

The use and philosophy of version control

May 19, 2016

With materials from Mozilla Science Lab

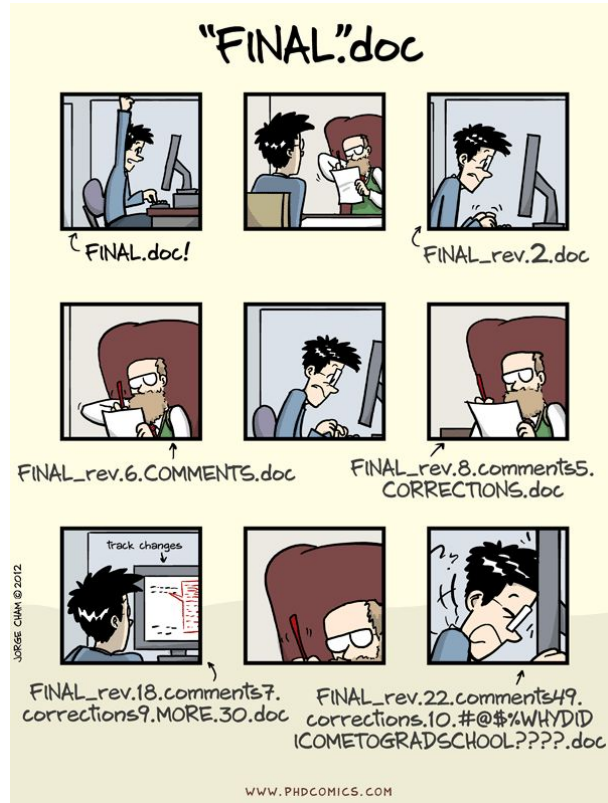


Here's what we're doing today

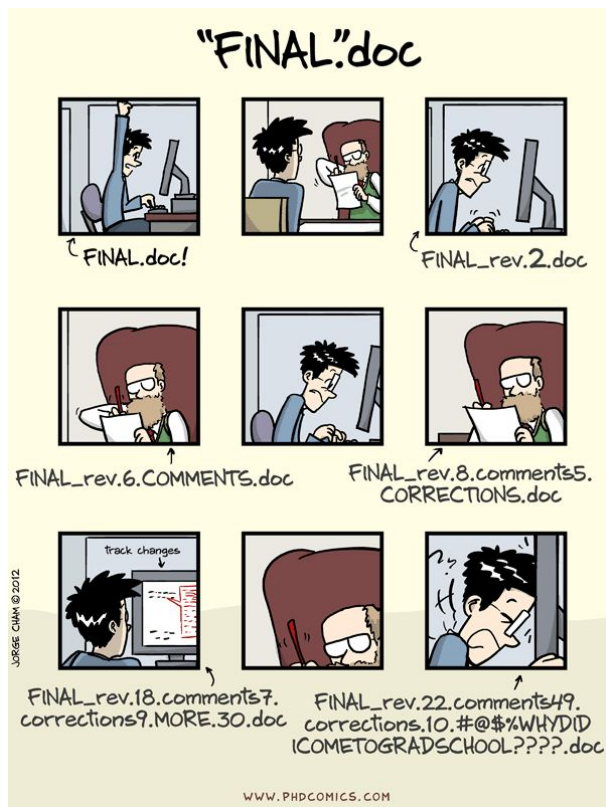
- **Introduction** to how Git works and **why** it can help you.
- **Workshop:** version control with Git



Version Control



Version Control



Name

Date modified

W phageParser-v1.docx

2015-12-21 10:21 ...

W Outline.docx

2016-02-03 4:22 AM

Outline.pdf

2016-02-03 4:22 AM

W WorkingDoc-SG.docx

2016-02-08 1:07 PM

W WorkingDoc-SG-MB.docx

2016-02-12 9:33 AM

W WorkingDoc-SG-MB_v2.docx

2016-02-14 6:45 PM

W WorkingDoc-SG-MB_v3.docx

2016-02-20 10:12 ...

W WorkingDoc-SG-MB_v4.docx

2016-02-26 9:33 PM

W WorkingDoc_v5.docx

2016-02-27 3:56 PM



Save only the changes.
The philosophy of version control.



Git & GitHub: how are they related?

