

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Kolaboratívne prototypovanie

[CollabUI]

Metodika verziovania

Tím: 24. CollabUI

Vedúci tímu: Ing. Eduard Kuric, PhD.

Členovia tímu: Adrián Nagy, Lukáš Vrba, Ján Kleň, Michal Melúch, Tomáš Mňačko, Peter Písecký, Miloslav Smetana

Študijný program: Inteligentné softvérové systémy

Ročník: 2017/2018

1. Úvod

Cieľom tejto metodiky je definovať základné postupy odovzdávania zdrojového kódu a udržiavania verzií v projekte Collab-UI. Metodika opisuje prácu s nástrojom na správu verzií Git spolu s použitou grafickou nadstavbou Sourcetree. Repozitár celej vyvíjanej aplikácie je uložený na portáli Bitbucket.

Touto metodikou sa riadia všetci členovia tímu č. 24 Collab-UI v rámci predmetu tímový projekt v akademickom roku 2017/2018.

2. Pojmy a skratky

- **Sprint** - iteračné obdobie projektu, v našom prípade 2. týždne
- **Epic** - konkrétna oblasť projektu, dekomponovaná na menšie oblasti
- **User Story/Backlog** - US - jedna funkcionálna jednotka projektu, ktorá sa skladá z niekoľkých taskov
- **Task** - úloha v rámci jedného story
- **Product owner** - PO - komunikuje so zákazníkom a zbiera od neho požiadavky, slúži ako mediátor pre komunikáciu s tímom a zákazníkom
- **Git** - distribuovaný systém riadenia revízií
- **Branch** - vetva - vývojová vetva projektu, v ktorej sa realizuje ucelená časť funkcionality.
- **Master branch** - hlavná vetva projektu, v ktorej je nasadená implementácia, ktorá úspešne prešla testovaním a zároveň ju akceptoval aj product owner.
- **Commit** - príspevok (najčastejšie implementačný) do vybranej vetvy projektu

3. Dôležité odkazy

- Repozitár projektu: https://bitbucket.org/agileuiteam/uiagile_repo
 - URL pre klonovanie: https://BB-NICKNAME@bitbucket.org/agileuiteam/uiagile_repo.git
- Systém riadenia revízií Git: <https://git-scm.com/>
- Nástroj pre pokročilú správu revízií: <https://www.sourcetreeapp.com/>

4. Pravidlá a postupy

Pri práci s verziami projektu je nutné riadiť sa nasledovnými pravidlami:

- Na správu verzií sa používa výhradne systém *Git* v spojení s hostingovou službou *Bitbucket*
- Na všetky úkony sa používa GUI rozhranie nástroja *Sourcetree*. Predchádza sa tak chybám, ktoré vznikajú pri používaní systému *Git* pomocou príkazového riadku. Využitie príkazového riadku je povolené len v špeciálnych prípadoch, ktoré nemôžu byť vyriešené pomocou nástroja *Sourcetree*.
- Závažné konflikty medzi jednotlivými príspevkami (t. j. commitmi) sa vždy riešia manuálne po konzultácii s autorom konfliktného kódu.
- Hlavná (angl. master) vetva projektu je v správe scrum mastera, ktorý do nej zlučuje zmeny až po ich úspešnom otestovaní akceptácií product ownerom. **Do master vetvy sa v priebehu šprintu necommituje žiadny kód.**

4.1 Základy práce s nástrojmi na správu verzií

Aby bolo možné pracovať s vzdialeným repozitárom projektu Collab-UI, je nutné mať nainštalovaný nasledujúci softvér:

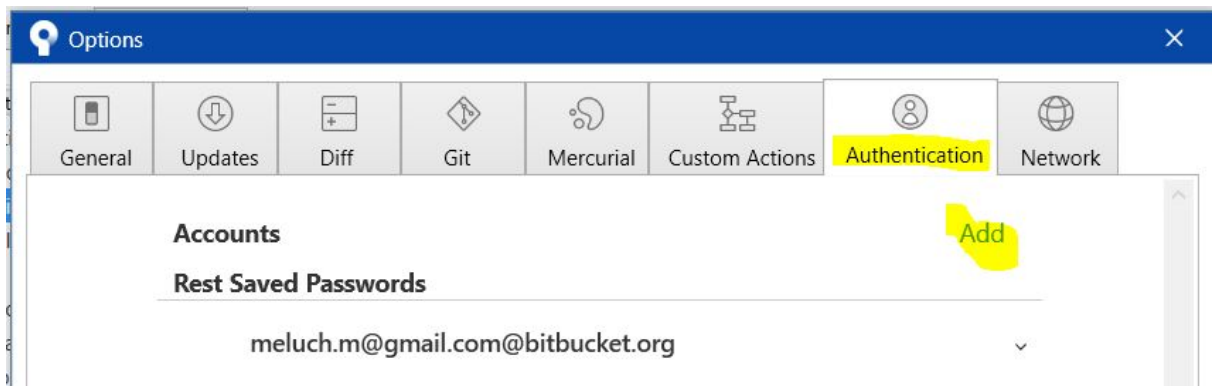
1. Systém riadenia revízií **Git** (stiahnuť [TU](#))
2. Grafický nástroj pre správu verzií **Sourcetree** od spoločnosti *Atlassian* (stiahnuť [TU](#))

Aby bolo možné pracovať so vzdialeným repozitárom uloženým na serveri Bitbucket, **je nutné mať vytvorené Atlassian konto**. Konto je možné si vytvoriť [TU](#).

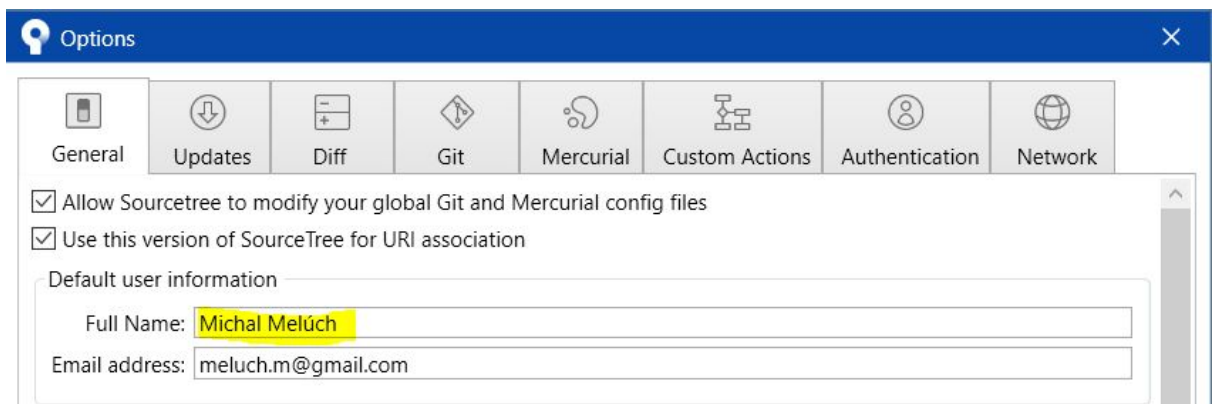
Po nainštalovaní uvedeného softvéru je nutné prepojiť aplikáciu *Sourcetree* so vzdialeným repozitárom projektu a Atlassian kontom vývojára.

Postup prihlásenia sa do svojho Atlassian konta v aplikácii *Sourcetree* je nasledovný:

1. Z hornej lišty zvolíme možnosť *Tools* a následne *Options*.
2. V pop-up okne, ktoré sa zobrazilo vyberieme predposlednú záložku s názvom *Authentication* a vyberieme možnosť pridať nové konto (*Add*):



