



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Advanced Git

Linux User Group Meeting 2015-1

Thursday, 23th April 2015 16:00 - 18:00

BFH - Room HG 408 (Quellgasse, Biel)

Daniel Baumann <daniel.baumann@bfh.ch>

IT System Engineer, Team Infrastructure Services

Overview

Part 1

- ▶ Quick Repetition of Terminology
- ▶ Git Repository Hosting at BFH
- ▶ Advanced Git Usage

Part 2

- ▶ Distributed Workflows
- ▶ Question and Answers

Aperitif

Repetition of Terminology

Repetition of Terminology

- ▶ Git Commit
- ▶ Git ID
- ▶ referencing objects
- ▶ fast-forward / non-fast-forward

Git Repository Hosting at BFH

Git Repository Hosting at BFH

git.bfh.ch

- ▶ documentation at <https://linux.bfh.ch/services/git/>, configuration available for reuse
- ▶ Debian 8 (jessie) Container with git, gitweb, gitolite3, irker
- ▶ initial key generation by user, import via ticket, after that selfservice for users via ssh

„Rant“ about GitLab

- ▶ not reasonably packagable (~200 Gems, ~half of them forked)
- ▶ security situation
- ▶ ruby; watching the other alternatives (~~gitolite~~, Gogs, Kallithea, ...)

Future for git.bfh.ch

- ▶ short-term: cgit instead of gitweb
- ▶ mid-term: integration with request-tracker and review-board
- ▶ key generation via browser, submission via IDM at selfhelp.bfh.ch

Advanced Git Usage

Finding broken stuff

Find the person to blame for

- ▶ git blame

Find the commit to blame for

- ▶ git bisect

Revert the broken commit

- ▶ git revert

Changing the Past

Editing Commits

- ▶ git amend
- ▶ git rebase and edit/reword commits
- ▶ tip: create an initial empty commit

- ▶ git filter branch

Reordering Commits

- ▶ git rebase and squash/remove etc. commits

Cutting Commits

- ▶ graft points

Branches and Remotes

Branches

- ▶ branches are cheap
- ▶ remote and local
- ▶ origin
- ▶ adding remote branches

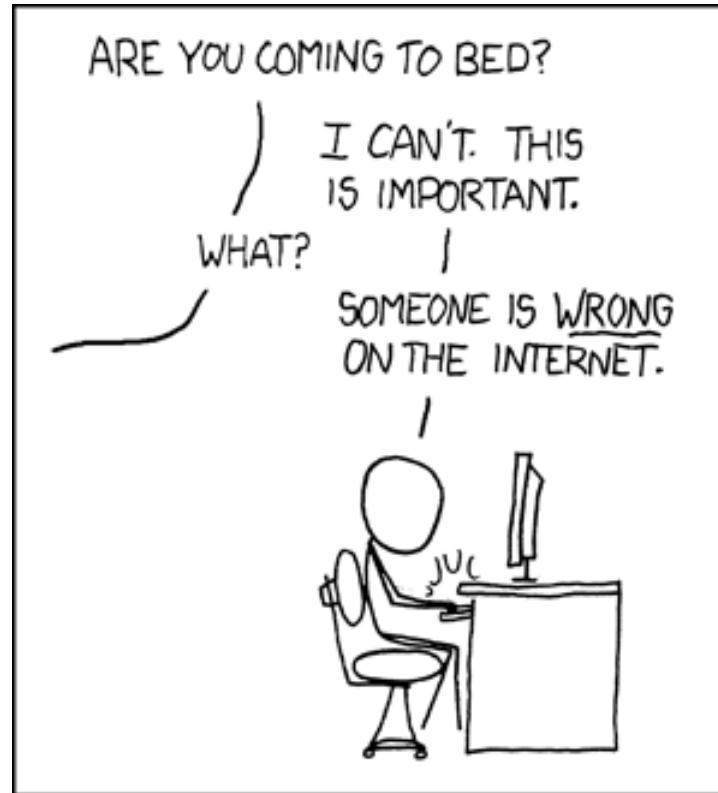
- ▶ fetching branches
- ▶ push other remote branches as local branches

Fetch commits

- ▶ `git cherry-pick`

Duty Calls

<https://xkcd.com/386/>



Tooltip: *What do you want me to do? LEAVE? Then they'll keep being wrong!*

Rebase versus Merge

- ▶ don't listen to the Internet, people are wrong
(or have very, very, veeeery specific workflows like kernel.org
and are in now way representative or good role models)
- ▶ if you don't, you're creating a mess for your and everyone else.
- ▶ always do `git pull -rebase`
- ▶ always do fast-forward merges (aka rebase), never non-fastforward merges (aka merges)

Distributed Workflow

Distributed Workflow

Sysadmin

- ▶ have one „central“, authoritative repository
- ▶ use two main branches, master and test
- ▶ everyone clones it and works on feature branches
- ▶ have a consistent feature branch naming scheme (tmp-*, bug-*, feature-*, whatever...)

User

- ▶ push feature branches asap/frequently
- ▶ always use rebase instead of merge
- ▶ rebase frequently, rebase often
- ▶ important: do „good“ commits only (wrt/ content, amount of content, and description)
- ▶ never rewrite history on „master“ branches, use revert (or filter-branch if needed)
- ▶ do pull requests for peer review (of „good“ commits/cleaned-up feature branches only)
- ▶ use developer origins („Signed-off-by“)

Random Stuff

Deployment via Git

- ▶ use ssh triggers, use preflight-tests
- ▶ grant people write (!= overwrite) access to your stuff. It's versioned.. what could possibly go wrong?
- ▶ ...

Questions and Answers

Thank You for Your Attention.

♥ Source Code is freely available

```
git clone git://git.bfh.ch/git/staff/bad9/other/talks.git
```