

# Git vcs - introduction

# *Agenda*

- VCS - overview
- Git snapshots and local changes
- SHA-1, changes markers
- Working directory, stage files and commit
- Git installation from sources
- Git configuration - global parameters
- Creating project - git init, adding changes and cloning - git clone
- .gitignore
- Branches - checkout, trivial and non-trivial merge, delete with and without changes
- Git commands

# Markers

- > some command
- >> output from command  
in brackets - additional information
- Q: - questions
- C: - study case
- E: - exercise

# *VCS overview*

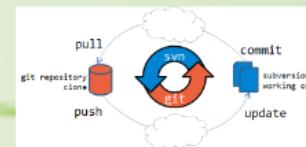
VCS definition - trivial

Helpful for developers

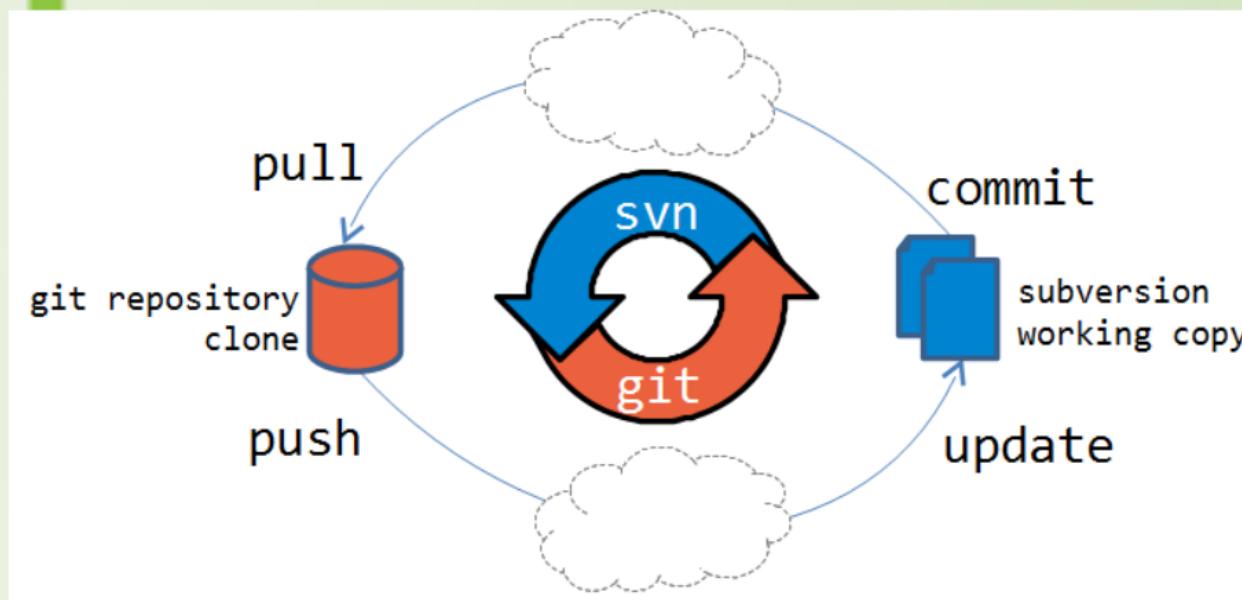
Examples (multi-dimensional) -  
ClearCase and Git

Basic flow:

- 1) My local changes
- 2) Publish changes
- 3) Update other changes



# push changes update other changes



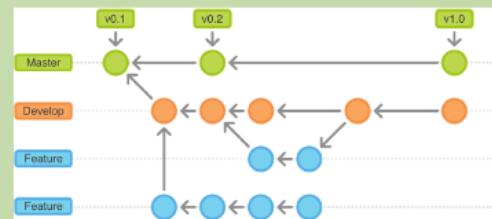
# *Git snapshots and local changes*

Two functions - client and server

client -> locally  
server -> remote repository

Q: is outgoing connection necessary during development process?

Client filesystem - based on snapshots



Q: is outgoing connection necessary  
during development process?