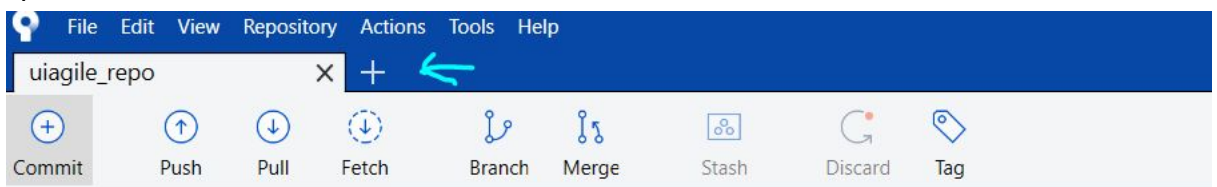
3. Zadáme svoje prístupové údaje, pomocou ktorých sa prihlasujeme aj na doménu <https://bitbucket.org/>.
4. V prvej záložke pup-up okna vyplníme svoje celé meno a priezvisko:



5. Aplikácia *Sourcetree* je teraz korektne prepojená s naším *Bitbucket* kontom.

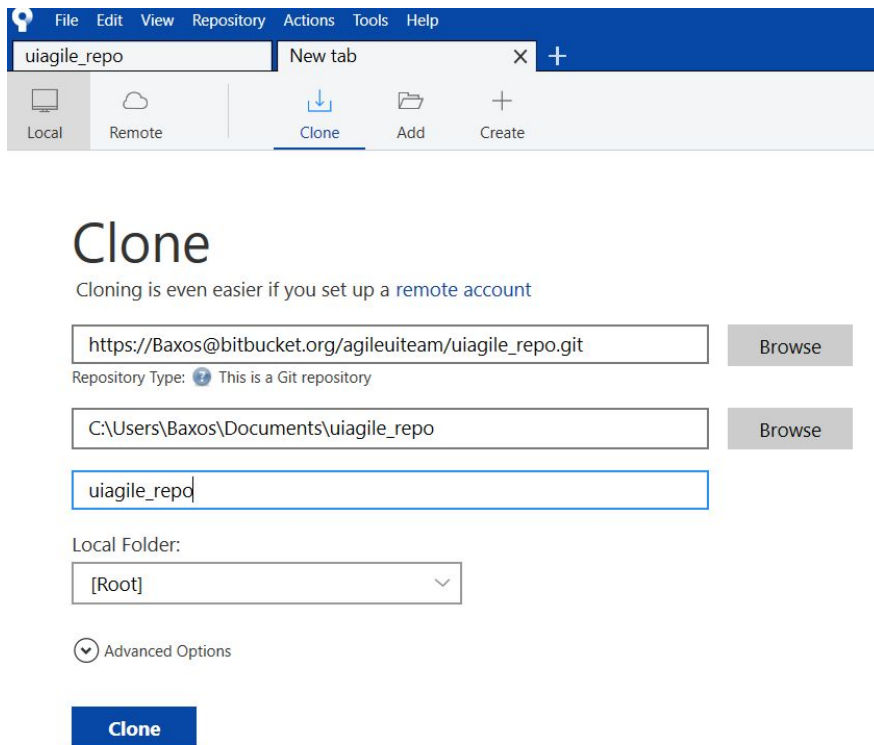
Postup pripojenia sa k vzdialenému repozitáru aplikácie je nasledovný:

1. Ak nás aplikácia sama nevyzve na klonovanie existujúceho repozitára, túto možnosť spustíme pomocou pridania novej záložky pod hornou lištou aplikácie:



2. Z horného menu pod názvom záložky vyberieme možnosť *Clone* a vložíme HTTPS URL Bitbucket repozitára, ktorá je uvedená v prehľade (*Overview*)

na stránke Bitbucket (odkaz [TU](#)).



