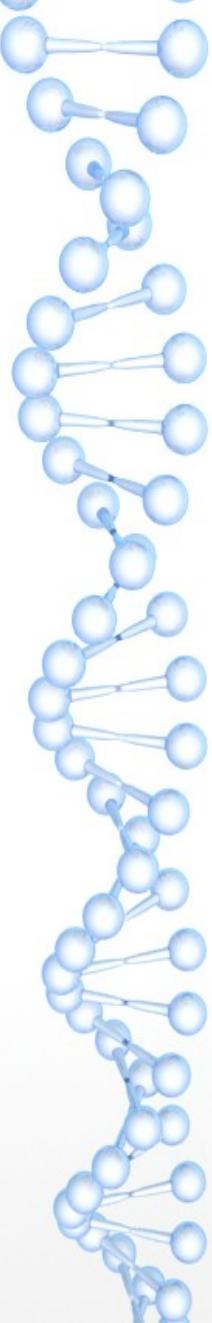


# GIT

## GIT primer

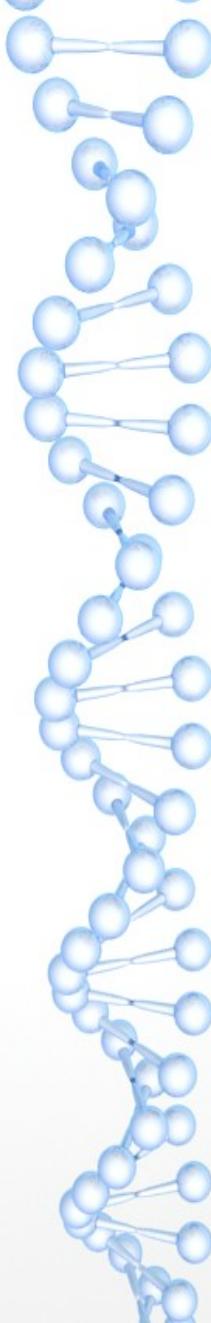
*Ako na GIT aj pre  
úplných začiatočníkov*

Miloš Korenčiak (MojaKomunita)



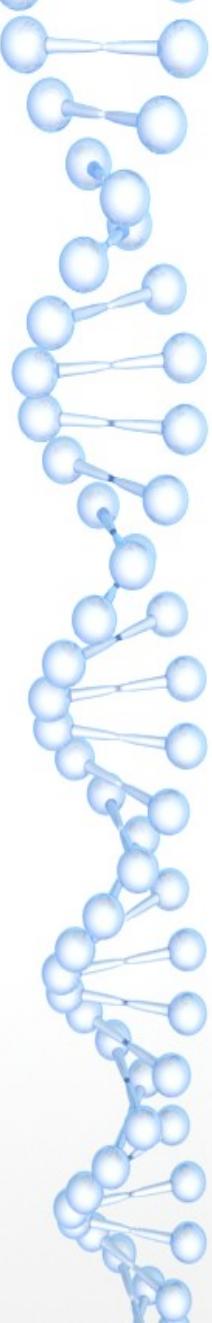
# Obsah

- 1. Čo (ne)urobí GIT za mňa?
- 2. Čo je vlastne GIT zač?
- 3. Tri stromy/databázy v GIT
- 4. Hands on lab (praktická ukážka toho najbežnejšieho na konzole)
- 5. Rôzne GUI pre GIT, aby sme to mali jednoduchšie
- 6. Sumár
- 7. Záver
- 8. Zdroje informácií



# 1. Čo (ne)urobí GIT za mňa?

- Neurobí za mňa prácu
- Nevyrieši komunikáciu v tíme ani nezhody v návrhu programu
- Nevyrieši každú kolíziu s kolegovým kódom
- Nevyrieši kóderove zabúdanie zálohovať /verzovať svoj program
- Manžérovi nepomôže zistit', či sa projekt stíha

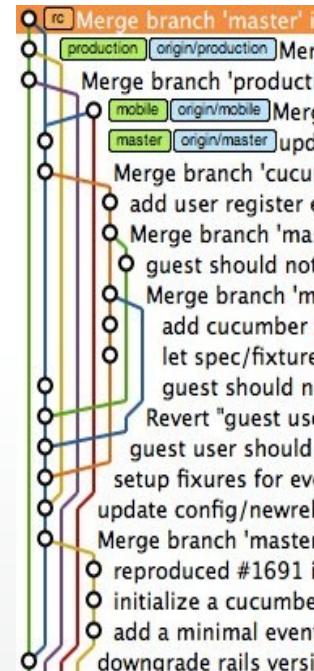
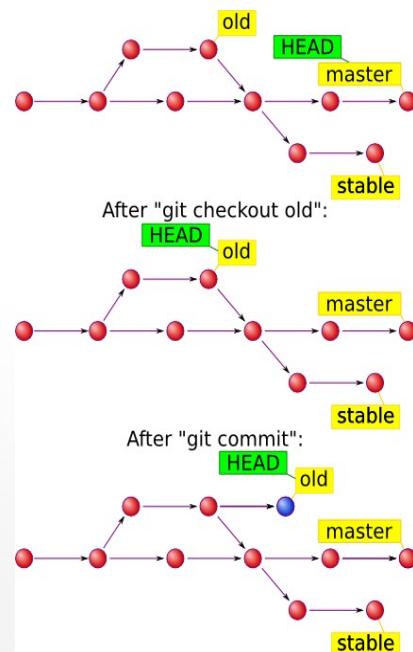


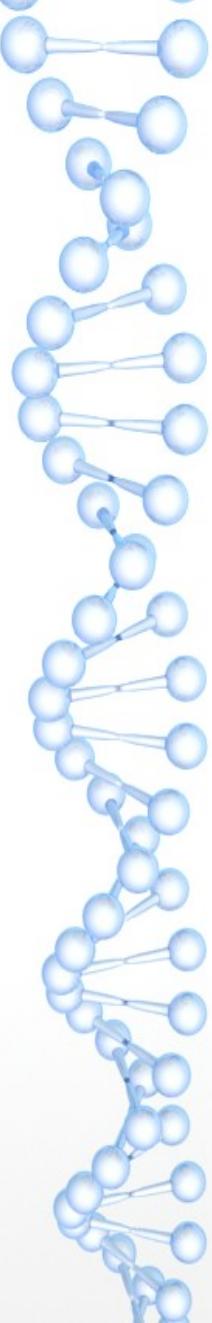
## 2. Čo je vlastne GIT zač?

- *a distributed revision control and source code management (SCM) system with an emphasis on speed*  
*(Wikipedia)*
- Umožňuje manažment kódu (SCM)
- Verzuje zdrojový kód
- Je distribuovaný
- Je rýchly :-) (za cenu...)
- Umožňuje mať rozrobených viac vecí naraz +prepínanie

# 2. Čo je vlastne GIT zač?

- Je to vlastne „strom“ commitov (check-point kódu)
- Priponá „stanice metra“
- Je uložený v internej databáze

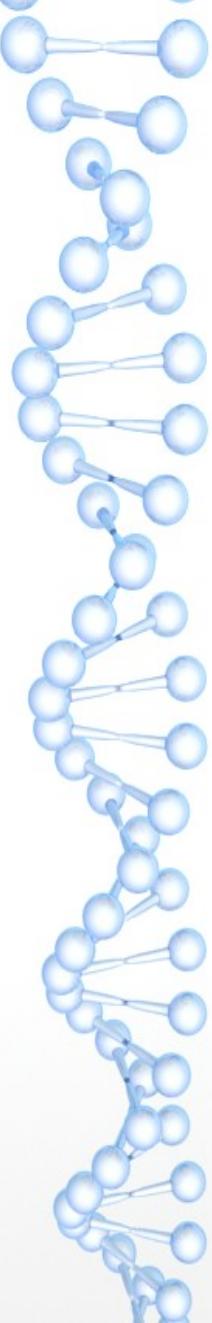




## 2. Čo je vlastne GIT zač?

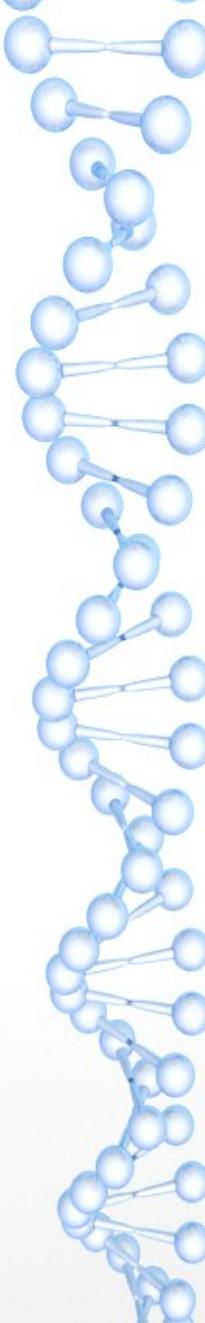
**Software tools for revision control are essential for the organization of multi-developer projects.**

([Wikipedia](#))



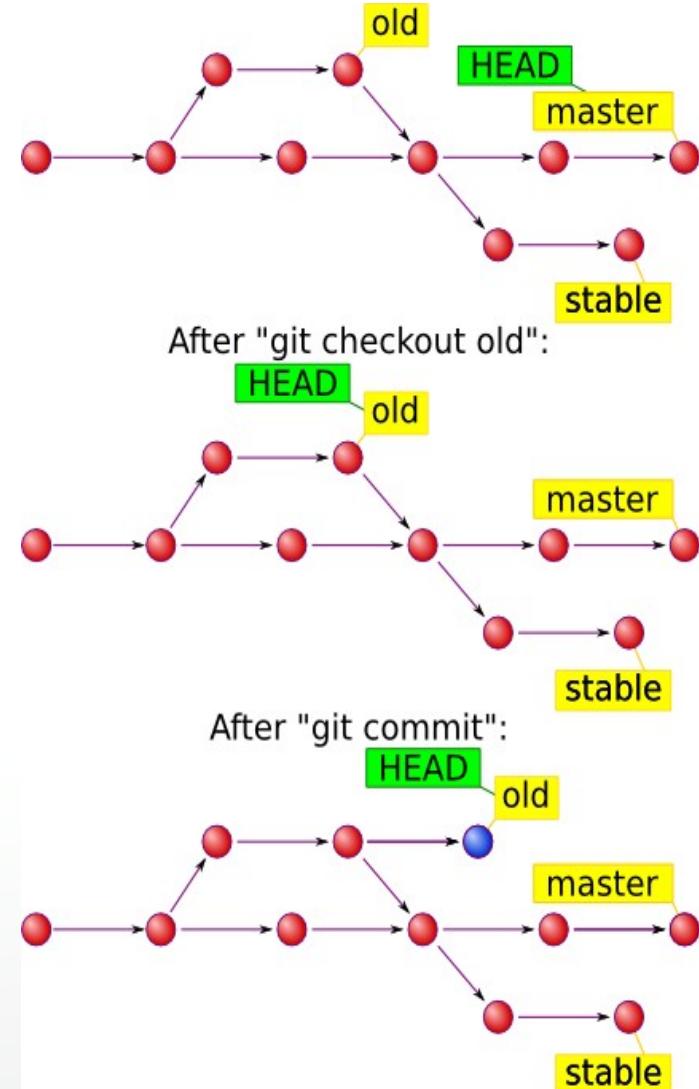
### 3. Tri stromy/databázy v GIT -working dir

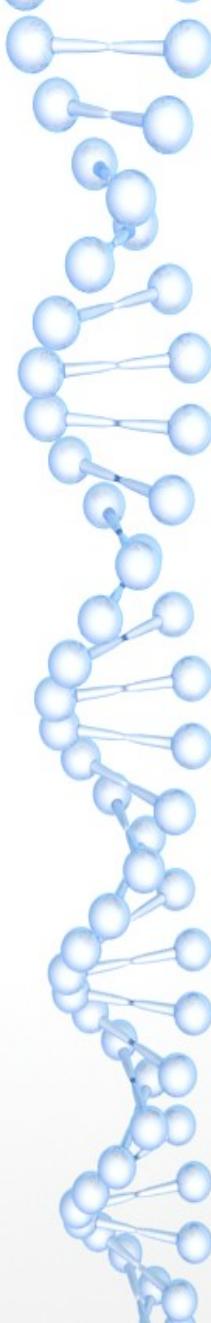
- Ten adresár v ktorom priamo programujeme
- Je v ňom teda fyzicky každý súbor projektu
- GIT prehľadáva aktuálny adresár a rekurzívne aj všetky podadresáre
- Zistuje, ktoré súbory oproti poslednému commitu pribudli (a potenciálne by sa mali pridať do repozitára), a ktoré súbory v repozitári (tzv. verzované /tracked súbory) sa zmenili
- GIT ho edituje pri checkout-e



### 3. Tri stromy/databázy v GIT -HEAD

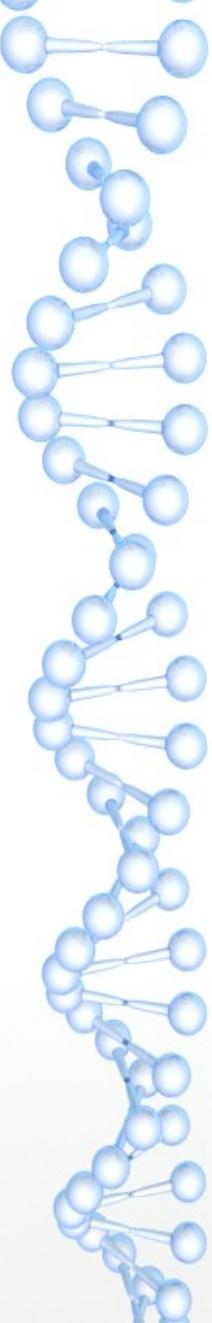
- Je to posledný commit v aktuálnej vetve „stromu“ (strom je celý GIT repozitár)
- Ak urobíme commit, stane sa synom aktuálneho HEAD commitu; HEAD ukazovateľ sa potom posunie na náš najnovší commit
- Ukazuje kde pracujeme





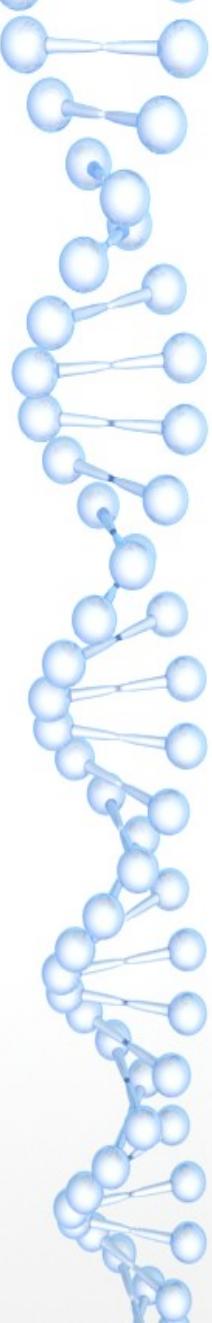
### 3. Tri stromy/databázy v GIT -index

- Je to miesto prípravy commitu (akýsi medzisklad)
- Pridávame tu zmeny, ktoré chceme zahrnúť do najbližšieho commitu
- Môžme tu pridať všetky zmeny zo súbora alebo len ich časť
- Môžeme pridať jeden alebo viac súborov
- Využívajú ho najmä GUI nástroje



### 3. Tri stromy/databázy v Git -ich spolupráca

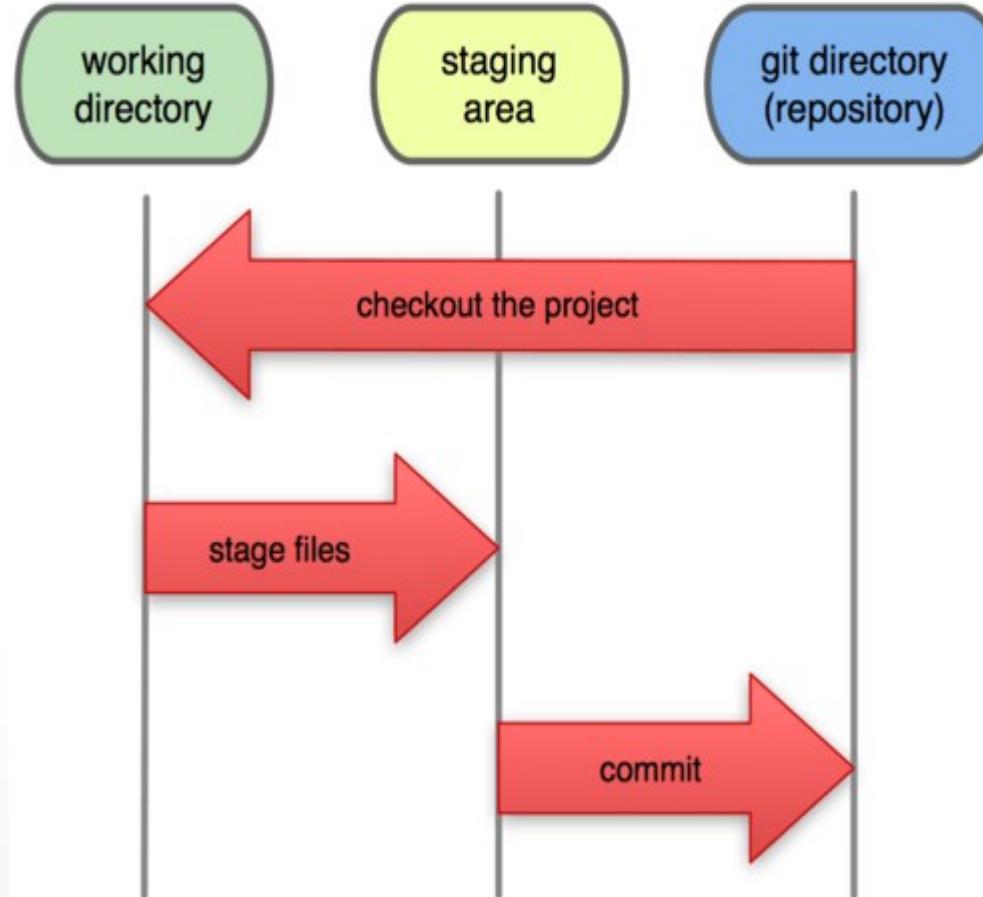
- Programovaním meníme working dir
- Príkazom git add (bude ukázaný neskôr) prenášame working dir zmeny do indexu
- Príkazom git commit (neskôr) prenášame index alebo aj working dir do stromu repozitára (až ten je verzovaný)
- IBA strom GIT repozitára je schopný priamo komunikovať so vzdialenosťmi GIT repozitármi

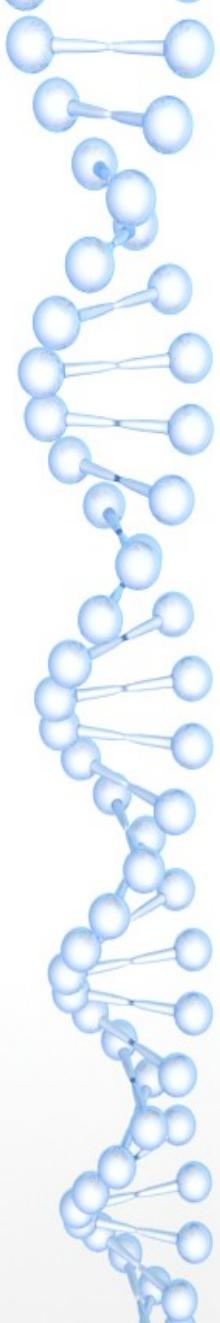


# 3. Tri stromy/databázy v Git -ich

## spoluhraná

### Local Operations

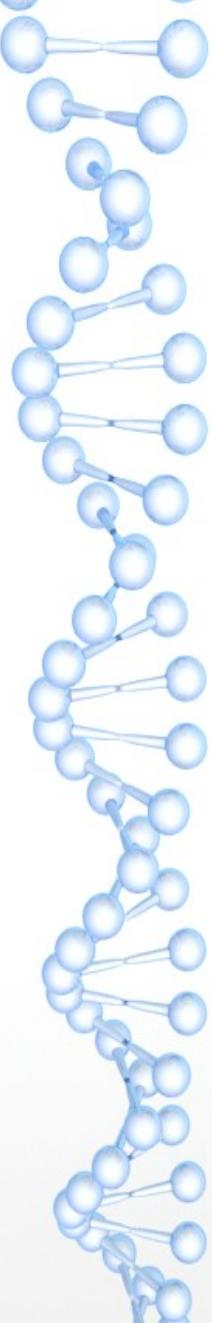




## 4. Hands on lab

(praktická ukážka v *git bash* konzole)

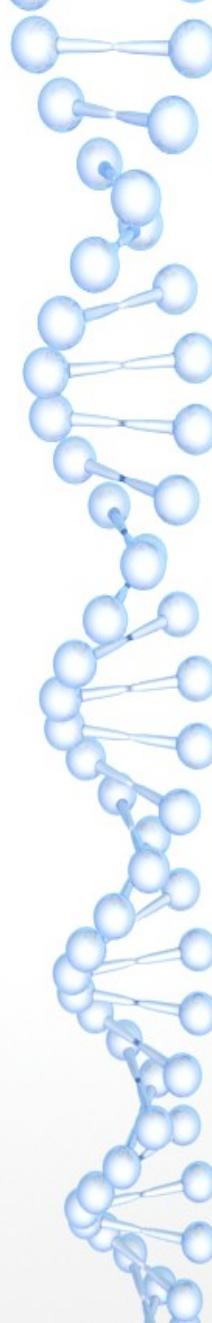
Predved'me si to na jednoduchom  
projekte ...



## 4.a) Vytvorenie repozitára

*\$ git init*

- Vytvorí v aktuálnom adresári lokálny git repozitár
- Jeho všetky dáta sú v jedinom *.git* skrytom podadresári



Stroj View Zariadenia Pomocník

Git Extensions

Start Dashboard Github Plugins Tools Help

Branches: Filter:

**Common Actions**

- [Open repository](#)
- [Clone repository](#)
- [Clone SVN repository](#)
- [Clone Github repository](#)
- [Create new repository](#)

**Recent Repositories**

Initialize new repository

Directory: G:\aa

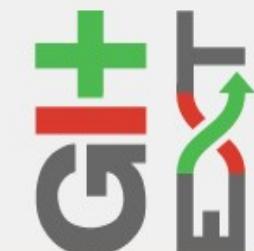
Repository type:

Personal repository

Central repository, no working dir (--bare --shared=all)

**Contribute**

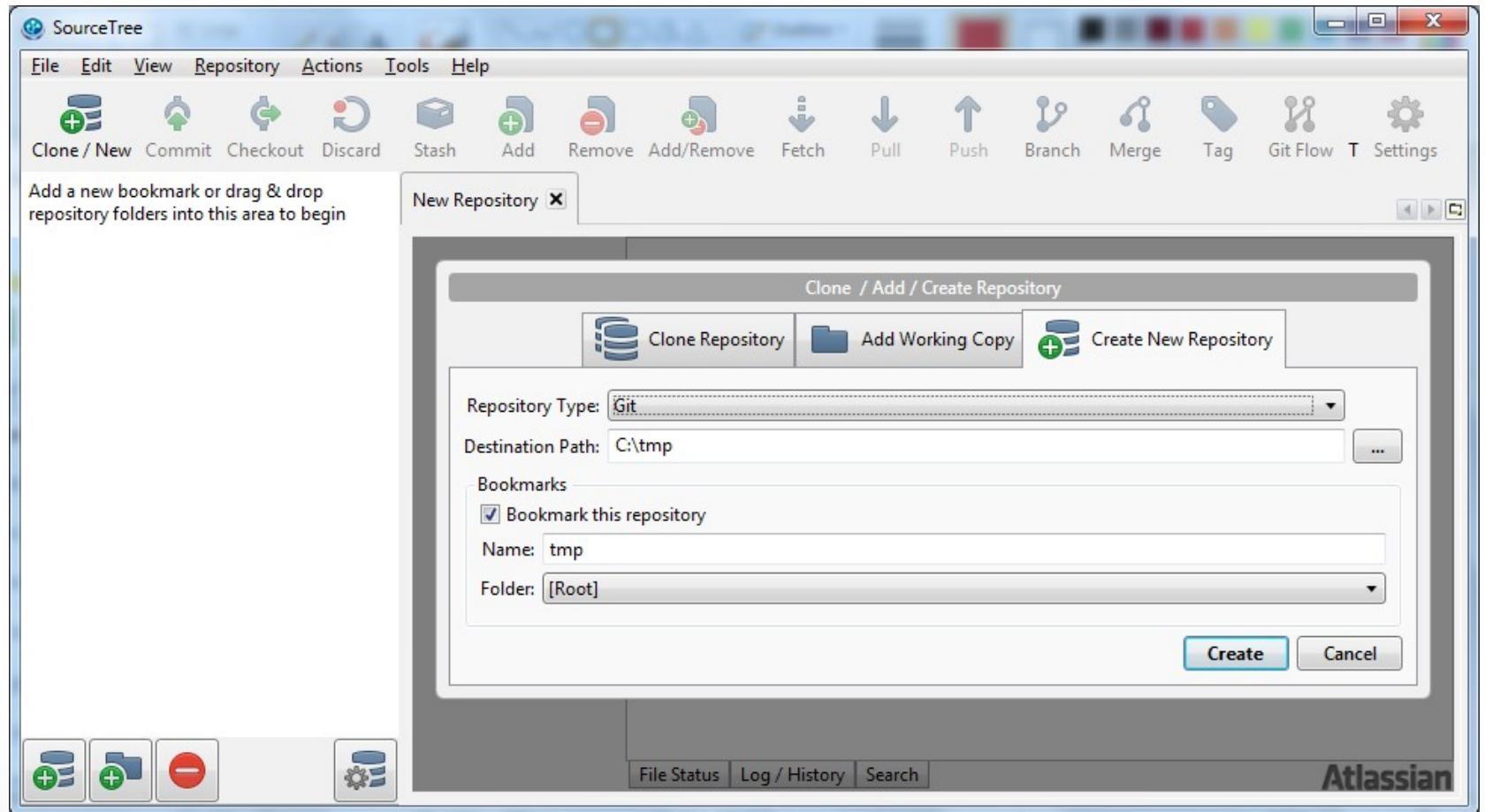
- [Develop](#)
- [Donate](#)
- [Translate](#)
- [Issues](#)

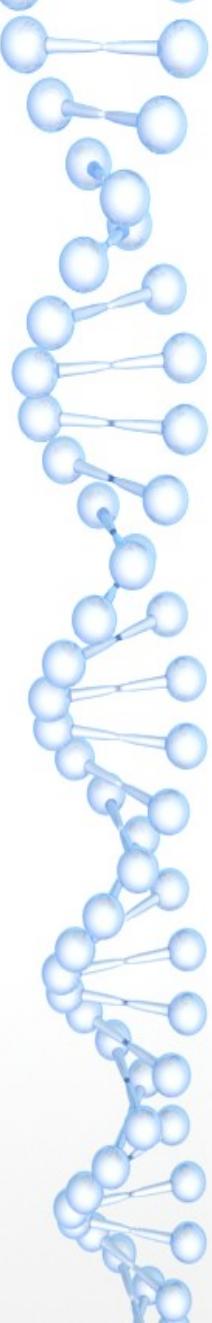


1:17 PM  
3/25/2014

14

Pravý ctrl





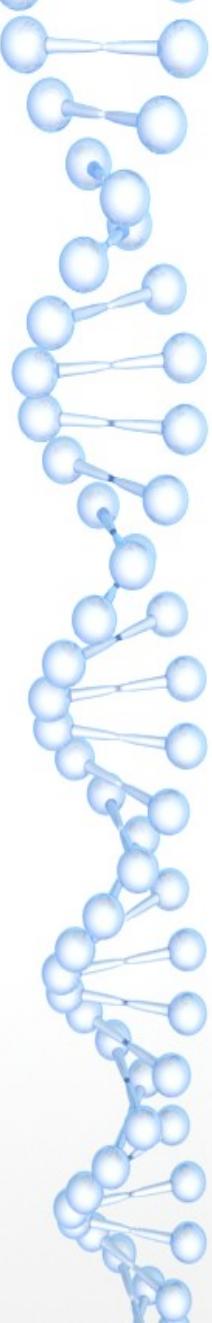
## 4.b) Stiahnutie repozitára

```
$ git clone https://github.com/vishnevskiy/battlenet.git  
$ git clone /home/m/kodenie/effective-django-tutorial
```

- Naklonuje repozitár z daného URI
- Zvyčajne sa používa zabezpečený protokol
- Zvyčajne tento protokol vyžaduje autorizáciu pri posielaní zmien pri budúcom *push*

- My použijeme Git tutoriál pre angularJS:  
(keywords do google):

# git angularjs tutorial phonecat



**GitHub** This repository Search Explore Features Enterprise Blog Sign up Sign in

angular / angular-phonecat ★ Star 1,112 Fork 2,138

Tutorial on building an angular application. <http://docs.angularjs.org/tutorial>

212 commits 4 branches 13 releases 38 contributors

branch: master angular-phonecat / +

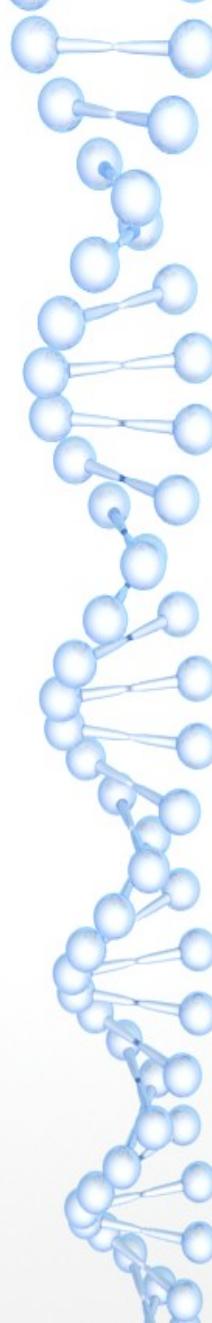
step-12 animations	3 days ago
petebacondarwin authored 3 days ago	latest commit fcb334371a
app step-12 animations	3 days ago
scripts chore(test-all): fix path to karma binary	3 months ago
test step-12 animations	3 days ago
.bowerrc chore(bower): ensure that bower doesn't try to interact with the user	4 months ago
.gitignore chore(bower): use bower for dependencies	7 months ago
.jshintrc chore(jshint): add protractor globals	7 months ago
.travis.yml chore(travis): firefox starts with an uppercase F	7 months ago

Code Issues Pull Requests Pulse Graphs

HTTPS clone URL <https://github.com/> You can clone with HTTPS or Subversion.

Clone in Desktop Download ZIP





Stroj View Zariadenia Pomocník

aa (master) - Git Extensions

Start Repository Navigate View Commands Github Plugins Tools Help

G:\aa\ master Commit Branches Filter:

▶ master hahaha babbbrak\_poradny 1 minute ago

**Clone**

Repository to clone: G:\aa

Destination: G:\bb

Subdirectory to create: aa

Branch: master

The repository will be cloned to a new directory located here:  
G:\bb\aa (New directory)

Repository type

Personal repository

Public repository, no working dir (--bare)

Initialize all submodules

Clone

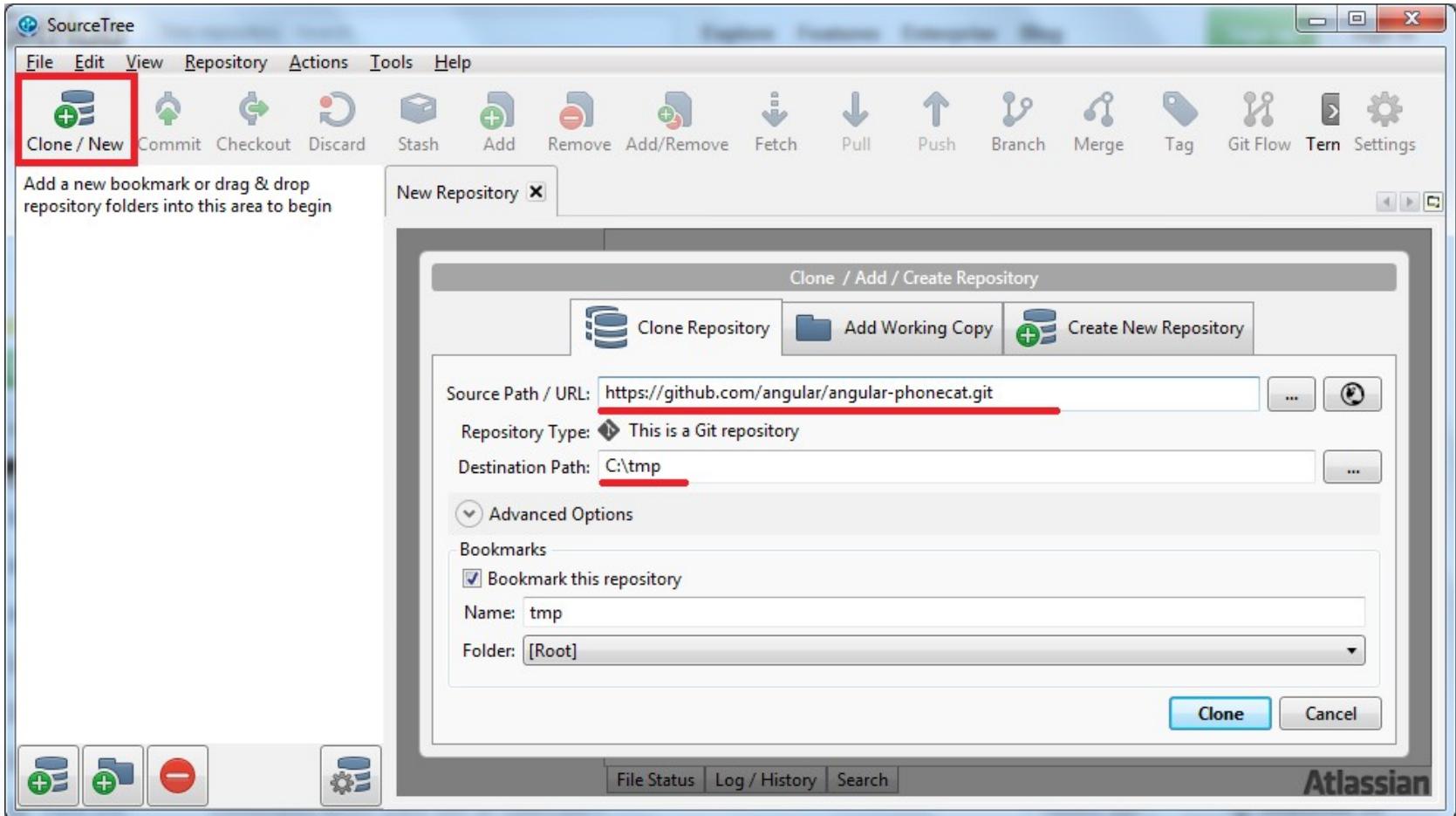
Contained in branches: [master](#)  
Contained in no tag

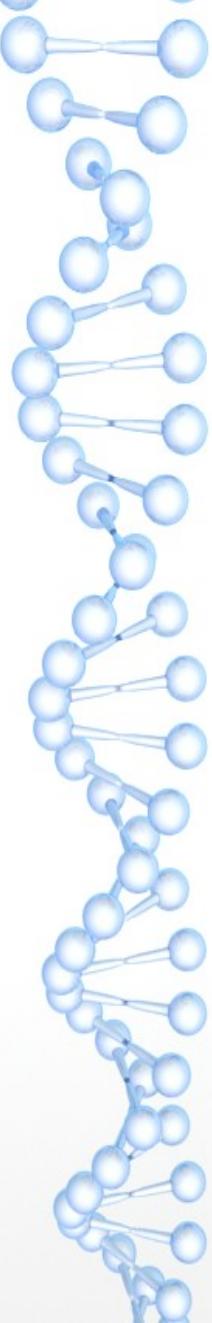
1:18 PM 3/25/2014

18

Pravý ctrl

Windows Taskbar icons: Start, Internet Explorer, File Explorer, Media Player, Firefox, Git Extensions.





## 4.c) Commit +Add

*\$ git status*

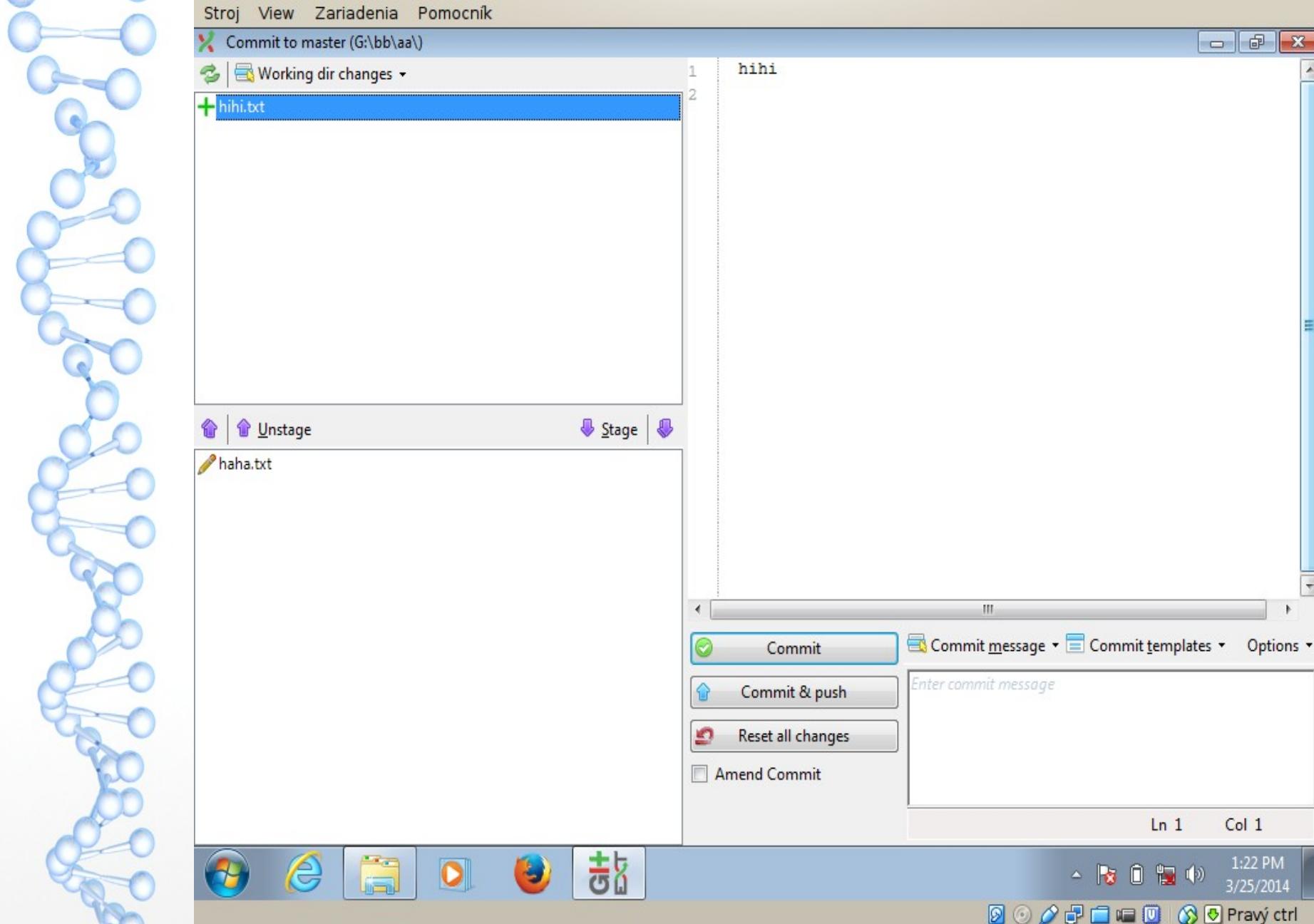
*\$ git add haha.txt*

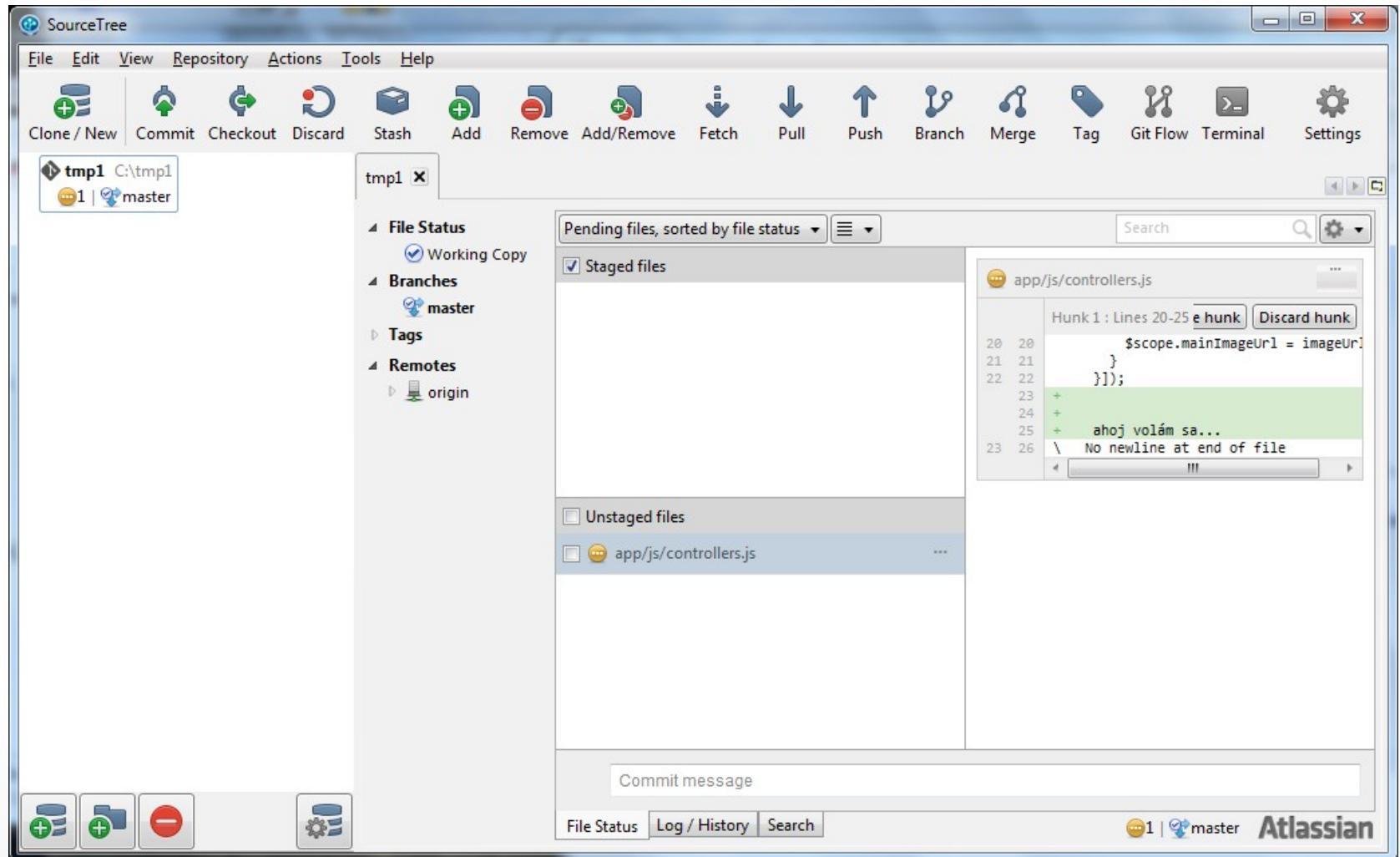
*\$ git commit*

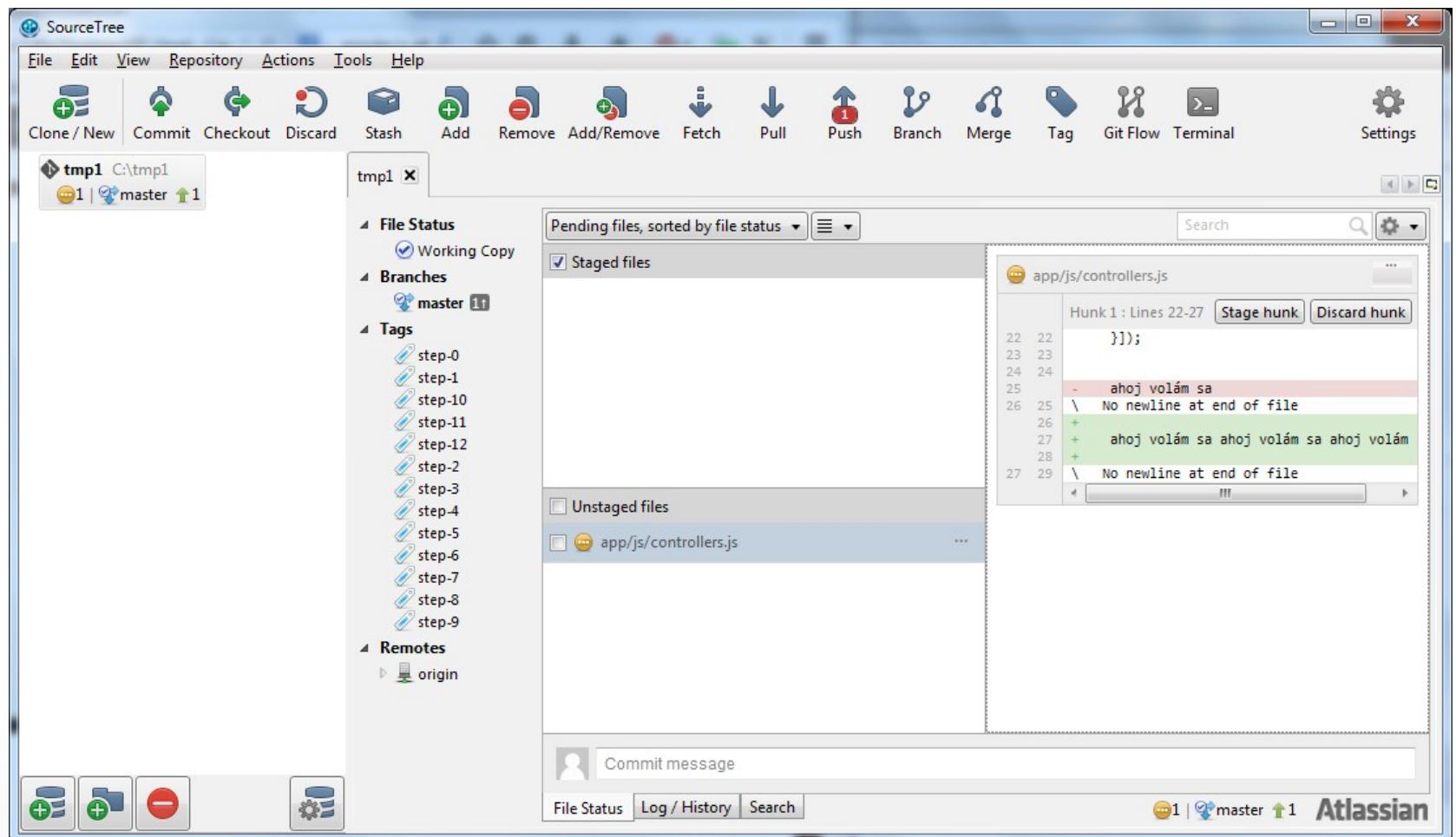
*\$ git commit -a*

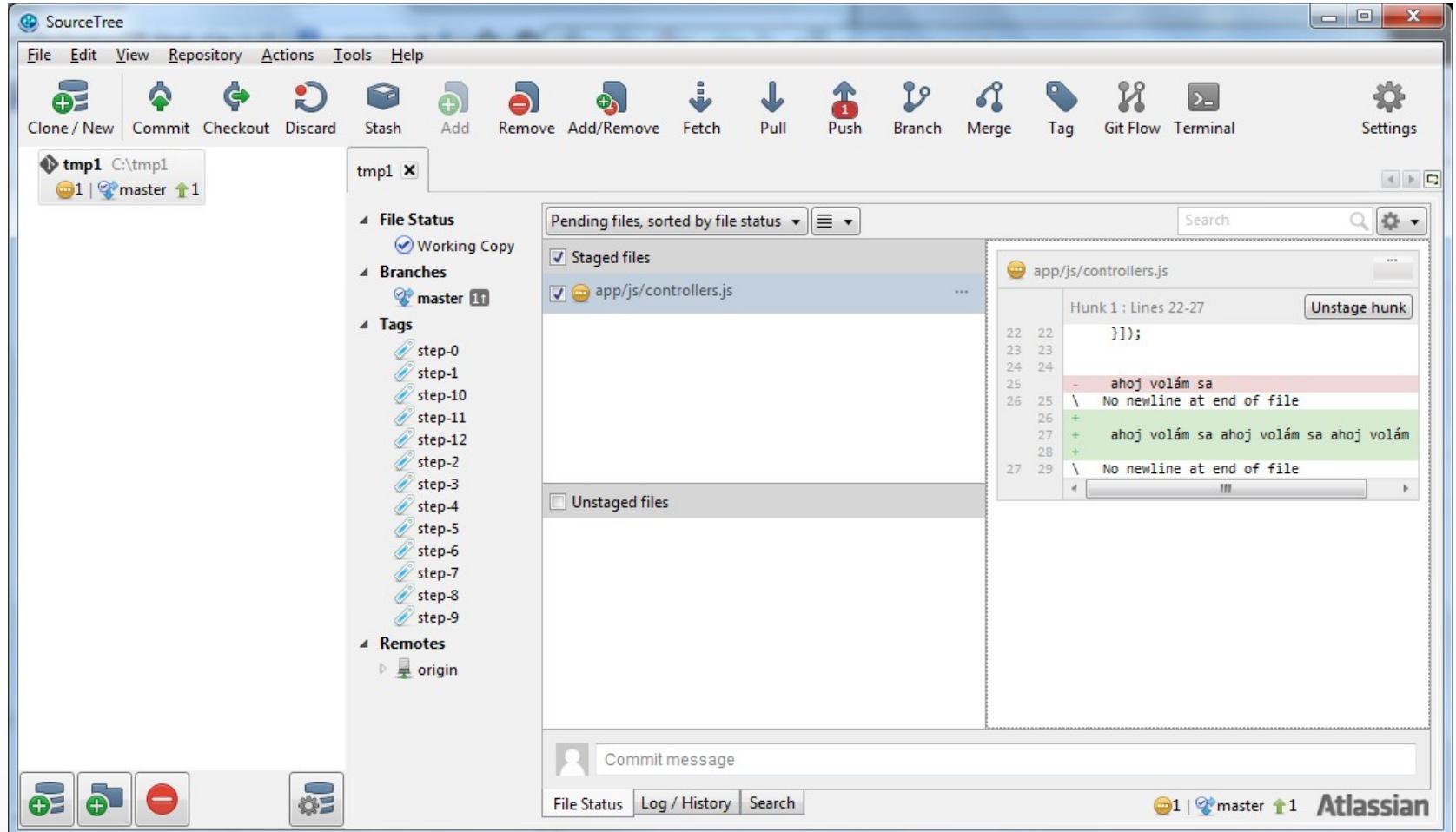
*\$ git commit -a -m „ahoj -som popis commitu“*