Client filesystem - based on snapshots



# ***SHA-1, changes markers***

SHA-1 as a checksum for each commit

HEAD as a pointer

Local, stage and remote

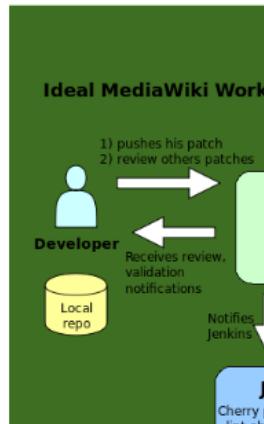
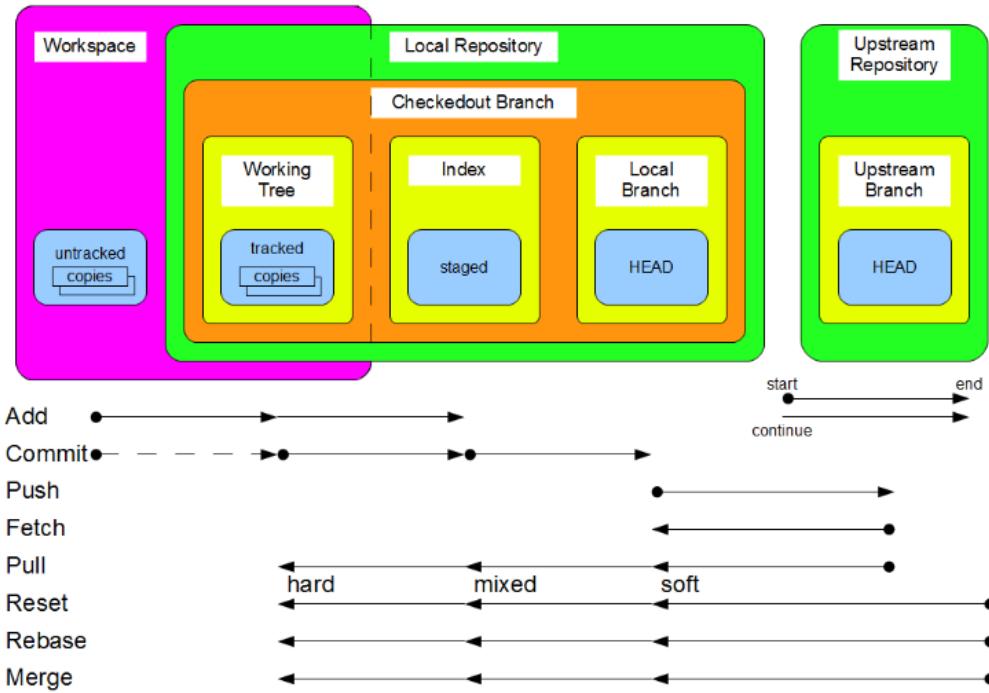
Basic Git flow:

- 1) Untracked
- 2) Tracked, staged
- 3) Commit and changing HEAD

E: comparison above flow with  
ClearCase (pseudo-code)



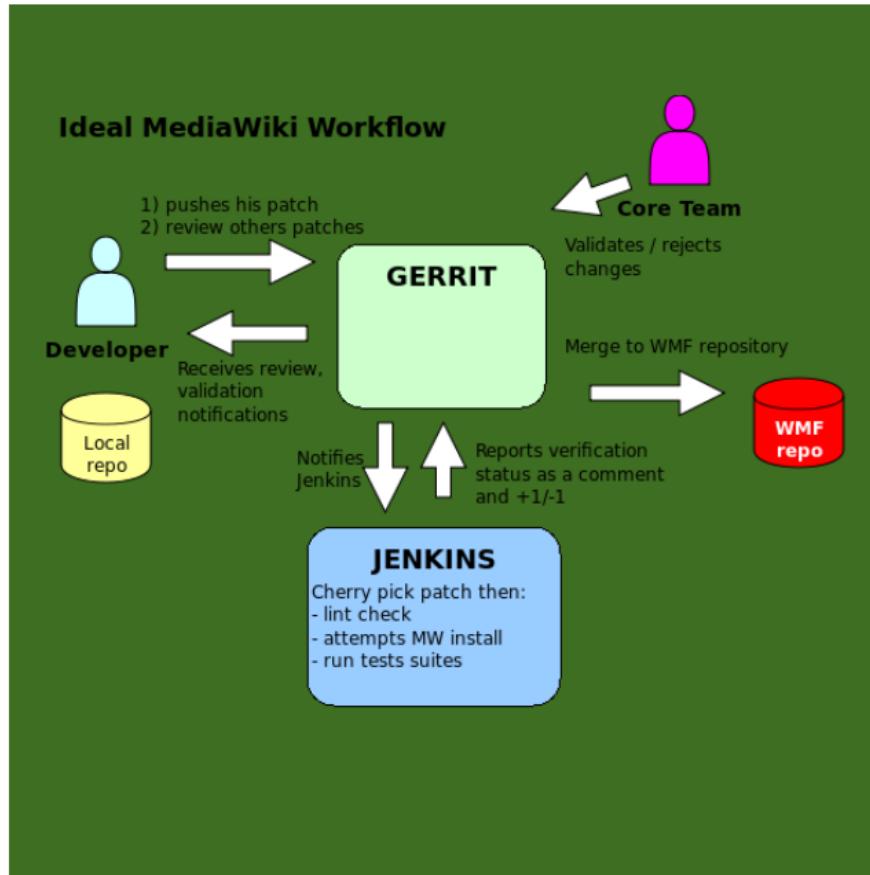
Add ●—  
Commit ●—  
Push  
Fetch  
Pull  
Reset  
Rebase  
Merge