- Git is the command line software handles version control on your repository.
- When you use GitHub, Git is always doing its thing behind the scenes.
- GitHub is a service that hosts your repository online and helps you work with contributors.
- It's a web interface for version control.

```
Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

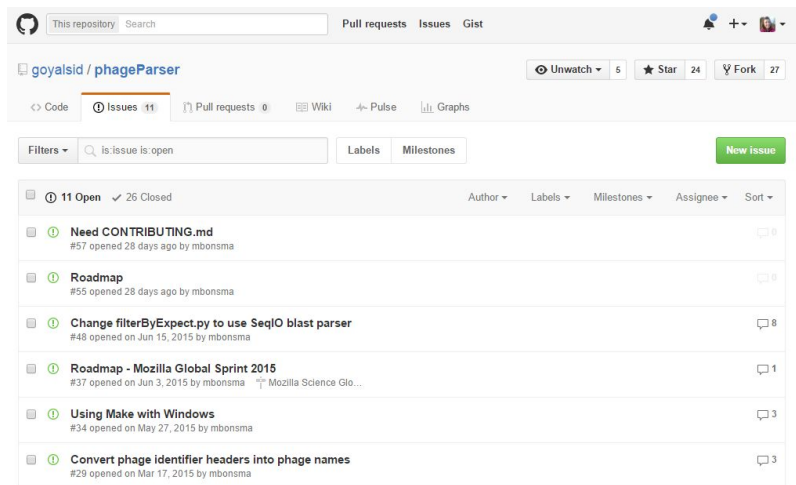
madeleine@MADELEINE-PC ~
$ cd Documents/GitHub/phageParser/

madeleine@MADELEINE-PC ~/Documents/GitHub/phageParser (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    last
    results.csv
    taxon_ids.csv

nothing added to commit but untracked files present (use "git add" to track)
```























Why I didn't use Git (for a long time):

- It was too hard.
- My current system of version control was working okay.



Why I didn't use Git (for a long time):

- It was too hard.
- My current system of version control was working okay.

Name	Size	Type
 1_Jan-16-15	2 items	Folder
 1_Jan-19-15	10 items	Folder
 1_Jan-20-15	17 items	Folder
 1_Jan-21-15	56 items	Folder
 1_Jan-22-15	341 items	Folder
 1_Jan-23-15	9 items	Folder
 1_Jan-26-15	11 items	Folder
 1_Jan-29-15	16 items	Folder
 1_Jan-30-15	1 item	Folder
 2_Feb-2-15	47 items	Folder
 2_Feb-9-15	2 items	Folder
 2_Feb-11-15	8 items	Folder
 2_Feb-16-15	23 items	Folder
 2_Feb-17-15	27 items	Folder
 2_Feb-18-15	8 items	Folder
 2_Feb-19-15	16 items	Folder
 2_Feb-20-15	34 items	Folder
 2_Feb-23-15	55 items	Folder
 2_Feb-24-15	1 item	Folder
 2_Feb-25-15	20 items	Folder

Why I started using Git:

- After a year of dated folders, I started spending a lot of time looking for files.
- I was forced to use GitHub for this group.
- I finally got comfortable with the commands.
- I realized the power of Git to save you from yourself.

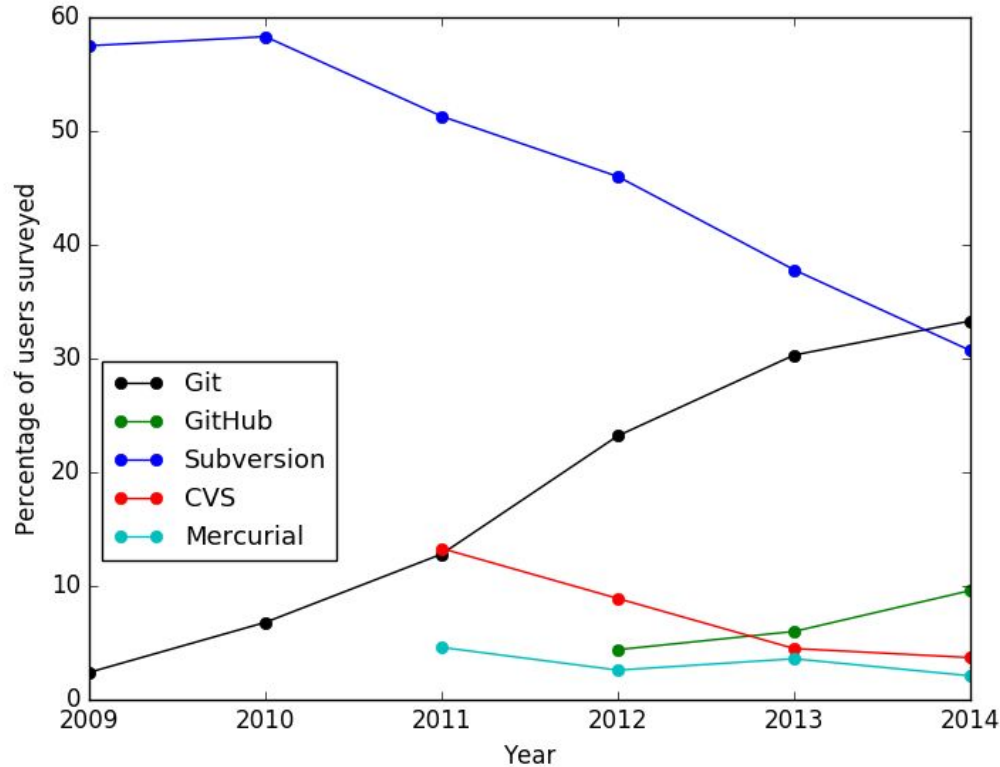


The slippery slope to full Git:

1. Realize version control (in any form) is necessary.
2. Experience frustration with manual version control.
3. Have colleagues who ask you to use GitHub (optional)
4. Overcome the (large) initial barrier to Git - confusing commands, unclear workflow, strange errors.
5. Have a “Git saved me” conversion experience.



Join the Git bandwagon!



Data from the 2013 and 2014 Eclipse community surveys and StackExchange (2009 data).



Save only the changes.
The philosophy of version control.



Version Control



Version control is like a time machine, taking you to any moment when your document was changed and saved.



Version Control... with Git

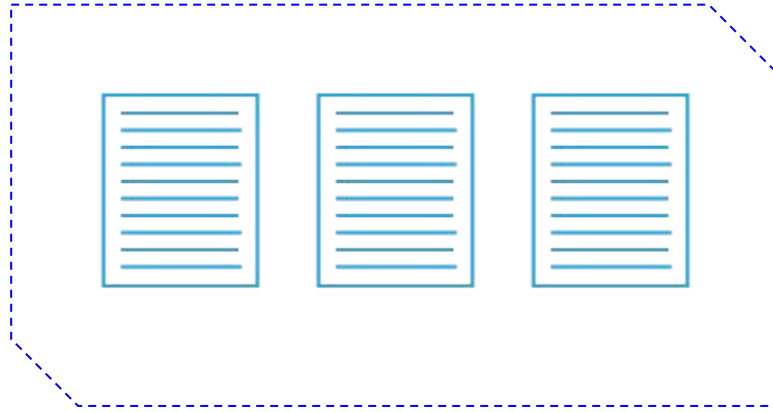


`return 0 instead of error`

A saved change is called a commit.



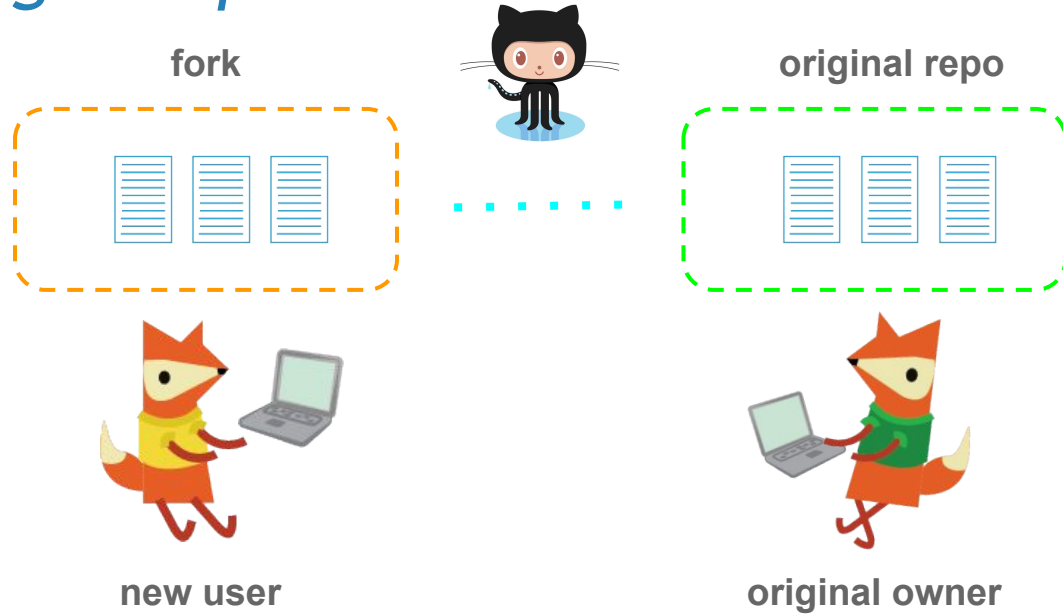
Where's my stuff?



