Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Tímový projekt

Story Teller

Metodika k verziovaniu

Vedúci projektu:	Ing. Karol Rástočný, PhD.
Názov tímu:	CoolStoryBro
Členovia tímu:	Bc. Jakub Ondik
	Bc. Patrik Januška
	Bc. Adam Neupauer
	Bc. Martin Olejár
	Bc. Miroslav Hurajt
Kontakt:	storyteller-04@googlegroups.com
Vypracoval:	Bc. Martin Olejár

1 Pravidlá pre vetvenie zdrojového kódu

V rámci hlavného repozitára projektu rozlišujeme tieto vetvy (angl. branch):

- **master** – produkčná vetva pre zákazníka,
- **dev** – vývojová vetva,
- vetvy pre jednotlivé používateľské príbehy, pre každý použ. príbeh existuje 1 vetva.

1.1 Vytvorenie vetvy pre používateľský príbeh

Na používateľskom príbehu sa pracuje v osobitnej vetve pre daný používateľský príbeh. Vetvu pre používateľský príbeh vytvára ten, kto je zaň zodpovedný. Pre názov tejto vetvy platí:

- píše sa v angličtine,
- má tvar "Číslo user story.Krátky popis user story", napr. 4567.Devices,
- v prípade, že sa krátky popis používateľského príbehu skladá z viacerých slov, tieto slová sa začínajú veľkým písmenom a píšu sa spolu, napr. 4575.PasswordRecovery.

Postup pre vytvorenie vetvy prostredníctvom Team Foundation Server (TFS) je nasledujúci:

1. prihlásime sa do TFS,
2. klikneme na **CODE** v hornom menu,
3. klikneme na názov vetvy,
4. zobrazí sa okno, v ktorom klikneme na možnosť **New branch...,**
5. v ďalšom okne pre vytvorenie vetvy vyplníme jej meno podľa definovaných pravidiel a v časti **Based on** vyberieme vetvu **dev.**

1.2 Pravidlá pre prácu vo vetve

Pre prácu s vetvami platia nasledujúce pravidlá, ktoré treba dodržiavať:

- priamo do vetiev **master** a **dev** sa nikdy nič nepridáva,
- pracuje sa iba v osobitnej vetve pre daný používateľský príbeh,
- pre aktualizáciu webovej stránky sa pracuje vo vetve Global.TeamWebsite,
- pred tým, ako začíname pracovať vo vetve, je potrebné stiahnuť si (angl. **pull**) zmeny, ktoré vykonal niekto iný do tejto vetvy, čím sa predíde neskorším konfliktom,
- pred tým, ako odovzdáme zmeny, resp. dáme **commit** do vetvy, skontrolujeme funkčnosť programu a spustíme všetky jednotkové testy
    - ak všetky testy prejdú a program funguje bez chýb, je možné dať commit do vetvy, inak treba pred commitom nedostatky opraviť
- je dobré dávať commit do vetvy čo najčastejšie, aby ostatní pracujúci v tej istej vetve mohli použiť pridaný kód a aby nevzniklo veľa konfliktov pri spájaní,
- popis commitu k určitej konkrétnej úlohe píšeme po anglicky v tvare *"[Task number] Popis pridanej funkcionality a iných dôležitých vecí"*, kde tento popis môže pre lepšie vyhľadávanie obsahovať kľúčové slová ako napr. added, fixed, removed
- pri viacerých úlohách je tvar popisu nasledujúci:
  "[Tasks number1, number2] Popis pridanej funkcionality a iných dôležitých vecí"
- ak sú vypracované všetky úlohy v rámci používateľského príbehu, pridáme vetvu **dev** do našej pracovnej vetvy (angl. **merge dev into branch**)

  - ak vzniknú konflikty, treba ich vyriešiť, vykonať jednotkové testy pre overenie zachovania funkcionality a potom dať ešte raz commit do pracovnej vetvy
- po vypracovaní všetkých úloh v rámci používateľského príbehu a spojení s **dev** sa riadime podkapitolou 1.4.

1.3 Pravidlá pre spájanie vetiev

Všetky vetvy pre používateľské príbehy sú spájané po ich dokončení s vetvou **dev**. Vetva **dev** sa spája s vetvou **master** po dokončení každého šprintu.

