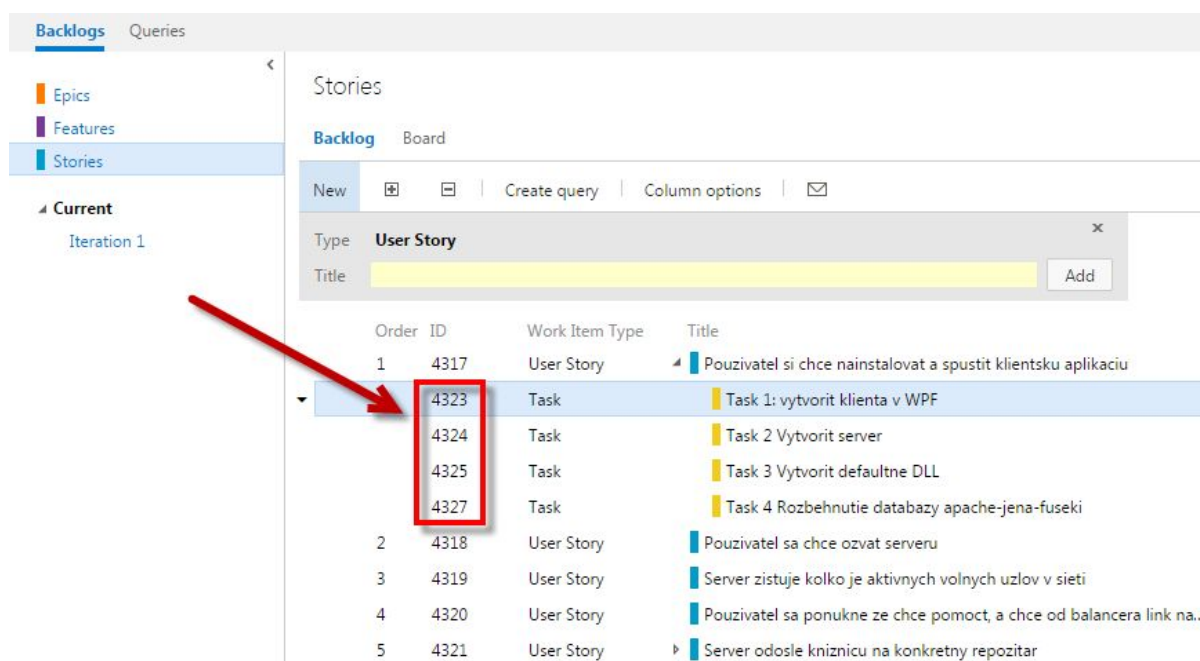


- **Z internetu**

Najnovšiu verziu gitu môžeme stiahnuť priamo zo stránky gitu na <https://git-scm.com/downloads>

## Pomenovanie vetví

- Každá nová vetva sa vytvára za účelom vytvorenia/opravenia nejakej funkcionality a jej názov by mal jednoznačne hovoriť o tom, čoho sa týka.
- Pomenovanie: **feature/'číslo\_user\_story'-'číslo\_tasku'**
  - napr. feature/4317-4327
  - ID taskov a user stories sa dajú nájsť v backlogu (treba správne nastaviť column options) alebo v detaile user story.



Order	ID	Work Item Type	Title
1	4317	User Story	Používateľ si chce nainštalovať a spustiť klientsku aplikáciu
	4323	Task	Task 1: vytvoriť klienta v WPF
	4324	Task	Task 2: Vytvoriť server
	4325	Task	Task 3: Vytvoriť defaultné DLL
	4327	Task	Task 4: Rozbehnutie databázy apache-jena-fuseki
2	4318	User Story	Používateľ sa chce ozvať serveru
3	4319	User Story	Server zisťuje koľko je aktívnych voľných uzlov v sieti
4	4320	User Story	Používateľ sa ponúka, že chce pomôcť, a chce od balancera link na...
5	4321	User Story	Server odoslal knižnicu na konkrétny repozitár

## Testovanie

Po dokončení ľubovoľnej funkcionality je potrebné, aby bola dôsledne otestovaná niekým druhým z tímu. Keď sa dokončia všetky tasky pre danú User Story, presunie sa do stavu 'Resolved', kde ktokoľvek z tímu otestuje správnosť danej funkcionality, prípadne skontroluje, či sú splnené **Acceptance criteria** (ak sú zadané). Následne, ak je všetko v poriadku, user story sa dá do stavu closed.

## **Odporúčaná literatúra**

Prehľad kľúčových pojmov a metodika gitu -

<http://labss2.fiit.stuba.sk/TeamProject/2010/team03is-si/data/metodiky/manazment-verzii.pdf>

Detailná dokumentácia ku gitu: <https://git-scm.com/doc>