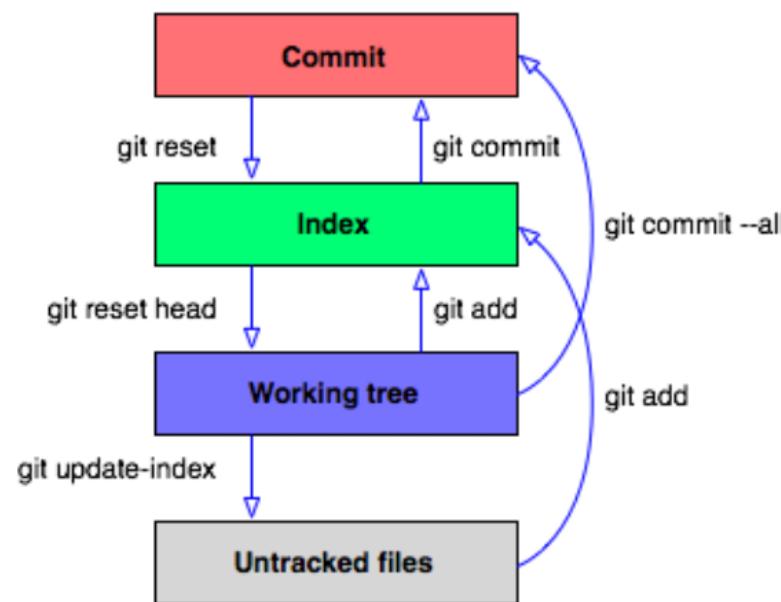
commit

0

1



# *Working directory, stage files and commit*





## *Git installation from the sources*

E: Check procedure for the newest Ubuntu on:  
<http://git-scm.com/book/en/Getting-Started-Installing-Git>

## *Git configuration - global parameters*

Signing configuration:

- > git config --global user.name "First Last"
- > git config --global user.email "first.last@domain.com"

Practical variables:

- > git config --global color.ui true
- > git config --global core.editor vi

## *Creating project - git init, adding changes and cloning - git clone*

Current directory on server became a repository

Adding .git metadata directory  
› git init

Repository is empty, changes can be set.

Now our repository can be cloned.

› git clone -b [branch-name] git@jittre.../name.git  
› git clone ssh://john@john.com/my-project.git

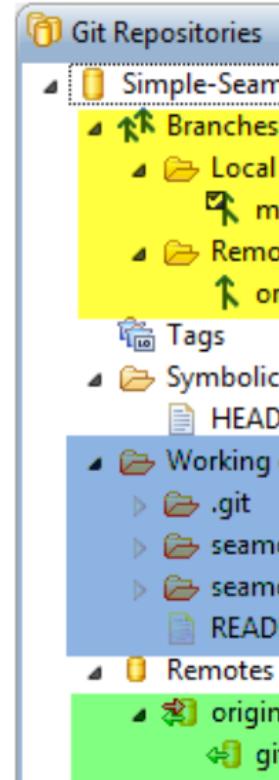
Q: is HTTP protocol available for connection?

