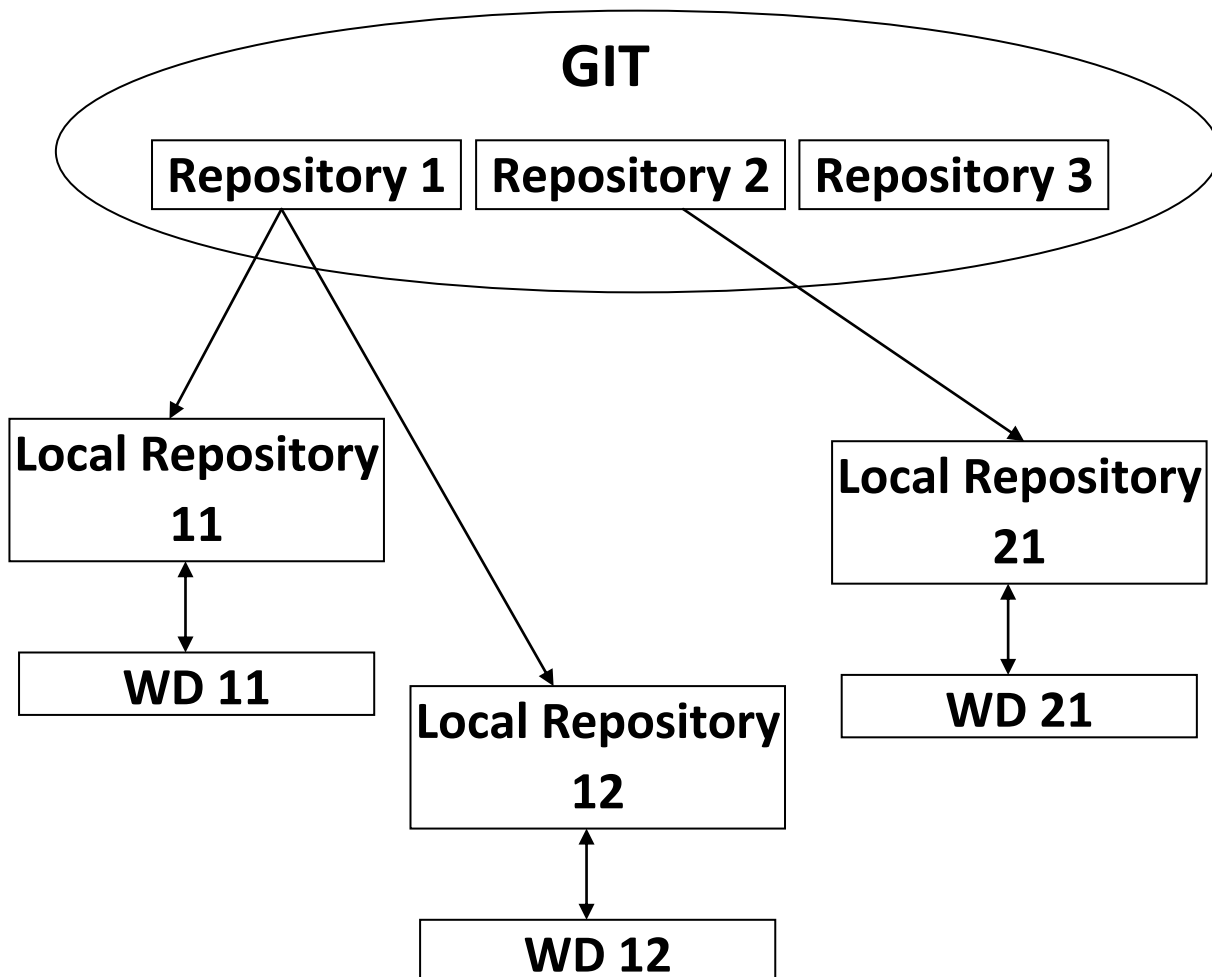


Obsah

GIT	2
Příkazy	3
Nástroje pro práci s GIT	4
Větvení - Git flow	10

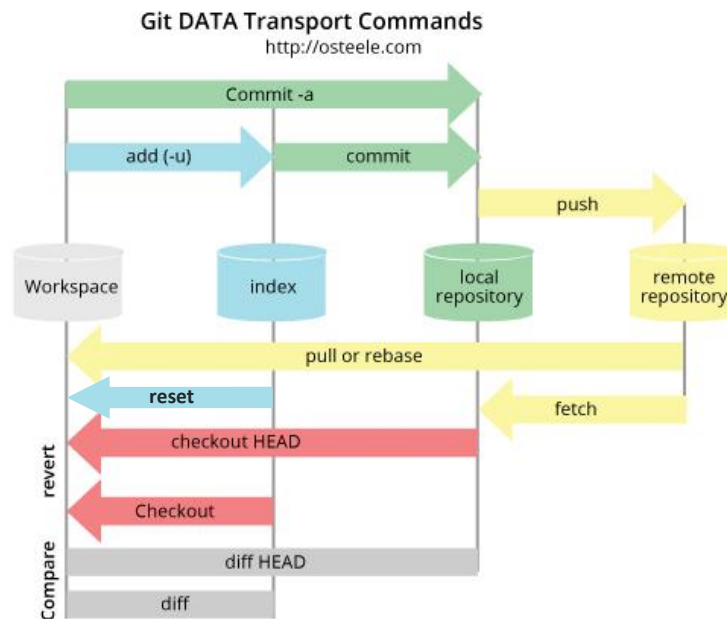
GIT

GIT je distribuovaný verzovací systém původně vyvinutý pro správu verzí jádra operačního systému Linux. Tento systém pro správu verzí je využíván od roku 2005. Dalo by se říct, že postatou GITu je jeho distribuovanost (každý vývojář má u sebe lokální kopii celé historie vývoje, na rozdíl od CVS - tam je historie pouze na serveru). Každá kopie repositáře je tak vlastně kompletní zálohou centrálního/vzdáleného repositáře.



Lokální kopie = lokální repositář + pracovní adresář (working directory WD)

Mezi lokálním repositářem a pracovním adresářem je ještě tzv. index. Do něj přidáváme nebo odebíráme modifikované soubory, které chceme přenést z pracovního adresáře do lokálního repositáře. Tím dosáhneme toho, že budeme přenášet pouze vybrané soubory, nikoli všechny modifikované soubory.



Základní příkazy:

add - přidání modifikovaného nebo nového souboru na index

commit - přenesení souborů z indexu do lokálního repositáře

commit -a - přenesení všech modifikovaných a nových souborů do lokálního repositáře

push - přenesení změn z lokálního do vzdáleného repositáře

reset - odebrání souborů/ů z indexu

pull/rebase/clone - přenesení změn ze vzdáleného repositáře do pracovního adresáře