1.4 Vytvorenie požiadavky na kontrolu používateľského príbehu

Po vypracovaní všetkých úloh v rámci používateľského príbehu, čo zahŕňa funkcionalitu, jednotkové testy, dokumentáciu modulov systému, kódu a doplnenie používateľskej príručky (ak sa dá), je nutné niekomu dať **pull request** podľa metodiky prehliadok kódu. Postup pre pull request je nasledujúci:

1. prihlásime sa do TFS,
2. klikneme na **CODE** v hornom menu,
3. klikneme na **Pull Requests** v menu,
4. klikneme vpravo na **New pull request,**
5. najprv vyberieme názov našej vetvy, ktorú je potrebné pridať do vetvy **dev**, napr.:
   Review changes in 4567.Devices relative to **dev**
6. vyplníme názov a v časti **Reviewers** vyberieme niekoho z tímu podľa metodiky prehliadok kódu,
7. klikneme na **New pull request** v dolnej časti.

V prípade, že chceme aktualizovať webovú stránku, používame vetvu Global.TeamWebsite, ktorú je potrebné pridať do vetvy **master**.

2 Používanie Git-u na verziovanie

V tímovom projekte je možné používať verziovací systém Git. Pre ďalšie časti tejto metodiky je potrebné si ho nainštalovať.

2.1 Inicializácia lokálneho repozitára

Pre pridávanie kódu a súborov do projektu je potrebné si vytvoriť lokálny repozitár. Postup sa skladá z nasledujúcich krokov:

1. vytvoríme nový priečinok kdekoľvek vo svojom počítači,
2. nastavíme sa v príkazovom riadku do nového priečinka a spustíme príkaz *git init,*
3. pre naklonovanie projektu do priečinka spustíme príkaz

   git clone https://tfs.fiit.stuba.sk:8443/tfs/StudentsProjects/_git/StoryTeller

2.2 Odovzdávanie zmien

Po úprave kódu alebo pridaní súborov do projektu skontrolujeme funkčnosť projektu a spustíme všetky jednotkové testy. Ak všetky testy prejdú a projekt je funkčný, môžeme odovzdať (angl. commit) aktuálnu verziu repozitára do vetvy, aby si ostatní členovia tímu mohli stiahnuť pridané časti projektu. Postup sa skladá z nasledujúcich krokov:

1. nastavíme sa v príkazovom riadku do priečinka StoryTeller, ktorý sa nachádza v našom vytvorenom priečinku pre lokálny repozitár,
2. spustíme príkaz *git add .*
3. spustíme príkaz *git commit -m "popis"* – popis odovzdania vyplníme podľa definovaných pravidiel, napríklad:

   git commit -m "[Task 4330] Added Angular controller for user profile with basic functions, removed useless dependencies"
4. spustíme príkaz *git push -u origin <názov vetvy>*.

2.3 Stiahnutie aktuálnej verzie vetvy

Pre stiahnutie aktuálnej verzie vetvy do lokálneho repozitára je postup nasledovný:

1. nastavíme sa v príkazovom riadku do priečinka StoryTeller, ktorý sa nachádza v našom vytvorenom priečinku pre lokálny repozitár,
2. spustíme príkaz *git pull.*

2.4 Zistenie aktuálneho stavu

Git umožňuje zobraziť stav lokálneho repozitára – 1 príkazom môžeme vidieť, ktoré súbory spolu s ich umiestnením sme pridali, modifikovali alebo odstránili. Postup je nasledovný:

1. nastavíme sa v príkazovom riadku do priečinka StoryTeller, ktorý sa nachádza v našom vytvorenom priečinku pre lokálny repozitár,
2. spustíme príkaz *git status.*

2.5 Riešenie problému konfliktov

Niekedy sa môže stať, že odovzdanie zmien v zdrojovom kóde nie je možné, pretože sa našli konflikty. Vtedy je nutné použiť nasledujúce príkazy:

1. *git stash*
2. *git pull*
3. *git stash apply*

V prípade, že je stále problém, treba vyriešiť priamo v kóde konflikty, ktoré sa do kódu dopísali. Po ich vyriešení zadáme príkazy:

1. *git add .*
2. *git merge*

V prípade úspechu dáme *git stash drop.*

3 Nástroj na prácu s vetvami

Po úvodných problémoch s Gitom sme začali používať nástroj SourceTree, ktorý ponúka prehľadné zobrazenie vetiev projektu a možnosti pre ich spájanie.