# *.gitignore*

All files are uploaded to the repository -  
not always

.gitignore configuration file allows to  
create some kind of private files

```
.
├── datafiles
│   └── data.txt
└── .git
    ├── branches
    │   └── COMMIT_EDITMSG
    ├── config
    ├── description
    ├── HEAD
    ├── hooks
    ├── index
    ├── info
    ├── logs
    └── objects
        └── refs
            └── test01
            └── test02
            └── test03
```



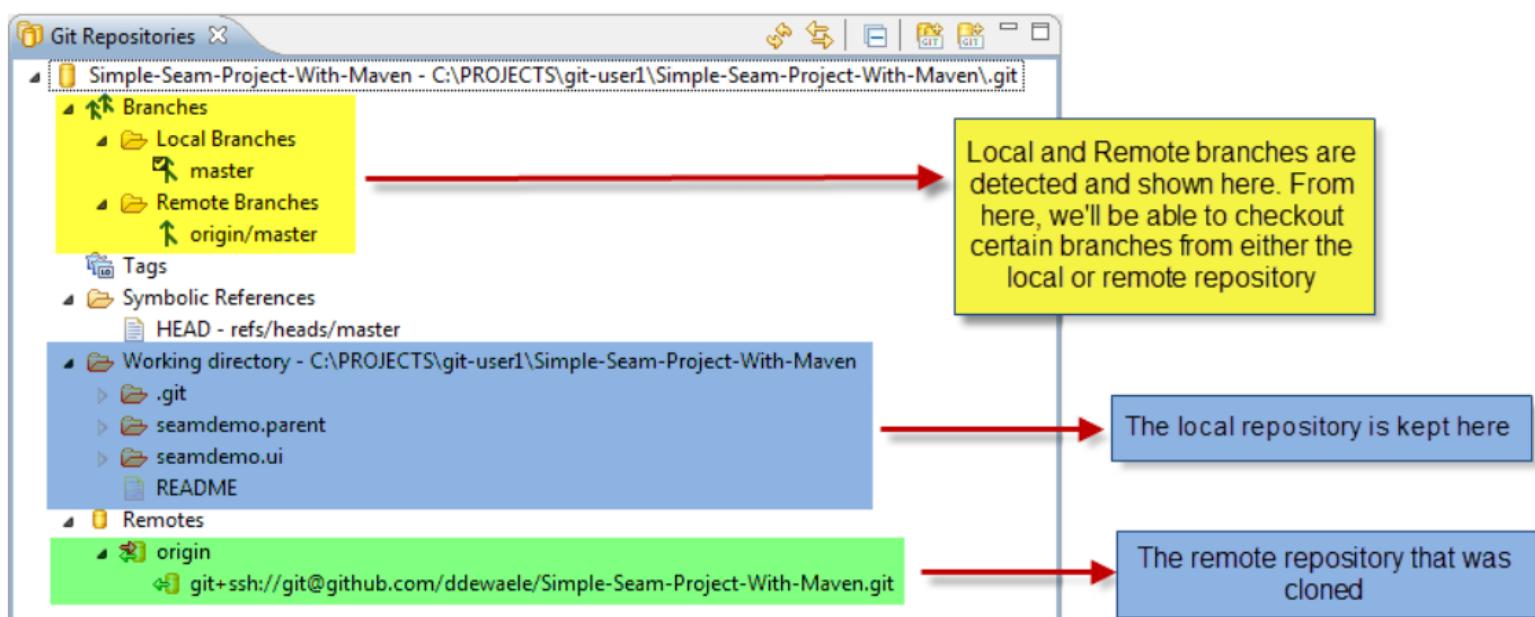
Q: what  
E: example

re

the repository -

s

file allows to  
private files



Q: what about .git directory and .gitignore?

E: example repository with .gitignore

## ***Branches - checkout, trivial and non-trivial merge, delete with and without changes***

branch - part of Git structure like stream in ClearCase

main branch - master (origin, remote)

checkout - switch between local branches

> git checkout testing

Q: which metadata files will be changed?

Creating and switching:

> git branch iss53

> git checkout iss53

Branches can be merged

E: two different branches with trivial merge

Unmerged s  
Q: when?

Merge after  
> git mergeto

Checking br  
> git branch

E: two differ  
non trivial m

Unmerged status for non-trivial changes

Q: when?

Merge after fix

> git mergetool

Checking branches

> git branch -v

E: two different branches with  
non trivial merge

ion  
commands?

## *Git commands*

**git diff/difftool** - simple or GUI tool for differences

> git difftool <SHA-1>...<SHA-1> filename

**git diff and git diff --staged** - comparing working directory with stage or stage with the snapshot

Q: are above command are always the same?

**git add** - adding new/modified files to stage

> git add -a

Q: change, add, change - what happened?