fetch - přenesení změn ze vzdáleného do lokálního repositáře

checkout/checkout HEAD - přenos změn z lokálního repositáře do pracovního adresáře

diff - porovnání souboru v pracovním adresáři s jeho kopií/vzorem v lokálním repositáři

Konflikty se řeší pouze při příkazů pull. Git sám udělá merge na základě reverzního algoritmu, v konfliktních souborech vyznačí změny a je nutné merge provést ručně - v používaném editoru nebo pomocí mergovacího nástroje.

Podrobnější popis všech příkazů gitů naleznete na <http://ndpsoftware.com/git-cheatsheet.html>.

Nástroje pro práci s GIT

Stejně jako u CVS se nejčastěji používají příkazy přímo na příkazové řádce, ale opět existují grafické nástroje pro práci s GITem - např. SmartGIT, TortoiseGIT, GITk, gitea. Případně mohou být přímo příkazy verzovacího systému GIT součástí nějakého vývojového prostředí - viz Oracle SQL developer nebo Netbeans.

Příkazová řádka

```
locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   a.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        abc_substring_bug.cpp
        user@server
no changes added to commit (use "git add" and/or "git commit -a")

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ |
```

Příkazem *status* jsme si vypsalí stav pracovního adresáře. Výpis rozlišuje mezi modifikovanými soubory a novými soubory, o nichž neví ani index, natož lokální či vzdálený repositář.

Příkazem *diff* si pak můžeme zobrazit rozdíl mezi modifikovaným souborem a souborem uloženým v lokálním repositáři.

```
locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git diff a.txt
warning: LF will be replaced by CRLF in a.txt.
The file will have its original line endings in your working directory
diff --git a/a.txt b/a.txt
index 414a23a..7e7aebc 100644
--- a/a.txt
+++ b/a.txt
@@ -1,9 +1,17 @@
-idsjasfl;asj
- sdf
- asf
+sf
+ asdf
+
+ as
+ fas<p>Lorem ipsum dolor sit amet consectetur molestie ac Quisque id Quisque. Nunc Curabitur non quam ante a habitasse.
ociis ligula eget et mollis. Convallis id tincidunt consequat congue Donec ultrices Phasellus non lobortis nec. Elit vit
<p>Vitae parturient consequat odio mauris orci feugiat Ut vitae nulla Nulla. Nulla ac nec senectus eget hendrerit a met
t adipiscing malesuada. Adipiscing mus urna et Praesent volutpat risus euismod sagittis Sed justo. Lacinia ipsum morbi u
t sapien Curabitur metus lorem dui. </p>
-<p>Nascetur In nibh dui id elit mauris Curabitur sollicitudin vel id. Aenean cursus tempor Vestibulum pellentesque enim
is Praesent non condimentum. Eros odio et arcu amet Nam Phasellus neque feugiat Nulla Fusce. Nullam adipiscing et at par
-<p>Pretium lorem wisi Vestibulum aliquam sed metus Phasellus et convallis pretium. Suscipit eros Pellentesque sem Pelle
lorem Vestibulum dictumst elit et ac. Lacus eleifend at tristique at tellus et semper dui laoreet laoreet. Ante dictum v
em Vestibulum dui leo vitae eros gravida. Tempus leo elit.</p>
+
+ asdf
+ fsda
+ fas
+ dfas
+ f
+ sf
+ as
\ No newline at end of file

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$
```

Zeleně jsou označeny přidané řádky, červeně smazané. Místa beze změn zůstávají bílá.

Pokud chceme přidat změny do lokálního repositáře, pak je nejprve musíme přidat na index pomocí příkazu `add`. Z následujícího obrázku je vidět, že můžeme přidávat nejen jednotlivé soubory, ale i celé adresáře. Git už za nás vyřeší to, že na index z požadovaného adresáře přidá pouze modifikované či nové soubory.

```
locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git add *
warning: LF will be replaced by CRLF in a.txt.
The file will have its original line endings in your working directory

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a.txt
        new file:   abc_substring_bug.cpp

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$
```