File Edit View Repository Actions Tools Help

uiagile_repo New tab x +

Local Remote Clone Add Create

Clone

Cloning is even easier if you set up a remote account

Repository Type: This is a Git repository

Local Folder:

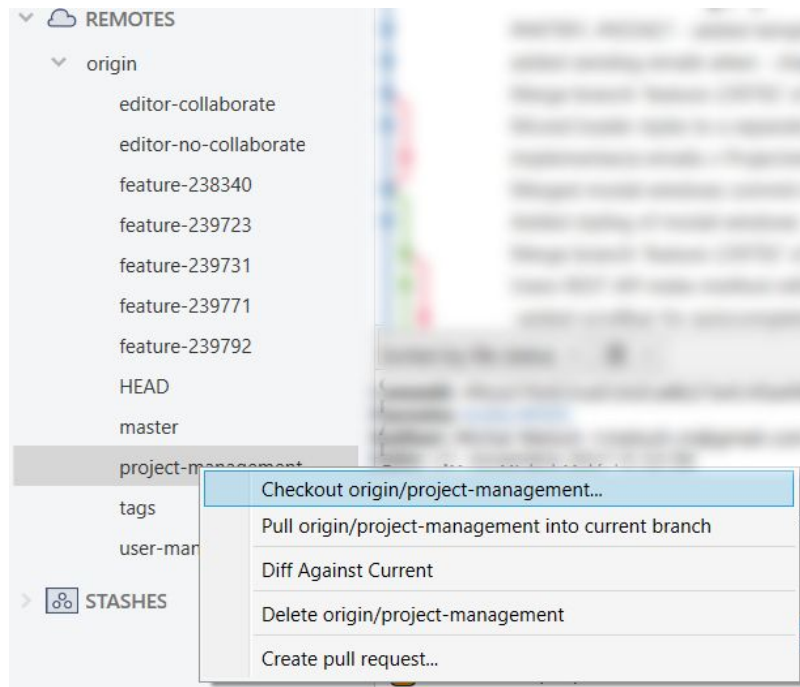
Advanced Options

3. Proces ukončíme tlačidlom *Clone* a počkáme kým sa repozitár naklonuje na nami zvolené lokálne úložisko.

4.2 Práca s vetvami projektu

Vetvy projektu sú vytvárané pomocou zoskupovania aktuálne riešených úloh (t. j. podľa scrum stories). Názvy vetiev musia byť čitateľné a zodpovedať charakteru riešených úloh - napr. *user-management* alebo *project-management*.

Rozlišujeme lokálne (tzv. *local*) a vzdialené (tzv. *remote*) vetvy. Vytvárať nové lokálne vetvy nie je pri bežnom vývoji potrebné - všetky vetvy vytvárame ako remotes (t. j. vzdialené) a ostatní členovia tímu si ich následne stiahnu do svojich počítačov pomocou možnosti *checkout* nachádzajúcej sa v bočnom menu pod záložkou *Remotes*:



Po checkoutnutí vzdialenej vetvy sa vytvorí jej lokálna kópia, ktorú môžeme vidieť v bočnom menu pod záložkou *Branches*. Prepínanie medzi takto vytvorenými lokálnymi verziami funguje taktiež pomocou príkazu *checkout* avšak vykonávame ho pod záložkou *Branches* (t. j. medzi lokálnymi vetvami)

Postup vytvárania novej vzdialenej vetvy (napr. pri začiatku nového šprintu) je podrobne opísaný v dokumentácii nástroja *Sourcetree* (odkaz [TU](#)).

4.3 Zásady prispievania do vetiev

Bežné prispievanie do vetiev projektu je realizované pomocou nasledovných *Git* príkazov dostupných v prostredí aplikácie *Sourcetree*:

- **Fetch** (automaticky po zapnutí aplikácie, manuálne v prípade potreby)
- **Pull** (pred začatím práce na projekte)
- **Commit** (po skončení menšej časti práce)
- **Push** (po skončení ucelenej časti práce)

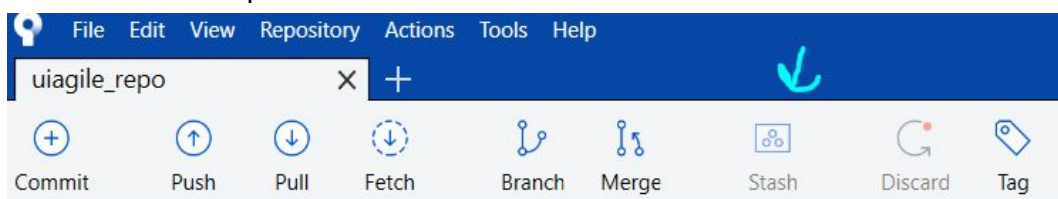
V prípade konfliktov medzi súbormi po príkaze *Push* je nutné tieto konflikty vyriešiť tak, aby zostali zachované zmeny obidvoch autorov. Následne je možné tieto súbory opakovane commitnúť a budú systémom *Git* akceptované.

