git and GitHub version control

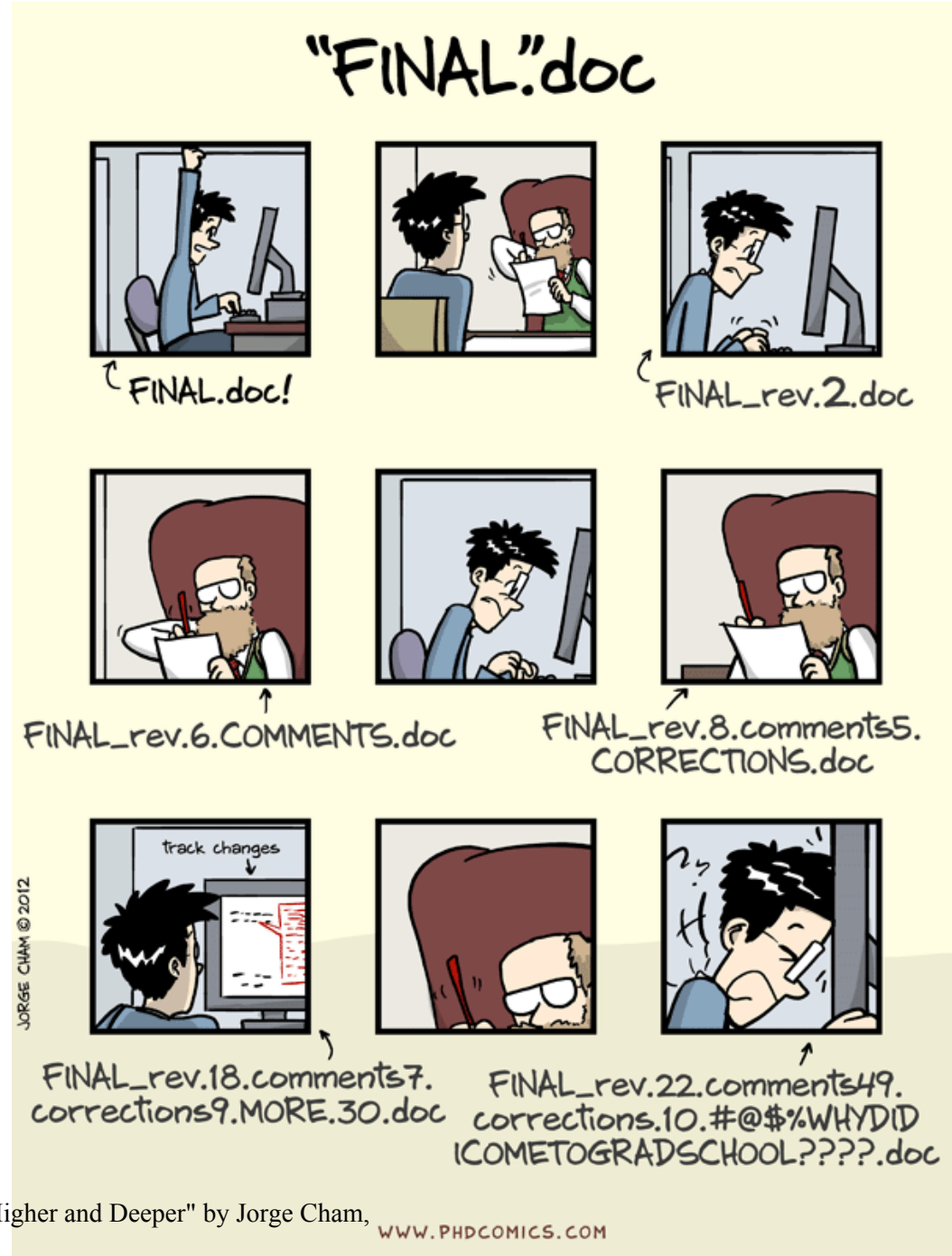
Hall A/C Analysis Workshop

June 26-7, 2017

Stephen Wood

Why version control?

```
$ git log --oneline --reverse
…
a0de8a6 Add figure captions
decea72 Final proofing, ready for advisor
3ce5c28 Make advisors corrections
69c0f4c Address advisors comments
152d0f5 Deal with advisors corrections
e093339 Why did I come to grad school?

$ git diff decea72 e093339
```



