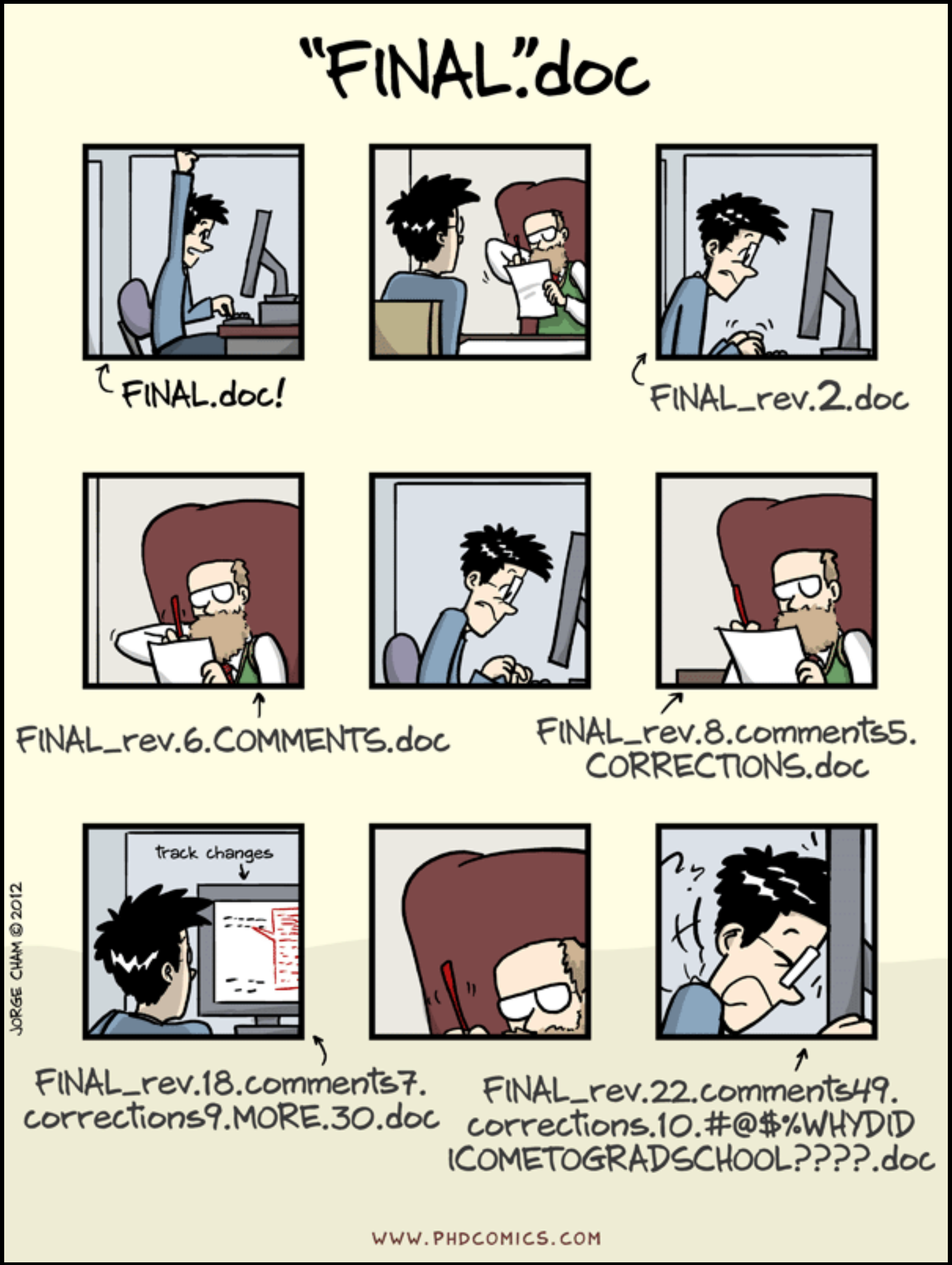