Working  
Directory

Test

**git commit** - commit changes from stage

> `git commit -m "Message"`

**git rm** - removing files from index

> `git rm --cached filename`

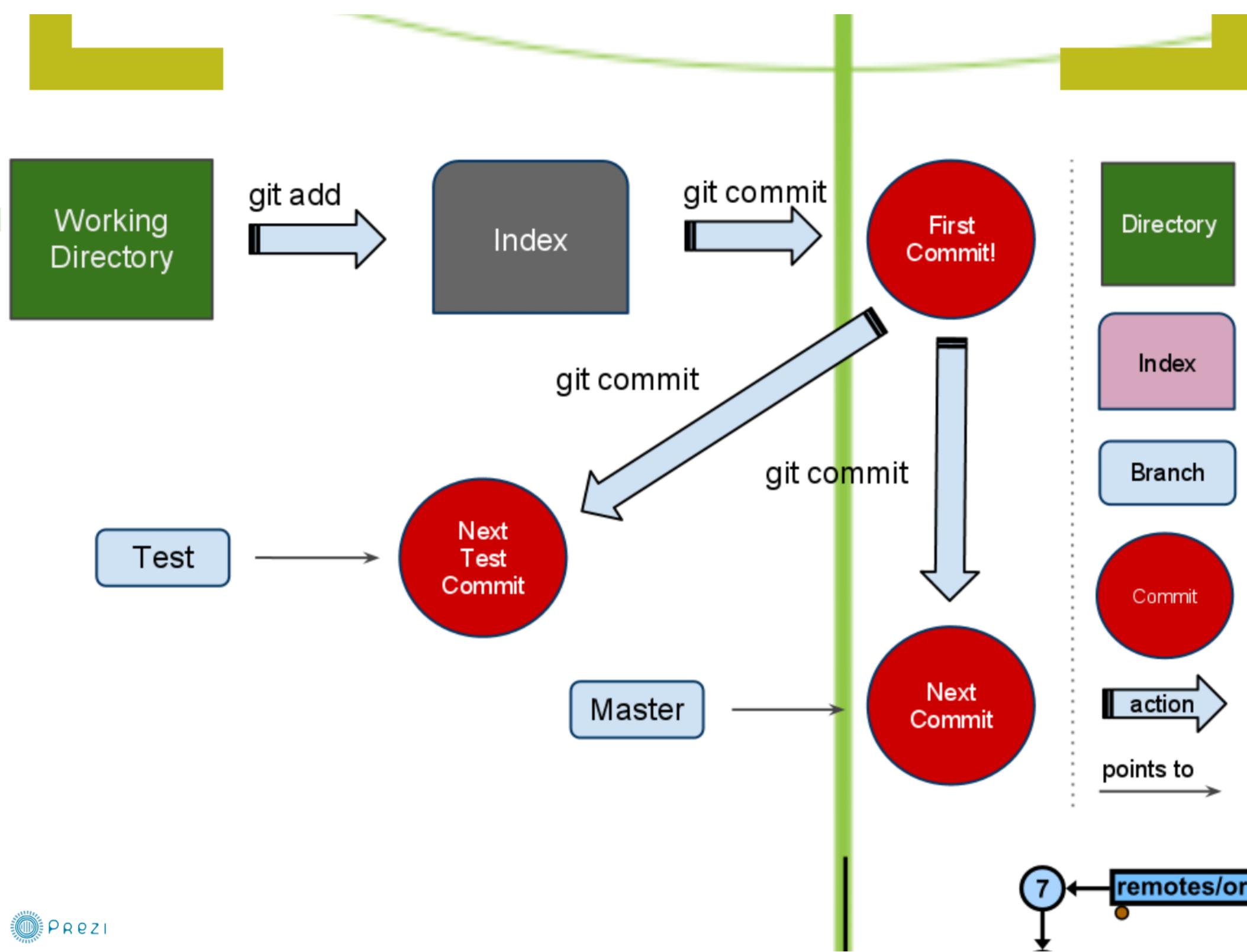
Q: what is the difference between git rm and rm?