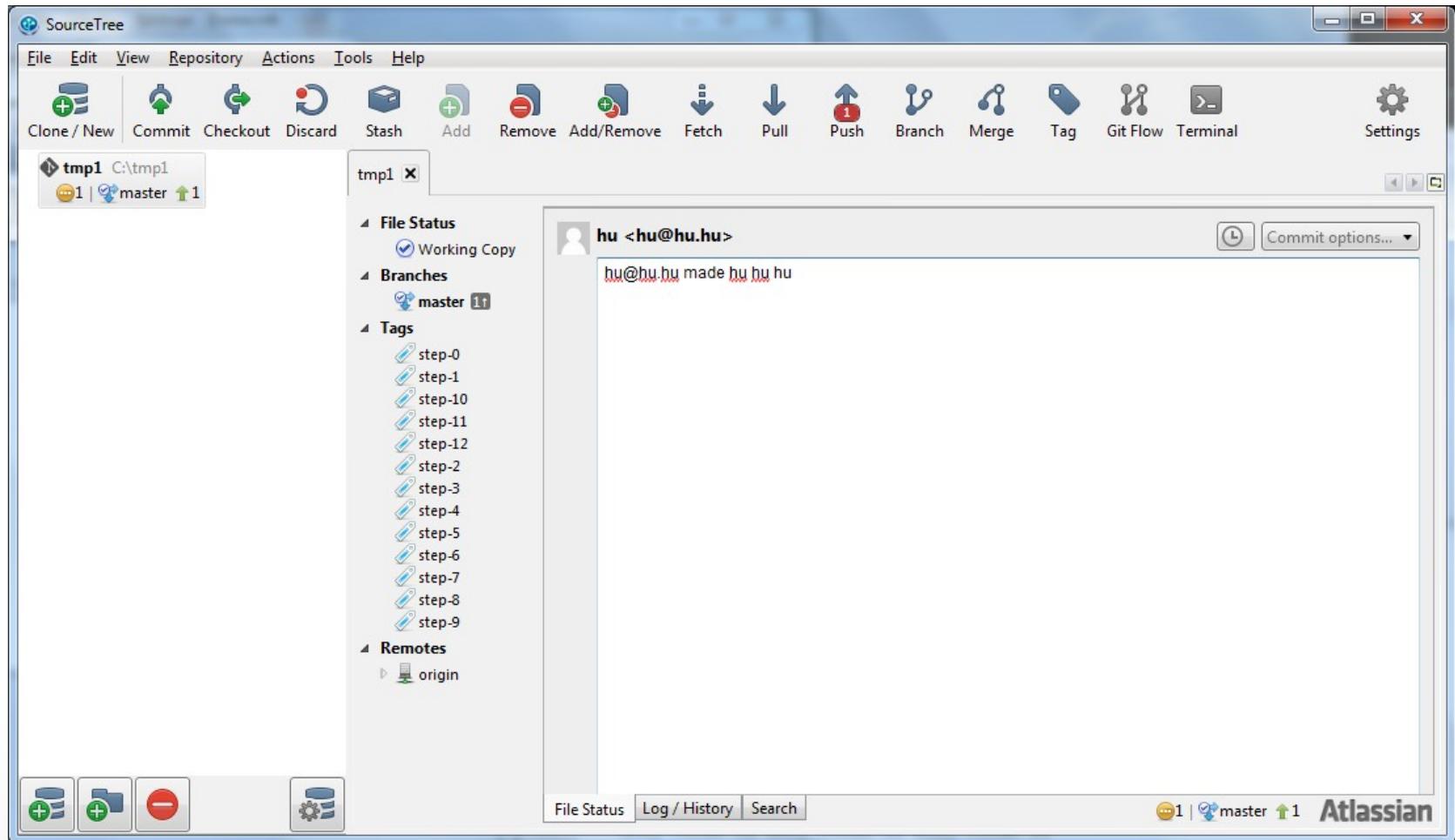
- Preskúmame stav repozitára
- Pridáme súbor, aby sa oň git staral
- Commit-neme jeho zmeny do git-u („Tri stromy git-u“)

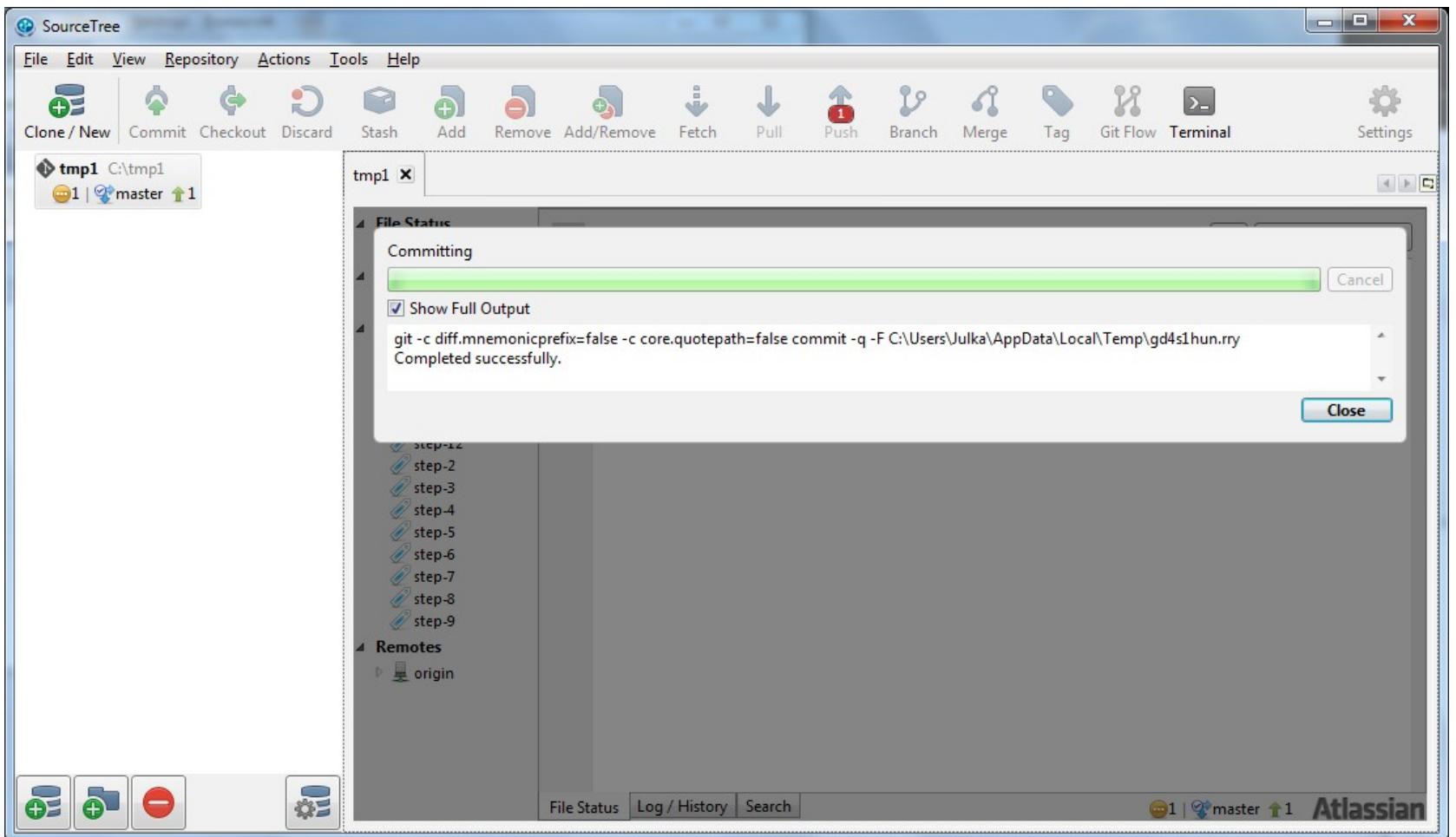


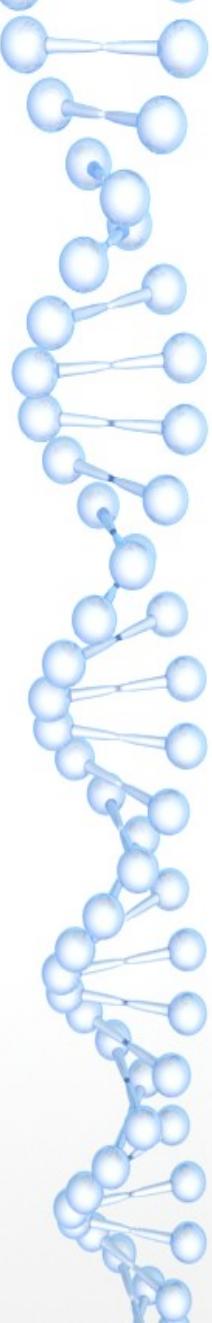








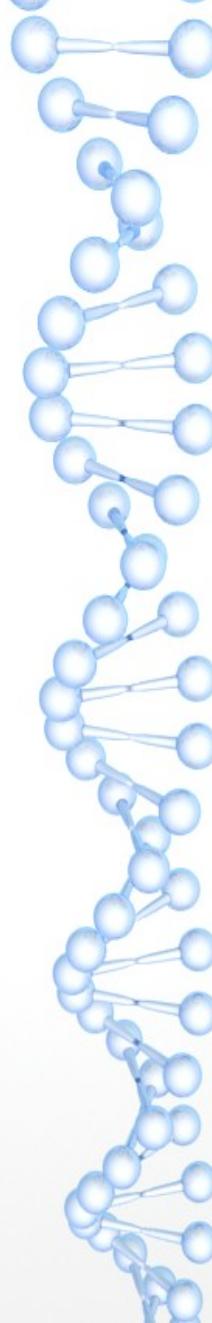




## 4.d) Push

*\$ git push*

- Pošle naše commity do vzdialeného repozitára
- Ak nemáme nijaké nové dátá, sme o tom informovaní
- Môže skončiť neúspešne (povieme pri Merge-ovani)



Stroj View Zariadenia Pomocník

aa (master) - Git Extensions

Start Repository Navigate View Commands Github Plugins Tools Help

G:\bb\aa\ master Commit Branches: Filter:

► master hihi added babbbrak\_poradny 4 seconds ago  
origin/master hahaha babbbrak\_poradny 7 minutes ago

**Push (G:\bb\aa\)**

Push to:  
 Remote origin   
 Url G:/aa

Push branches Push tags Push multiple branches

Branch:  
Branch to push [All] to

Pull Push

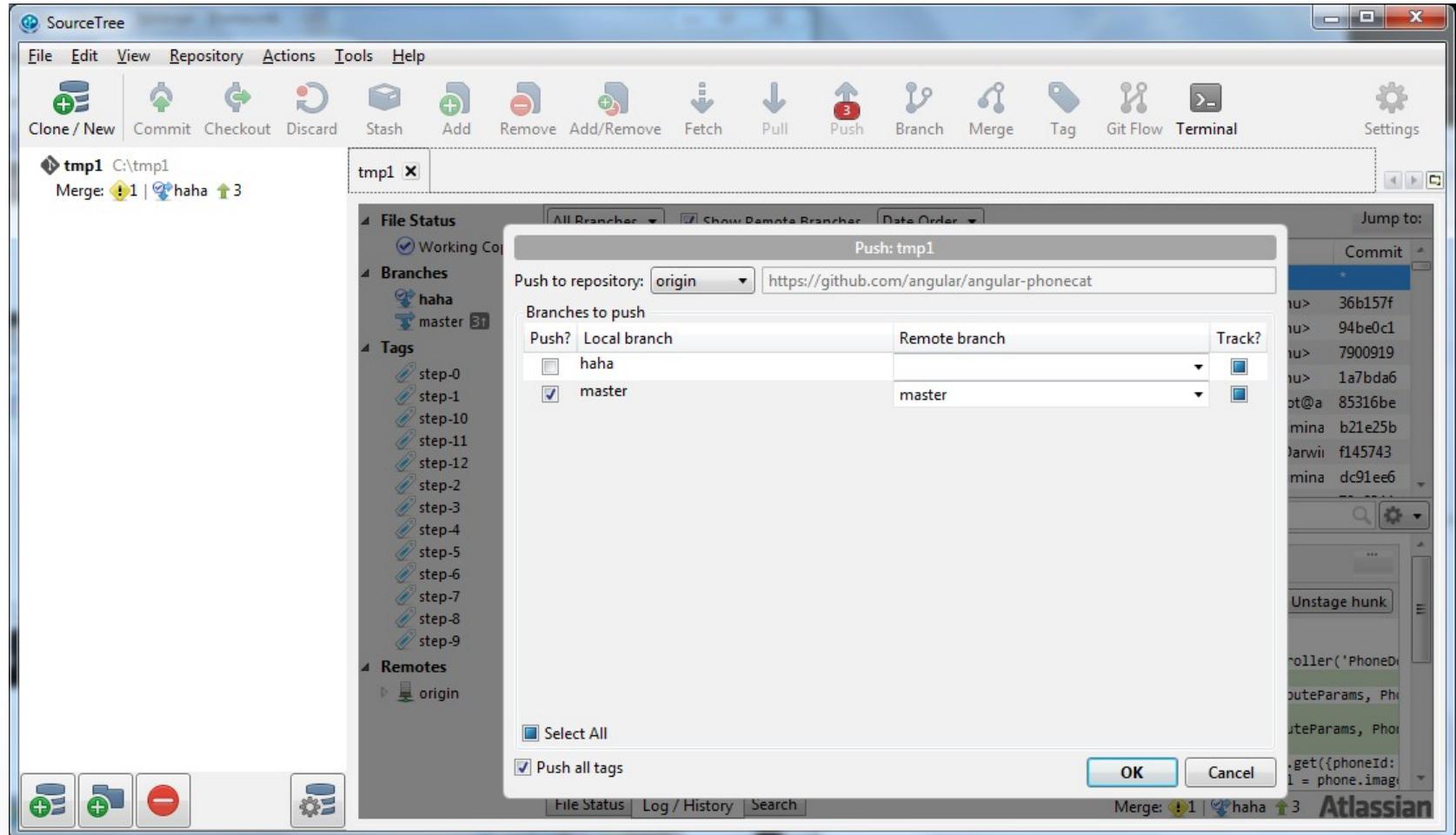
Commit File Author: Date: Commit Parent(s):  
hihi added

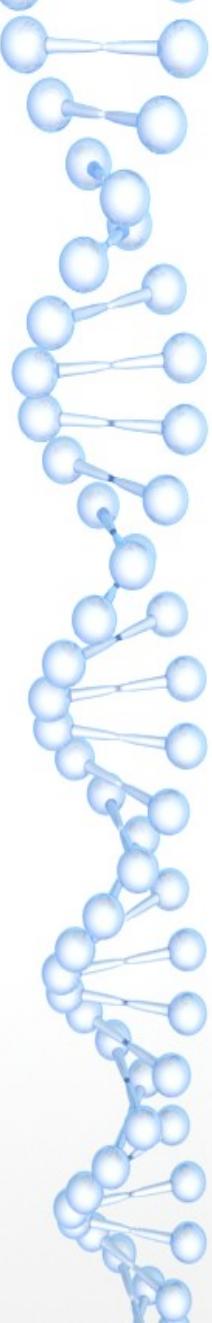
Contained in branches: [master](#)  
Contained in no tag

1:23 PM 3/25/2014

28

Pravý ctrl

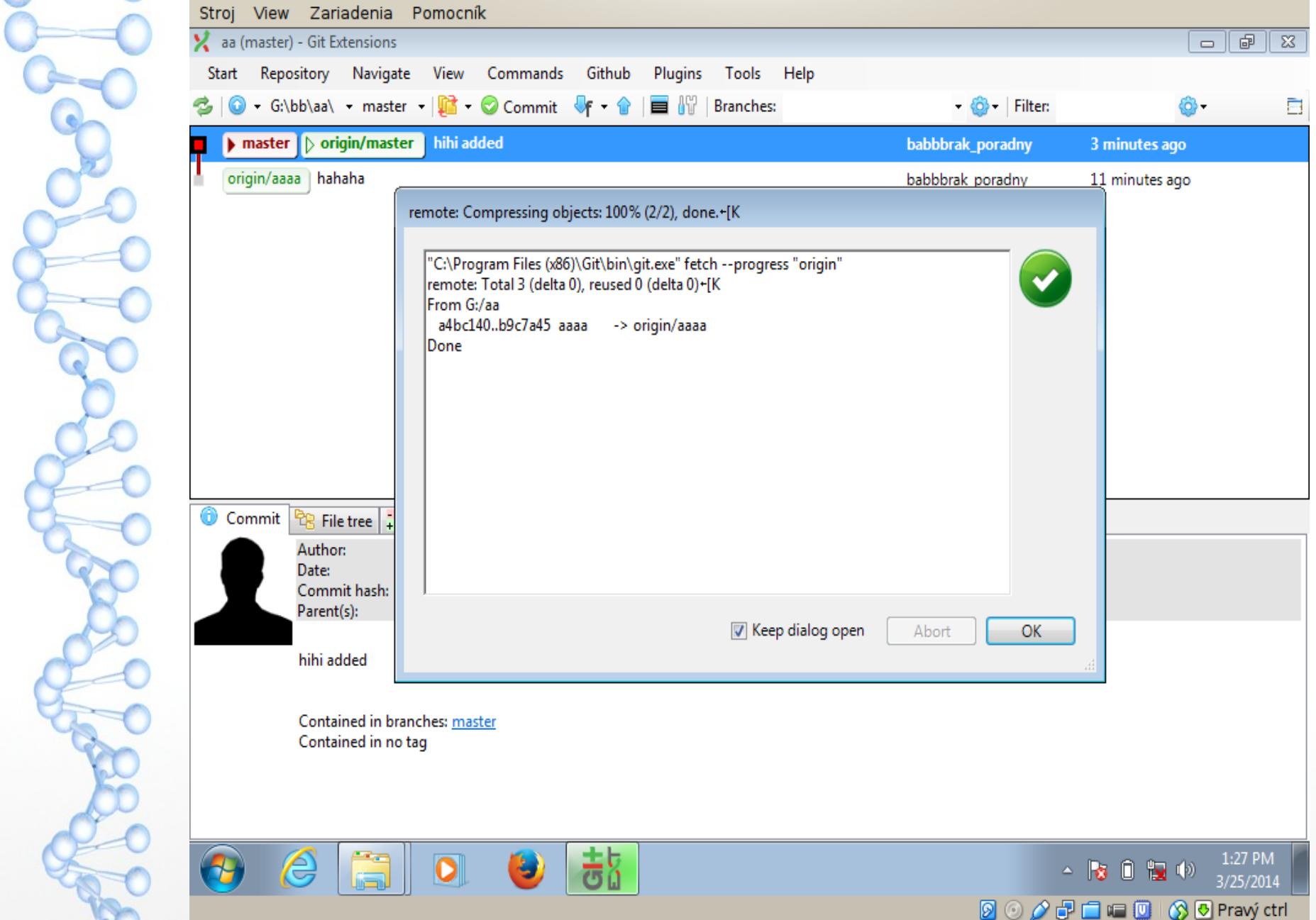


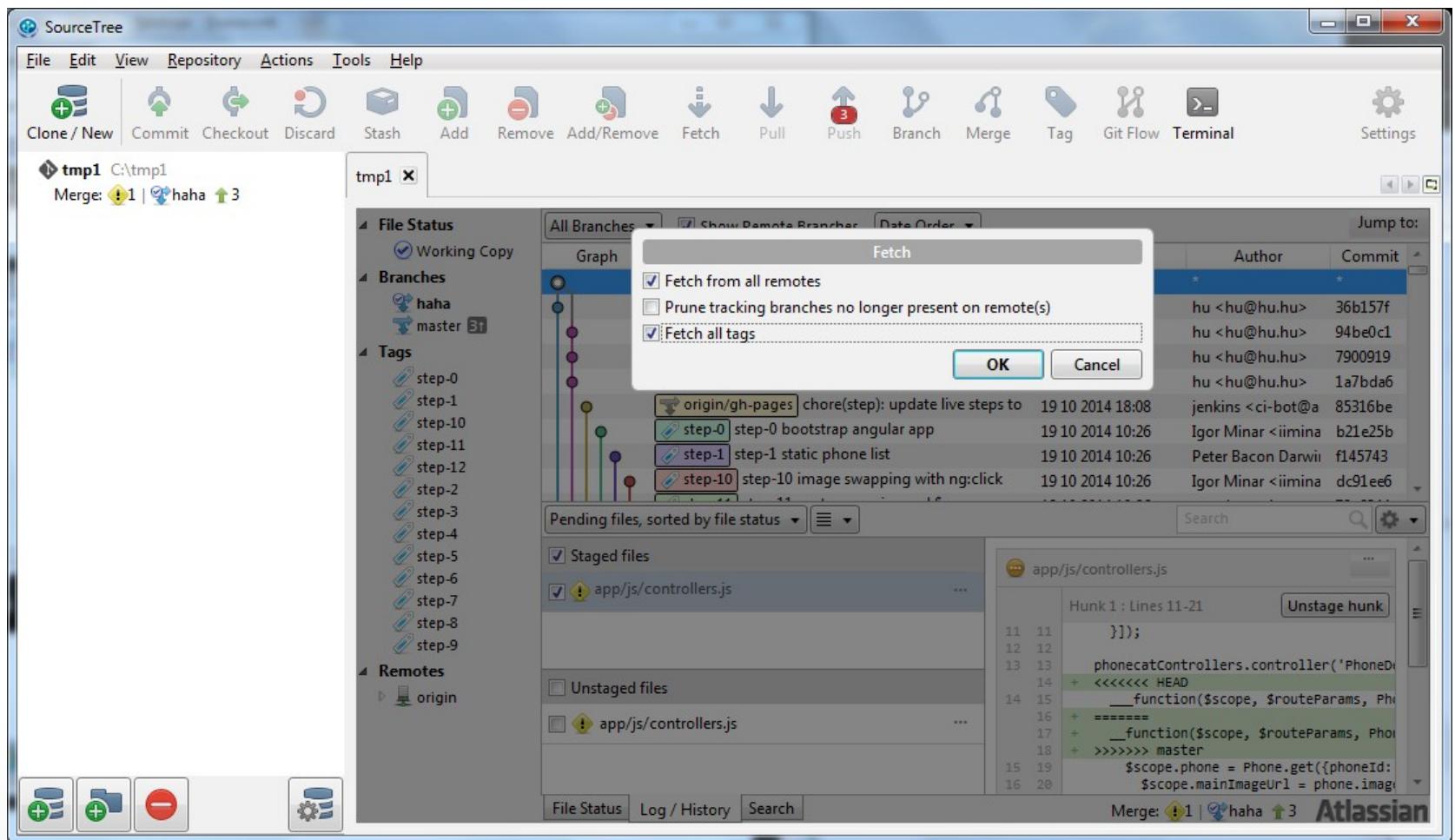


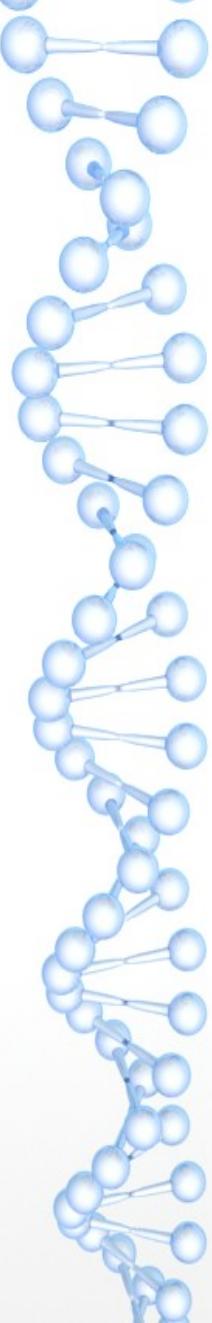
## 4.e) Fetch

*\$ git fetch*

- Získa nové commity zo vzdialeného repozitára
- Nič nezmení u nás
- Nikdy nespôsobí „problémy“



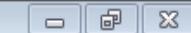




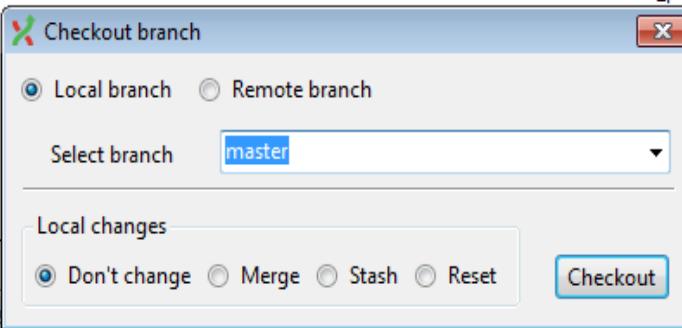
## 4.f) Checkout

`$ git checkout {meno vety}`

- Nikdy nespôsobuje komunikáciu so vzdialeným servrom
- Prepne nás na inú lokálne prístupnú vetvu
- *Working dir* sa upraví podľa vety, na ktorú sa checkout-ujeme
- Môžeme prísť o dátu napr. vo *working dir* -zvykne sa (podľa prepínačov) opýtať, či to tak chceme



stash	WIP on master: c76ca5a added hihi double	babbbrak_poradny	33 seconds ago
	untracked files on master: c76ca5a added hihi double	babbbrak_poradny	33 seconds ago
	index on master: c76ca5a added hihi double	babbbrak_poradny	34 seconds ago
master	added hihi double	babbbrak_poradny	12 minutes ago
aaaa	hihi added second time	babbbrak_poradny	16 minutes ago
	hihi added	babbbrak_poradny	20 minutes ago
	hahaha	babbbrak_poradny	27 minutes ago



Commit

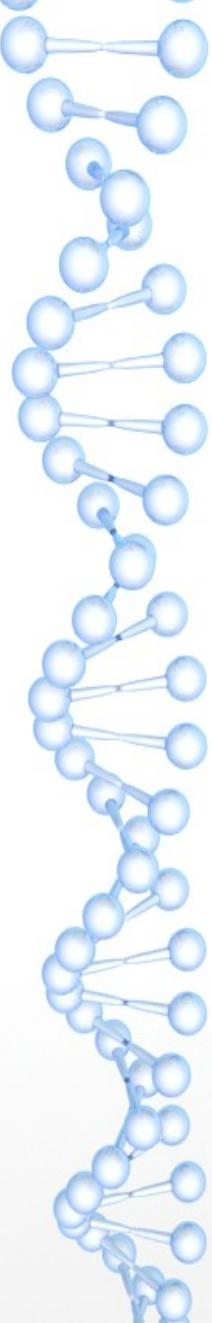
Author: babbbrak\_pora  
Date: 16 minutes ago (3/25/2014 1:26:17 PM)  
Commit hash: b9c7a451b10211613a50b94e341b8eb1f45d272a  
Parent(s): a4bc140d21

hihi added second time

Contained in branches: aaaa

Contained in no tag





SourceTree

File Edit View Repository Actions Tools Help

Clone / New Commit Checkout Discard Stash Add Remove Add/Remove Fetch Pull Push Branch Merge Tag Git Flow Terminal Settings

**tmp1 C:\tmp1**  
Merge: !1 | hahaha ↑3

**tmp1 x**

All Branches Show Remote Branches Date Order Jump to:

**File Status** Working Copy

**Branches** hahaha master B1

**Tags** step-0 step-1 step-11 Checkout step-10... Details... step-12 step-2 step-3 step-4 step-5 step-6 step-7 step-8 step-9

**Remotes** origin

Graph Description Date Author Commit

Graph	Description	Date	Author	Commit
step-1	step-1 static phone list	19.10.2014 10:26	Peter Bacon Darwin	f145743
step-10	step-10 image swapping with ng:click	19.10.2014 10:26	Igor Minar <iiminar@gmail.com>	dc91ee6
step-11	step-11 custom service and \$resource	19.10.2014 10:26	smather <simon.rn>	78e6344
step-12	origin/master origin/HEAD step-	19.10.2014 10:26	Peter Bacon Darwin	fcb3343
step-2	step-2 angular template with repeater	19.10.2014 10:26	Peter Bacon Darwin	7c1e0bb
step-3	step-3 interactive search	19.10.2014 10:26	Peter Bacon Darwin	cf24f6a
step-4	step-4 phone ordering	19.10.2014 10:26	Peter Bacon Darwin	b2ad58f
step-5	step-5 XHR and dependency injection	19.10.2014 10:26	Peter Bacon Darwin	16bf1de
step-6	step-6 phone images and links	19.10.2014 10:26	Peter Bacon Darwin	1ec9012

Sorted by file status

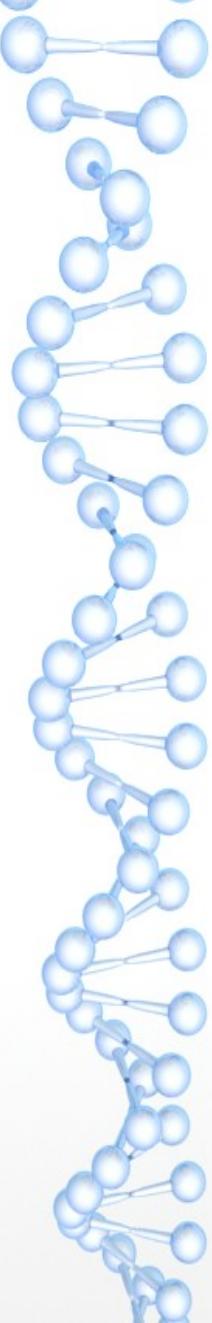
Commit: dc91ee6a2d0cef88e070cafacb2e9d92333baed4 [dc91ee6]  
Parents: 6ae92c2ff  
Author: Igor Minar <iiminar@gmail.com>  
Date: 19. októbra 2014 10:19:49  
**Committer:** Peter Bacon Darwin <pete@bacondarwin.com>  
**Commit Date:** 19. októbra 2014 10:26:20

app/css/app.css  
app/js/controllers.js  
app/partials/phone-detail.html

Hunk 1 : Lines 17-26 nk

17	17	function(\$scope,
18	18	\$http.get('pho
19	19	\$scope.phone
20	20	\$scope.mainI
21	21	});
22	22	\$+
23	23	\$+
24	24	\$+
25	25	}
26	26	]);

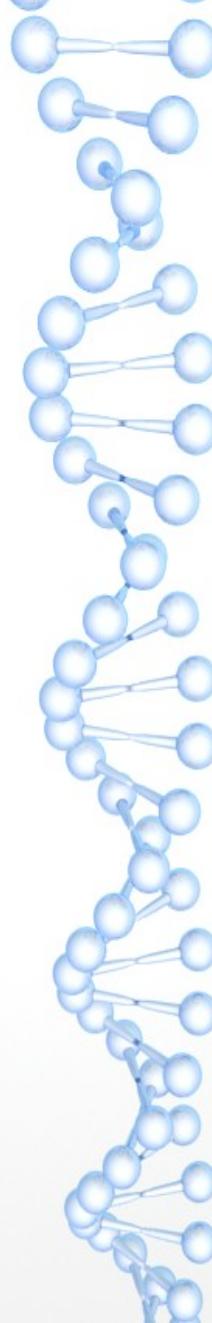
Merge: !1 | hahaha ↑3 Atlassian



## 4.g) Pull

\$ *git pull*

- Urobí niečo ako fetch (stiahnutie) a následný checkout na najnovší commit vo vzdialenom repozitári
- Tiež detektuje zmenu vo *working dir*-ak máme niečo necommitnuté, varuje nás
- **Jeho cieľom je čo nainteligentnejšie ZOSYNCHRONIZOVАŤ našu lokálnu a vzdialenú vetvu**
- Robí ešte skrytý Merge...



The screenshot shows the Git Extensions interface with a 'Pull' dialog open. The dialog includes a commit graph on the left and various configuration options on the right.

**Pull from:**

- Remote origin
- Url G:/aa

**Branch:**

- Local branch master
- Remote branch (dropdown menu)

**Merge options:**

- Merge remote branch into current branch
- Rebase current branch on top of remote branch, creates linear history (use with caution)
- Do not merge, only fetch remote changes

**Tag options:**

- Follow tagopt, if not specified, fetch tags reachable from remote HEAD
- Fetch no tag
- Fetch all tags

**Buttons at the bottom:**

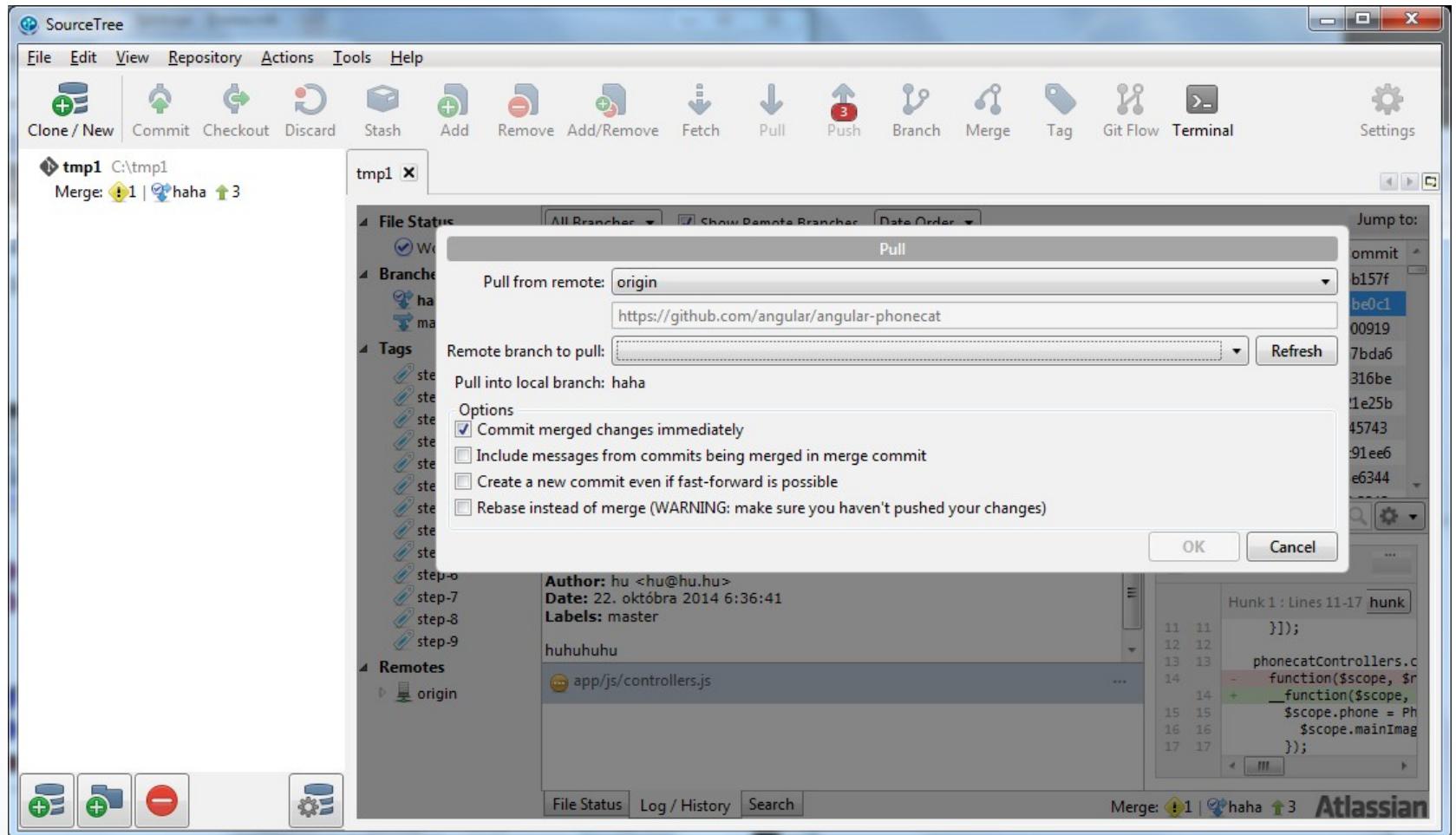
- Solve conflicts
- Stash changes
- Auto stash
- 

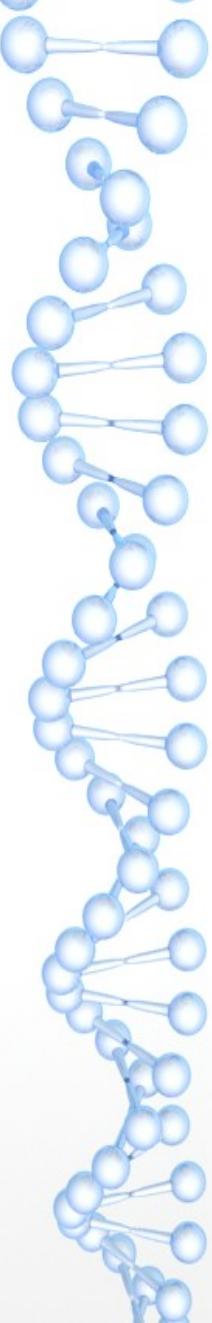
**Commit Graph Legend:**

- Red square = current branch
- Green circle = new commit
- .AAA postfix for files if merge conflicts occur

**Commit Graph Description:**

The graph illustrates a merge operation between a local branch (master) and a remote branch (origin). The local branch has commits a, b, c, d, e, f, g. The remote branch has commits a, b, c, d, e, f. A merge commit g is shown as the result of the merge, which includes all commits from both branches. Labels indicate the local branch is 'current' and the remote branch is 'other'. A note says 'Hover to see scenario when fast forward is possible.'

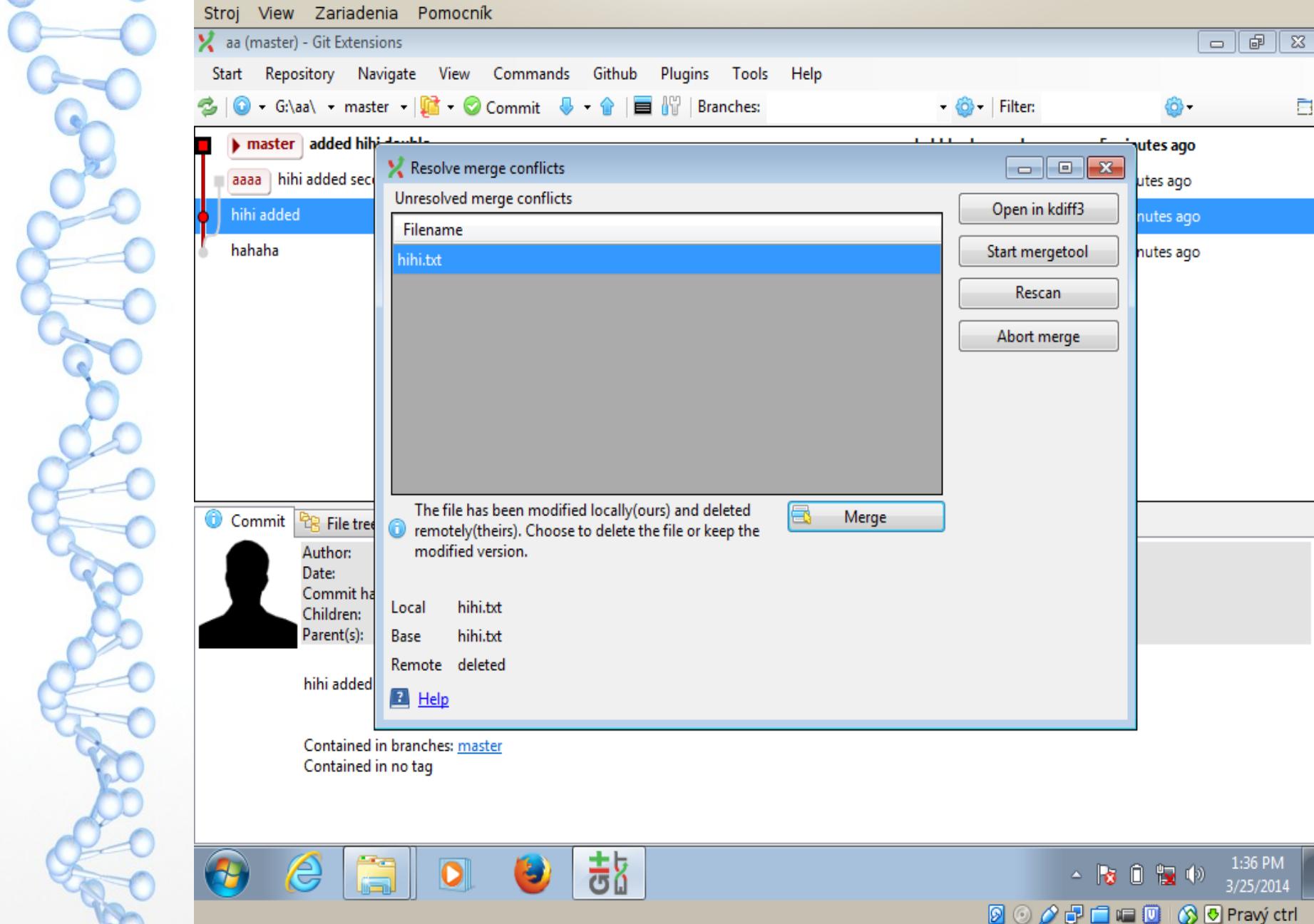


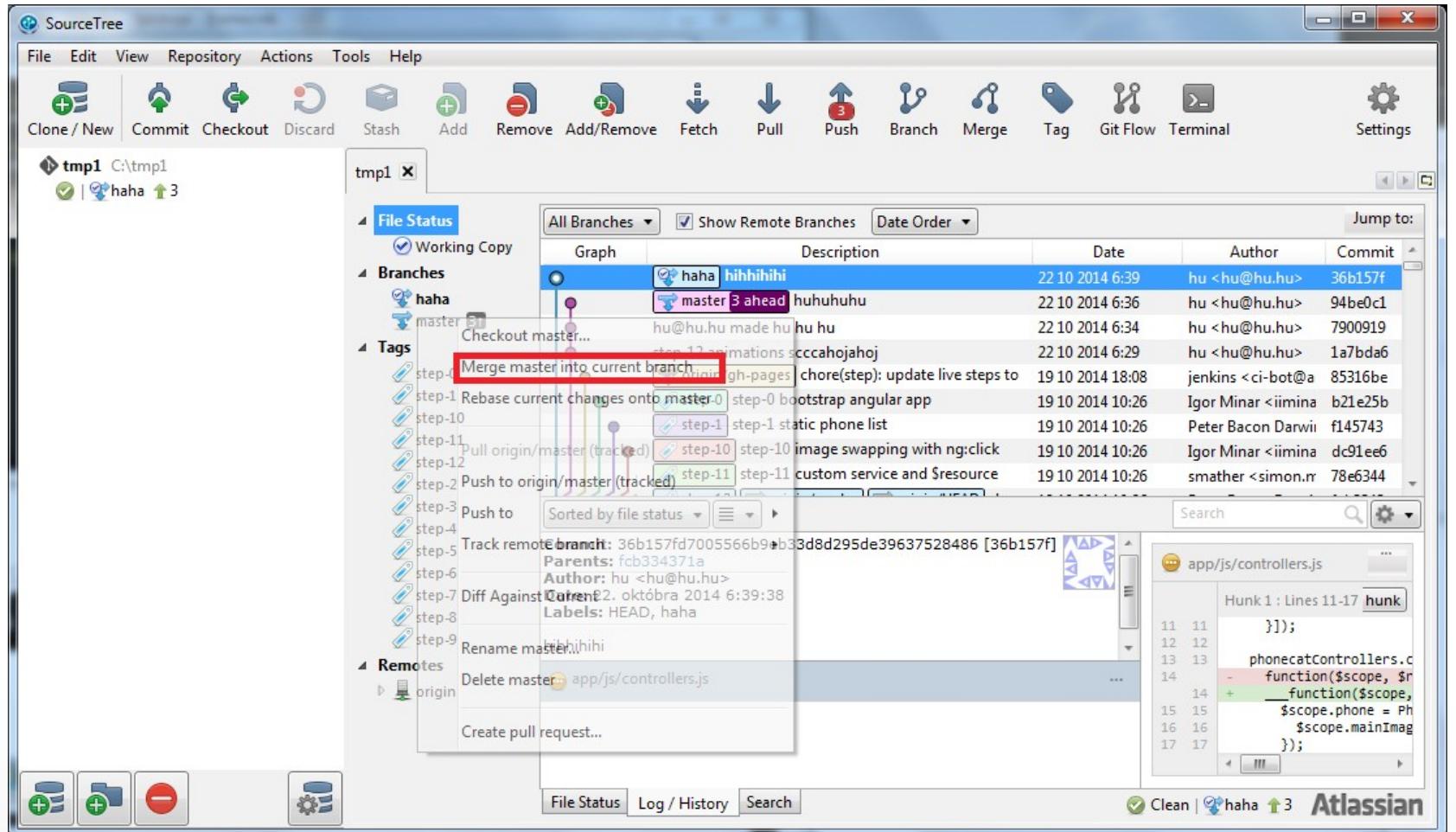


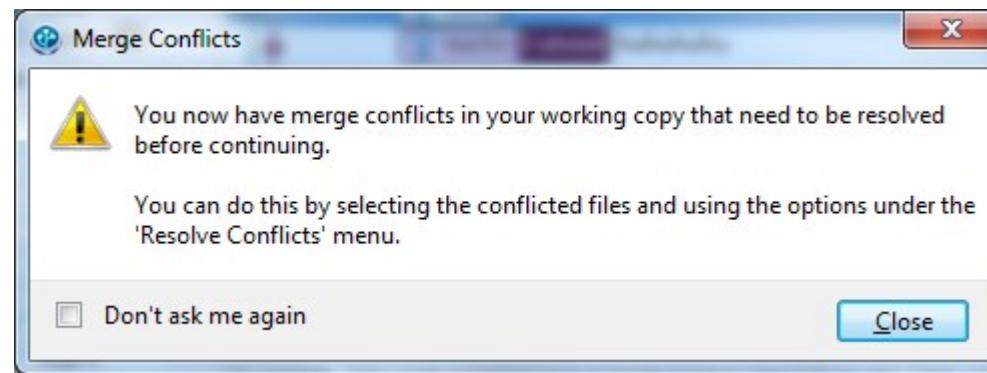
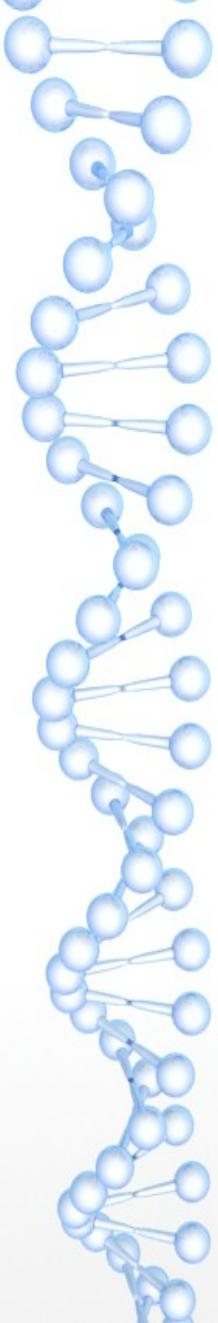
## 4.h) Merge-ovanie cez kdiff3

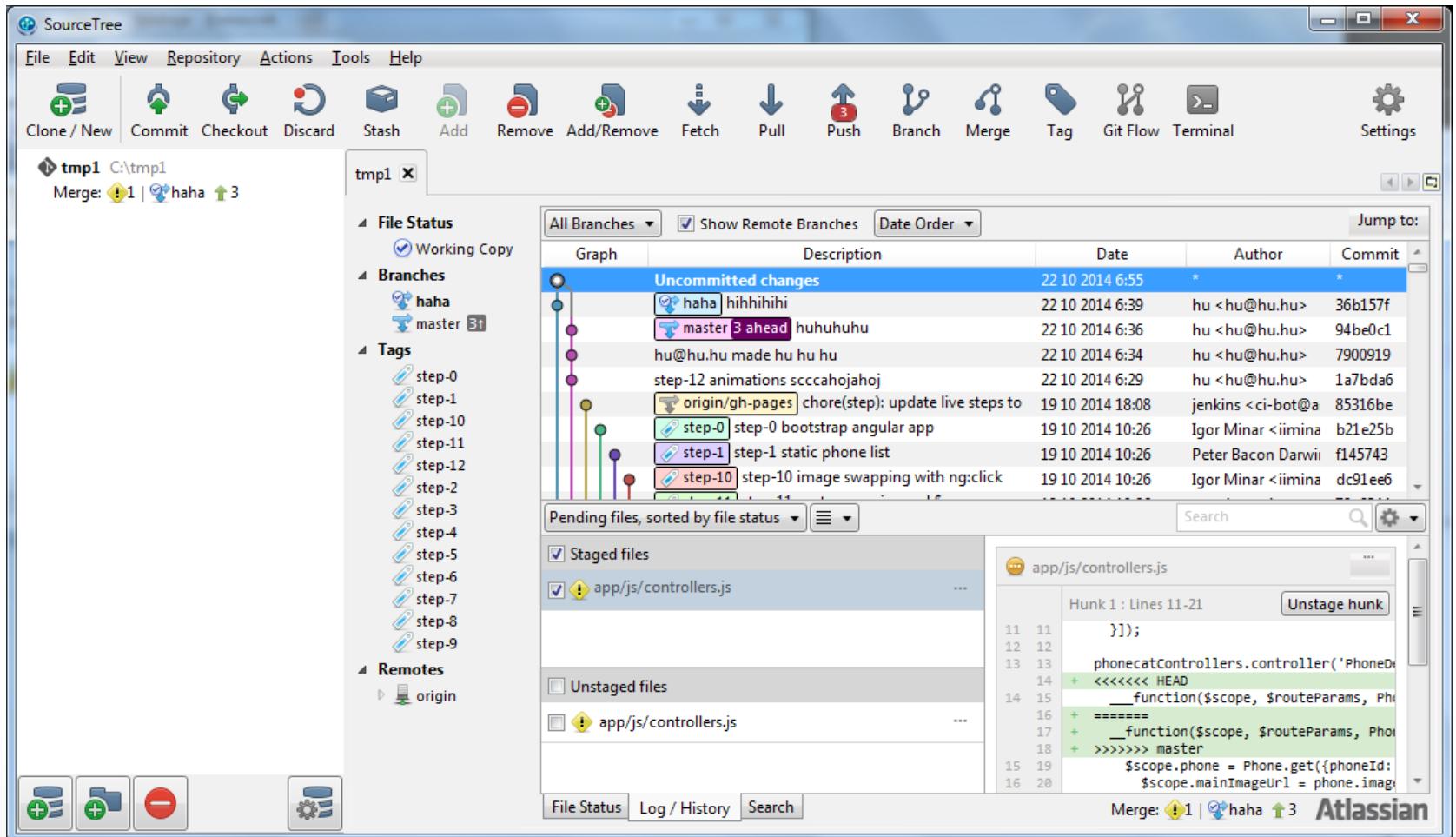
`$ git merge {iná vetva}`

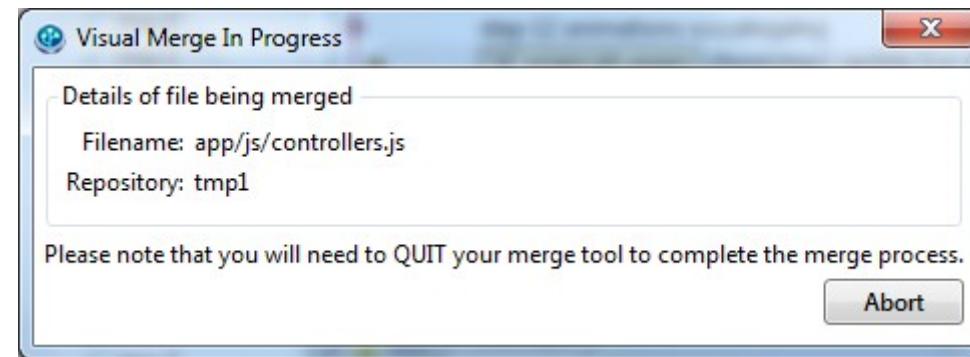
- Spojí aktuálnu a {inú vetvu}
- Musí riešiť (prípadné) **kolízie kódu** -triviálne prípady rieši sám; neautomatizovateľné necháva kóderovi =**konfliktný kód**
- Konfliktný kód prenecháva špecializovaným programom -napr. *kdiff3* (konfigurovateľné)
- Je často najobávanejšou operáciou začiatočníkov











KDiff3 controllers.js.BASE.1816.js <-> controllers.js.LOCAL.1816.js <-> controllers.js.REMOTE.1816.js - KDiff3

File Edit Directory Movement Diffview Merge Window Settings Help

A (Base): C:\tmp1\app\js\controllers.js.BASE.1816.js ...  
Top line 14 Encoding: System Line end style: DOS  
B: C:\tmp1\app\js\controllers.js.LOCAL.1816.js ...  
Top line 14 Encoding: System Line end style: DOS  
C: C:\tmp1\app\js\controllers.js.REMOTE.1816.js ...  
Top line 14 Encoding: System Line end style: DOS

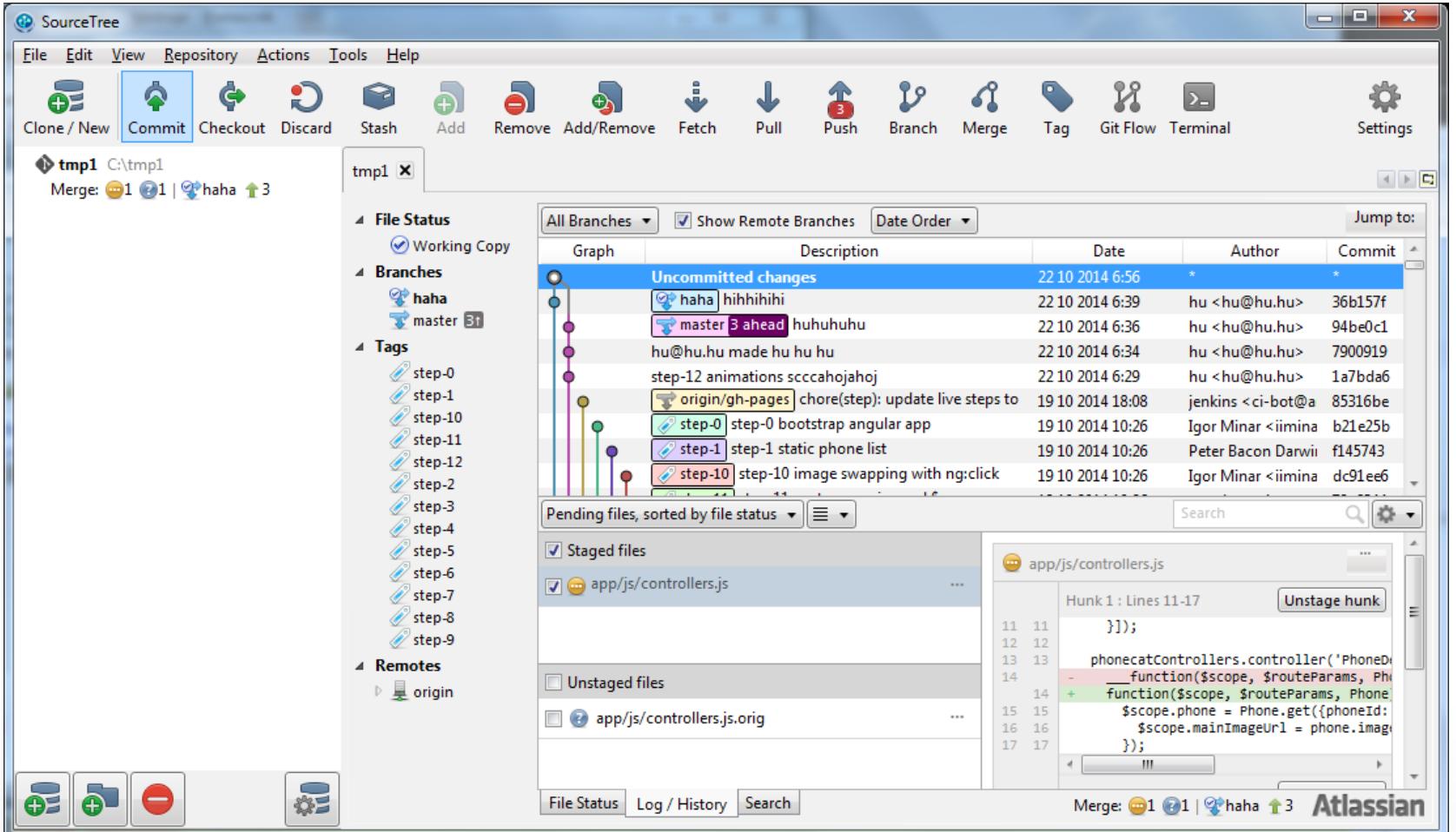
```
function($scope, $routeParams, $Phone) {
  ...
  $scope.phone = $Phone.get({phoneId: $routeParams.phoneId});
  $scope.mainImageUrl = phone.images[0];
  ...
  $scope.setImage = function(imageUrl) {
    ...
    $scope.mainImageUrl = imageUrl;
  }
}
]);
```

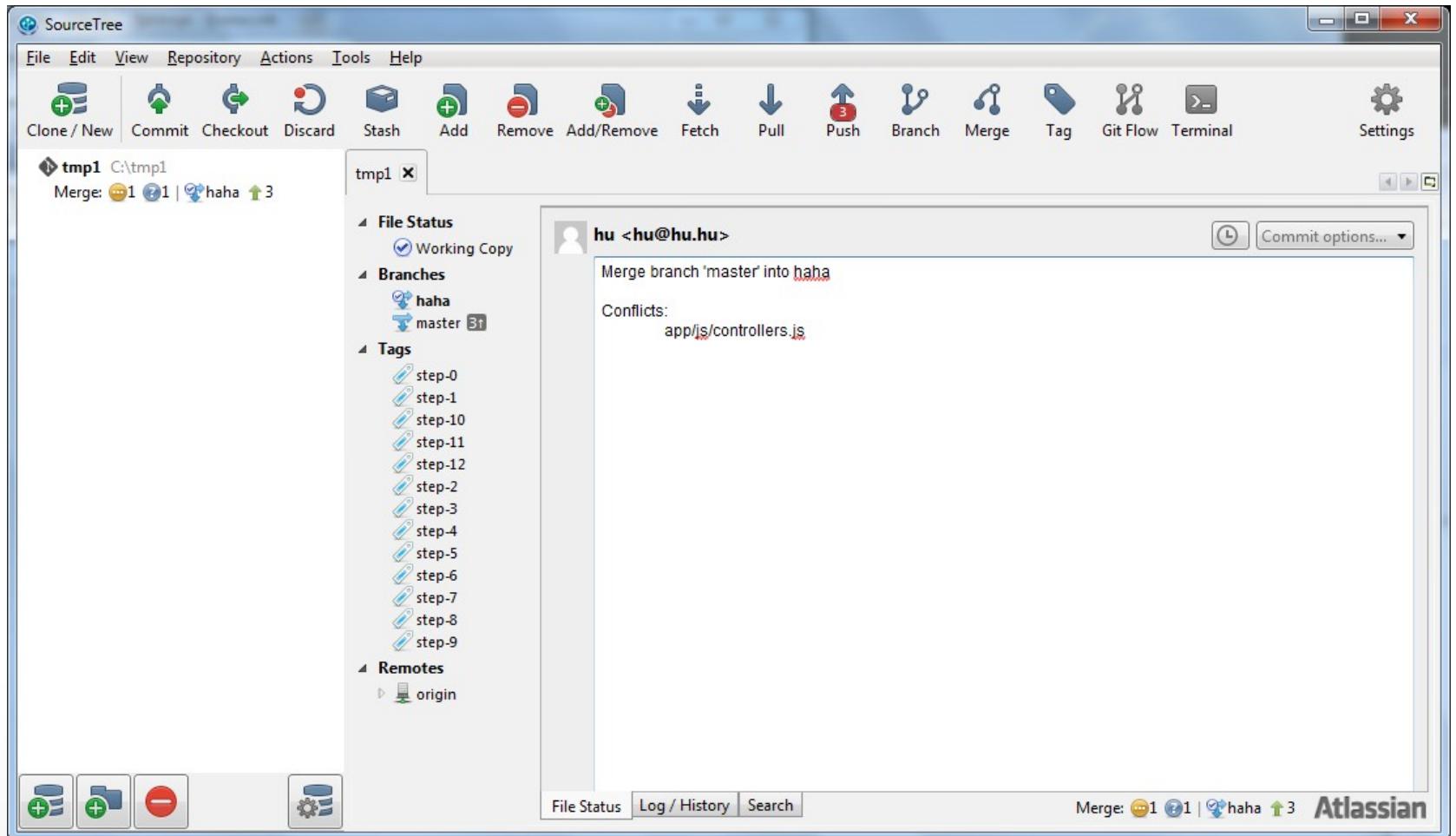
Output: C:\tmp1\app\js\controllers.js Encoding for saving: Codec from C: System Line end style: DOS (A, B, C)

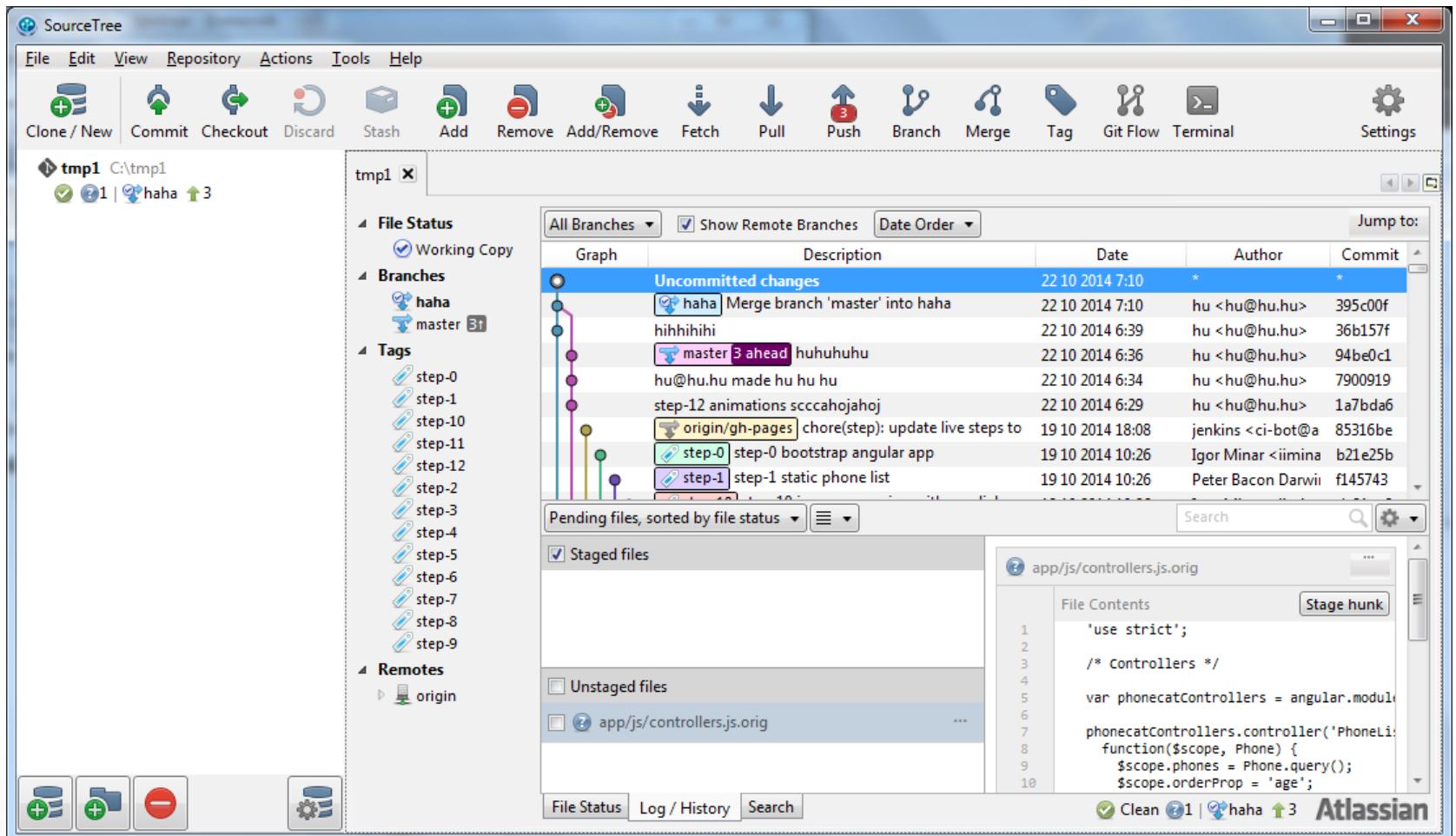
? <Merge Conflict>

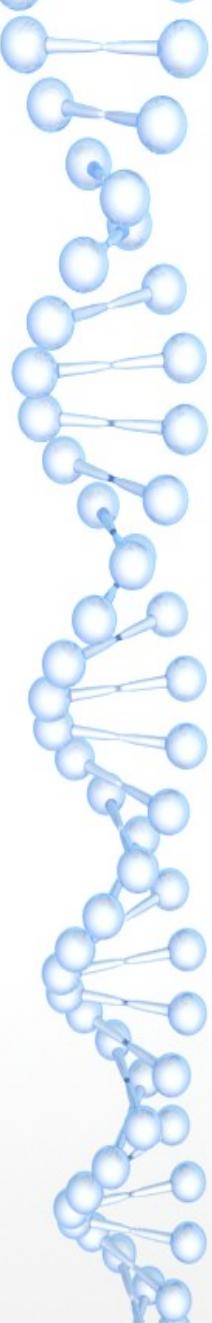
```
function($scope, $routeParams, $Phone) {
  ...
  $scope.phone = $Phone.get({phoneId: $routeParams.phoneId}, function(phone) {
    ...
    $scope.mainImageUrl = phone.images[0];
  });

  $scope.setImage = function(imageUrl) {
    ...
    $scope.mainImageUrl = imageUrl;
  }
}
]);
```





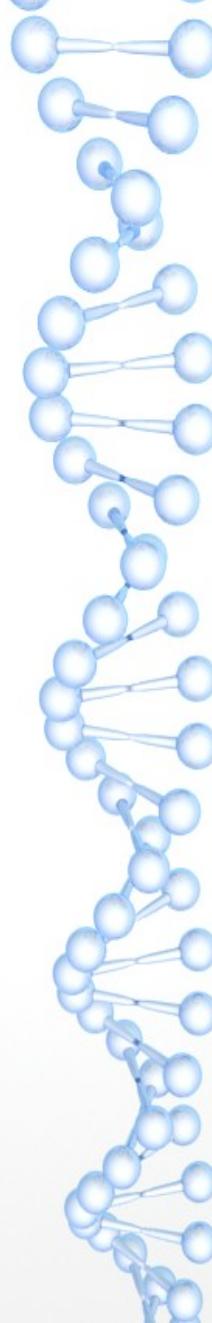




## 4.i) Stash

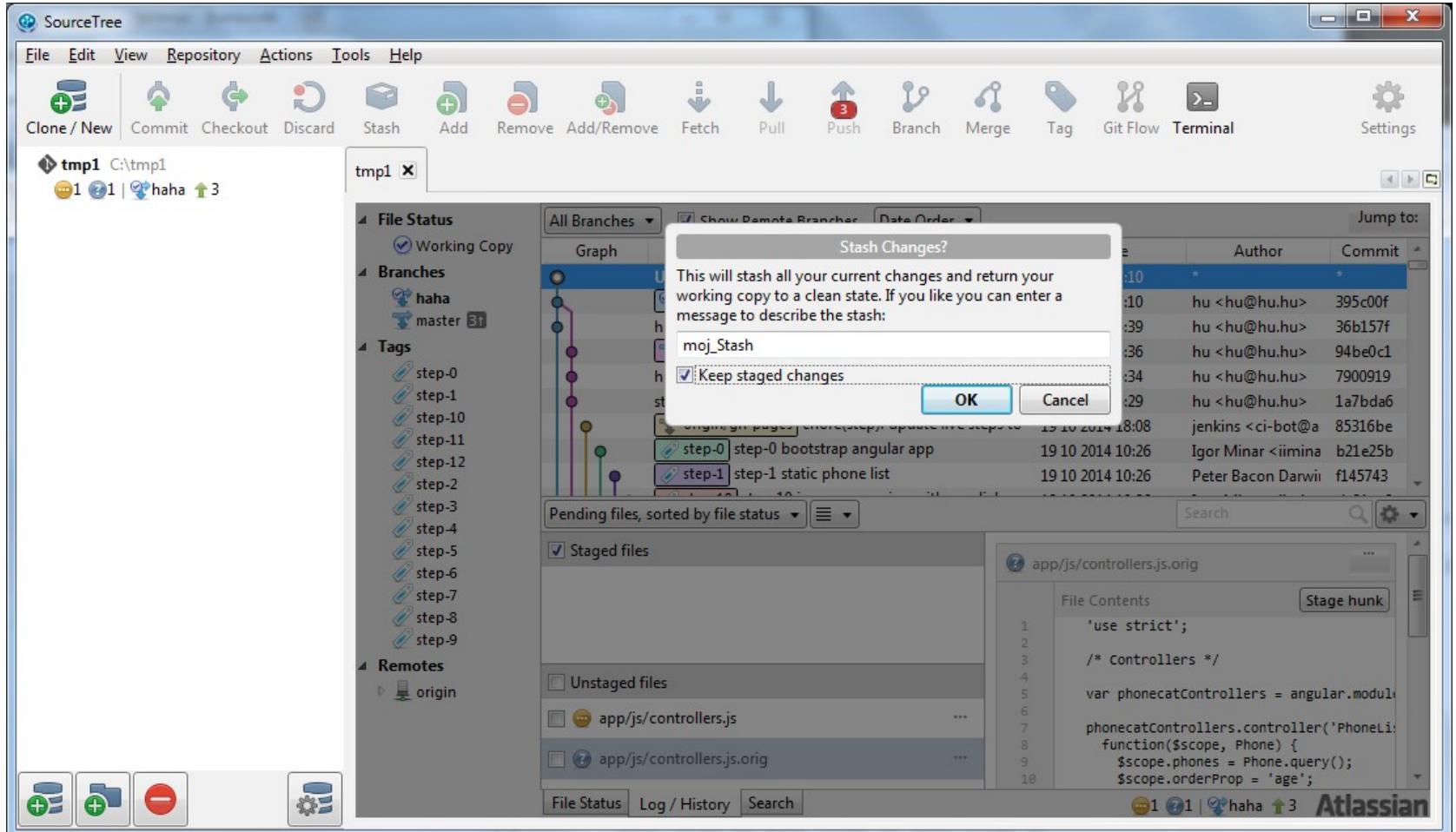
*\$ git stash*

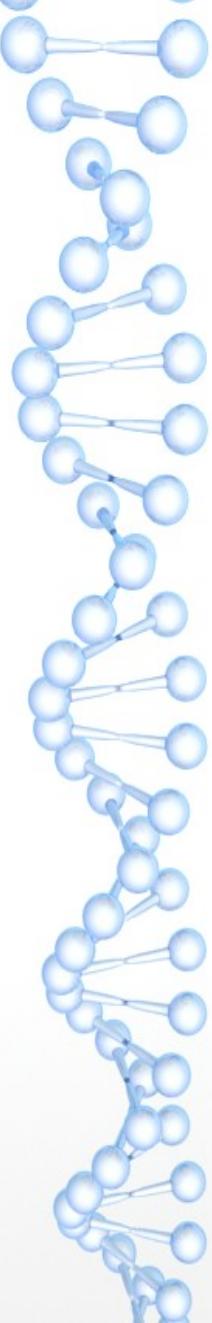
- Umožňuje narýchlo odsunúť aktuálne ešte necommit-nuté working dir zmeny
- Dá sa k nim hocikedy vrátiť cez *\$ git stash pop*
- Funguje na princípe zásobníka
- Vhodný pri operatívnych zásahoch  
(použiteľný na rýchle prepínanie sa medzi viac rozrobenými vecami)



The screenshot shows the Git Extensions interface with the following details:

- Toolbar:** Stroj, View, Zariadenia, Pomocník, aa (master) - Git Extensions.
- Menu Bar:** Start, Repository, Navigate, View, Commands, Github, Plugins, Tools, Help.
- Branches:** master, Commit (1).
- File List:** G:\aa\, master, Commit (1).
- Commit History:** master added hihi double, aaaa hihi added second time, babbbrak\_poradny hihi added, babbbrak\_poradny hahaha.
- Process Dialog:** "C:\Program Files (x86)\Git\bin\git.exe" stash save -u  
Saved working directory and index state WIP on master: c76ca5a added hihi double  
HEAD is now at c76ca5a added hihi double  
Done
- Commit Details:** Author: [empty], Date: [empty], Commit hash: [empty], Children: [empty], Parent(s): [empty].
- Status:** Contained in branches: master, Contained in no tag.