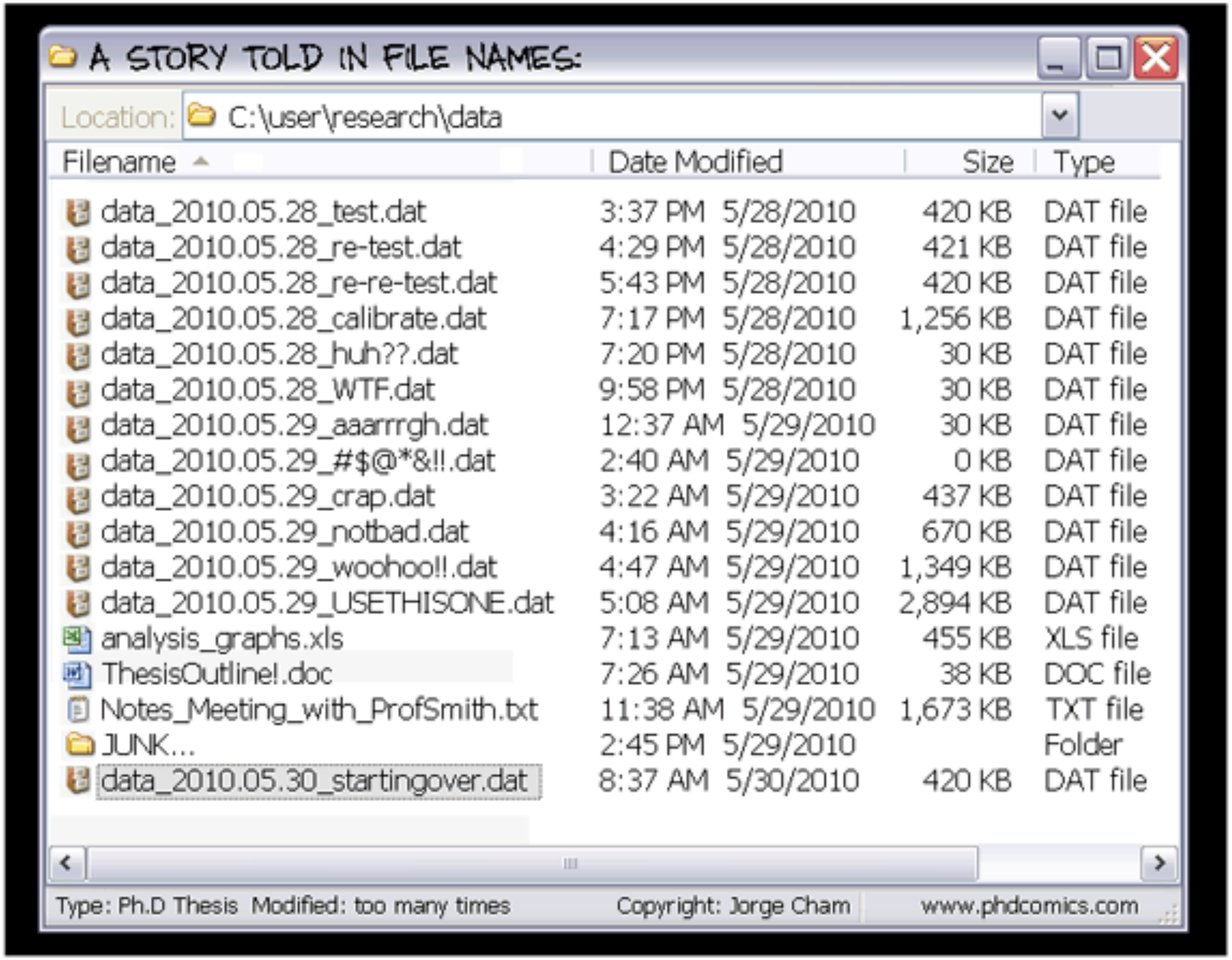
Version Control with Git & GitHub

