UNIVERSITY *of* WASHINGTON

CSE 583

Version Control I & Submitting Homework

David Beck[1,2,3], Joseph Hellerstein[1,3], **Bernease Herman[1]**, Colin Lockard[3]

[1]eScience Institute
[2]Chemical Engineering
[3]Computer Science

October 4, 2018

Survey the room

Raise your hand if…

- You've never used source / version control
- You've used source control, but not Git
- Beginner in Git as an individual (`git add`, `git commit`)
- Moderate/Advanced in Git as an individual (`git rebase`)
- Beginner in Git as collaborative team (`git push, git pull`)
- Moderate/Advanced in Git as collaborative team (PR, hooks)

Agenda

1. Questions from last lecture (Procedural Python)
2. Version Control, Git, and GitHub
3. Hands on practice with Git & GitHub for individuals
4. **On separate video capture:** Turning in homework

Questions from last lecture on Procedural Python?

Why use version control?

Compare writing software with writing a manuscript.

Use <u>undo</u> to revert to a previous state

Use <u>track changes</u> when sharing document with advisor

Still, word processors can still be frustrating.

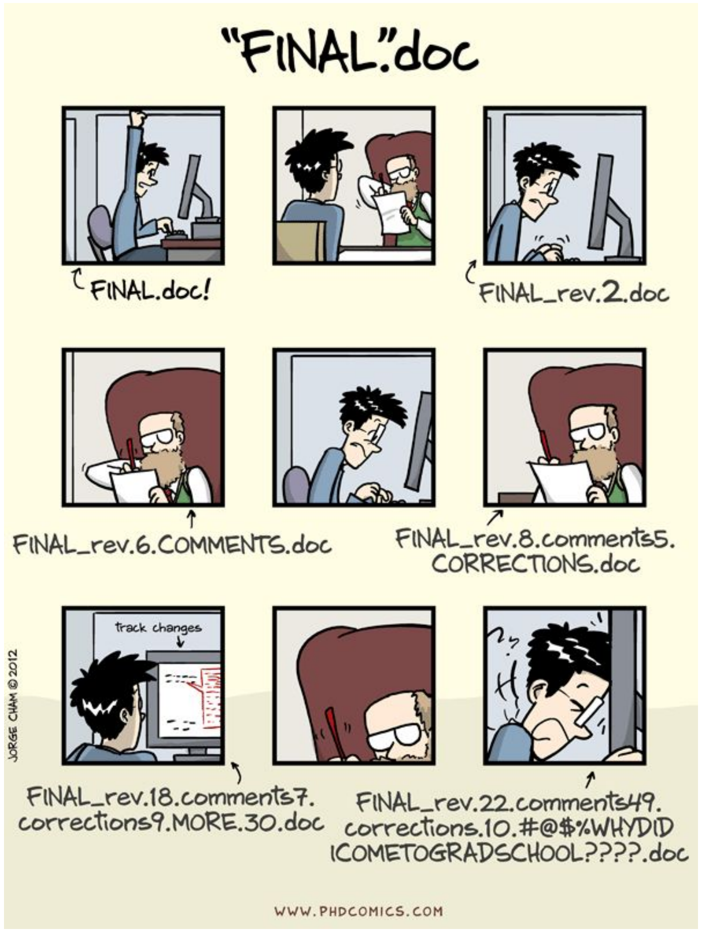
Intelligently <u>combine changes</u> made concurrently

<u>Record reasoning</u> (why? for code, what?) for changes

Efficient use of <u>file storage</u>

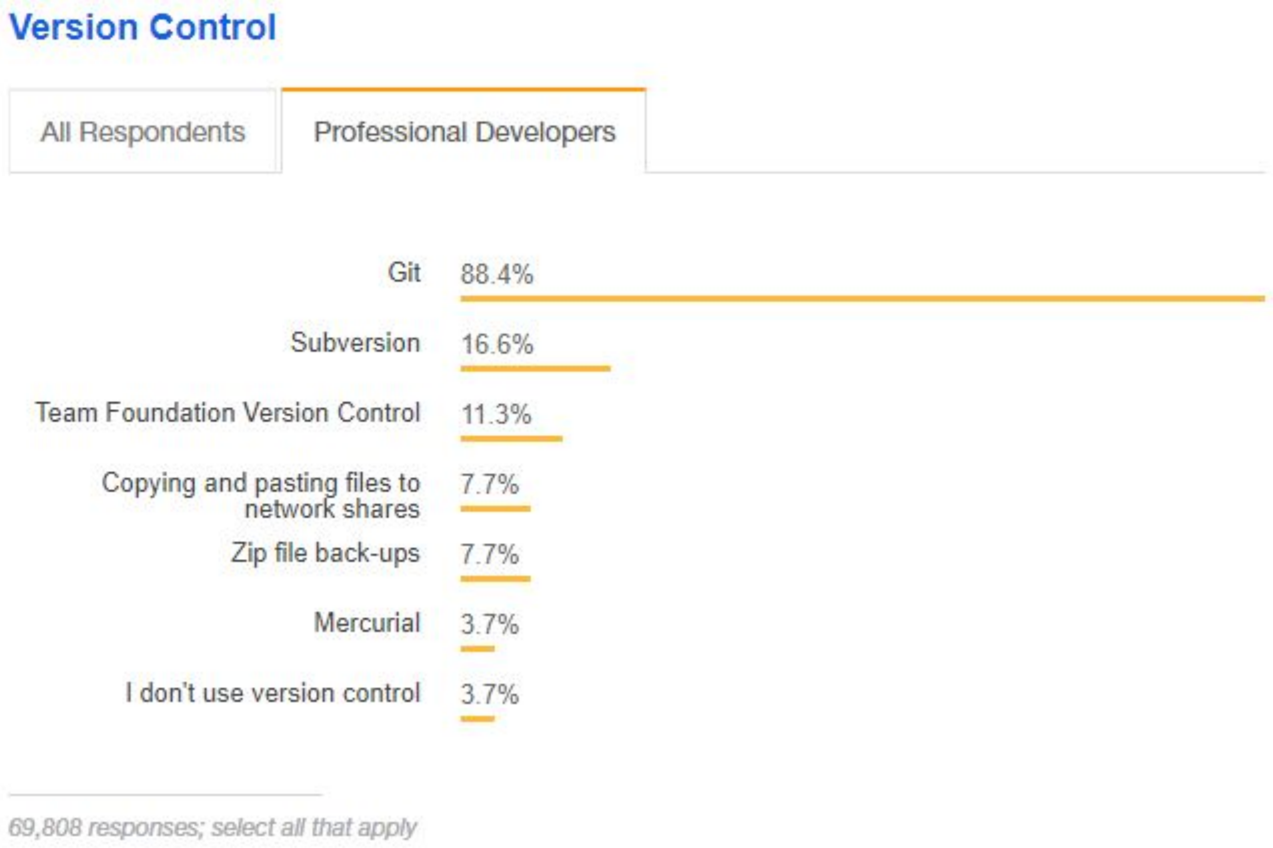
Modern version control systems can address all of these.

Tracking versions and efficient storage


"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.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

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

Stack Overflow Developer Survey 2018



Git	88.4%
Subversion	16.6%
Team Foundation Version Control	11.3%
Copying and pasting files to network shares	7.7%
Zip file back-ups	7.7%
Mercurial	3.7%
I don't use version control	3.7%

69,808 responses; select all that apply

Git is a de facto standard for VCS

Benefits (**++++++**)

- **Performance:** operations in Git are optimized and fast
- **Flexibility:** doesn't require use of a particular workflow
- **Security:** protection against untraceable changes
- **Popularity:** employment, available on many platforms
- **Distributed:** can be used offline, no need for server

Downsides (**-**)

- **Distributed:** not ideal for large files, merging changes

Version control in the cloud, GitHub

A working copy of your repository stored on GitHub's servers connected and accessible via the internet

- Others can download and use your code
- Online repository is suitable as central repository
- Social features, i.e. issue tracker, comments, notifications

Alternatives: Atlassian's BitBucket, GitLab, SourceForge

Hands on Git / GitHub practice I

INTRODUCTION TO GIT

*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
> git config [options]
> git init
> .gitignore
```

1. Make Changes

(use your preferred editor and tools.)

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]
```

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

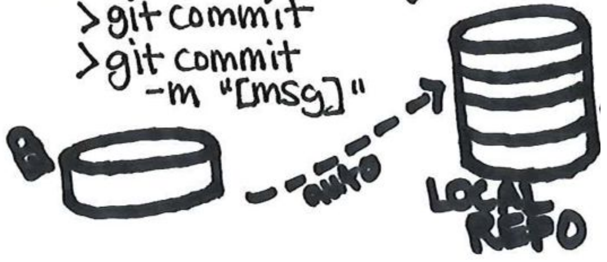
(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make changes