SourceTree

File Edit View Repository Actions Tools Help

Clone / New Commit Checkout Discard Stash Add Remove Add/Remove Fetch Pull Push Branch Merge Tag Git Flow Terminal Settings

**tmp1 C:\tmp1**

tmp1 x

File Status Working Copy

Branches haha master

Tags step-0 step-1 step-10 step-11 step-12 step-2 step-3 step-4 step-5 step-6 step-7 step-8 step-9

Remotes origin

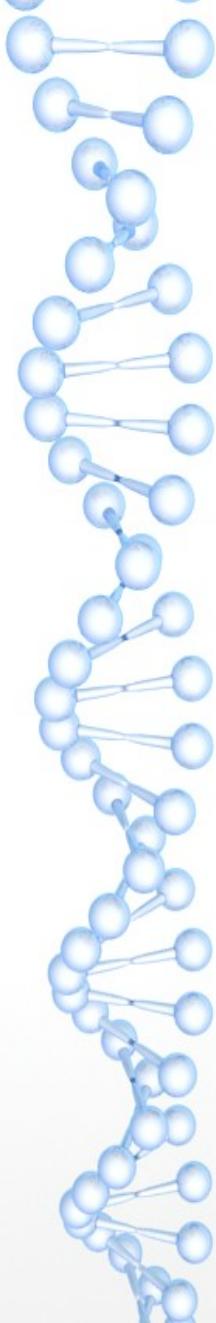
Stashes On haha: moj\_Stash

app/js/controllers.js

Hunk 1 : Lines 12-18

```
12 12 phonecatControllers.controller('PhoneDetailCtrl', ['$scope', '$routeParams', 'Phone',  
13 13 function($scope, $routeParams, Phone) {  
14 14 - $scope.phone = Phone.get({phoneId: $routeParams.phoneId}, function(phone) {  
15 15 + $scope.phone = Phone_____.get({phoneId: $routeParams.phoneId}, function(phone) {  
16 16 $scope.mainImageUrl = phone.images[0];  
17 17 });  
18 18});
```

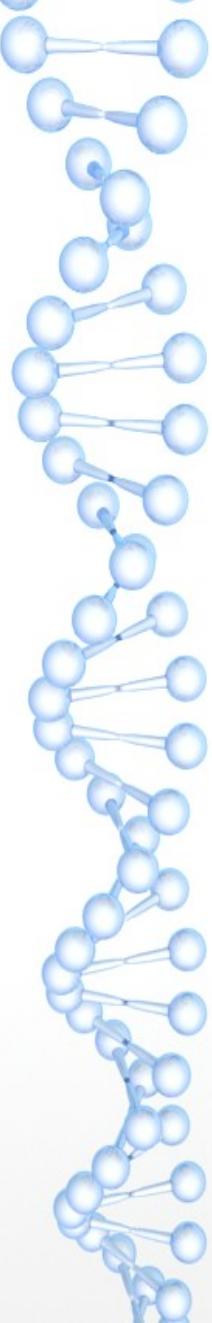
File Status Log / History Search On haha: moj\_Stash Clean 1 | haha 3 Atlassian



4.j) Čo ked' spravím chybu? (opravy bežných chýb)

$$\begin{aligned}
 C &= \alpha + \beta + \gamma \\
 C &\cdot (T \cdot S (\Omega \cdot 10^7) + 3\alpha + 2 \cdot 3 \ln(11)) \\
 C &\cdot (T \cdot S \cdot \log \frac{\beta}{\gamma + \delta} + 3\alpha + 6 \ln(11))^2 \\
 C &\cdot \left[ \sqrt{\sum_{x_1}^{x_n} \alpha dx + \frac{3[(3+7)(1+6+3\beta)]}{(5+\gamma)(8+2)+1} + 6 \ln(11)} \right]^2 \\
 C &\cdot \left[ \sqrt{\sum_{x_1}^{x_n} \frac{(3+7)(1+6+3\beta)}{(5+\gamma)(8+2)+1} dx + \frac{3[(3+7)^2 + (1+6+3\beta)^2]}{(5+\gamma)(8+2)+1} + 6 \ln(11)} \right]^2 \\
 C &\cdot \left[ \sqrt{\sum_{x_1}^{x_n} \frac{(3+7)^2 + (\beta - 180^\circ)^2 + 3\beta}{(5+\gamma)(8+2)+1} dx + \frac{3[(3+7)^2 + (\beta - 180^\circ)^2]}{(5+\gamma)(8+2)+1} + 6 \ln(11)} \right]^2 \\
 C &\cdot \sqrt{\sum_{x_1}^{x_n} \frac{\sqrt{3+7x + (\beta - 180^\circ)^2 + 3\beta}}{10\Omega + 6\beta - 1} dx + \frac{3\sqrt{3+7x + (\beta - 180^\circ)^2 + 3\beta}}{10\Omega + 6\beta - 1} + 6 \ln(11) + \log \beta} \\
 C &\cdot \sqrt{\left[ \sum_{x_1}^{x_n} \alpha dx + \frac{3\alpha \left[ (3\beta - 7)(1+6+3\beta) + 3\beta \right]}{(5+\gamma)(8+2)} + \frac{3\beta \left[ (3+7)(1+6+3\beta) + 3\beta \right]}{10\Omega + 6\beta - 1} + \log \beta \right]^2} \\
 C &=
 \end{aligned}$$

$$\frac{\sqrt{2}}{2} =$$

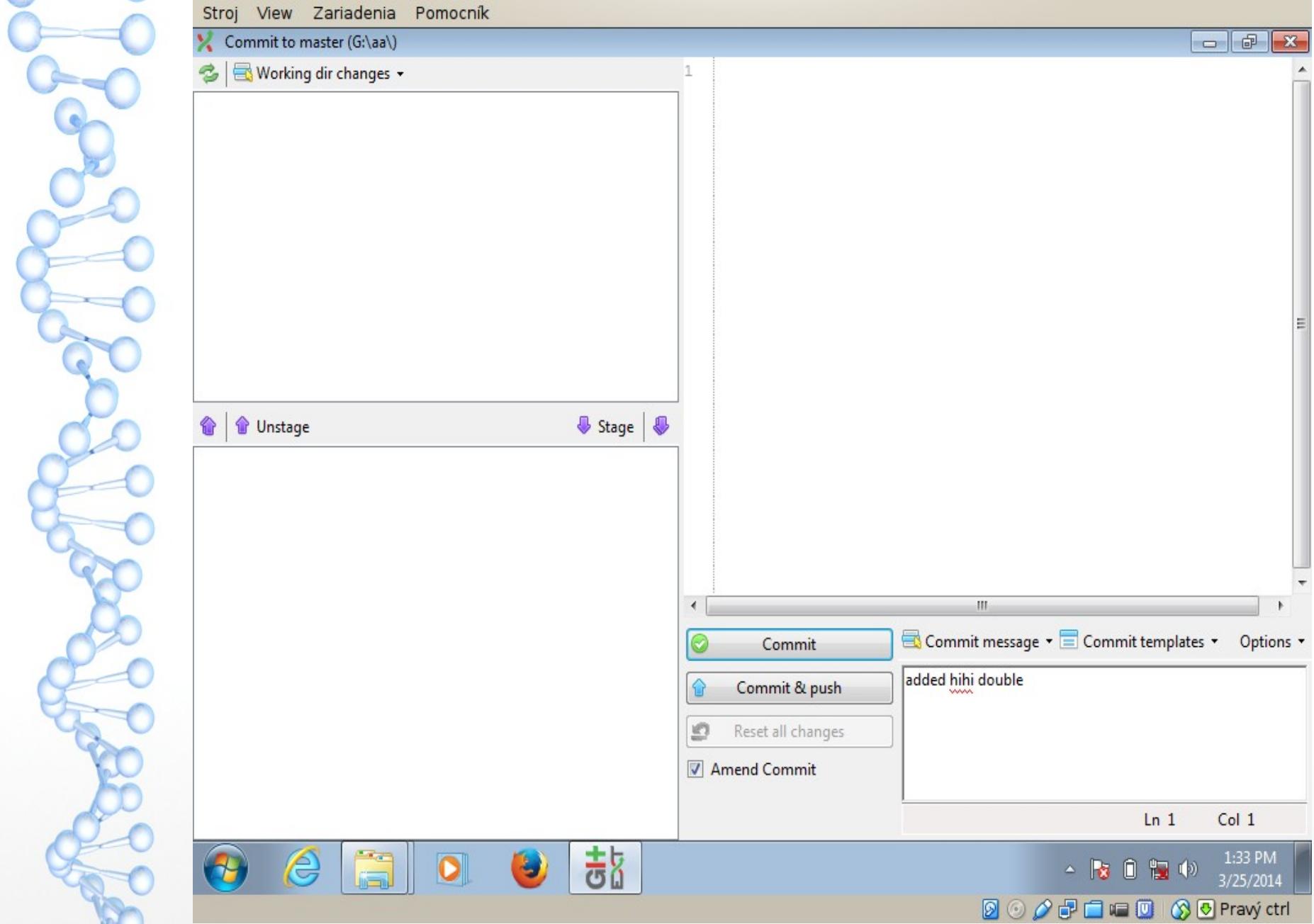


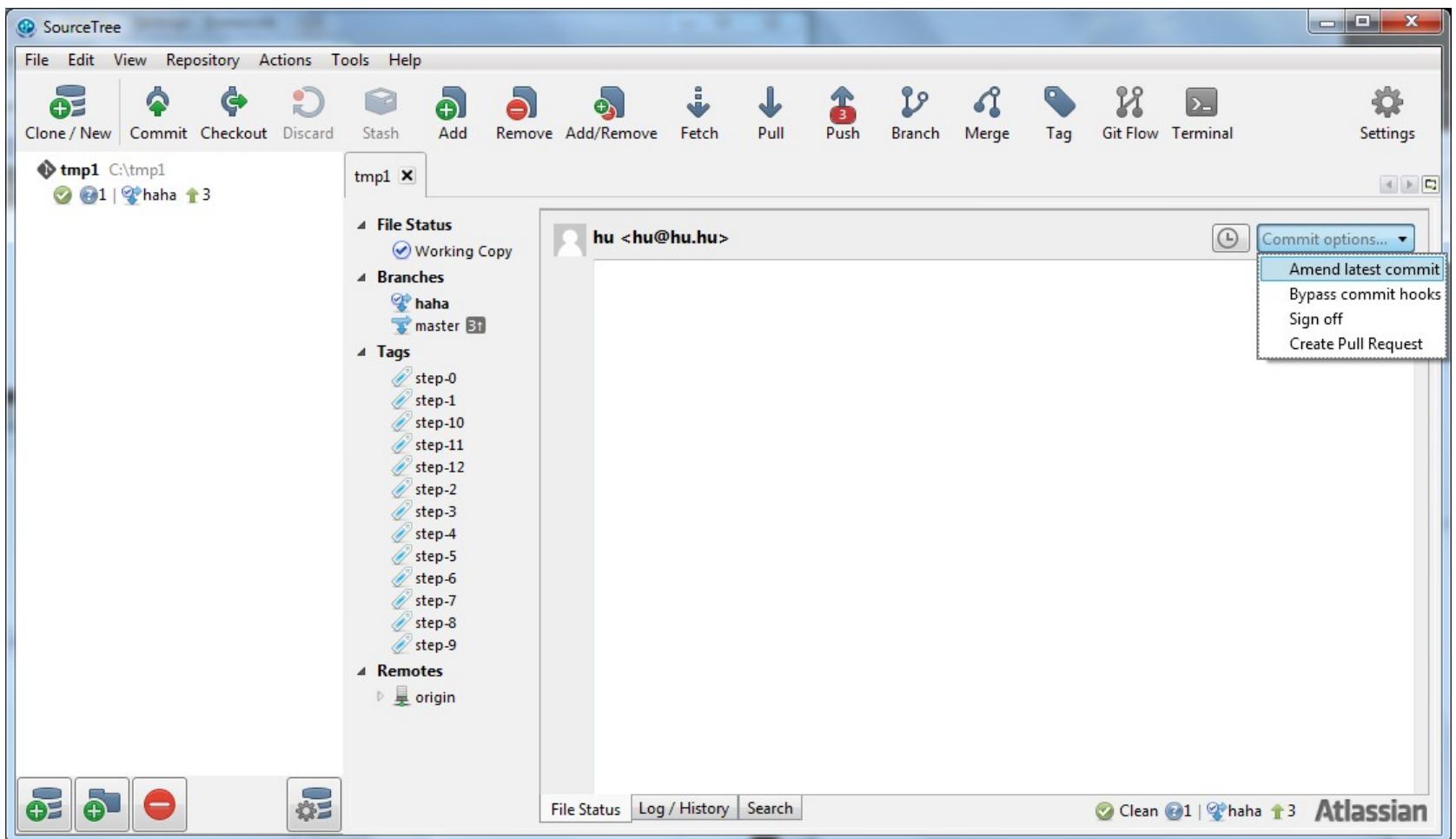
## 4.j) Čo ked' spravím chybu? (opravy bežných chýb)

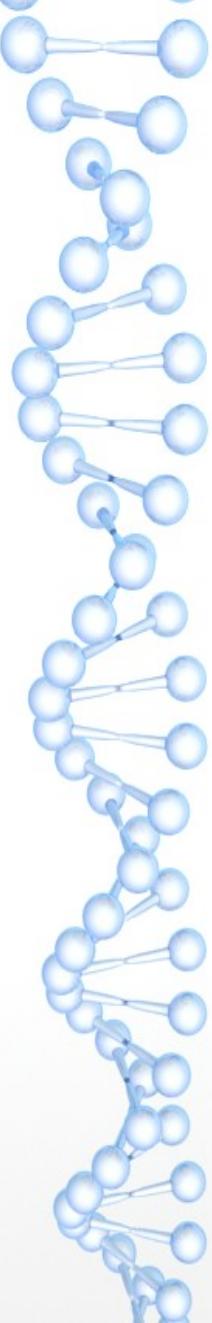
- V commite som urobil pravopisnú chybu a zabudol súbor:

```
$ git add {zabudnutý súbor}
```

```
$ git commit –amend # pozmeni posledny commit
```





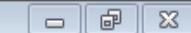


## 4.j) Čo ked' spravím chybu? (opravy bežných chýb)

- Posledný commit je CELÝ zle a chcem ho zrušiť (napr. aj pri zlom merge -chcem zrušiť merge)

```
$ git reset --hard HEAD^
```

```
$ git reset --soft HEAD^
```



```
c:\ MINGW32:/g/aa
Welcome to Git <version 1.8.3-preview20130601>

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

m@A8 /g/aa <master>
$
```

Commit

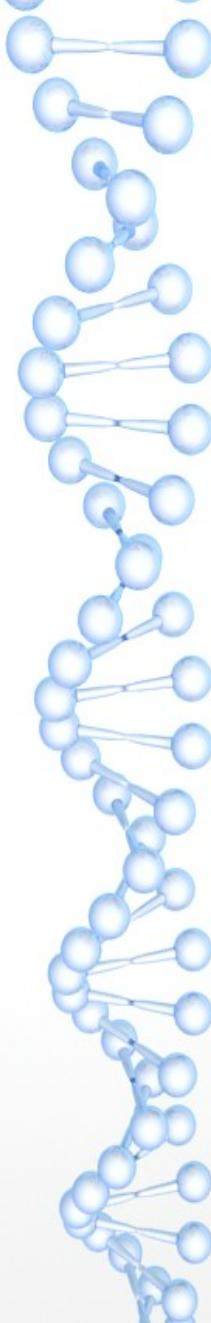
Parent(s): [1f9c15519b](#)

added hihi double

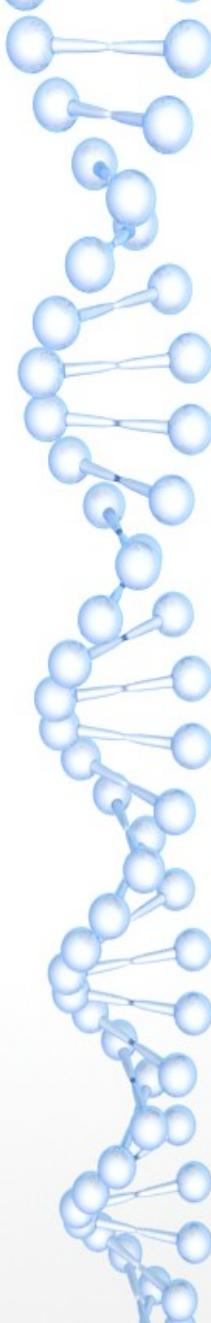
Contained in branches: [master](#)

Contained in no tag





```
Julka@A6 /c/tmp1 <haha!MERGING>
$ #git reset --hard \!?
Julka@A6 /c/tmp1 <haha!MERGING>
$ #git reset --hard HEAD
Julka@A6 /c/tmp1 <haha!MERGING>
$ git reset --hard HEAD
HEAD is now at 36b157f hihhihihi
Julka@A6 /c/tmp1 <haha>
$
```

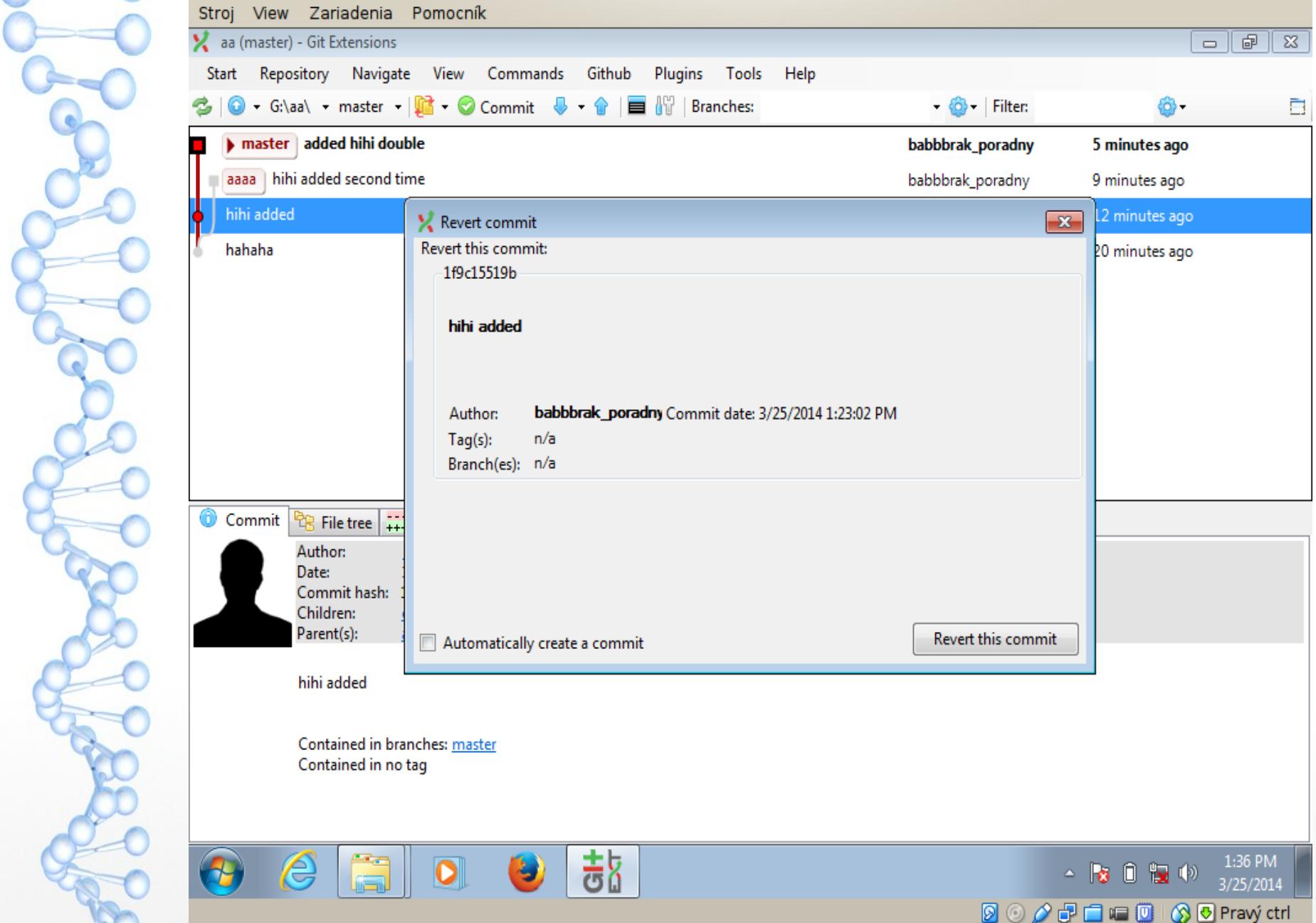


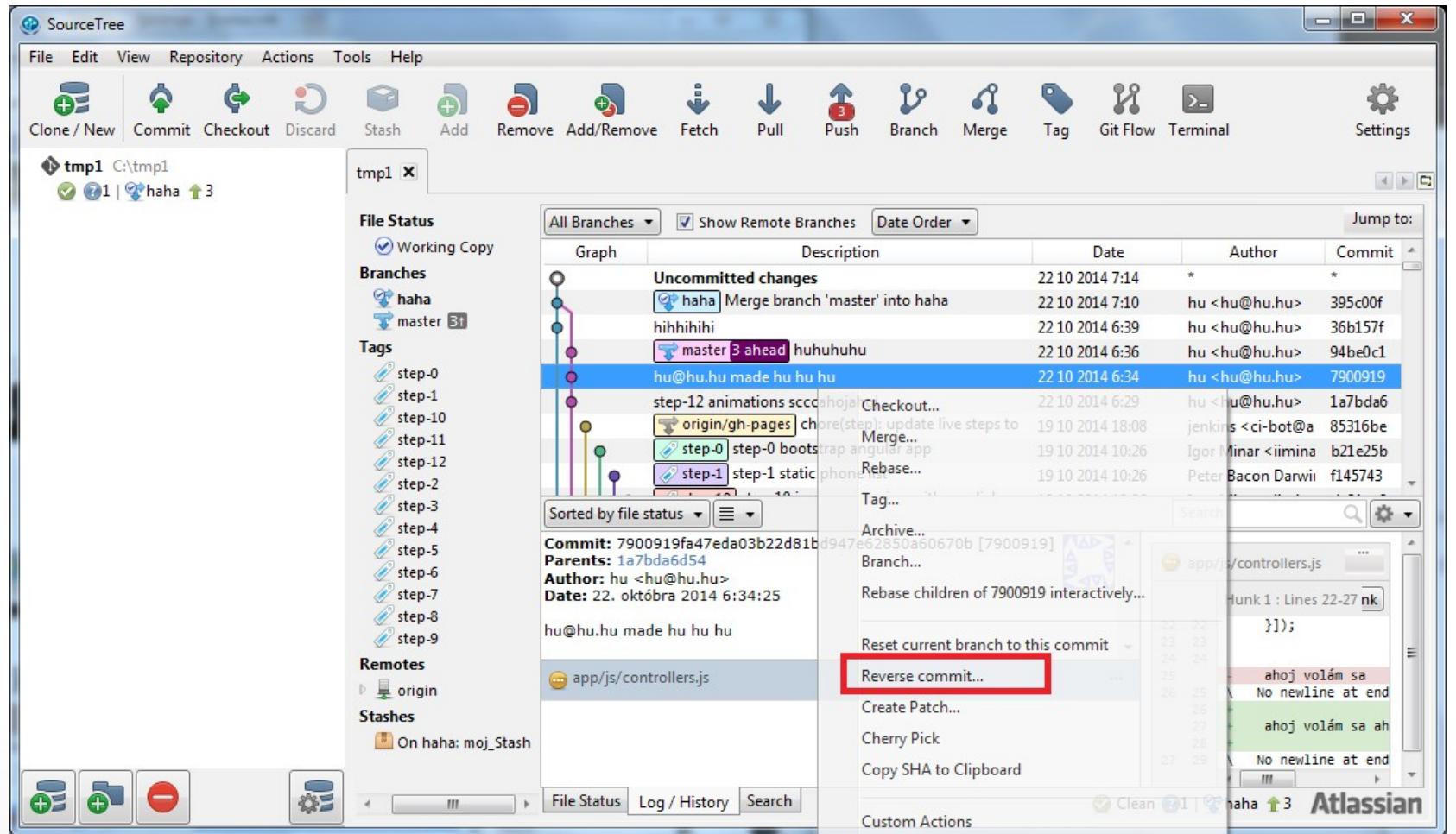
## 4.j) Čo ked' spravím chybu? (opravy bežných chýb)

