



6, 7 october 2016 - ROMA

Git for Programmers



Primož Gabrijelčič

<http://primoz.gabrijelcic.org>



Kudos

Pro Git

- Scott Chacon and Ben Straub
- *Pro Git*
- <https://git-scm.com/book/en/v2>
- Creative Commons Attribution Non Commercial Share Alike 3.0



A little bit of history

Version Control Systems

- File backups
 - __history
- Local VCS
 - database of changes
 - RCS, SCCS
- Centralised VCS
 - developer checks out project
 - CVS, Subversion, Perforce, Visual SourceSafe
- Distributed VCS
 - Developer mirrors the repository
 - BitKeeper, Git, Mercurial, Darcs



A decorative background graphic consisting of several overlapping, rounded rectangular bands in various colors (blue, green, red, yellow) against a light gray grid pattern. A thin black horizontal bar runs across the middle of the slide, partially obscuring the graphic.

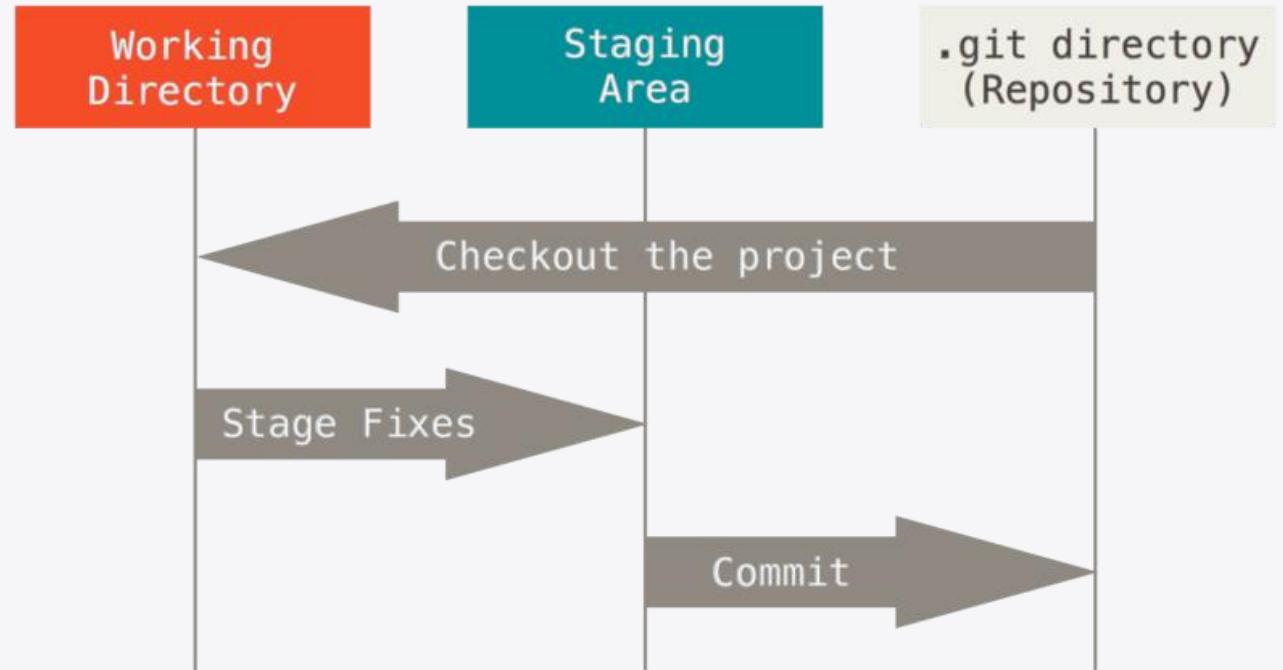
Git

Git

- Mini filesystem
- Stream of snapshots
 - Only changed files are stored
- Almost all operations are local
 - Repository is mirrored locally
- Checksums
 - SHA-1
 - File name = data hash

File states

- Committed
- Modified
- Staged
 - Can be skipped



Local configuration

- git config --list
- git config --global core.editor "C:\Program Files (x86)\PSPad editor\PSPad.exe"
- git config --edit --global
- git help config
- c:\Users\gabr\.gitconfig

Whitespace and newlines

- `git config --global core.autocrlf true`
 - commit: CRLF -> LF
 - checkout: LF -> CRLF
- for Windows-only projects
 - `git config --global core.autocrlf false`
 - CRLF stays CRLF
- `core.whitespace`
 - LF-delimited