Příkazu `commit` je nutné doplnit o log/komentář pro daný commit. Vložený komentář bude vložen ke všem souborům, kterých se commit týká.

```
locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git commit -m "oprava pokus a vložení automatu pro příjem abc"

[master fa1653f] oprava pokus a vložení automatu pro příjem abc
 2 files changed, 65 insertions(+), 3 deletions(-)
 create mode 100644 abc_substring_bug.cpp

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

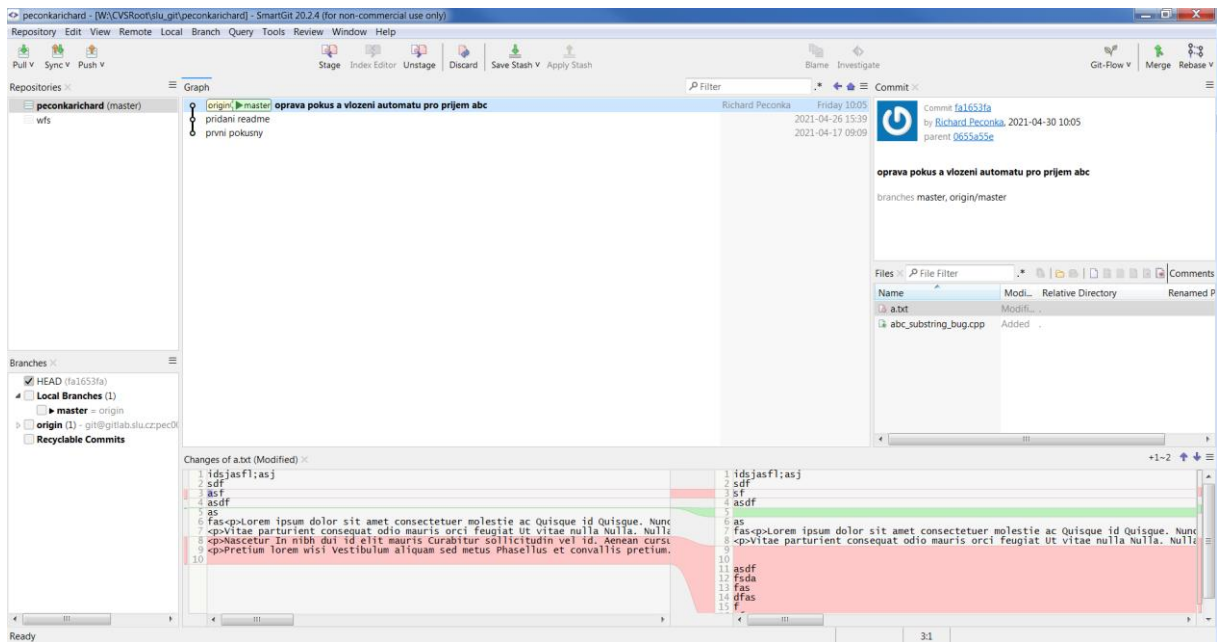
nothing to commit, working tree clean

locadmin@peconkatps MINGW64 /w/CVSRoot/slu_git/peconkarichard (master)
$
```

Git vypíše kolik souborů jsme commitovali a kolik změn náš commit vyvolal. Vypsáním status se pak dozvíme, že naše větev je 1 commit od stavu hlavní větve v centrálním repositáři a že máme provést příkaz `push`, abychom publikovali naše změny do vzdáleného repositáře a tím je zpřístupnili ostatním vývojářům v týmu.

SmartGIT