- Ak už je push-nutý nejaký commit a chcem nad ním urobit' „undo“

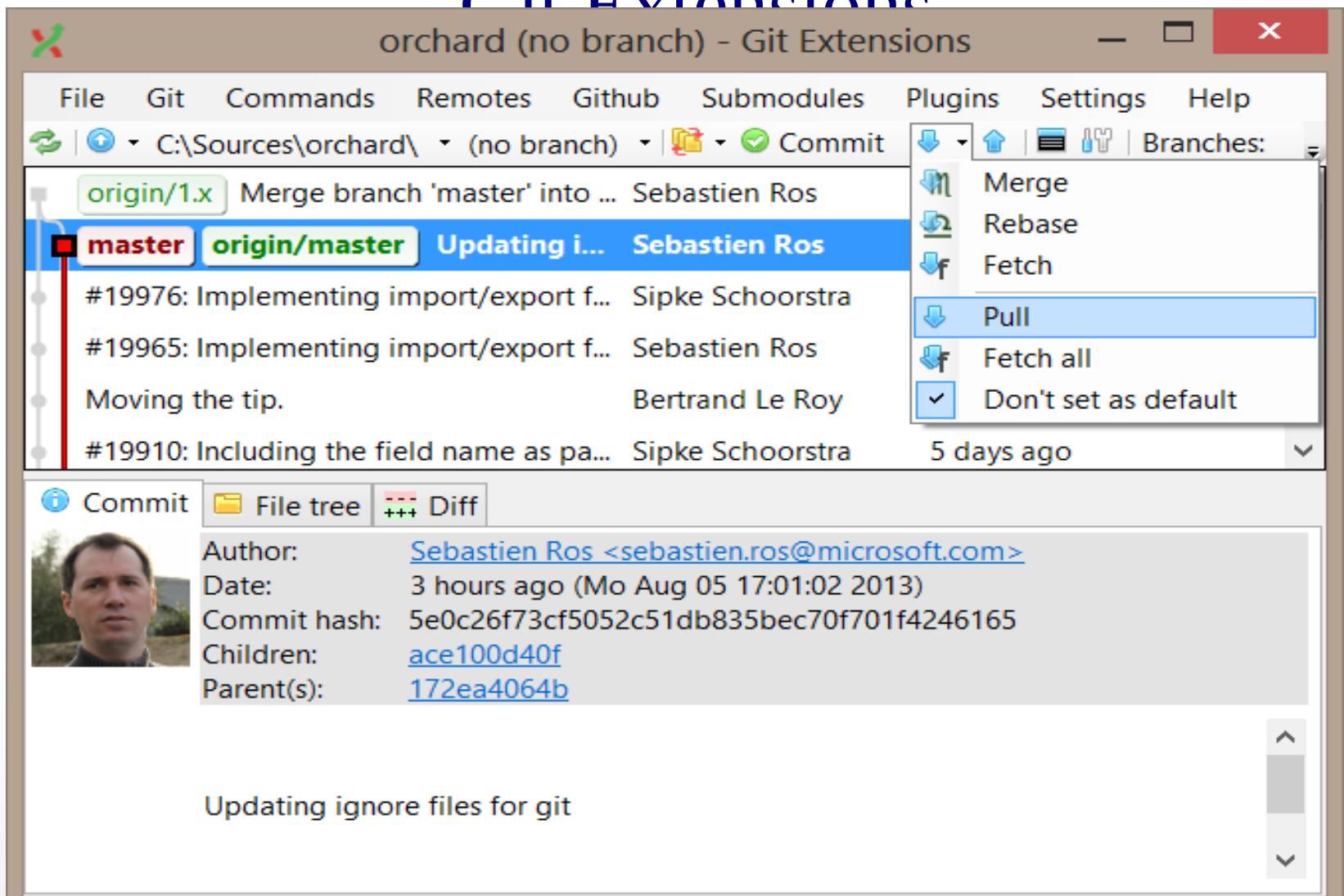
*\$ git revert {oznacenie zleho commitu -napr. Hash}*

- NIKDY nedeletujeme commity, ktoré sú už push-nuté! Niekoľko iných ich mohol pull-núť a zlý commit je v histórií jeho commitov. Vzdialený repozitár toto zmazanie preto zvyčajne nedovolí.

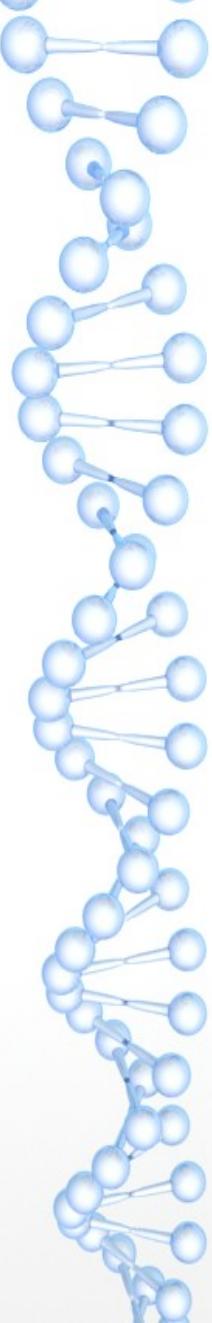




# 5. Rôzne GUI pre git -Windows - Git Extensions



# 5. rôzne GUI pre git -Windows -GitHub



The image shows a GitHub desktop application window. On the left sidebar, under the 'github' section, there is a list of repositories: 'niik' (with a profile picture) and 'reactiveui'. The 'reactiveui' repository is highlighted with a blue background. The main pane displays a list of repositories under the 'ReactiveUI' organization:

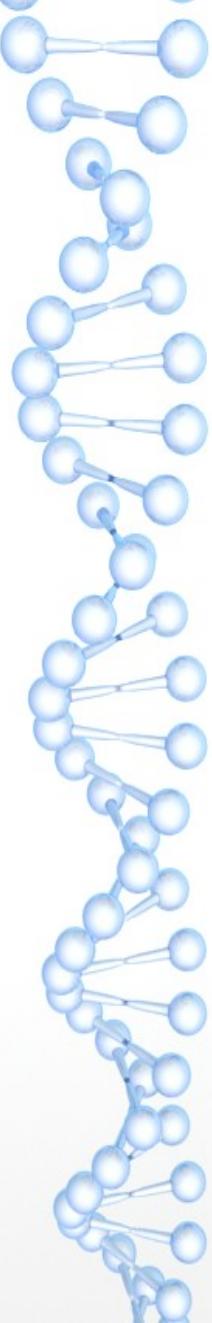
- reactiveui/ReactiveUI
- reactiveui/ReactiveUI.Samples
- reactiveui/RxUIProjectTemplate
- reactiveui/RxUI\_QCon

**ReactiveUI**

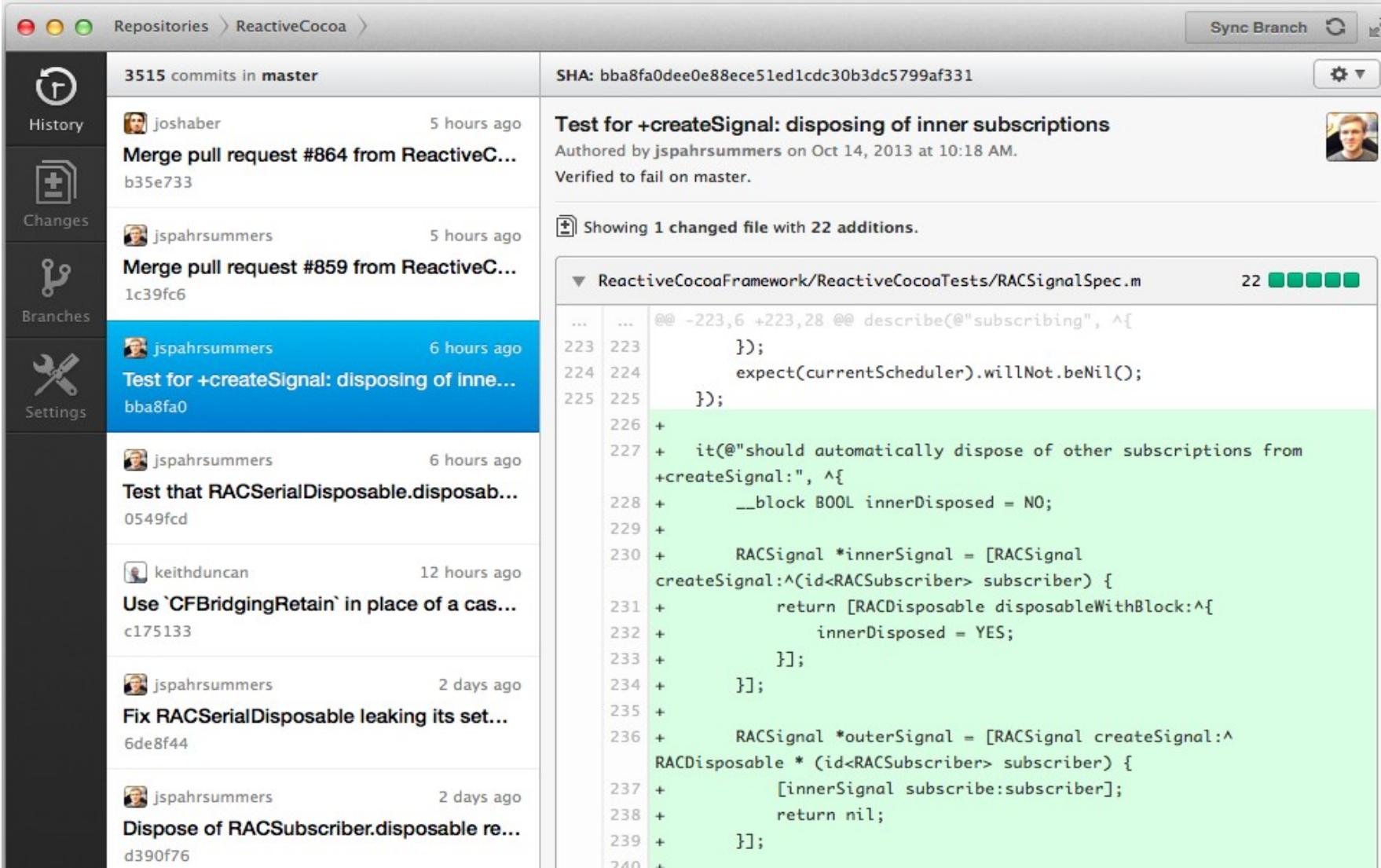
Use the Reactive Extensions for .NET along with Silverlight, WPF, or Windows Phone to create elegant, testable User Interfaces.

This library is organized into several high-level assemblies:

- **ReactiveUI** - Core library that doesn't rely on any particular UI framework. `ReactiveObject`, the base `ViewModel` object, as well as `ReactiveCollection`, a more awesome `ObservableCollection`, is in here.
- **ReactiveUI.Xaml** - Classes that require references to a Xaml'y framework, like WPF or WinRT. `ReactiveCommand`, an implementation of `ICommand`, as well as the `UserError` classes are in this assembly.
- **ReactiveUI.Blend** - This class has several Blend Behaviors and Triggers that make attaching `ViewModel` changes to Visual State Manager states.
- **ReactiveUI.Routing** - A screens and navigation framework as well as `ViewModel` locator. This framework helps you to write applications using IoC containers to locate views, as well as navigating back and forwards between views.



# 5. Rôzne GUI pre git -MacOS -GitHub



The screenshot shows the GitHub for Mac application interface. The title bar reads "Repositories > ReactiveCocoa >". On the left, there's a sidebar with icons for History (a circular arrow), Changes (a document with a plus), Branches (a gear), and Settings (a wrench). The main area displays a list of commits in the "master" branch:

- 3515 commits in master (SHA: bba8fa0dee0e88ece51ed1cdc30b3dc5799af331)
- joshaber 5 hours ago: Merge pull request #864 from ReactiveCocoa/b35e733
- jspahrsummers 5 hours ago: Merge pull request #859 from ReactiveCocoa/1c39fc6
- jspahrsummers 6 hours ago: Test for +createSignal: disposing of inner subscriptions (bba8fa0) - This commit is highlighted with a blue background.
- jspahrsummers 6 hours ago: Test that RACSerialDisposable.disposable...
- keithduncan 12 hours ago: Use `CFBridgingRetain` in place of a cas...
- jspahrsummers 2 days ago: Fix RACSerialDisposable leaking its set...
- jspahrsummers 2 days ago: Dispose of RACSubscriber.disposable re...

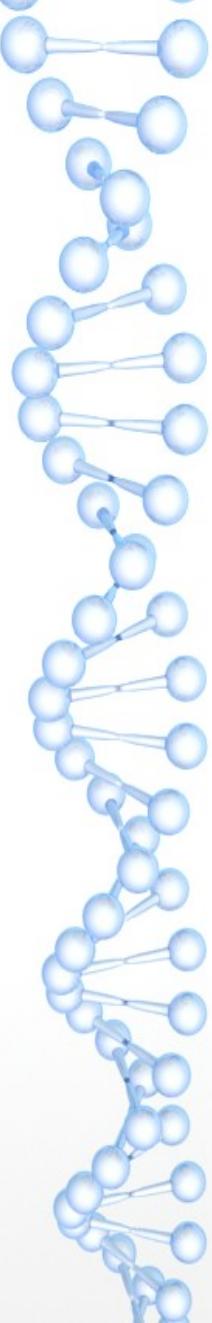
Details for the highlighted commit (bba8fa0):  
Author: jspahrsummers on Oct 14, 2013 at 10:18 AM.  
Verified to fail on master.

Showing 1 changed file with 22 additions.

ReactiveCocoaFramework/ReactiveCocoaTests/RACSignalSpec.m

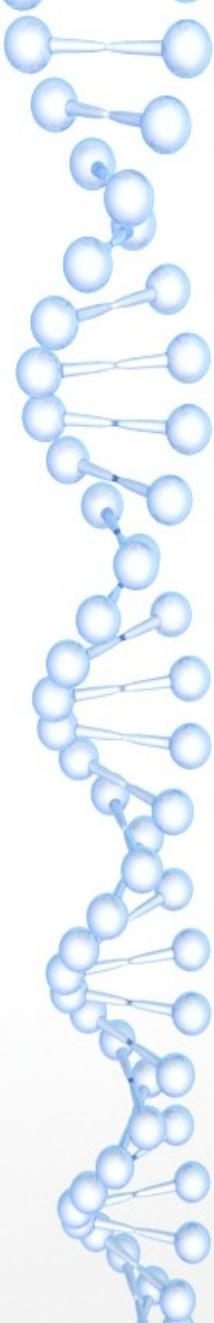
```
@@ -223,6 +223,28 @@ describe(@"subscribing", ^{
223     expect(currentScheduler).willNot.beNil();
224   });
225 });
226 +
227 + it(@"should automatically dispose of other subscriptions from
228 + +createSignal:", ^{
229 +   __block BOOL innerDisposed = NO;
230 +
231 +   RACSignal *innerSignal = [RACSignal
232 + createSignal:^(id<RACSubscriber> subscriber) {
233 +     return [RACDisposable disposableWithBlock:^{
234 +       innerDisposed = YES;
235 +     }];
236 +
237 +     RACSignal *outerSignal = [RACSignal createSignal:^{
238 +       RACDisposable * (id<RACSubscriber> subscriber) {
239 +         [innerSignal subscribe:subscriber];
240 +       return nil;
241 +     }];
242 +   }];
243 + }
```

# 5. Rôzne GUI pre git -Linux -qgit



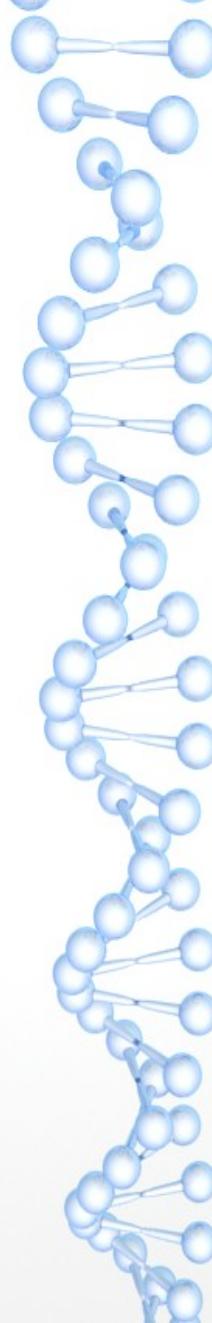
The image shows a screenshot of the qgit graphical user interface for Git. The interface has a menu bar with File, Edit, View, Actions, and Help. A toolbar above the main window includes icons for file operations like Open, Save, and Commit. The main area is divided into several panes:

- Git tree:** On the left, it shows the project's directory structure with files like deaf, core, dbDumps, logs, static, .gitignore, countries.json, data\_class\_m..., howToStart.txt, langs.json, lessons-data...., LICENSE, manage.py, nepocujuci\_ci..., nepocujuci\_ci..., nepocujuci.xmi, README.md, and requirements.txt.
- Rev list:** This pane contains a list of commits. The commit "v0.122 Feature #2209" is selected and highlighted in red. Other commits listed include "Working directory changes", "master fixed the test for 'make all parameters optional' in UserSet", "origin/master v0.123 fix Feature #2209", "v0.121 removed required params for UserSet method", "v0.120 fixed security -using requests", "changed getting videoSize through requests module", "updated token and validation way", "added requests as dependency", "v0.119 less verbose methods for data classes", and "SECURING the external https call (FB API, iTunes); fixed DictionarySe...".
- Graph:** To the left of the Rev list, there is a small graph icon.
- Short Log:** This tab is selected in the Rev list pane.
- Author:** This tab is also present in the Rev list pane.
- Author Date:** This tab is also present in the Rev list pane.
- Code Diff:** The bottom half of the interface displays the code differences between the current commit (v0.122) and its previous version (v0.121). The diff shows additions and deletions in the core/admin.py file, specifically in the admin.site.register\_view() function and the GetLanguages class.
- Log Message:** Below the code diff, a log message for the v0.122 commit is shown: "SECURING the external https call (FB API, iTunes); fixed DictionarySe...".
- Tag:** At the bottom left, it says "Tag: v0.122 [Feature #2209]".