repository (or repo): the collection of files where you store your work



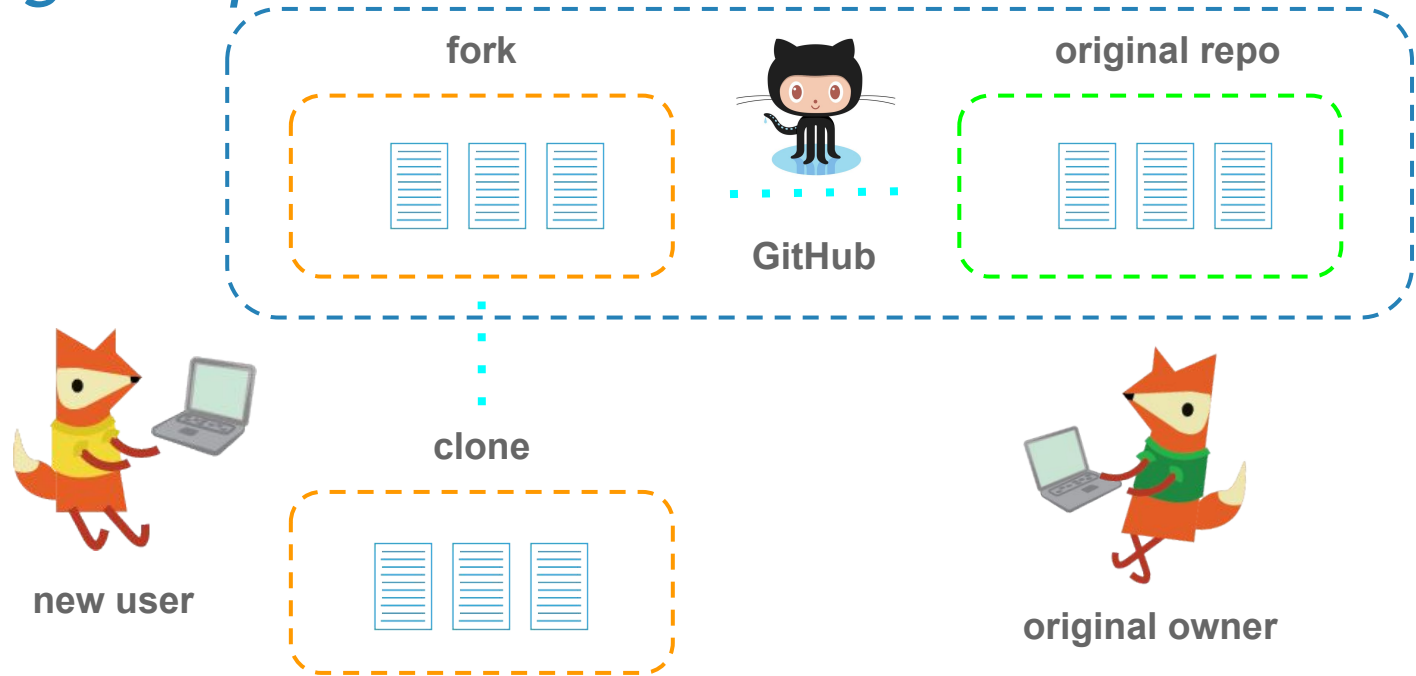
Forking a Repo



A fork is a complete copy of a repo made from one user's account to another account, often for repurposing or independent work.



Cloning a Repo



A clone is a complete copy of a repo on a specific machine. Usually the cloned copy is kept in sync with the original.



Your turn!

- Configure Git
- **Fork** UofTCoders/studyGroup
- **Clone** onto your machine
- Make and **commit** changes
- View **history**



Resources

Here are a few things I found helpful in learning Git/GitHub and in making this presentation, or things others have suggested for learning Git/GitHub.

Tutorials:

- Beginner:
 - [Got 15 minutes and want to learn Git?](#)
 - [GitHub collaboration workflow](#) - Mozilla Science Lab
 - ~15 minute read, 45 minute (paired) exercise
 - Beginner: [Introduction to Git](#) (command line) - Software Carpentry - 2-4 hours to work through, covers many features of Git
 - [Introduction to Git](#) (command line) - UofT Coders - ~45-60 minute exercise
- Intermediate: [More on Git](#) (command line) - UofT Coders
 - ~45-60 minute exercise

For reference:

- [Helpful wiki](#) for using GitHub in the browser.
- [Atlassian tutorials](#) for learning Git (they come highly recommended).
- The ultimate Git reference: the [Pro Git book](#).