"Piled Higher and Deeper" by Jorge Cham,

git resources

Software Carpentry Workshop @ JLab
https://swc-osg-workshop.github.io/2017-05-17-JLAB/novice/git/

Hall A/C Analysis Workshop, 2013 – Ed Brash talk
http://hallaweb.jlab.org/data_reduc/AnaWork2013/Ed-Git-Scons-Introduction.pptx

Git information in Hall C Wiki (links to lots of tutorials)
https://hallcweb.jlab.org/wiki/index.php/Git_Howto

Understanding the Git Workflow
https://sandofsky.com/blog/git-workflow.html

git – setting defaults

Start up workshop virtual machine

```
git config --global user.name “your name”
git config --global user.email xxx@jlab.org
git config --global core.editor ”emacs”  (or nano or vim)
git config --global push.default simple

cat ~/.gitignore
```

Simple exercise – retrieve hallc_replay

(needed for Hall C workshop sessions)

```
cd ~
git clone https://github.com/JeffersonLab/hallc_replay

cd hallc_replay
ls
git branch

git checkout –b workshop
git branch

git checkout master
git checkout –b temp
{emacs/vim/nano} somefile.txt
git status
git add somefile.txt
git status
git commit somefile.txt
git status

git checkout workshop
ls                              (See that somefile.txt is gone.)
git log                         (See the history)
```

(An editor will pop up. Enter info and save)

To update:

```
git checkout master
git pull origin master
git checkout workshop
git merge master
```

Just to practice
Not need to setup hallc_replay

The ".gitignore" file

Avoid including "derived" files and editor junk in git repository.

.o files, executables, log files, intermediate files, etc.

Create a file "`.gitignore`" with one line for each thing to ignore.

```
*~
\#*\#
*.o
*.so
*.log
*.root
ROOTfiles/

git add .gitignore
git commit .gitignore
```

GitHub.com

github.com is a git server with many added features

GitHub != git -- GitHub not needed for personal projects without collaborators

By default, projects on GitHub are public

JeffersonLab has a corporate account. Allows private repositories, most are public.
(private projects still visible to JLab users)

Anyone can setup unlimited projects repositories.
Price = all your stuff is public

GitHub.com - collaboration

To collaborate on a Jlab GitHub project (e.g. analyzer, hcana, hallc_replay, …) get your own GitHub account.

Install your public ssh key on GitHub.

Visit project page – e.g. https://github.com/JeffersonLab/hallc_replay

Fork the project (top right)

Clone the “fork”

```
git clone git@github.com:YOURUSERNAME/hallc_replay.git
cd hallc_replay
git remote add --track master upstream https://github.com/JeffersonLab/hallc_replay

Git checkout –b myaddedfeatures
  Do stuff
git push origin myaddedfeatures
```

On hallc_replay project page on GitHub, select “myaddedfeatures” branch and click “New pull request”. Wait for project manager to merge changes. Update your “master” branch.

See https://hallcweb.jlab.org/wiki/index.php/Analyzer/Git for detailed “hcana” information

Rebasing / Editing history

```
Starting from “master branch”

git checkout –b workbranch
```

Do some work, lots of commits

```
git rebase –i master~/hallc
```

Edit the list and save

```
pick d8f7fa2 Start working on file a.
pick ae17be0 Start working a fileb
pick 082c27e More work on filea, didn't finish, had to go home
pick 2ee43df More work on file b
pick 393b333 Fix typo in file b
pick d5a2c39 Finish filea
pick d8df357 Something else
```

```
pick d8f7fa2 Start working on file a.
squash 082c27e More work on filea, didn't finish, had to go home
squash d5a2c39 Finish filea
pick ae17be0 Start working a fileb
squash 2ee43df More work on file b
squash 393b333 Fix typo in file b
edit d8df357 Something else
```