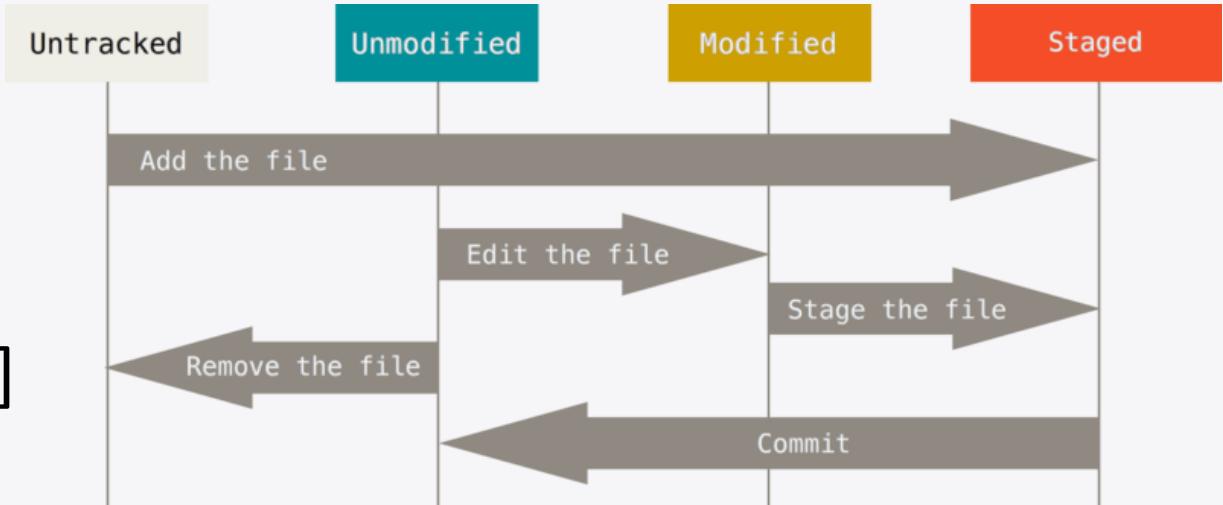
Repositories

Getting a repository

- Initializing new repository
 - cd project_folder
 - git init
 - git add files
 - git commit -m “initial commit”
- Clone existing repository
 - git clone <url>
 - http://, https://, ssh://, git://

Files

- git status
- git add <mask>
- git diff [--staged]
- git commit [-a] [-m “message”]
- git rm [--cached]
- git mv <from> <to>
 - git rm + git add
- git commit



Source: <https://git-scm.com/book/en/v2/Getting-Started-Git-Basics>

.gitignore

- <https://github.com/github/gitignore/blob/master/Delphi.gitignore>

Changelog

- git log [-p] [--stat] [--graph]

Ooops!

- `git commit --amend`
 - AKA Ooops, I forgot that file!
- `git reset HEAD <file>`
 - AKA ‘unstage’
 - `git config --global alias.unstage “reset HEAD -”`



Remotes

No programmer is an island

Remotes

- git clone
- git remote [-v]
 - *origin* = name for the default remote repository
- perfectly valid: more than one remote
 - git remote add <remote_name> <url>

Pull & Push

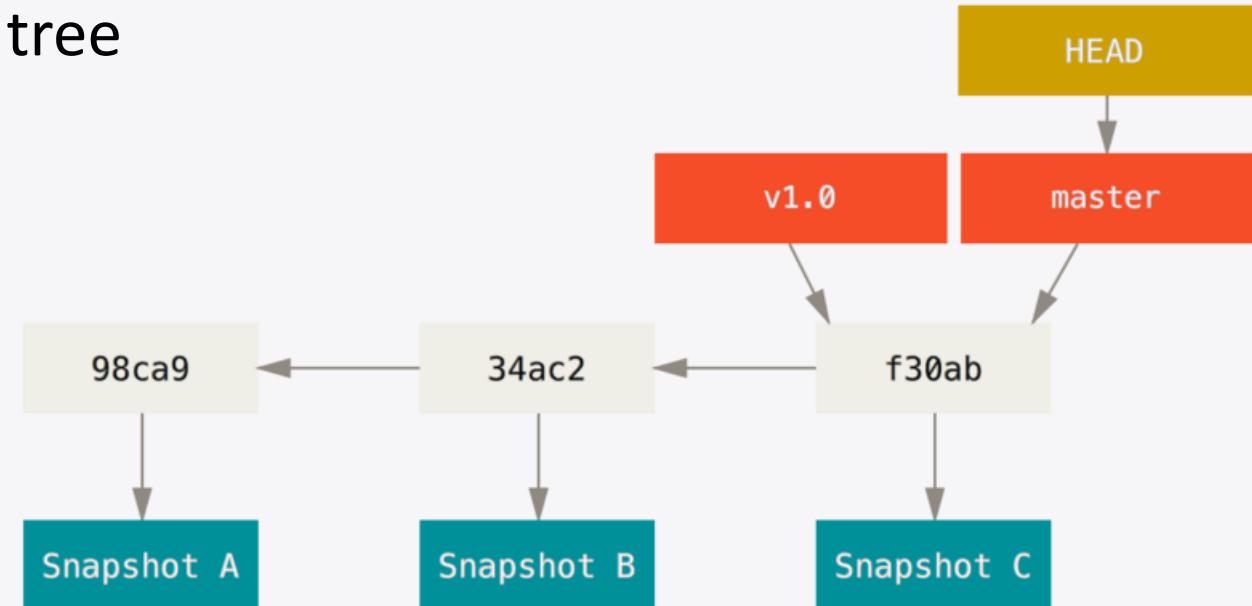
- `git fetch [<remote_name>]`
 - Retrieves the data you don't yet have
 - New data goes to the repository, not to your working area
- `git pull`
 - fetch + merge
- `git push [<remote_name>]`