3.1 Inicializácia lokálneho repozitára

Postup pre inicializácia lokálneho repozitára, teda naklonovanie globálneho repozitára do priečinka v počítači, je nasledujúci:

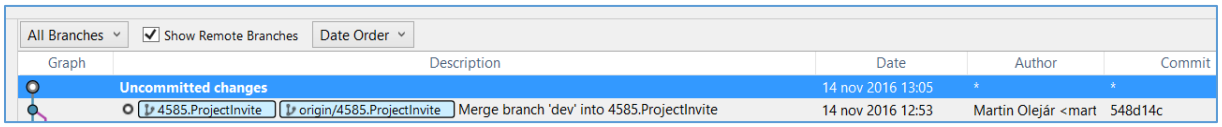
1. v SourceTree klikneme v hornom menu na **Clone / New,**
2. vyplníme **Source Path / URL** nasledujúcim odkazom:
   https://tfs.fiit.stuba.sk:8443/tfs/StudentsProjects/StoryTeller/_git/StoryTeller
3. vyplníme **Destination Path**, t.j. cestu k priečinku, kde bude uložený lokálny repozitár,
4. potvrdíme tlačidlom **Clone.**

3.2 Práca vo vetve

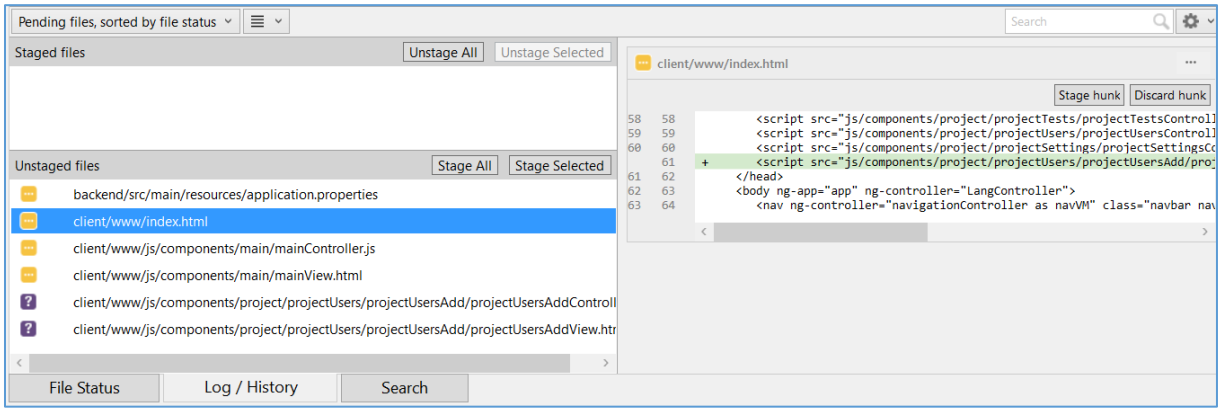
Pracovnú vetvu si otvoríme tak, že v časti BRANCHES klikneme dvakrát na danú vetvu. Pri úplne prvom otváraní vetvy je nutné túto akciu vykonať v časti REMOTES.

Tlačidlo **Fetch** v hornom menu slúži na aktualizáciu zobrazovaných commitov v nástroji. Toto tlačidlo treba stlačiť vždy, keď začíname pracovať, alebo priebežne pre aktuálne zobrazenie. Takto môžeme zistiť, či niekto iný vykonal commit v našej pracovnej vetve.

Všetky zmeny, ktoré vykonáme vo vetve, si môžeme v nástroji pozrieť. Najprv klikneme na **Uncommitted changes** podľa obr. 1. V dolnej časti (na obr. 2) sa nám zobrazia všetky súbory, ktoré sme modifikovali. Po kliknutí na nejaký súbor vidíme vpravo všetky zmeny, ktoré sme v tomto súbore vykonali.