Správy opisujúce jednotlivé commity musia byť stručné a vecné ale zároveň by mali dostatočne výstižne opisovať problém, ktorý bol v rámci daného commitu riešený.

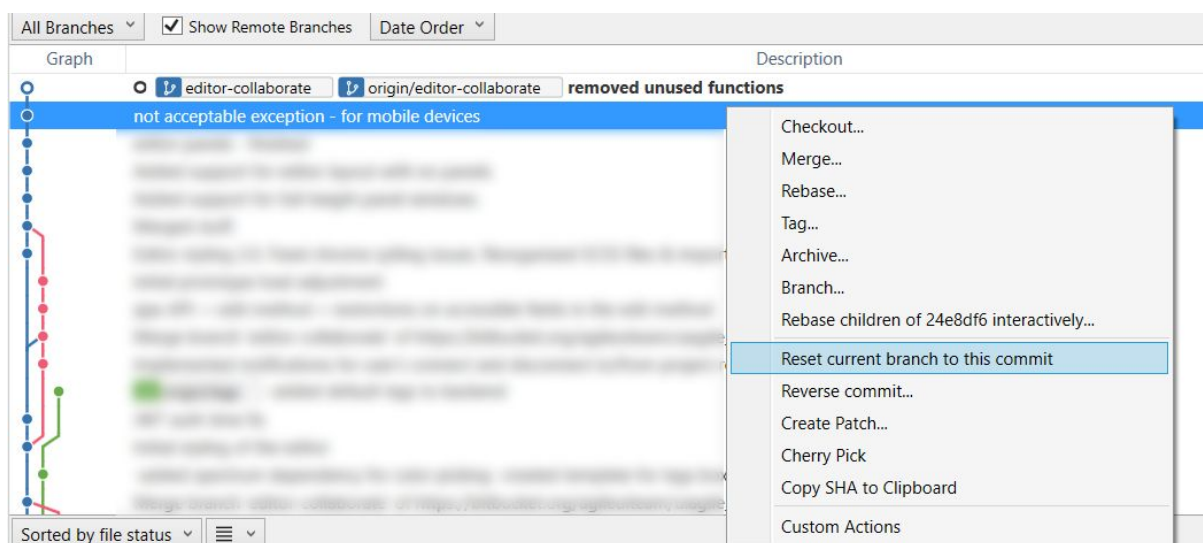
4.4 Návrat ku starším verziám projektu

V prípade potreby návratu ku staršej verzii projektu (napr. v prípade zavedenia závažných chýb novšou verziou) je možné sa vrátiť ku staršiemu commitu pomocou nasledovných krokov:

1. Lokálne zmeny si v prípade potreby uložíme pomocou príkazu *Stash* z horného menu aplikácie:



2. Zvolíme želaný starší commit v hlavnom prehľade a v jeho kontextovom menu zvolíme možnosť *Reset current branch to this commit*:



3. V nasledujúcom pop-up okne zvolíme mód *Hard* - t. j. nahradenie lokálnych zmien - a potvrdíme voľbu.
4. Stav projektu sa teraz nachádza v stave, v akom bol pri danom commitu. Lokálne zmeny, ktoré sme predtým vykonali sú stále dostupné v *Stashi*, ktorý sme vytvorili v prvom kroku tohto procesu. Zoznam dostupných stashov môžeme vidieť pod záložkou *Stashes* v ľavom bočnom menu aplikácie:

