



# Version Control with git and GitHub

un-do and re-do for research projects

---

Robert Forkel

Quantitative Methods – Spring School 2017

Max Planck Institute for the Science of Human History

# Table of contents

1. Quantitative Methods and Software Development
2. Version Control
3. git
4. GitHub
5. An Example


# Quantitative Methods and Software Development

---

# Your paper is a software project

When you are using quantitative methods, your paper is (also) a software project!

- even if you don't write **code** yourself!
- although typically you do
- there are **dependencies**
- be prepared to encounter **bugs**!
- there are **build artifacts**
- created by some sort of **workflow**
- there are collaborators (even if only your future self (see below))
- it takes (a lot) longer than estimated 😊  
⇒ your code will stay with you (a lot) longer than expected!



But having the same problems as  
software developers is actually a good  
thing!

# Tools and Best Practices

| Problem                                | Best Practice        | Tool                   |
|--|----------------------|------------------------|
| collaborative editing<br>of text files | version control      | <b>git</b> /GitHub     |
| un-do                                  | version control      | <b>git</b>             |
| re-do                                  | build automation     | <b>make</b>            |
| maintenability                         | documentation        | the <b>README</b> file |
| replicability and<br>correctness       | testing              |                        |
| big/relational data                    | relational databases | SQLite/PostgreSQL      |

# 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

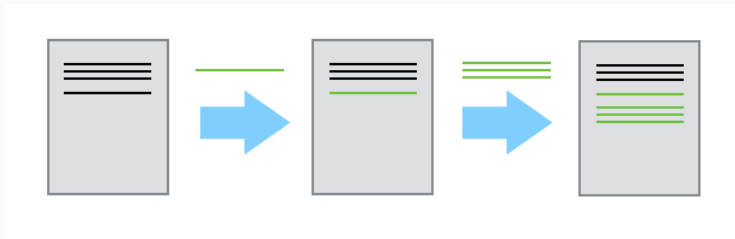
JORGE CHAM © 2012

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham, [www.phdcomics.com](http://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.
- **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

- repository** Some disc space where a vcs stores the full history of commits of a project and information about who changed what, when.
- change set** group of changes 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.
- 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 mentioned above

```
forkel@ssh.mpg.de@dlt5502178l: ~/venvs/dplace/dplace
(dplace)dlt5502178l:~/venvs/dplace/dplace$ git log -n 3 -- dplace_app/api_views.py
commit 076e4d6bd282c229af317c0494d8e656cfd95d4
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 <hbibiko@ssh.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

## git requires some discipline



|   | 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.

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`
- $\text{\LaTeX}$ , Bib $\text{\TeX}$
- 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.

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>

## An Example


---









# Repository layout


Branch: master ▾ **qmss-2016 / example-paper /**

New file Upload files Find file History

 **xrotwang** added readme Latest commit 558495c just now

..

|   |  |                |
|---|--|----------------|
|  <b>config</b>         | added semi-realistic content to config files | 16 minutes ago |
|  <b>data</b>           | Update datapoint for Marshallese             | 33 minutes ago |
|  <b>README.md</b>      | added readme                                 | just now       |
|  <b>paper.tex</b>      | added example paper skeleton                 | 40 minutes ago |
|  <b>references.bib</b> | Update datapoint for Marshallese             | 33 minutes ago |
|  <b>supplement.tex</b> | Update datapoint for Marshallese             | 33 minutes ago |

 **README.md**

## Example paper

- `paper.tex` : LaTeX source for the paper
- `supplement.tex` : LaTeX source for the supplements
- `references.bib` : BibTeX source for the bibliography
- `data` : directory for the (raw) data files
- `config` : directory for configuration files

Figure 5: Exemplary repository layout.

## Transparent data modification

**Update datapoint for Marshallese** commit message Browse files

xrotwang committed 4 minutes ago 1 parent 97b8d68 commit 4f39f7fbd7a80988581f42fe2e576baee6553bf5

Showing 3 changed files with 15 additions and 2 deletions.

- example-paper/data/wals-41A.tab +1 -1
- example-paper/references.bib +9 -0
- example-paper/supplement.tex +5 -1

2 example-paper/data/wals-41A.tab diff of changes in one file View

|         | @@ -143,7 +143,7 @@ map Mapudungun         | 3 Three-way contrast                                     | -38.0 -72.0 Araucanian | Araucanian |
|---------|--|--|------------------------|------------|
| 143 143 | mhi Marathi 2 Two-way contrast             | 19.0 76.0 Indic Indo-European                            | Nominal Categories     |            |
| 144 144 | mrg Margi 2 Two-way contrast               | 11.0 13.0 Biu-Mandara Afro-Asiatic                       | Nominal Categories     |            |
| 145 145 | mar Maricopa 5 Five (or more)-way contrast | 33.166666667 -113.166666667 Yuman                        | Hokan Nomi             |            |
| 146 146 | -msh Marshallese 2 Two-way contrast        | 7.11666666667 171.05 Oceanic Austronesian                | Nominal Cate           |            |
| 147 147 | +msh Marshallese 2 Four-way contrast       | 7.11666666667 171.05 Oceanic Austronesian                | Nominal Cate           |            |
| 147 147 | mrt Martuthunira 2 Two-way contrast        | -20.833333333 116.5 Western Pama-Nyungan                 | Pama-Nyungan           |            |
| 148 148 | mau Maung 3 Three-way contrast             | -11.916666667 133.5 Iwaidjan Iwaidjan                    | Nominal Cate           |            |
| 149 149 | may Maybrat 3 Three-way contrast           | -1.333333333 132.5 North-Central Bird's Head West Papuan |                        |            |

**Figure 6:** Logically related changes bundled in one `commit`