Program WinCVS je grafickým prostředím pro práci s verzovacím systémem GIT. Prostředí aplikace je v duchu vývojových prostředí. Prostředí aplikace je rozděleno do oken pro správu jednotlivých repositářů a větví v rámci konkrétního repositáře. Dále máme k dispozici graf dané větve, včetně všech commitů v dané větvi ve formě popisu commitu (message). U konkrétního commitu pak máme k dispozici informaci o uživateli i čase, kdy daný commit proběhl. Následuje informaci o souborech, jejichž změny byly commitem dotčeny, včetně zobrazení změn formou porovnání verzí daného souboru.



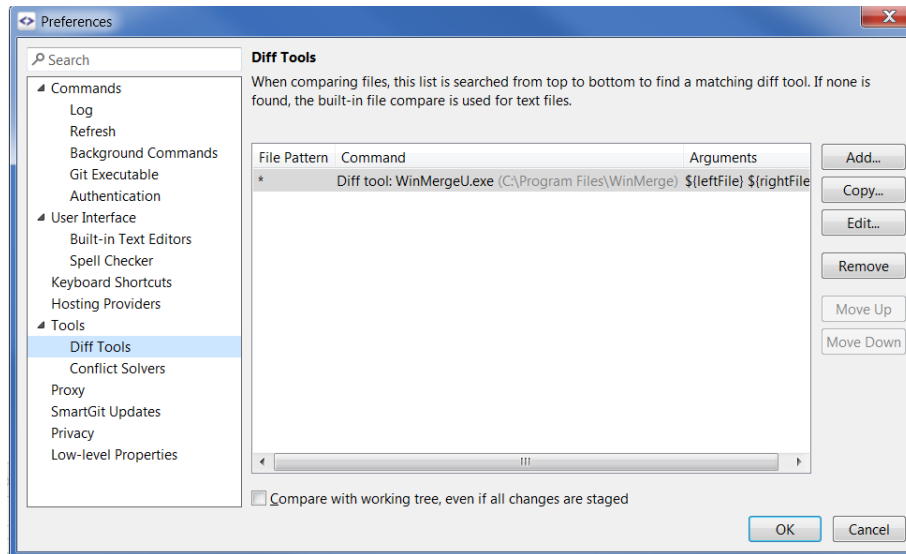
Jednotlivé příkazy jsou dostupné buď z hlavního menu aplikace nebo přes tlačítka na tlačítkové liště, příp. přes kontextové menu vyvolané přes pravé tlačítko myši. V rámci hlavního menu je odděleno menu pro přepínání mezi repositáři. Příkazy GITu jsou odděleny do skupin podle toho, zda pracujeme s lokálním repositářem (Local) nebo se vzdáleným repositářem (Remote). Zvláštní skupinu pak tvoří příkazy pro práci s větvemi (Branch). Do této skupiny je včleněn také příkaz pro tvorbu značek Add tag, pro vytváření releasů.