Obrázok 1: Zoznam vykonaných commitov

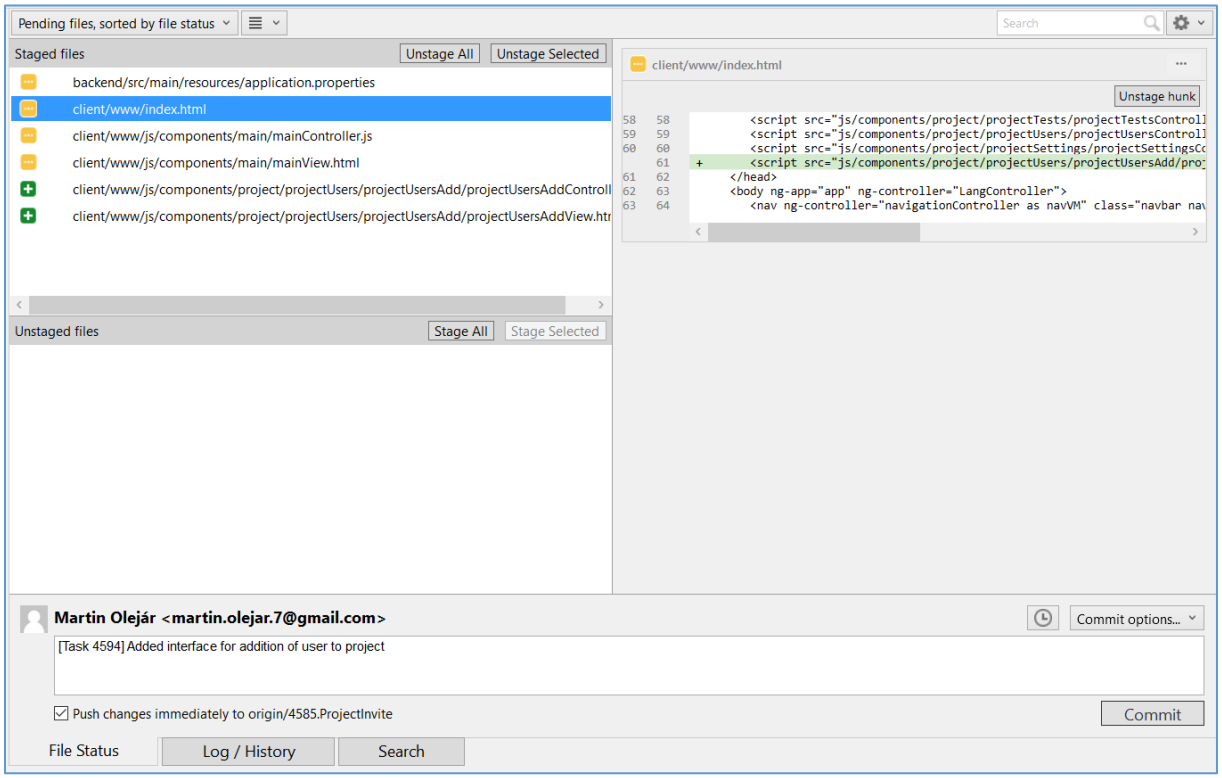


Obrázok 2: Zobrazenie zmien vo vetve

3.3 Odovzdávanie zmien

Po úprave kódu alebo pridaní súborov do projektu skontrolujeme funkčnosť projektu a spustíme všetky jednotkové testy. Ak všetky testy prejdú a projekt je funkčný, môžeme odovzdať (angl. commit) aktuálnu verziu repozitára do vetvy, aby si ostatní členovia tímu mohli stiahnuť pridané časti projektu. Postup sa skladá z nasledujúcich krokov:

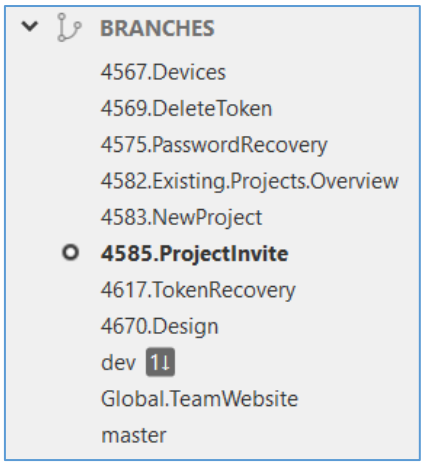
1. klikneme v dolnej časti na **File Status,**
2. v časti **Unstaged files** uvidíme všetky modifikované súbory, klikneme na **Stage All,**
3. v dolnej časti vyplníme popis commitu podľa definovaných pravidiel,
4. zaškrtneme **Push changes immediately to ...,**
5. výsledná situácia je zobrazená na obr. 3, môžeme si tiež prezerať vykonané zmeny v jednotlivých súboroch
6. pre potvrdenie commitu klikneme na **Commit.**



Obrázok 3: Odovzdanie zmien

3.4 Stiahnutie aktuálnej verzie vetvy

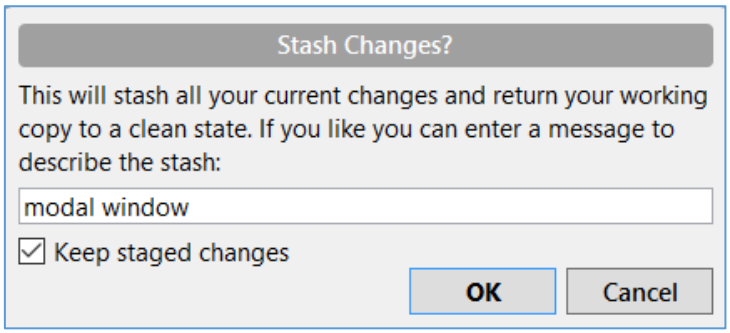
Ak niekto iný vykonal zmeny vo vetvách, je potrebné si tieto zmeny stiahnuť. Najprv klikneme na **Fetch** v hornom menu. Pri každej vetve, ktorá nie je lokálne aktuálna, teda niekto iný ju modifikoval a vykonal commit a jeho zmeny nie sú v lokálnom repozitári, sa zobrazí počet vykonaných commitov a šípka dole ako na obr. 4. Stiahnutie aktuálnej verzie vetvy vykonáme tak, že sa prepneme do danej vetvy a klikneme na tlačidlo **Pull** v hornom menu.



Obrázok 4: Zobrazenie aktuálnych vetiev

3.5 Odloženie vykonaných zmien vo vetve

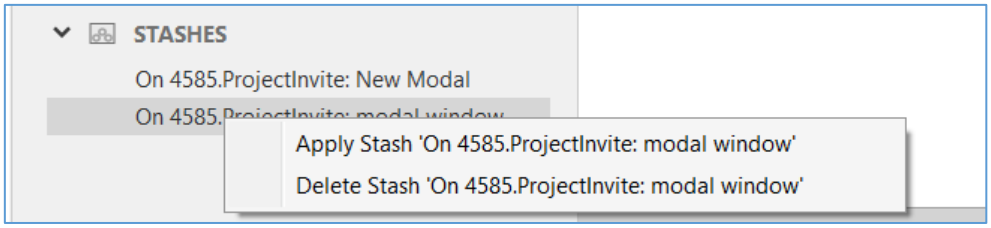
Ak pracujeme v nejakej vetve a potrebujeme vykonať nejaké zmeny v inej vetve, teda musíme sa prepnúť do inej vetvy, je potrebné si predtým vykonané zmeny odložiť (angl. **stash**). V hornom menu klikneme na **Stash**, zobrazí sa modálne okno na obr. 5, v ktorom vyplníme správu pre popis stash-u a zaškrtneme **Keep staged changes**. Potvrdíme kliknutím na **OK**.



Obrázok 5: Odloženie zmien

Potom sa môžeme prepnúť do inej vetvy a vykonávať v nej ľubovoľné zmeny aj robiť commity. Zmeny, ktoré sme si odložili pomocou stash, si môžeme vrátiť naspäť do vetvy. Postup je nasledujúci:

1. prepneme sa do danej vetvy, t.j. dvakrát klikneme na názov vetvy,
2. v časti STASHES (na obr. 6) klikneme pravým tlačidlom na názov stash-u (v našom prípade *modal window*) a klikneme na **Apply Stash...**



Obrázok 6: Aplikovanie stash-u