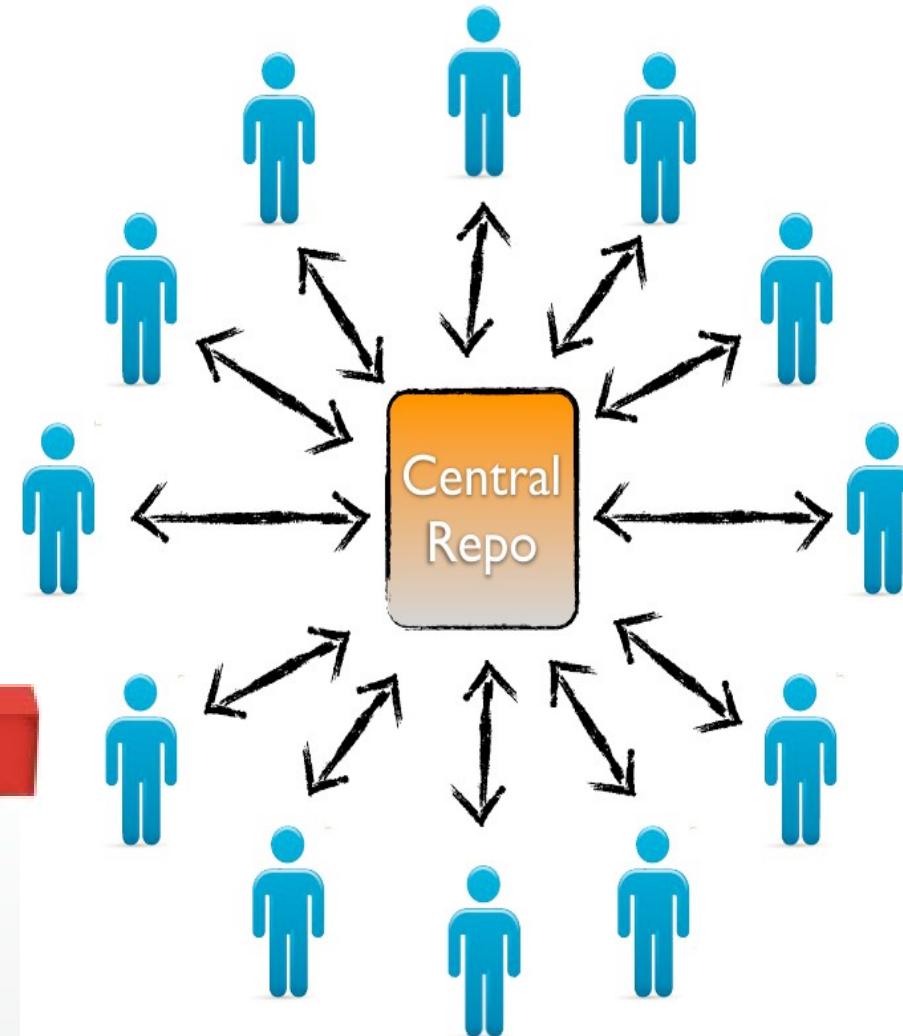


## 6. „Tvar“ developerskej komunity

- **Centrálny repozitár** (rovný s rovným) -zvykne byť pri closed source repozitároch
- **Diktátor** (niekto zaraďuje kód do hlavného stromu)  
-bežné pri OpenSource projektoch; ak sa ukáže diktátor zlý, urobí sa fork =iný „diktátor“ si založí vlastnú vetvu
- Vždy môže byť s **Mirroringom** (server má svoj mirror  
-podobné CDN / cloutu) -ponúka napr. SourceForge;  
lepšia geografická distribúcia
- Existuje aj možnosť miešaného closed +open vývoja



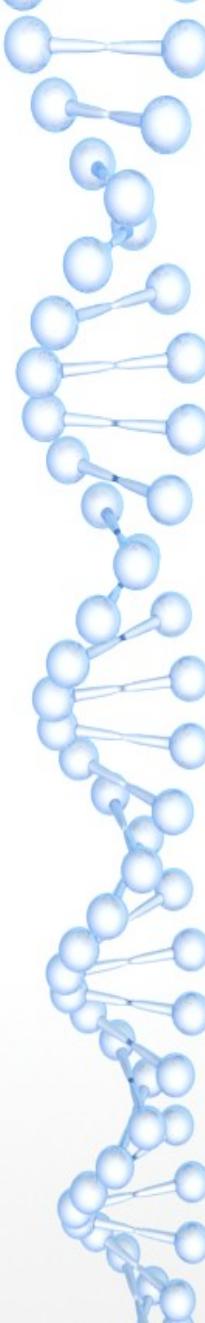
## 6. „Tvar“ dev komunity -centrálny renozitár



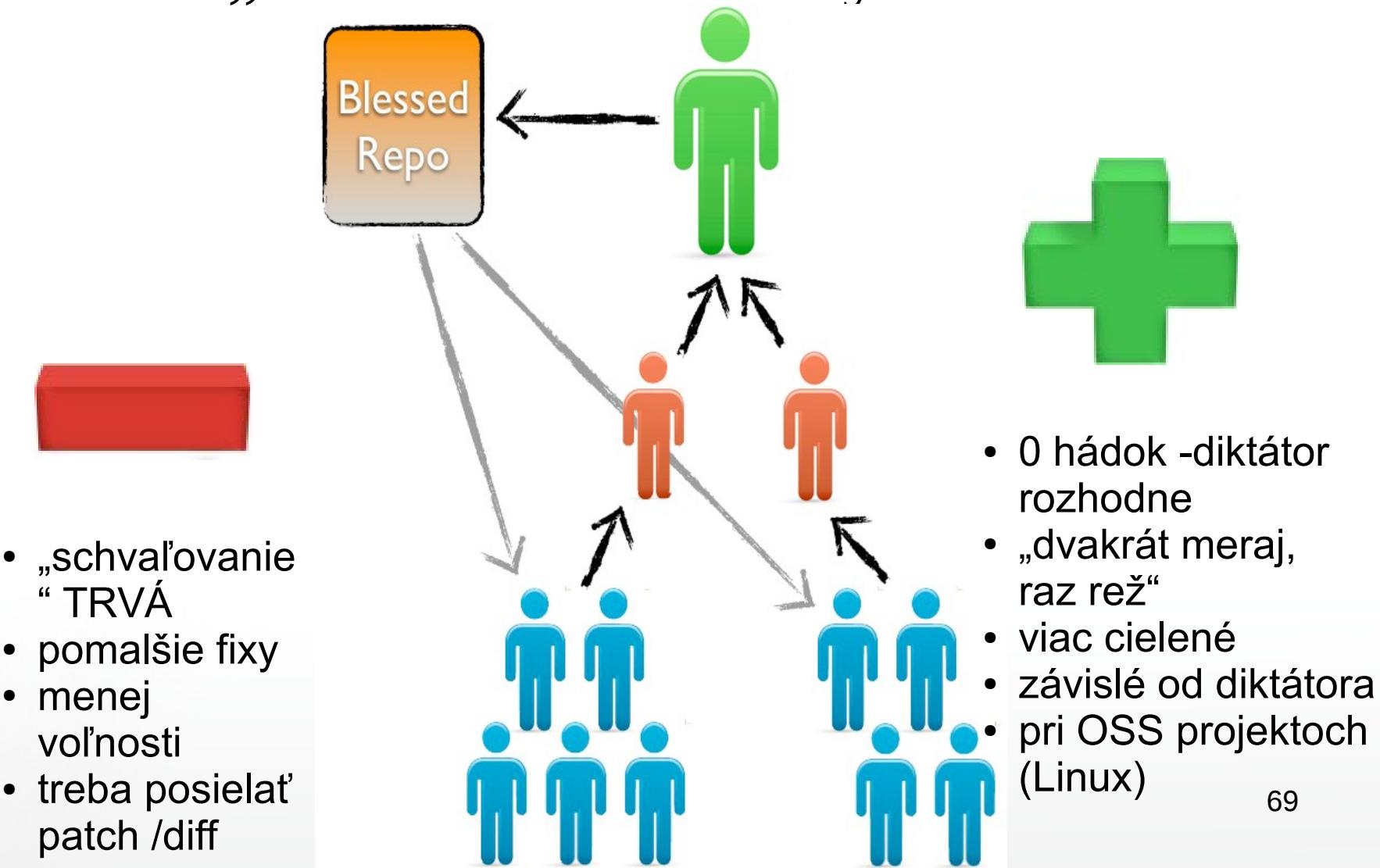
- stabilita buildu
- kolízie závýmu

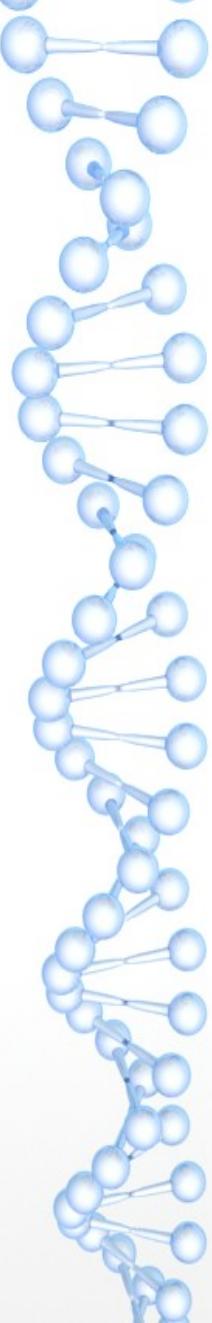


- rýchlosť
- jednoduchosť
- v TB internet banking<sup>68</sup>

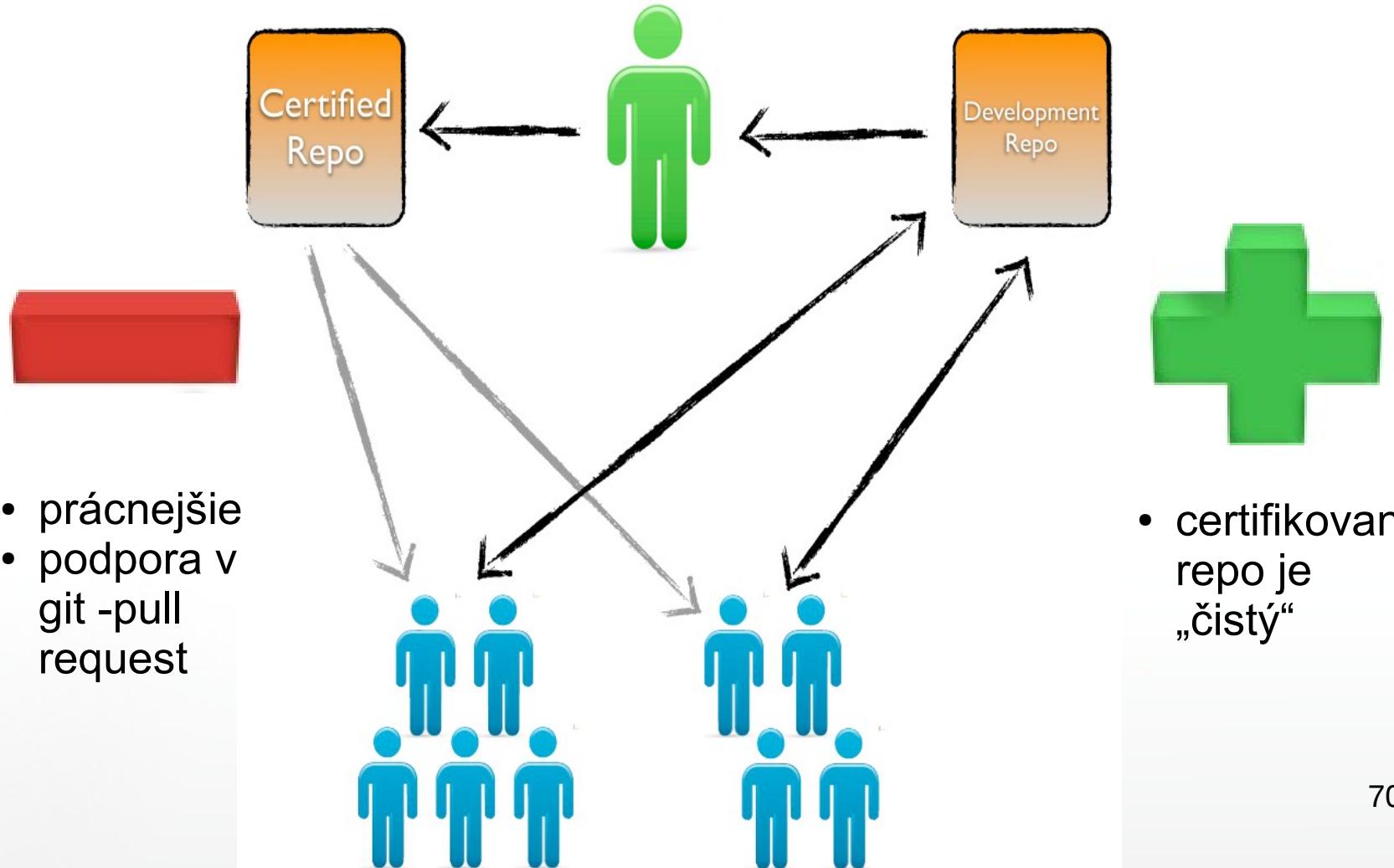


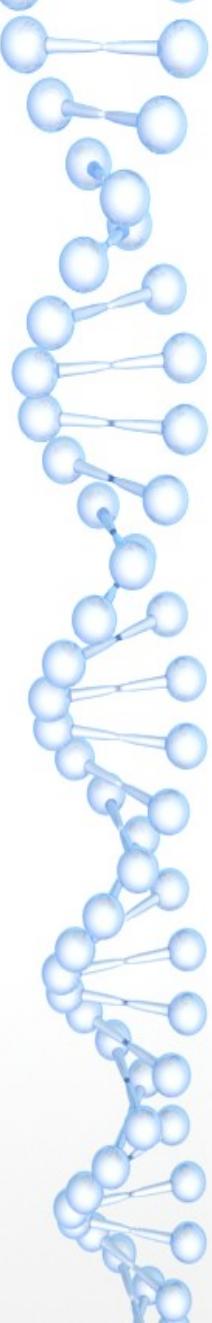
## 6. „Tvar“ dev komunity -diktátor



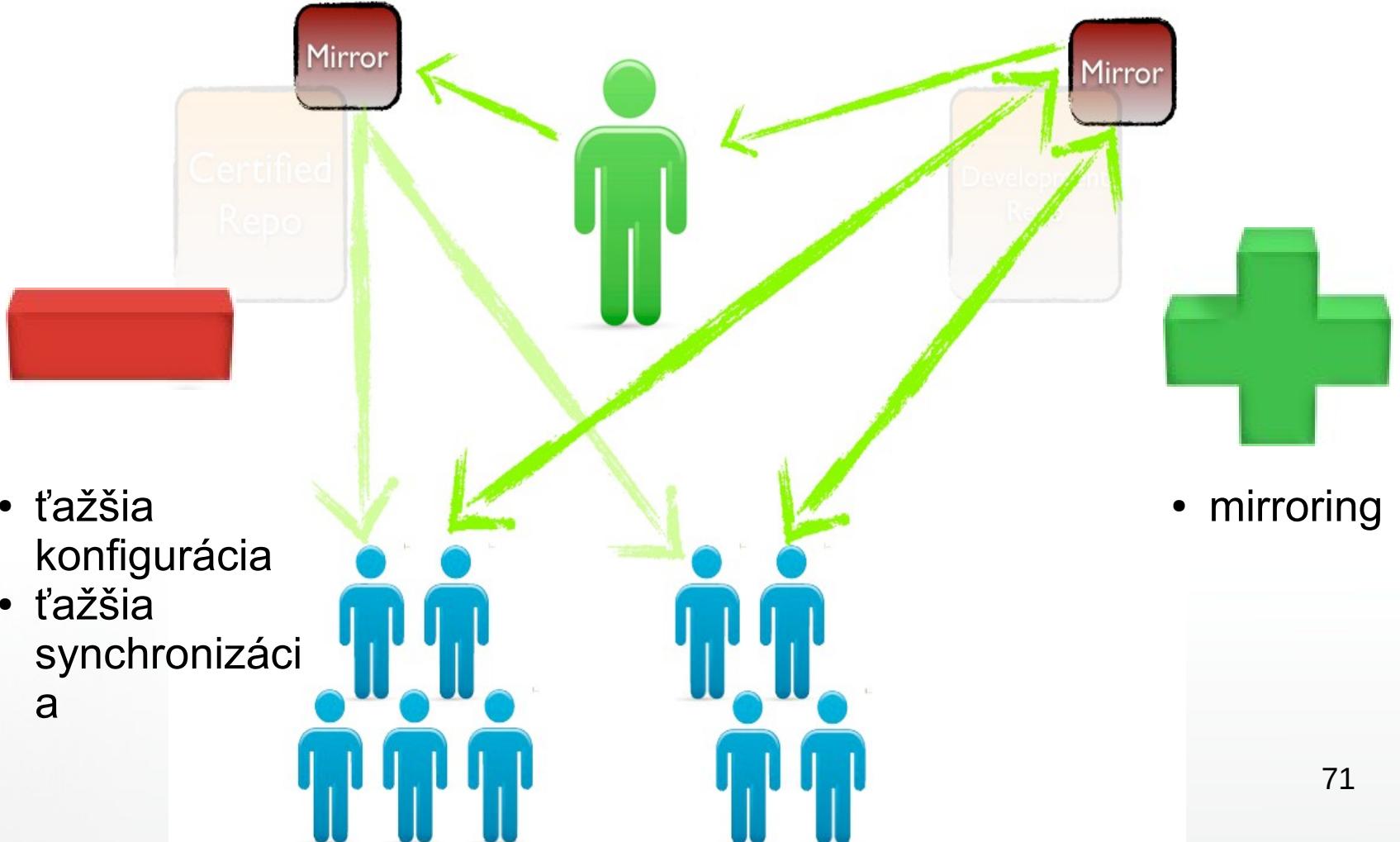


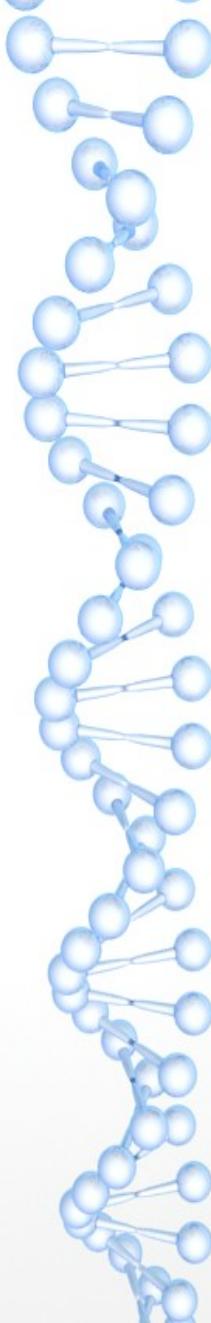
## 6. „Tvar“ dev komunity -diktátor v2



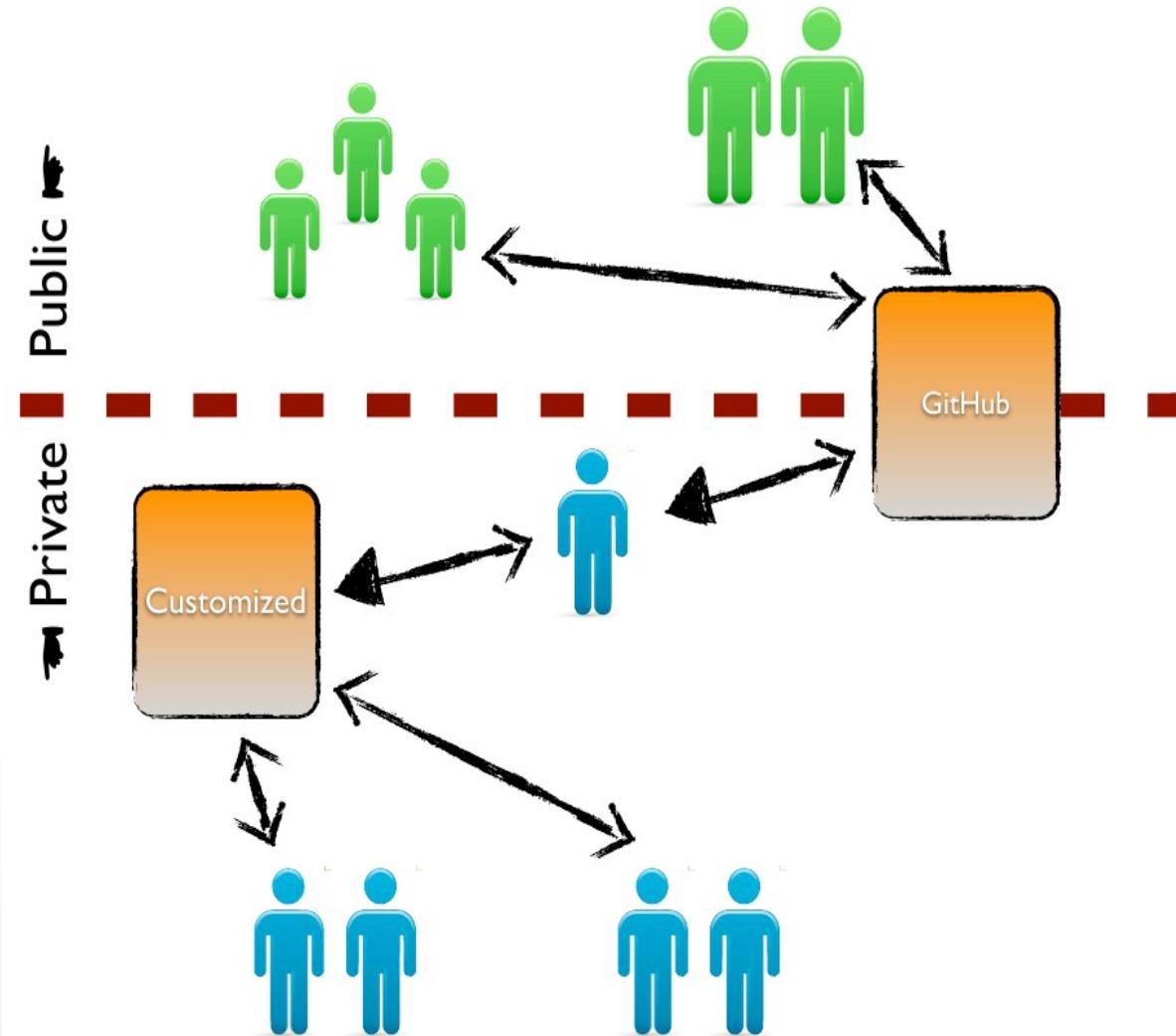


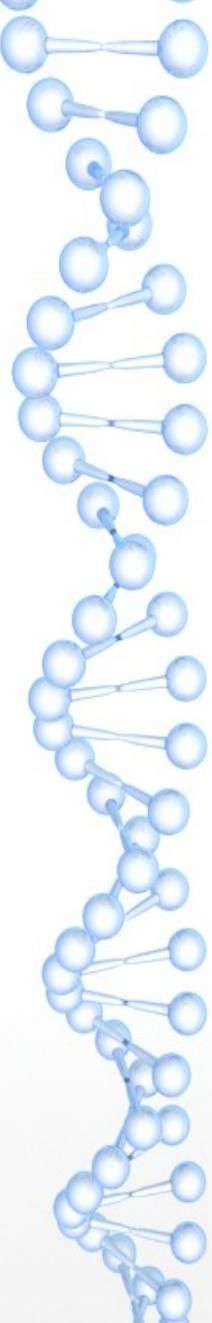
## 6. „Tvar“ dev komunity -diktátor v2 +mirror





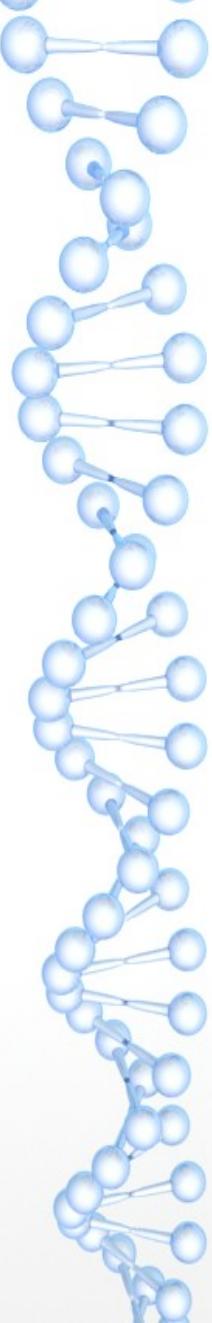
## 6. „Tvar“ dev komunity -open +closed





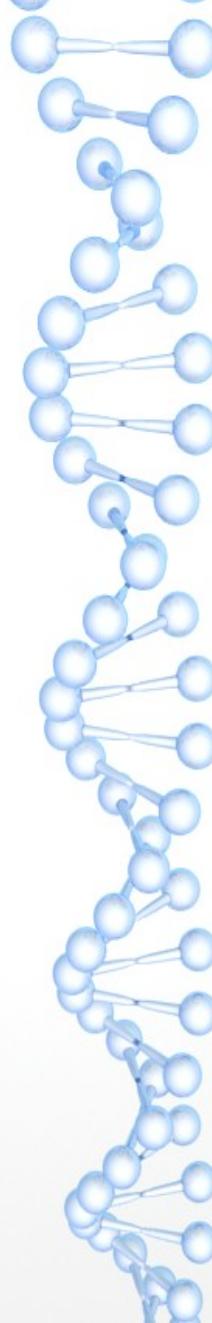
## 6. Sumár (alebo „Čo už vieme“)

1. V čom je git dobrý
2. Ako vnútorne funguje GIT
3. Ako vytvoriť a naklonovať GIT repozitár
4. Ako komunikovať so vzdialeným repozitárom
5. Ako commitnúť zmeny, pridať súbor, merge-ovať vetvy
6. Ako napraviť najzvyčajnejšie chyby
7. Ktorý GUI nástroj je fajn

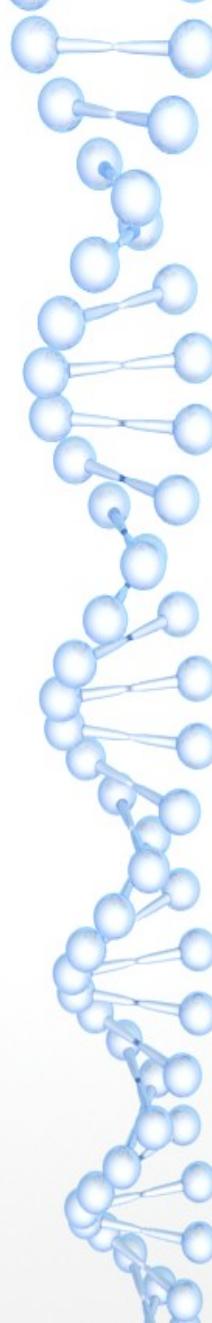


## 7. Záverom

- Najvhodnejšia príručka ku GIT je *Pro GIT* od Scotta Chacona: je v češtine, používa veľa obrázkových príkladov, ľahko čitateľná, obsahuje takmer celú user dokumentáciu GIT-u
- Git dokáže robiť len *automatizovanú* prácu  
-nemožno od neho čakať „telepatiu“ (viď Merge-ovanie)
- GUI nástroje sú fajn, ale vždy je lepšie vidieť im na prsty



**Otázky?**



Ďakujem za pozornosť!