Remote	Local	Branch	Query	Tools	Review
Pull...				Alt+Page Down	
Fetch				Alt+Shift+Page Down	
Fetch More...					
Synchronize...				Ctrl+Y	
Push				Alt+Page Up	
Push To...				Alt+Shift+Page Up	
Push Up To...					
Add...					
Rename...				F2	
Delete...					
Submodule					
Subtree					
Set Depth...					
Properties...					

Local	Branch	Query	Tools	Review	Window	Help
Stage				Ctrl+T		
Unstage				Ctrl+Shift+T		
Index Editor				Ctrl+Alt+T		
Ignore...				Ctrl+I		
Edit Ignore File...						
Rename...				F2		
Toggle 'Assume Unchanged'						
Toggle 'Skip Worktree'						
Resolve...				Ctrl+Shift+V		
Commit...				Ctrl+K		
Use Message for Commit				Ctrl+Shift+K		
Reset...				Ctrl+R		
Reset Advanced...				Ctrl+Shift+R		
Undo Last Commit...				Ctrl+Alt+R		
LFS						
Discard...				Ctrl+Z		
Remove...				Ctrl+-		
Delete...				Delete		
Clean Working Tree...						
Stash All...				Ctrl+S		
Stash Selection...						
Apply Stash...				Ctrl+Shift+S		
Rename Stash...				F2		
Drop Stash...						
Run Garbage Collector...						

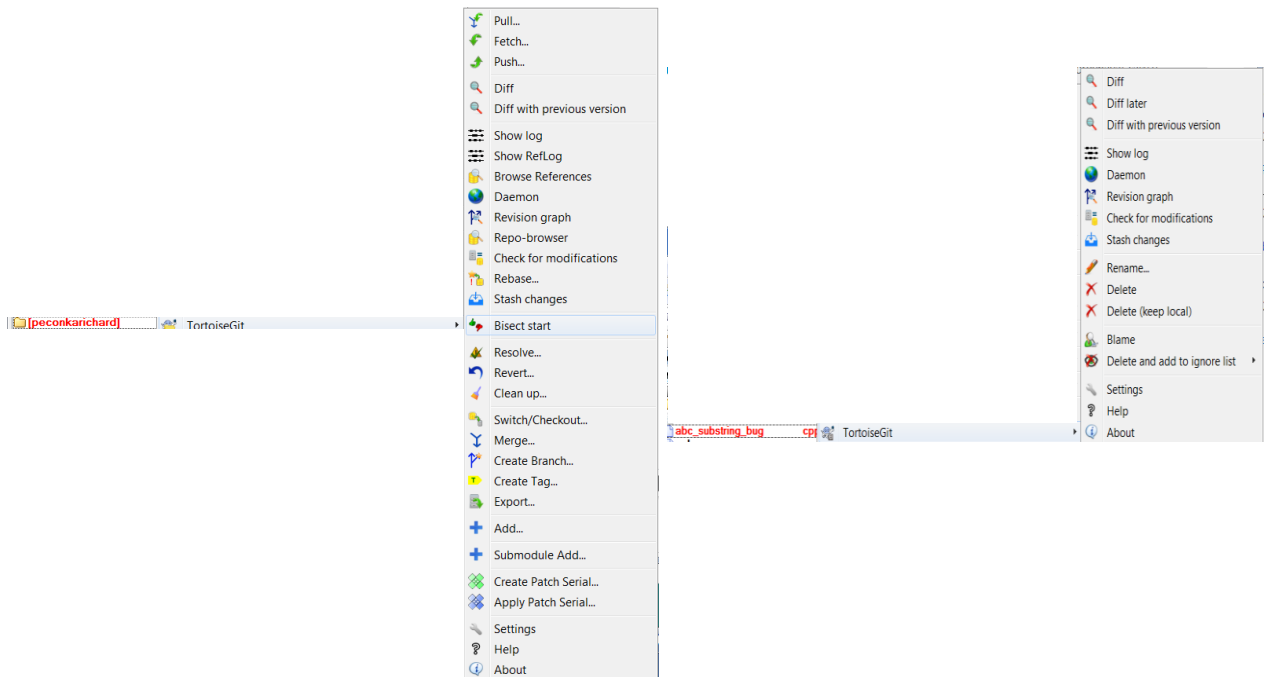
Branch	Query	Tools	Review	Window	Help
Git-Flow					
Check Out...				Ctrl+G	
Merge...				Ctrl+M	
Cherry-Pick...				Ctrl+Shift+M	
Revert...				Ctrl+Alt+M	
Rebase					
Bisect					
Abort					
Add Branch...				F7	
Add Tag...				Shift+F7	
Rename...				F2	
Delete...					
Edit Commit Message...				F2	
Edit Author...					
Squash Commits...				Ctrl+J	
Set Tracked Branch...					
Stop Tracking...					