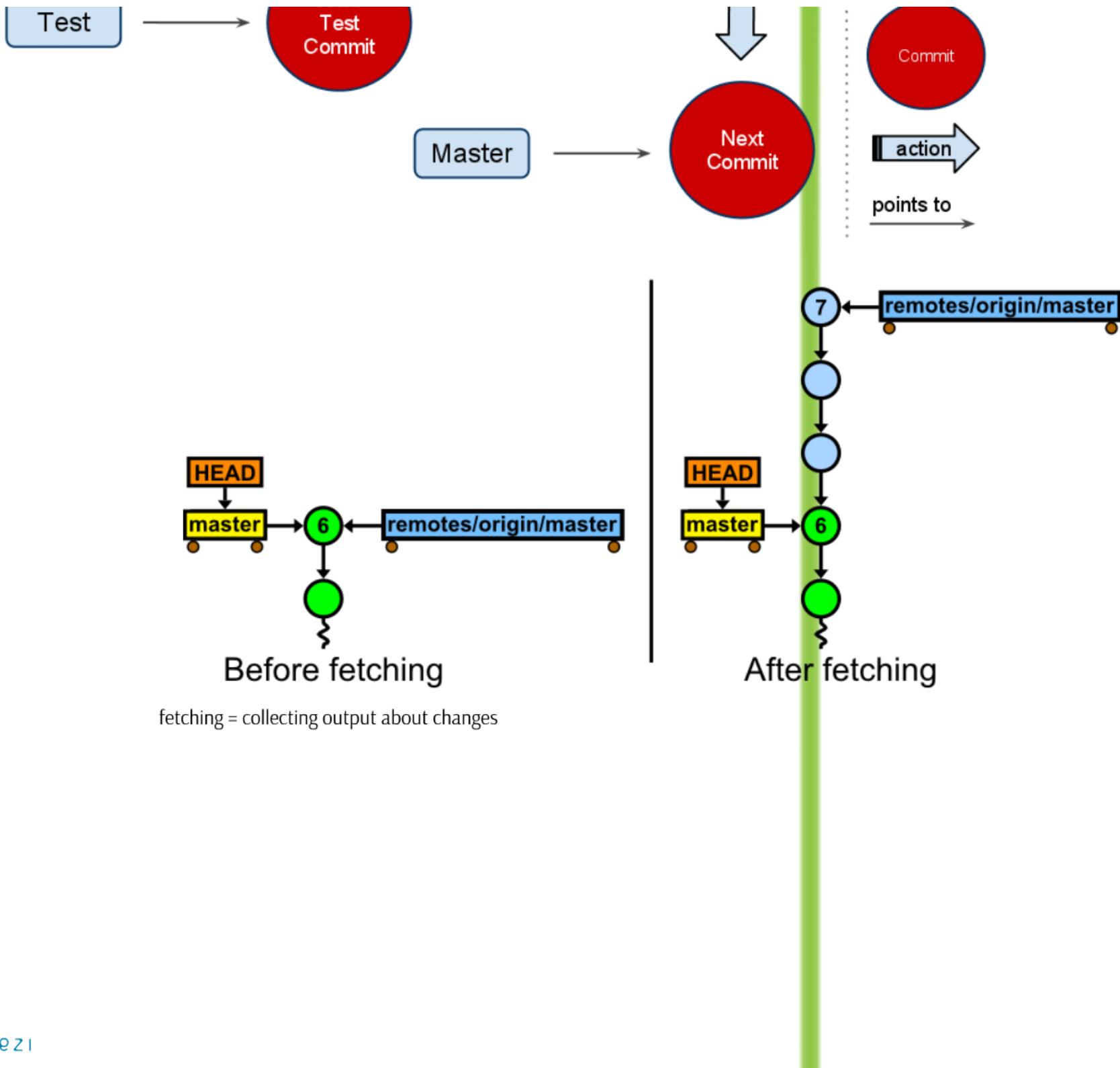
**git mv** - rename filename

> `mv README.txt README`

> `git rm README.txt`

> `git add README`





## **git log** - history of changes

- > `git log -p -2`
- > `git log --stat`

## **git commit** - basic commit

- > `git commit -s`
- > `git commit --amend --no-edit`

## **git reset** - removing files from stage

- > `git reset HEAD filename`

## **git checkout on files** - restoring previous version

Q: what is the difference between two above commands?

git pull does a git fetch followed by a git merge.

## **git tag** - tags versioning, not based on SHA-1

```
> git tag -a v1.4 -m 'my version 1.4'  
> git show v1.4  
> git tag -a v1.2 -m 'version 1.2' 9fce02  
> git push origin --tags  
> git push origin v1.2
```

Lightweight tag that is basically a branch that never moves.

