



Version Control with git and GitHub

un-do and re-do for research projects

Robert Forkel

Quantitative Methods – Spring School 2016

Max Planck Institute for the Science of Human History

Table of contents

1. Version Control

2. git

3. GitHub

Version Control

Version Control – which problem does it solve?

There are only two hard things in Computer Science: cache invalidation and naming things. (Phil Karlton)

"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc



JORGE CHAM © 2012

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham, www.phdcomics.com

Version Control – just save the changes

Instead, we could keep the same file name, but record changes:

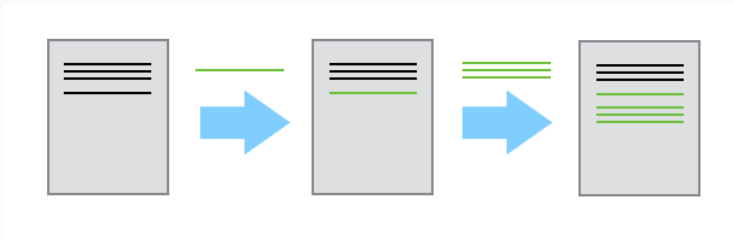


Figure 2: Version control systems save changes – and thus allow un-do and re-do!

Version Control – the basics

- So in the simplest case, version control systems track **what** changed **when**.
- Adding comments whenever you save a state adds the **why**.
- Working collaboratively on documents requires adding the **who**.
- Modern version control systems track changes of a whole directory tree – not just of a file.
- **Note:** Version control is not the same as backup!
- But if you backup your repository properly, you get a wayback machine added on top of a backup.

Version Control – terminology

- change set** group of changes to files that will be added to a single commit in a version control repository.
- commit** record a change set in a version control repository. As a noun, the result of committing, i.e. a recorded change set in a repository.
- repository** Some disc space where a vcs stores the full history of commits of a project and information about who changed what, when.
- merge** (a repository): To reconcile two sets of changes to a repository.
- conflict** A change made by one user of a version control system that is incompatible with changes made by other users. Helping users resolve conflicts is one of version control's major tasks.

Adapted from the Software Carpentry lesson on git.

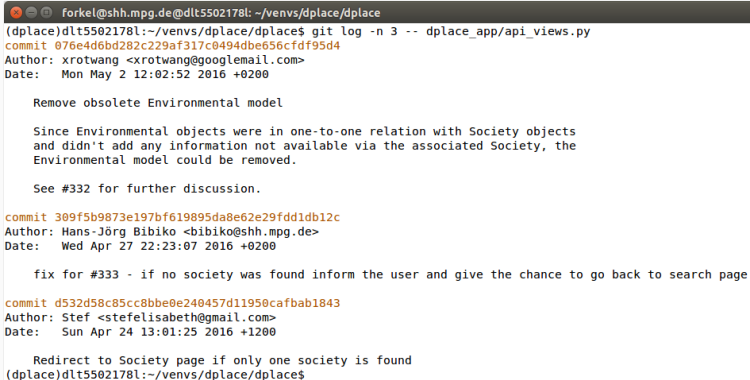
git

What is git?

`git` is a **dvcs** – a distributed version control system.

- So it does all the things mentioned above ...
- ...in a distributed way, i.e.:
 - every repository copy (clone) contains the complete history
 - **commit** = save a change (add/edit/delete) in your local copy
 - **pull/push/synchronize** = exchange changes with other copies

git does all the things outlined above:



```
forkel@shh.mpg.de@dlt5502178l: ~/venvs/dplace/dplace
(dplace)dlt5502178l:~/venvs/dplace/dplace$ git log -n 3 -- dplace_app/api_views.py
commit 076e4d6bd282c229af317c0494dbe656cfd95d4
Author: xrotwang <xrotwang@googlemail.com>
Date: Mon May 2 12:02:52 2016 +0200

    Remove obsolete Environmental model

    Since Environmental objects were in one-to-one relation with Society objects
    and didn't add any information not available via the associated Society, the
    Environmental model could be removed.

    See #332 for further discussion.

commit 309f5b9873e197bf619895da8e62e29fdd1db12c
Author: Hans-Jörg Bibiko <bibiko@shh.mpg.de>
Date: Wed Apr 27 22:23:07 2016 +0200

    fix for #333 - if no society was found inform the user and give the chance to go back to search page

commit d532d58c85cc8bbe0e240457d11950cafbab1843
Author: Stef <stefelisabeth@gmail.com>
Date: Sun Apr 24 13:01:25 2016 +1200

    Redirect to Society page if only one society is found
(dplace)dlt5502178l:~/venvs/dplace/dplace$
```

Figure 3: git log command

Of course this requires some discipline ...



Figure 4: “Git Commit” by xkcd, <https://xkcd.com/1296/>

What goes into the repository?

Technically all files could be put under version control.

- your code, of course
- configuration files!
- the raw data, preferably in formats amenable to diff
- output of `pip freeze` or the equivalent command in R

What goes into the repository?

But bonus points (automatic merging, meaningful diffs) for line-based text formats:

- documentation in markdown, e.g. `README.md`
- \LaTeX , Bib \TeX
- CSV
- nexus, newick
- INI
- IPython Notebooks, i.e. pretty-printed JSON

What goes into the repository?

Rule of thumb: Whatever can be generated automatically doesn't go in version control.

But: In research, output of one workflow step is often input for the next. To make it possible to execute the workflow starting anywhere, keep intermediate results as well in version control. Also often, manual editing of intermediate artefacts is necessary, and version control is the right tool to track this!

What if multiple users make changes to the same file?

- If the file has a line-based (plain-text) format and the users changed different sections, chances are high that git can automatically **merge** the changes correctly.
- If they conflict for the same line(s) and you understand the file, you can semi-manually resolve the conflict with a merge/diff-tool picking lines from either version.
- Otherwise pick/create the 'right' version by hand and **commit** that to the repository.

GitHub

What is GitHub?

- **GitHub** is a commercial hosting service for git repositories.
- It provides a rich web-interface for git repositories (browsing & comparing files/history, wikis, bug tracking, reviews, comments).
- It also provides **GitHub Desktop** – a desktop application (for Windows and OSX) as a well-integrated GUI for git and GitHub.

First steps

- download & install <https://desktop.github.com>
- alternatively, install git and set it up to work with GitHub
<https://help.github.com/articles/set-up-git/>
- sign up for a user account at <https://github.com>
- visit and clone <https://github.com/shh-dlce/qmss-2016>
- Follow the Software Carpentry lesson on “Version Control with git” at <http://swcarpentry.github.io/git-novice/index.html>

git and GitHub is becoming the de-facto standard for collaboration in software development and research, and is already quite well integrated

- On your desktop: GitHub Desktop (see above)
- in RStudio:
<http://www.datasurg.net/2015/07/13/rstudio-and-github/>
- in Overleaf: <https://www.overleaf.com/blog/195-new-collaborate-online-and-offline-with-overleaf-and-git-beta>