

Git Good

Linus Berglund

February 17, 2021

devies

What is Git?

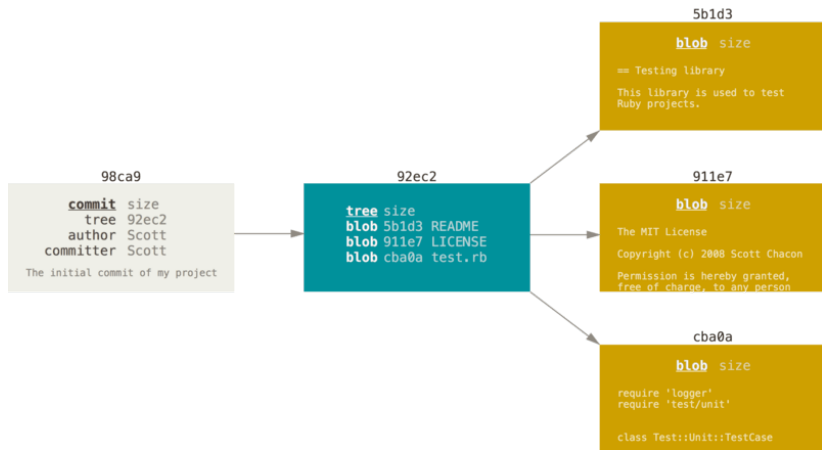


Git TL;DR

- ▶ Distributed version control system (VCS)
- ▶ Created by Linus Torvalds
- ▶ Motto: *Local branching on the cheap*

- ▶ Each commit builds upon the previous
- ▶ Stores snapshots, not differences
- ▶ Everyone has a full mirror of the repository

Tree model



Data model

▶ `.git/HEAD`

Data model

- ▶ `.git/HEAD`
- ▶ `.git/hooks`

Data model

- ▶ `.git/HEAD`
- ▶ `.git/hooks`
- ▶ `.git/objects`

Data model

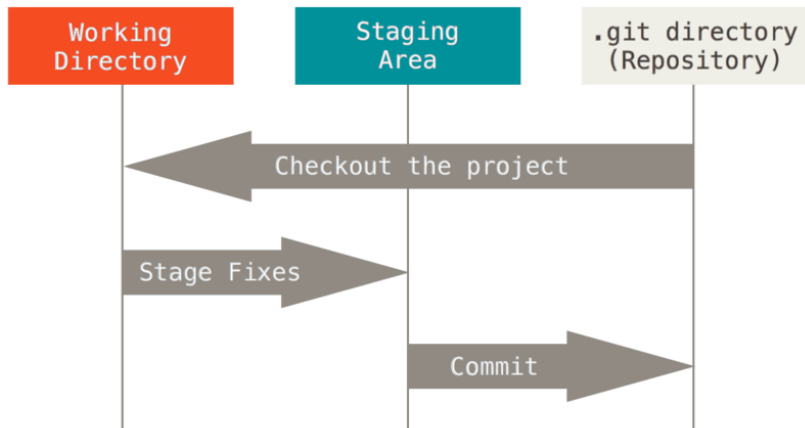
- ▶ `.git/HEAD`
- ▶ `.git/hooks`
- ▶ `.git/objects`
- ▶ `.git/refs`

Data model

- ▶ `.git/HEAD`
- ▶ `.git/hooks`
- ▶ `.git/objects`
- ▶ `.git/refs`

- ▶ **Live Demo!**

Three working areas



Checkout

`git checkout` is very useful

Checkout

git checkout is very useful

- ▶ `git checkout branch_name`

Checkout

git checkout is very useful

- ▶ `git checkout branch_name`
- ▶ `git checkout -b new_branch`

Checkout

git checkout is very useful

- ▶ `git checkout branch_name`
- ▶ `git checkout -b new_branch`
- ▶ `git checkout .`