V rámci nastavení si pak můžeme prostředí upravit, jak jsme zvyklí. Např. změnit jednotlivé klávesové zkratky nebo definovat svůj oblíbený externí nástroj pro porovnání změn jednotlivých souborů.

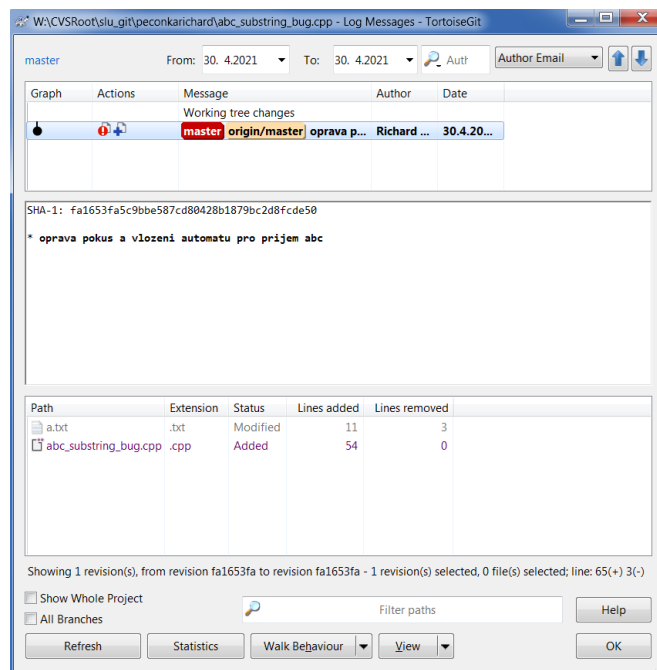


TortoiseGIT

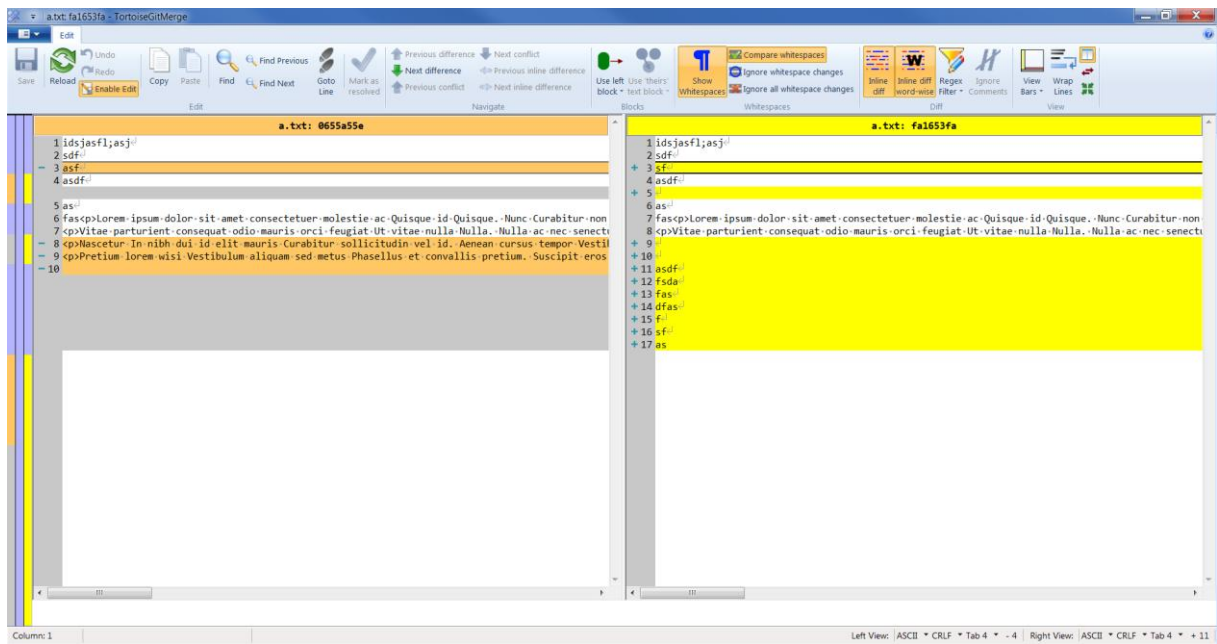
Prostředí TortoiseGIT, stejně jako TortoiseCVS, není ucelenou aplikací, jak jsme u vývojových prostředí zvyklí. Prostředí TortoiseGIT je dostupné z kontextového menu pro jednotlivé soubory a adresáře. Jeho obsah se mění podle toho, co vybraný adresář nebo soubor reprezentuje. Na rozdíl od TortoiseCVS je TortoiseGIT dostupný v rámci samostatného podmenu kontextového menu, ale i tak jednotlivé položky odpovídají kontextu a zobrazí se nám různá menu pro repositář či konkrétní soubor.



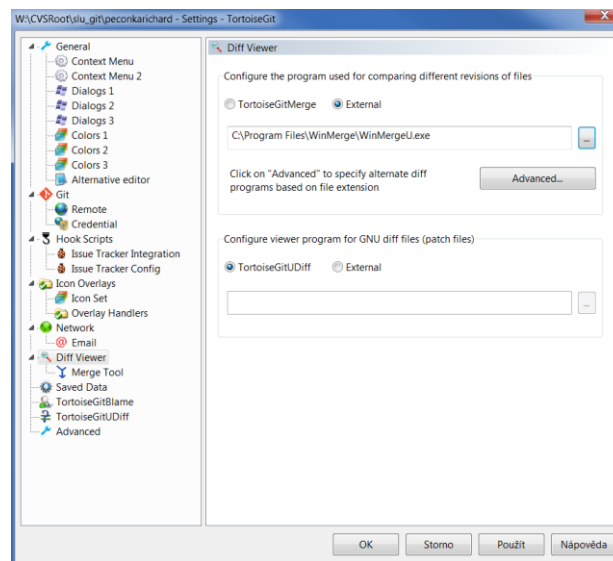
Historii jednotlivých commitů si pak můžeme zobrazit volbou Show log.