Toby Hodges

Software Carpentry // CERN // 27 & 28 November 2019


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDIDICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

http://phdcomics.com/comics/archive.php?comicid=1531



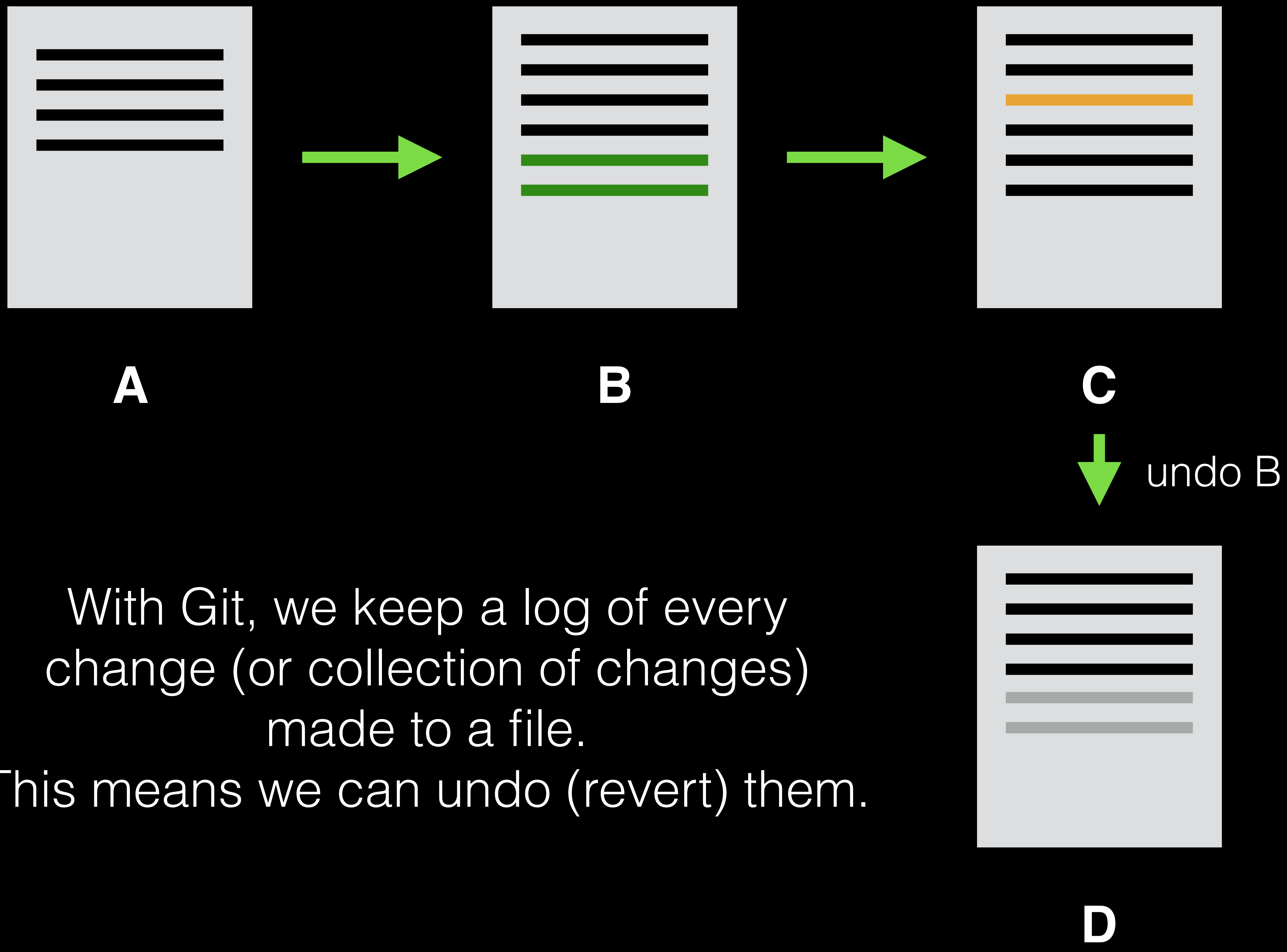
A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#$@*&!!.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline!.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