Checkout

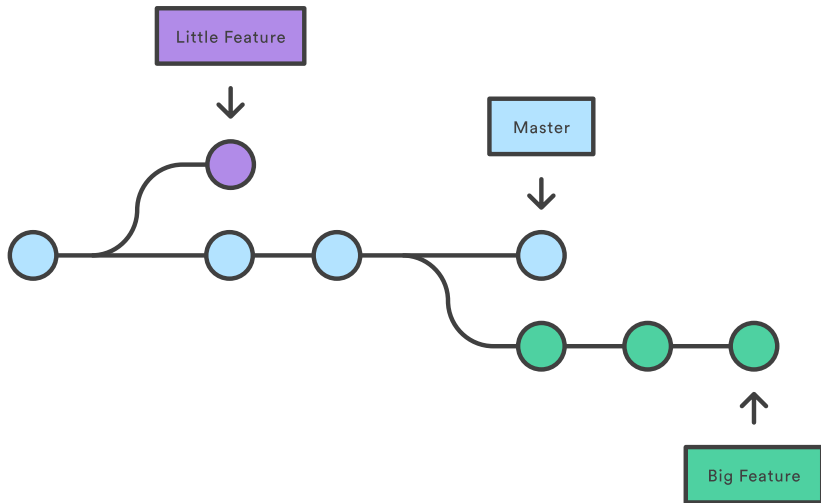
git checkout is very useful

- ▶ `git checkout branch_name`
- ▶ `git checkout -b new_branch`
- ▶ `git checkout .`
- ▶ `git checkout origin/master README.md`

Checkout

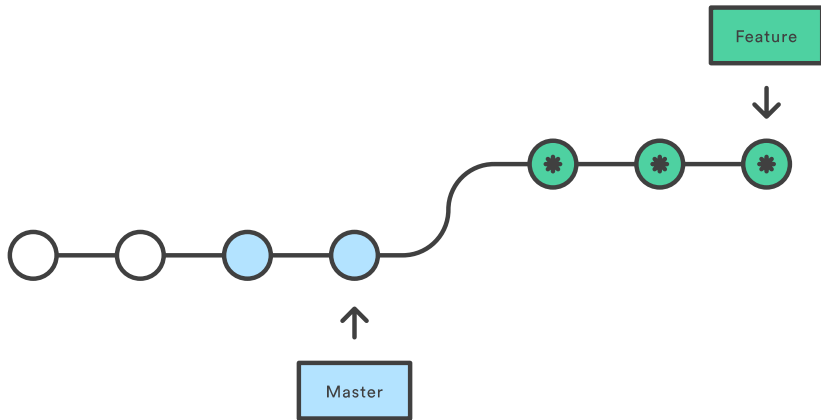
Command	Scope	Common use cases
git reset	Commit-level	Discard commits in a private branch or throw away uncommitted changes
git reset	File-level	Unstage a file
git checkout	Commit-level	Switch between branches or inspect old snapshots
git checkout	File-level	Discard changes in the working directory
git revert	Commit-level	Undo commits in a public branch
git revert	File-level	(N/A)

Branches



Rebasing

Rebasing the feature branch onto master



* Brand New Commit

Commit messages

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Commit messages

- ▶ Separate subject from body with a blank line

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters
- ▶ Capitalize the subject line

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters
- ▶ Capitalize the subject line
- ▶ Do not end the subject line with a period

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters
- ▶ Capitalize the subject line
- ▶ Do not end the subject line with a period
- ▶ Use the imperative mood in the subject line

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters
- ▶ Capitalize the subject line
- ▶ Do not end the subject line with a period
- ▶ Use the imperative mood in the subject line
- ▶ Wrap the body at 72 characters

Commit messages

- ▶ Separate subject from body with a blank line
- ▶ Limit the subject line to 50 characters
- ▶ Capitalize the subject line
- ▶ Do not end the subject line with a period
- ▶ Use the imperative mood in the subject line
- ▶ Wrap the body at 72 characters
- ▶ Use the body to explain what and why vs. how

Commit messages - example

```
$ git log --oneline -5 --author cbeams --before "Fri Mar 26 2009"
```

```
e5f4b49 Re-adding ConfigurationPostProcessorTests after its brief removal in r814. @Ignore-ing the testCgl  
2db0f12 fixed two build-breaking issues: + reverted ClassMetadataReadingVisitor to revision 794 + eliminat  
147789f Tweaks to package-info.java files  
22b25e0 Consolidated Util and MutableAnnotationUtils classes into existing AsmUtils  
7f96f57 polishing
```

Yikes. Compare that with these more recent commits from the same repository:

```
$ git log --oneline -5 --author pwebb --before "Sat Aug 30 2014"
```

```
5ba3db6 Fix failing CompositePropertySourceTests  
84564a0 Rework @PropertySource early parsing logic  
e142fd1 Add tests for ImportSelector meta-data  
887815f Update docbook dependency and generate epub  
ac8326d Polish mockito usage
```

Which would you rather read?

The former varies in length and form; the latter is concise and consistent.

The former is what happens by default; the latter never happens by accident.