Branching

Branches

- A branch is a pointer into commit tree
- ‘git init’ creates ‘master’ branch
- git branch <branch_name>
- git log --oneline --decorate
- git checkout <branch_name>
- git checkout --b <new_branch>
- git branch [--merged]
- git branch --d <branch_name>
- branch = lightweight
 - A single file containing 40-char SHA-1 checksum of the commit



Merging

- You **always** work on a branch that is checked out
- `git merge <branch_name>`
- Resolving merge conflicts
 - <<<<< HEAD
Changed in the master.
=====
 - This is a branch.
>>>>> branch1
- `git mergetool`
- `git commit`

Types of branches

- Long-Running branches
 - master/develop
 - feature
 - git-flow
- Topic branches
 - Short-lived
 - A fix for an issue = a branch

Remote branches

- origin/master
- git fetch <remote>
- git push <remote> <branch>
- git config push.default current
- git checkout --b <new_branch>
- git push --origin <new_branch>
- git push

Tracking branch

- `git checkout --b <new_branch> <remote>/<remote_branch>`
- `git checkout --track <remote>/<remote_branch>`
- `git checkout <remote_branch>`
- `git branch -u <remote>/<remote_branch>`

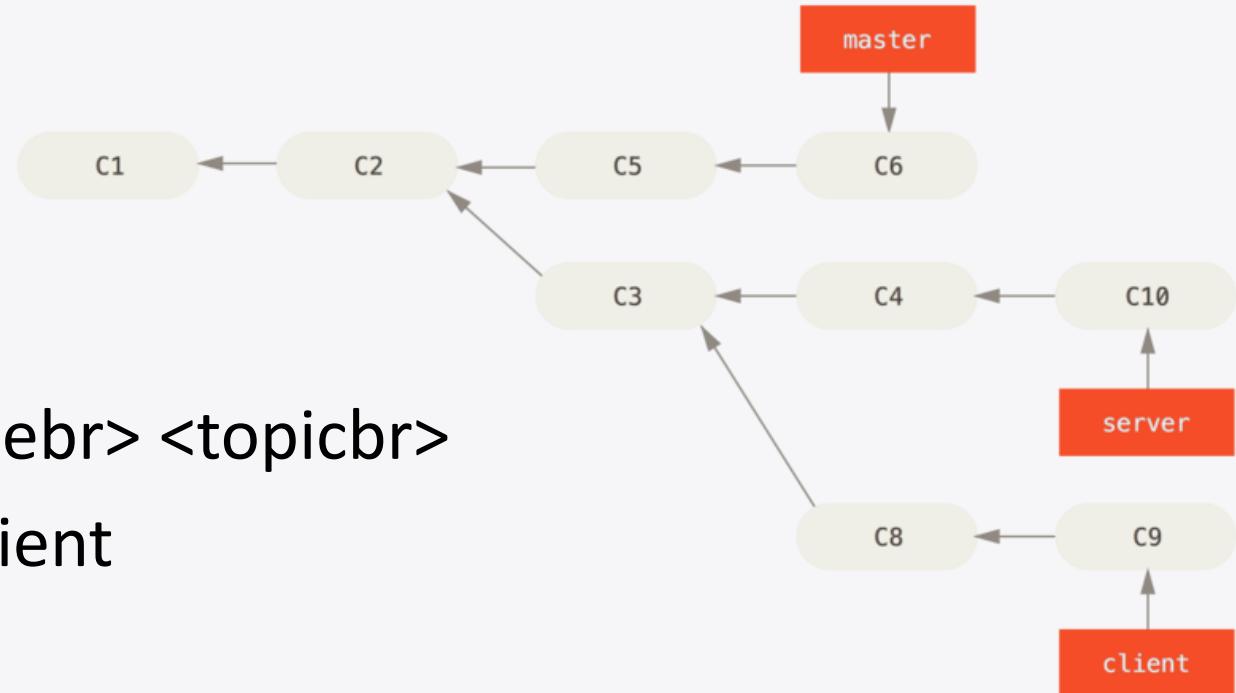
- `git pull`

Rebase

- Take changes committed on one branch ...
 - ... and replay them on another one.
- git checkout experimental
- git rebase master
- Do not rebase commits that exist outside your repository.