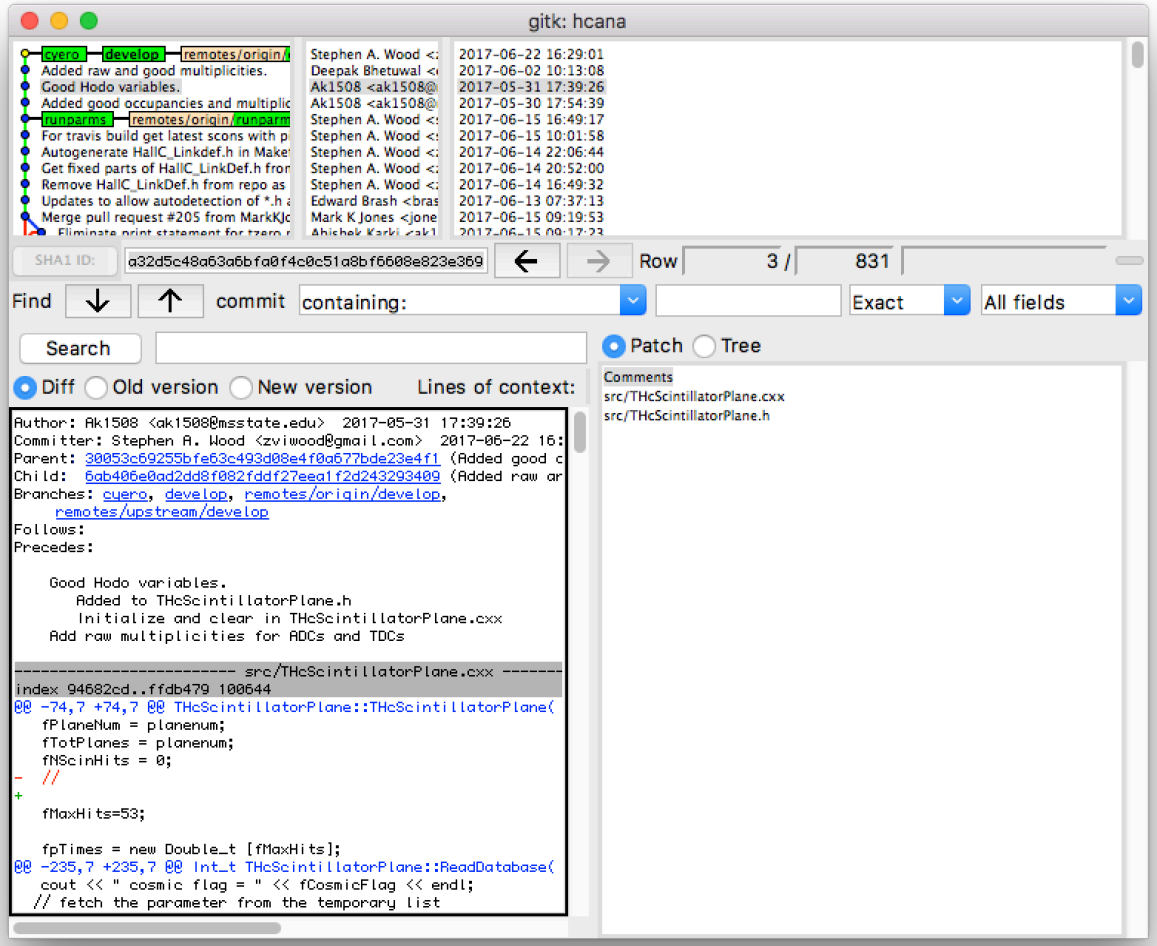
```
git log --oneline
078c9b0 Create filec
b64f990 Create fileb
b3a7efc Create filea
5376520 Initial commit

git checkout master
git merge workbranch
```

gitk – a useful utility

To install on virtual machine

```
sudo yum install gitk
```

```
cd ~/hallc
gitk
```



Advice

Use git for everything

- Don’t need a server (GitHub) for personal projects
- Text/Tex – Reports, publications, theses

Don’t work in the ”master” (“develop” in case of hcana) branch

- Keep master branch in sync with public server

Keep crap out of repository with .gitignore

Use branches liberally

Commit early and often

Rewrite your commit history before putting in public repository

- Make first lines of commit comments great

(Not real advice)

Learn how to resolve conflicts when merging and rebasing

Learn and follow “rules” of projects you join

Read online git tutorials, practice