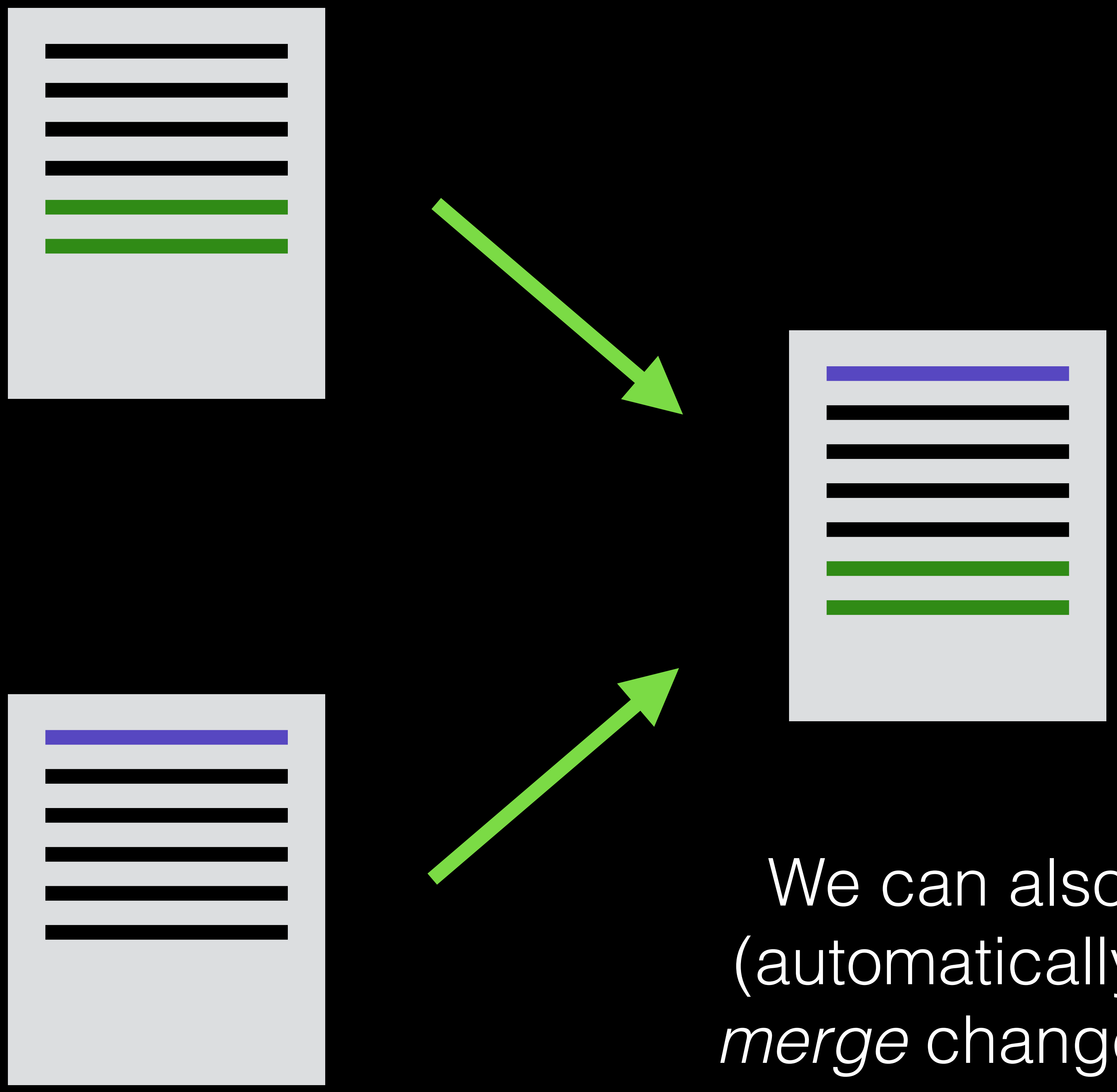
http://phdcomics.com/comics.php?f=1323



With Git, we keep a log of every change (or collection of changes) made to a file.
This means we can undo (revert) them.

Changes from person 1

Changes from person 2



We can also (automatically) *merge* changes from multiple collaborators

Git != GitHub

- Git
  - Command-line software for managing file versions and project development
- GitHub
  - Web service hosting projects and providing extra features e.g. to facilitate collaboration

Today

- Follow along with what I do
- I can’t teach you everything in half a day (or even two days)
- I can teach you >90% of what you **need**
- …and make it less scary along the way

Acknowledgments

- Material inspired by/based on work by
  - Luis Pedro Coelho
  - Holger Dinkel & Grischa Toedt
  - Software Carpentry
  - Code Refinery

Let’s get started!