3-branch rebase

- `git rebase [--onto <branch>] <basebr> <topicbr>`
- `git rebase --onto master server client`





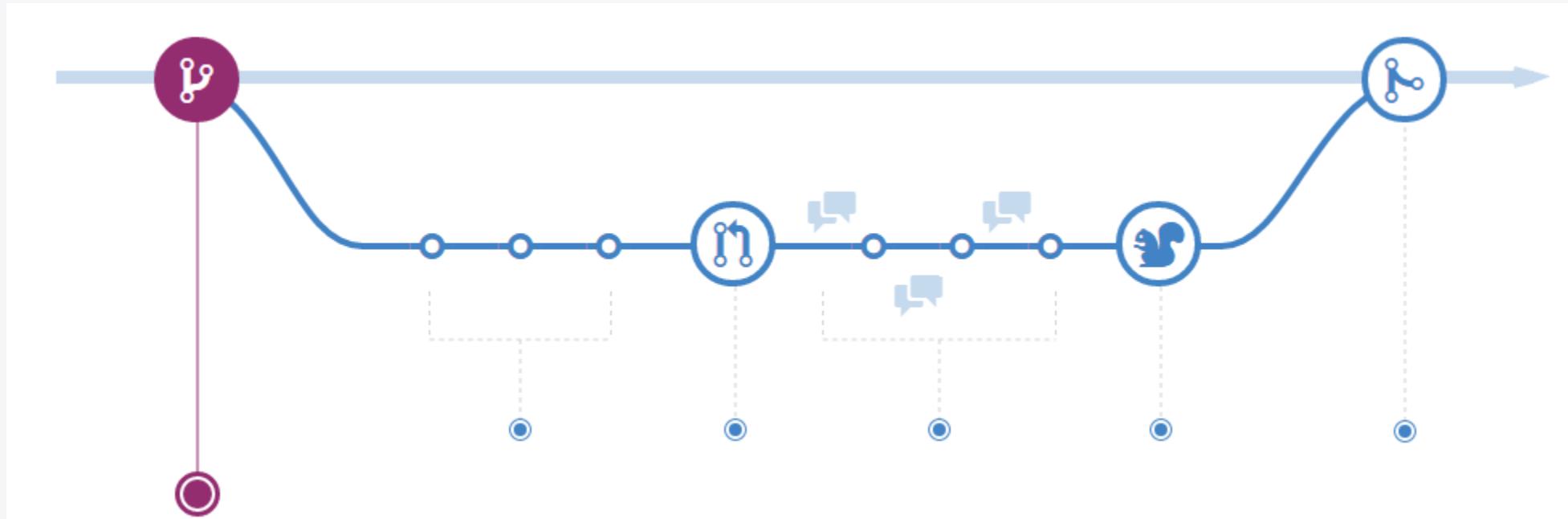
Workflows

A decorative background graphic at the bottom of the slide features a grid pattern over a series of overlapping, semi-transparent circles in various colors (blue, green, orange, yellow). A horizontal black bar runs across the middle of the slide, partially obscuring the circles.

GitFlow



GitHub Flow



Source: <https://guides.github.com/introduction/flow/>



The background features a stylized graphic of overlapping, rounded rectangles in various colors (blue, green, red, yellow) on a white grid pattern. A horizontal black bar runs across the middle of the slide, partially obscuring the graphic. The word "Tools" is positioned on this bar in a large, bold, black sans-serif font.

Tools

Tools

- Git for Windows
- TortoiseGit
- GitHub Desktop
- GitKraken



Resources

Resources

- [https://en.wikipedia.org/wiki/Version control](https://en.wikipedia.org/wiki/Version_control)
- <https://git-scm.com>
- <https://github.com>
- <https://bitbucket.org>
- <https://git-for-windows.github.io>
- <https://git-scm.com/downloads/guis>
- <https://desktop.github.com>
- <https://www.gitkraken.com>
- <https://tortoisegit.org>
- <https://git-scm.com/book/en/v2>
- <https://wiert.me/2016/07/14/git-cheat-sheet/>

Workflows

- <http://jeffkreeftmeijer.com/2010/why-arent-you-using-git-flow/>
- <https://yakiloo.com/getting-started-git-flow/>
- <https://github.com/nvie/gitflow>
- <https://www.git-tower.com/learn/git/ebook/en/command-line/advanced-topics/git-flow>
- <http://danielkummer.github.io/git-flow-cheatsheet/>
- <http://endoflineblog.com/gitflow-considered-harmful>
- <https://guides.github.com/introduction/flow/>



Questions?