Git GUIs

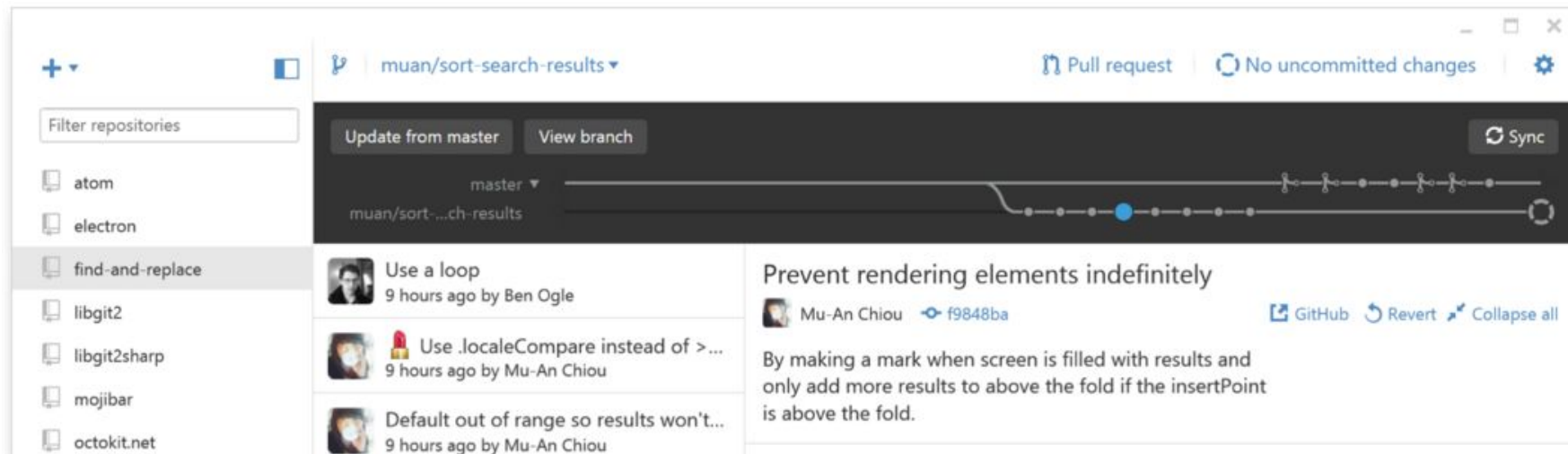
SourceTree - people I trust tell me it's great.

The screenshot displays the SourceTree Git GUI interface. At the top is a toolbar with icons for Fetch, Pull, Push, Branch, Merge, Tag, Git Flow, Explorer, and Terminal. Below the toolbar, the left sidebar shows the project structure for 'studyGroup', including File Status, Branches (with sub-items like code-of-conduct, gh-pages, local, new_posts, reg_exp_branch, SWC_app), Tags, Remotes (origin, upstream), and Stashes. The main area shows a commit history table with columns for Graph, Description, and Date. The current commit is highlighted in blue and shows a merge of upstream/gh-pages and origin/gh-pages into origin/HEAD, resulting in the gh-pages branch. The commit message is 'Merge pull request #87 from linamnt/pythonclasses'.

Graph	Description	Date
	origin/new_posts new_posts Add post for Jan 28 coworking session	19 Jan 2016 17:51
	upstream/gh-pages origin/gh-pages origin/HEAD • gh-pages Merge pull request #87 from linamnt/pythonclasses	19 Jan 2016 17:39
	added event posting for pythonclasses	19 Jan 2016 17:37
	Merge pull request #86 from linamnt/pythonclasses	19 Jan 2016 17:30
	Updated ruby lessons to mesh with the website	14 Jan 2016 19:03
	Merge pull request #85 from jennaleeb/patch-1	14 Jan 2016 18:09
	fixed format of intro	11 Jan 2016 16:50
	missing comma	11 Jan 2016 16:48
	added link to lesson.md	11 Jan 2016 16:47
	Update lesson.md	11 Jan 2016 15:29
	MoreCamelCase	11 Jan 2016 15:25
	typo	11 Jan 2016 15:12

Git GUIs

GitHub Desktop - it used to be bad so I stopped using it. I hear it's better now.



Git GUIs

Git comes with 'gitk' and git gui, a graphical history viewer and graphical commit maker.