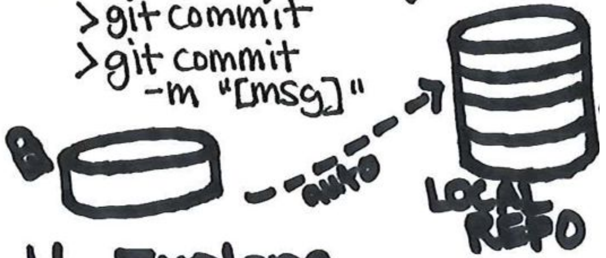
(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

(Repeat 1-4 as desired.)

Time to remember your GitHub logins

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

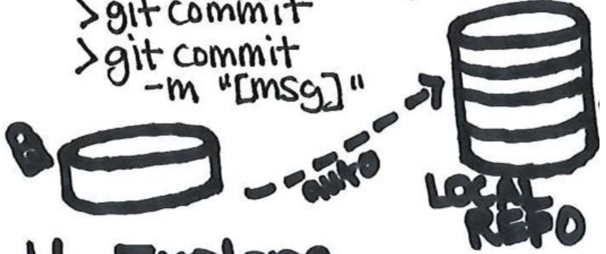
(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



REMOTE REPO

4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

(Repeat 1-4 as desired.)

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

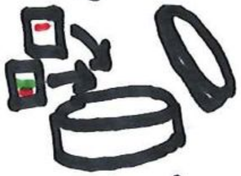
```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

(use your preferred editor and tools.)

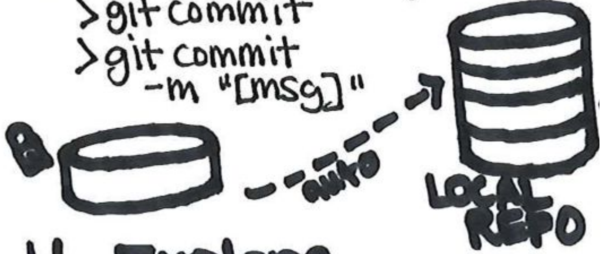
2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```



3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



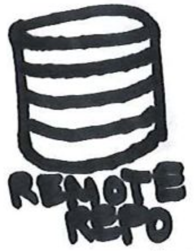
4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

(Repeat 1-4 as desired.)

5. Add remote

```
>git remote add [name][url]
>git remote -v
```



INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

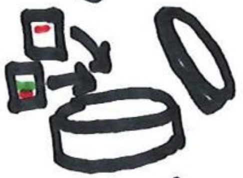
```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

(use your preferred editor and tools.)

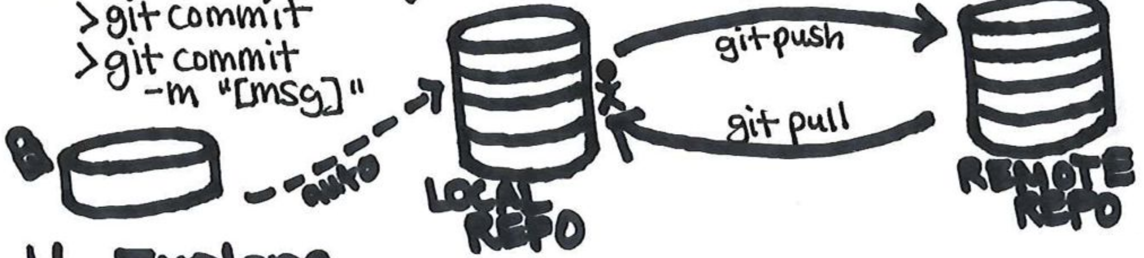
2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```



3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

5. Add remote

```
>git remote add [name][url]
>git remote -v
```

(Repeat 1-4 as desired.)

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

(use your preferred editor and tools.)