Aliases:

```
git config --global alias.co checkout  
git config --global alias.ci commit  
git config --global alias.st status  
git config --global alias.br branch  
git config --global alias.hist 'log --pretty=format:"%h %ad | %s%d [%an]" --  
graph --date=short'  
git config --global alias.type 'cat-file -t'  
git config --global alias.dump 'cat-file -p'
```

# Troubleshooting

W18/hsl/ob

```
ning: git log --  
HEAD  
ning: git remote  
pository.  
is. Please manually  
I try again.
```

```
- git push origin  
Counting objects: 23, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (14/14), done.  
Writing objects: 100% (18/18), 19.7 KB | 0 bytes/s, done.  
Total 18 (delta 8), reused 0 (delta 0)  
remote: Resolving deltas: 100% (8/8)  
remote: Processing changes: refs: 1, done  
remote:  
remote: ERROR: In commit f2f8571746975589091aa8dfbc87a  
remote: ERROR: committer email address john.rygard@ericsson.se  
remote: ERROR: does not match your user account.  
remote:  
remote: ERROR:  
remote: ERROR: The following addresses are currently registered:  
remote: ERROR: rthbuilder_local@mailman.ericsson.se  
remote:  
remote: ERROR: To register an email address, please visit:  
remote: ERROR: https://host/#/settings/contact  
remote:  
remote:  
remote:  
To ssh://rthbuilder@host:29418/rt/tools  
! [remote rejected] master -> master (bad主人)  
error: failed to push some refs to 'ssh://rthbuilder@host:29418/rt/tools'
```

```
-> git push origin  
Counting objects: 23, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (14/14), done.
```

```
remote:  
remote: New Changes:  
remote: https://host/96131  
remote:  
To ssh://elukmin@host:29418/hsl/obx  
 * [new branch] HEAD -> refs/publish/master/elukmin  
Switched to branch 'master'  
Deleted branch 'elukmin'
```

```
-> git push origin  
Counting objects: 23, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (14/14), done.  
Writing objects: 100% (18/18), 1.97 KiB | 0 bytes/s, done.  
Total 18 (delta 8), reused 0 (delta 0)  
remote: Resolving deltas: 100% (8/8)  
remote: Processing changes: refs: 1, done  
remote:  
remote: ERROR: In commit f2f85817469f7558090a1e8dfb1c874fbb6b336f  
remote: ERROR: committer email address johan.nygard@ericsson.com  
remote: ERROR: does not match your user account.  
remote: ERROR:  
remote: ERROR: The following addresses are currently registered:  
remote: ERROR: rthbuilder_local@mailman.lmera.ericsson.se  
remote: ERROR:  
remote: ERROR: To register an email address, please visit:  
remote: ERROR: https://host/#/settings/contact  
remote:  
remote:  
To ssh://rthbuilder@host:29418/rt/tools  
! [remote rejected] master -> master (invalid committer)  
error: failed to push some refs to 'ssh://rthbuilder@host:29418/rt/tools'
```



```
-> git review -f -v
2013-10-23 16:44:44.639552 Running: git log --color=never --oneline HEAD^1..HEAD
2013-10-23 16:44:44.664705 Running: git remote
2013-10-23 16:44:44.679572 Running: git branch -a --color=never
2013-10-23 16:44:44.704771 Running: git branch --color=never
2013-10-23 16:44:44.726552 Running: git log HEAD^1..HEAD
Found topic 'elukmin' from parsing changes.
2013-10-23 16:44:44.750652 Running: git rev-parse --git-dir
2013-10-23 16:44:44.766286 Running: git remote update origin
Fetching origin
2013-10-23 16:44:46.441407 Running: git rebase -i remotes/origin/master
2013-10-23 16:44:47.338928 Running: git reset --hard ORIG_HEAD
2013-10-23 16:44:47.381378 Running: git remote update origin
Fetching origin
2013-10-23 16:44:49.295238 Running: git config --get color.ui
2013-10-23 16:44:49.310993 Running: git log --color=auto --decorate --oneline elukmin --not
remotes/origin/master
2013-10-23 16:44:49.337534 Running: git push origin HEAD:refs/publish/master/elukmin
remote: Resolving deltas: 100% (1/1)
remote: Branch refs/heads/master:
remote: You need 'Push' rights to upload code review requests.
remote: Verify that you are pushing to the right branch.
remote: User: elukmin
remote: Please read the documentation and contact an administrator
remote: if you feel the configuration is incorrect
remote: Processing changes: refs: 1, done
To ssh://elukmin@host:29418/playground
! [remote rejected] HEAD -> refs/publish/master/elukmin (cannot upload review)
error: failed to push some refs to 'ssh://elukmin@host:29418/playground'
```

```
-> git revert HEAD
[master f249d4c] Revert "Updated task-siu.bb file as a result of wr4 and wr5 rootfs comparison"
1 file changed, 39 insertions(+), 35 deletions(-)

-> git status
# On branch master
```