Exercise

Now it's your turn…

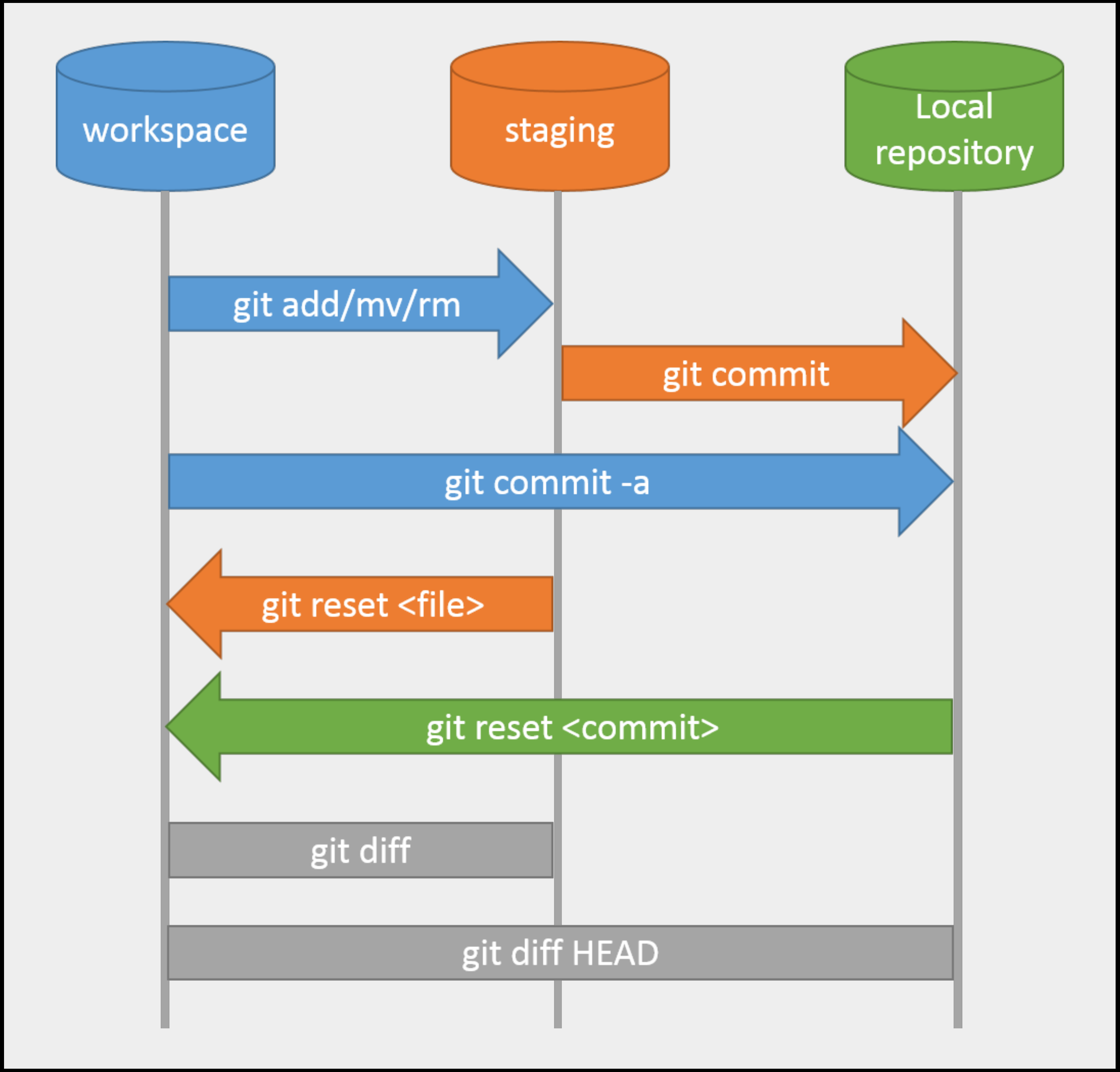
1. add a line, "`* enjoy!`" to the end of `instructions.txt`

2. add and commit this change to your project history

Exercise

1. make some changes to `README.md`
2. commit them to the repository
3. make some more changes
4. discard the most recent changes to `README.md`
5. revert `README.md` to the state it was in *before* you started this exercise

Okay, so what just happened?



(image courtesy of ‘research bazaar’
https://raw.githubusercontent.com/resbaz/lessons/master/git/git-local.png)

Exercise

1. Split into pairs - A & B

2. A adds B to her repository

3. B clones A's repository
   (make sure to keep the folder names different!)

   - `cd ..`

   - `git clone https://github.com/userA/repo.git ~/Desktop/userA-repo`

4. B makes some changes <u>locally</u>, then adds, commits, pushes (during this, A makes **no** changes)

Exercise

1. B changes first line of hummus.md, adds, commits, **pushes before A**

2. A also changes that line, adds, commits

3. A should now try to push the changes…

Conflict resolution

```
$ cat file_with_merge_conflict
<<<<<<< HEAD
this is the file content in the local version
=======
this is the conflicting content that
was retrieved from the remote repository when
we tried to pull changes
>>>>>>> 25484e4dbed3e259db9a64d51ec7a74552998036
```

Exercise

- Swap roles and repeat the previous process, creating and resolving a merge conflict
- remember to commit and push the changes when you’re done

Suggested reading

- The Git Parable - http://tom.preston-werner.com/2009/05/19/the-git-parable.html
- Git Concepts Simplified - http://gitolite.com/gcs.html
- Software Carpentry Git Lesson - http://swcarpentry.github.io/git-novice/
- Code Refinery Git Lesson - https://coderefinery.github.io/git-intro/
- GitHub Flow - http://scottchacon.com/2011/08/31/github-flow.html