



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences

## Git Advanced for TI Staff

Thursday, 10th September 2015 09:00 - 10:30

BFH - Room HG 4.08, Quellgasse, Biel

Daniel Baumann <daniel.baumann@bfh.ch>

IT System Engineer, Team Infrastructure Services

# Overview

## **Part 1**

- ▶ Quick Repetition of Terminology
- ▶ Some more Git commands

## **Part 2**

- ▶ Distributed Workflows
- ▶ Question and Answers

## Repetition of Terminology

## Repetition of Terminology

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

# 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`



# Shortcuts

## **commit**

- ▶ no need for precision: head~10
- ▶ no need to repeat commit message: -C HEAD

## **partial commits/checkouts**

- ▶ commit -p
- ▶ checkout -p

## **reopen commits**

- ▶ git reset HEAD~

# 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

## More Information

### **Documentation, Links, etc.**

- ▶ <https://linux.bfh.ch/software/git>
- ▶ <https://linux.bfh.ch/services/git>

### **Personal Support**

- ▶ Git Hotline (and general IT-Support :): 031 848 48 48
- ▶ [bfh-linux-users@lists.bfh.ch](mailto:bfh-linux-users@lists.bfh.ch)

### **Next up**

- ▶ Events, Dates, etc.: <https://linux.bfh.ch>
- ▶ Linux Install Events in Biel (1.10.2015) und Burgdorf (12.10.2015)
- ▶ Linux User Group Meeting (October 2015)

Thank You for Your Attention.

♥ Source Code is freely available

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