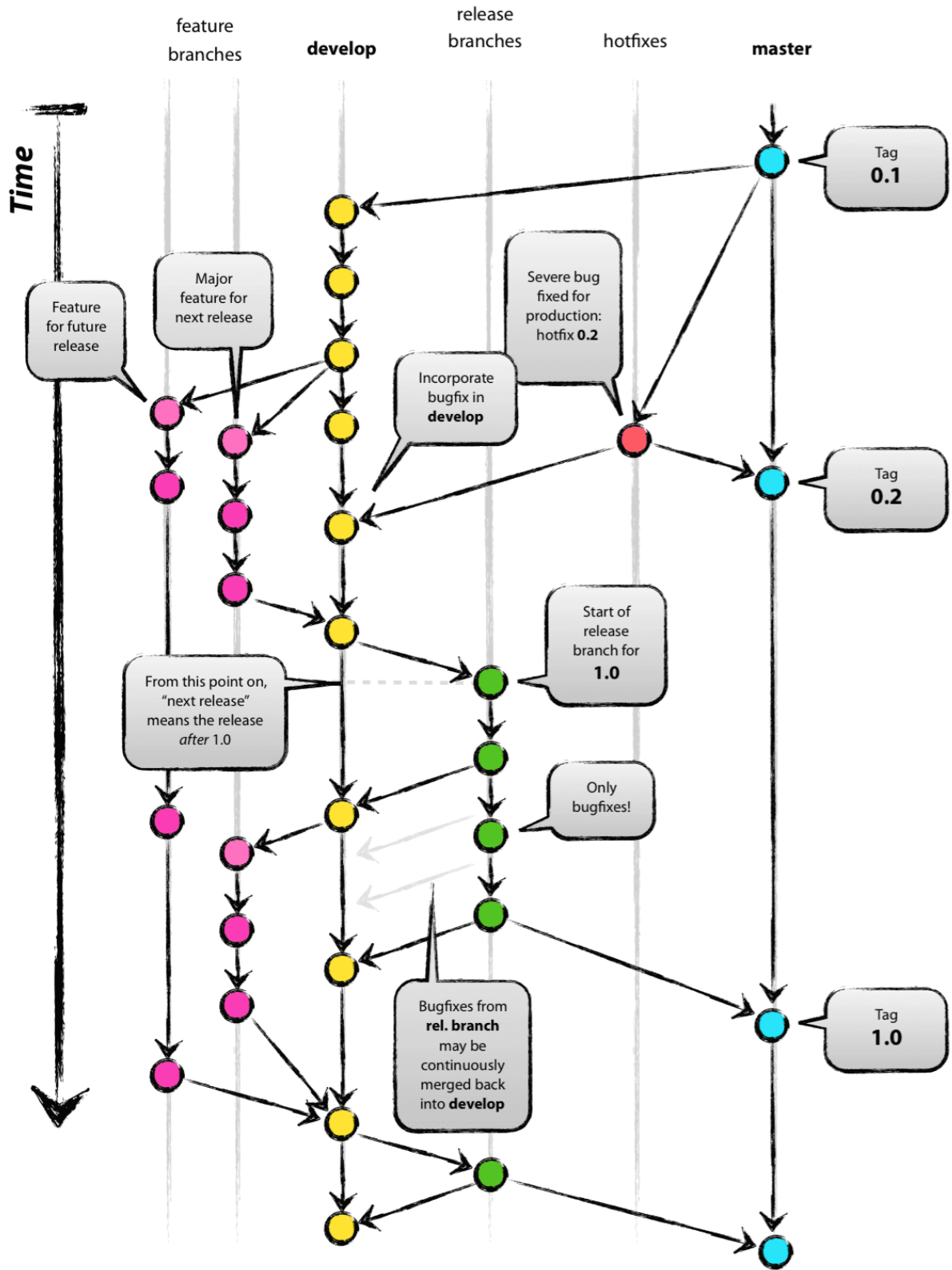
Z tohoto místa pak výběrem konkrétního souboru můžeme vyvolat interní program pro porovnání verzí TortoiseGitMerge.



Z obrázku je patrné, že operace tohoto srovnávacího nástroje voláme tlačítky z tlačítkové lišty. Funkce odpovídají standardním srovnávacím nástrojům, jak jsme již dříve ukazovali u programu WinMerge. V rámci nastavení programu můžeme samozřejmě změnit tento nástroj na libovolný externí program.



Větvení - Git flow



<https://nvie.com/posts/a-successful-git-branching-model/>