Issue:

```
-> git review -f -v
2013-08-09 11:38:09.591054 Running: git log --
color=never --oneline HEAD^1..HEAD
2013-08-09 11:38:09.613177 Running: git remote
No 'gitreview' file found in this repository.
We don't know where your gerrit is. Please manually
create a remote named gerrit and try again.
```

Solution:

```
-> touch .gitreview
-> vi .gitreview
-> cat .gitreview
[gerrit]
host=host
port=29418
project=rt/tools.git
defaultbranch=master
defaultremote=origin
```

```
remote: Resolving deltas: 100% (1/1)
remote: Branch refs/heads/master:
remote: You need 'Push' rights to upload code review requests.
remote: Verify that you are pushing to the right branch.
remote: User: elukmin
remote: Please read the documentation and contact an administrator
remote: If you feel the configuration is incorrect
remote: Processing changes: refs: 1, done
To ssh://elukmin@host:29418/playground
 ! [remote rejected] HEAD -> refs/publish/master/elukmin (cannot upload review)
error: failed to push some refs to 'ssh://elukmin@host:29418/playground'
```

```
-> git revert HEAD
[master f249d4c] Revert "Updated task-siu.bb file as a result of wr4 and wr5 rootfs comparison"
1 file changed, 39 insertions(+), 35 deletions(-)
```

```
-> 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 directory clean
```

```
-> git commit -s
# 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 directory clean
```

```
-> git rebase -i origin/master
Successfully rebased and updated refs/heads/master.
```

```
-> git push origin
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:
```

```
git config --global push.default matching
```

```
To squelch this message and adopt the new behavior now, use:
```

```
git config --global push.default simple
```

```
See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)
```

```
Counting objects: 19, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1006 bytes | 0 bytes/s, done.
Total 7 (delta 5), reused 1 (delta 0)
To git@github.com:DUSg2/wrl-siu
 e868a53..f249d4c master -> master
```

Q: pick or squash for revert?

[On local branch (different than master):  
-> git reset --hard master  
HEAD is now at 4d376df Merge branch 'master' of ssh://host:29418/hsl/obx]

# *Gerrit*

Git business implementation  
with accesses/changes/trees management

DEMO

E: revert few last commits from the sandbox  
E: checkout sandbox to the directory with  
example name

# Links

- <http://steveko.wordpress.com/2012/02/24/10-things-i-hate-about-git/>
- <http://chodounsky.net/2013/05/01/compare-files-with-git-diff-tool/>
- <http://pl.wikibooks.org/wiki/Git/Przypadki>
- <http://sethrobertson.github.com/GitFixUm/>
- <http://tylerfrankenstein.com/code/how-git-rm-all-deleted-files-shown-git-status>
- <http://christoph.ruegg.name/blog/git-howto-revert-a-commit-already-pushed-to-a-remote-repository.html>
- <http://gitready.com/intermediate/2009/03/16/rolling-back-changes-with-revert.html>
- <https://www.atlassian.com/git/tutorial/>
- <http://git-scm.com/book/en/>
- <https://coderwall.com/p/crj69a>
- <http://365git.tumblr.com/post/472624933/the-four-buckets-how-git-considers-content>
- <https://blogs.atlassian.com/2013/04/git-flow-comes-to-java/>
- <https://review.typo3.org/Documentation/error-prohibited-by-gerrit.html>
- <http://longair.net/blog/2009/04/16/git-fetch-and-merge/>
- <http://gitref.org/index.html>
- [http://githowto.com/tagging\\_versions](http://githowto.com/tagging_versions)
- <http://blog.iqandreas.com/git/how-to-move-tags/>
- [root.cern.ch/drupal/content/git-tips-and-tricks](http://root.cern.ch/drupal/content/git-tips-and-tricks)
- <http://blog.mikepearce.net/2010/05/18/the-difference-between-git-pull-git-fetch-and-git-clone-and-git-rebase/>

# Git vcs - introduction