2. Stage changed files

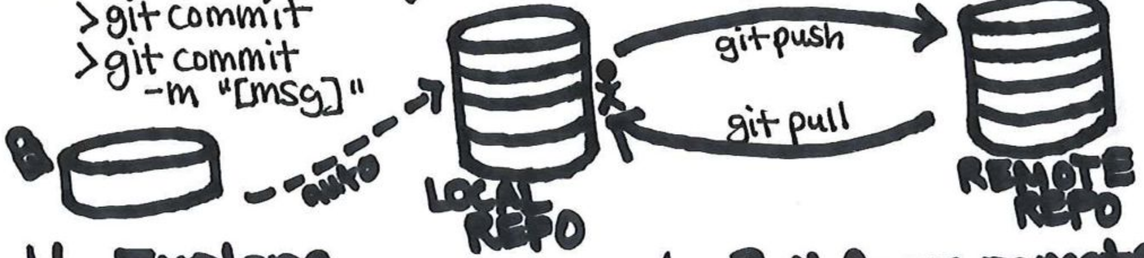
```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```

5. Add remote

```
>git remote add [name][url]
>git remote -v
```



4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

6. Pull from remote

```
>git fetch [remote][branch]
>git pull [remote][branch]
```

(Repeat 1-4 as desired.)

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

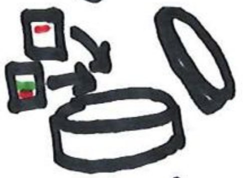
```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

(use your preferred editor and tools.)

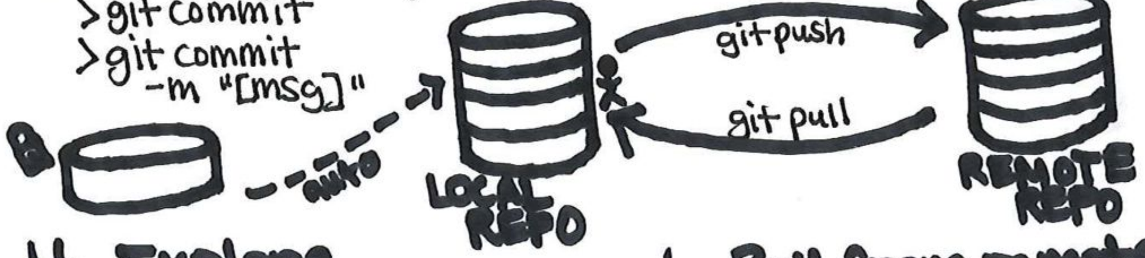
2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```



3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

(Repeat 1-4 as desired.)

5. Add remote

```
>git remote add [name][url]
>git remote -v
```

6. Pull from remote

```
>git fetch [remote][branch]
>git pull [remote][branch]
```

7. Push to remote

```
>git push [remote][branch]
```

INTRODUCTION TO GIT
*(and some GitHub)

0. Set up

```
>git config [options]
>git init
>.gitignore
```

1. Make Changes

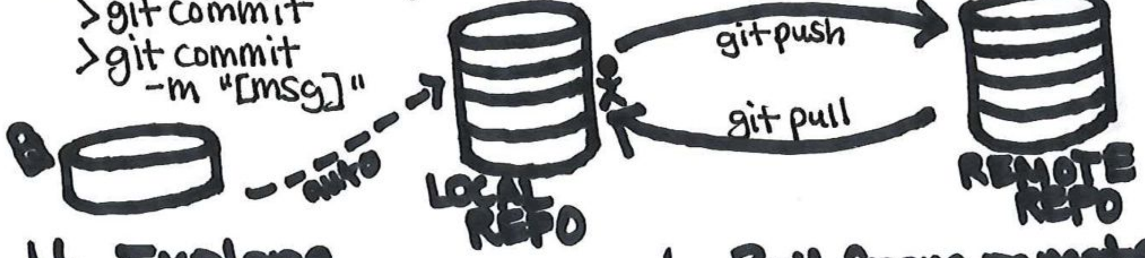
(use your preferred editor and tools.)

2. Stage changed files

```
>git add
>git add -A
>git rm [path]
```

3. Create snapshot

```
>git commit
>git commit
   -m "[msg]"
```



4. Explore

```
>git status
>git log [options]
>git show [sha1]
```

(Repeat 1-4 as desired.)

5. Add remote

```
>git remote add [name][url]
>git remote -v
```

6. Pull from remote

```
>git fetch [remote][branch]
>git pull [remote][branch]
```

7. Push to remote

```
>git push [remote][branch]
```

BONUS: Conflicts

TIP: pull before commit to minimize conflicts!

```
>git merge
```

Questions on version control?

(Note that some topics will be covered in later VCS sessions)