Useful commands

Git builtins

- ▶ `git stash`
- ▶ `git diff`

- ▶ `git blame`
- ▶ `git reflog`
- ▶ `git bisect`

Aliases

- ▶ `gs` instead of `git status`
- ▶ `git lol` instead of `git log --graph --decorate --pretty=oneline --abbrev-commit`

Useful non-git tools

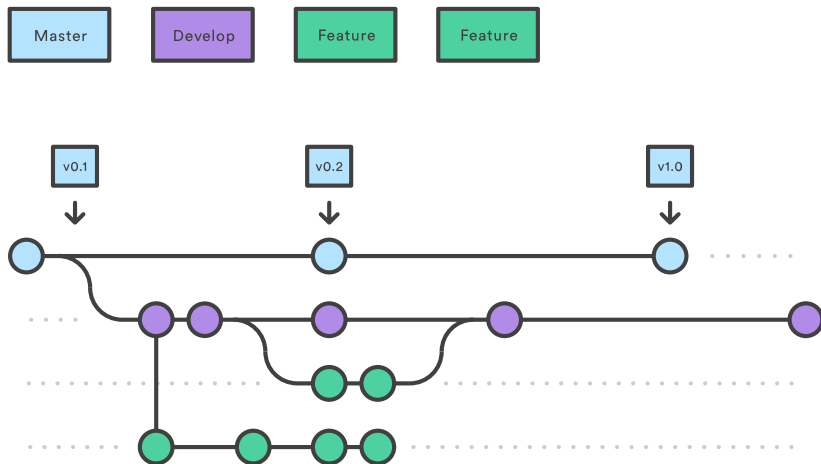
Used in demo

- ▶ Tmux - `tmux`
- ▶ Fuzzy Find - `fzf`
- ▶ exa (colorful alternative to `ls`) - `exa`
- ▶ bat (colorful alternative to `cat`) - `bat`

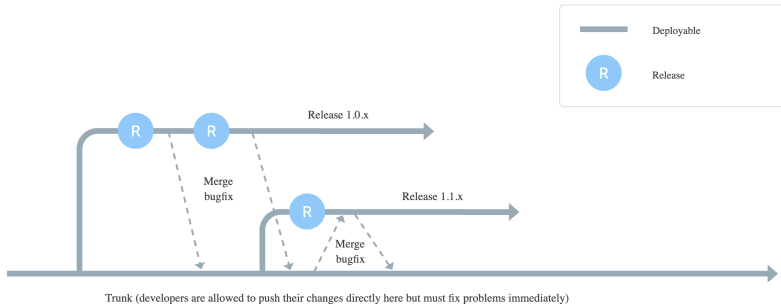
Other cool tools

- ▶ Tig `tig`
- ▶ Ripgrep (alternative to `grep` and `git grep`) - `rg`

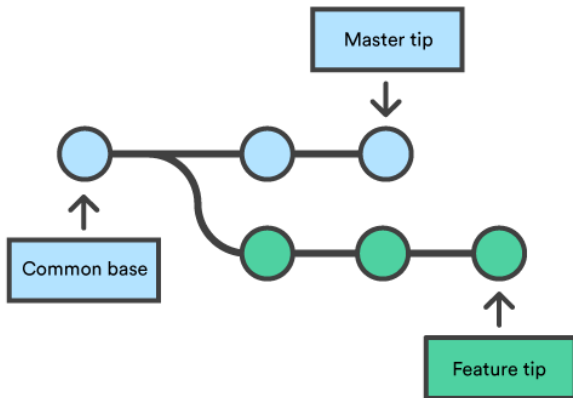
Workflows - Git Flow



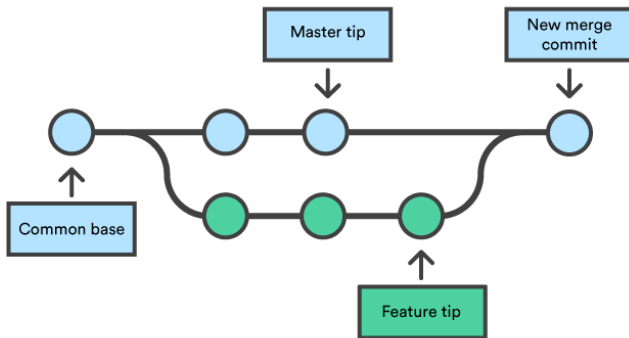
Workflows - Trunk Based Development



Merge commit (without PR)



Merge commit (without PR)



Merge commit (PR)

Merging master into the feature branch

