

CIS 4930/6930-002

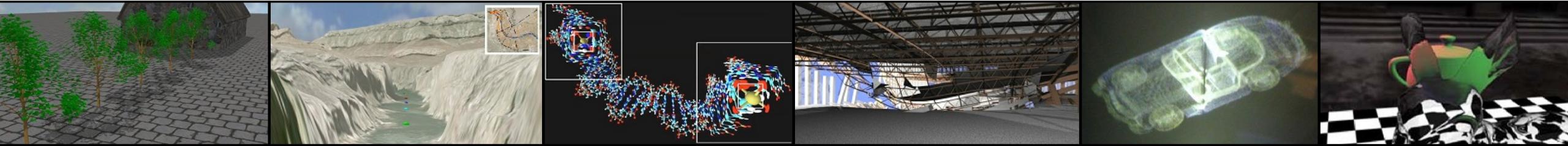
DATA VISUALIZATION



Introduction to Git

Paul Rosen
Assistant Professor
University of South Florida

(slide acknowledgments: <http://excess.org/article/2008/07/ogre-git-tutorial/>)



GIT IS A DISTRIBUTED **VERSION-CONTROL** SYSTEM

Terminology: In git-speak, a “version” is called a “commit.”

Git keeps track of the history of your commits, so you can go back and look at earlier versions, or just give up on the current version and go back some earlier version.

Can be used to implement a variety of software configuration management models and workflows



GIT IS A **DISTRIBUTED** VERSION-CONTROL SYSTEM

You keep your files in a *repository* on your local machine.

You synchronize your repository with a remote repository on a server (in our case, GitHub).

You protect your code from system crashes by synchronizing with the server.

If you move from one machine to another, you can pick up the changes by synchronizing with the server.

If you work on a team, other people's uploads can be synchronized using the server.



GIT TOOLS

A collection of many tools

Very flexible

You can do anything the model permits

Including shooting yourself in the foot

Need to understand the underlying model



GROUPS OF **GIT** COMMANDS

Setup and branch management

init, checkout, branch, clone

Modify

add, delete, rename, commit

Get information

status, diff, log

Create reference points

tag, branch

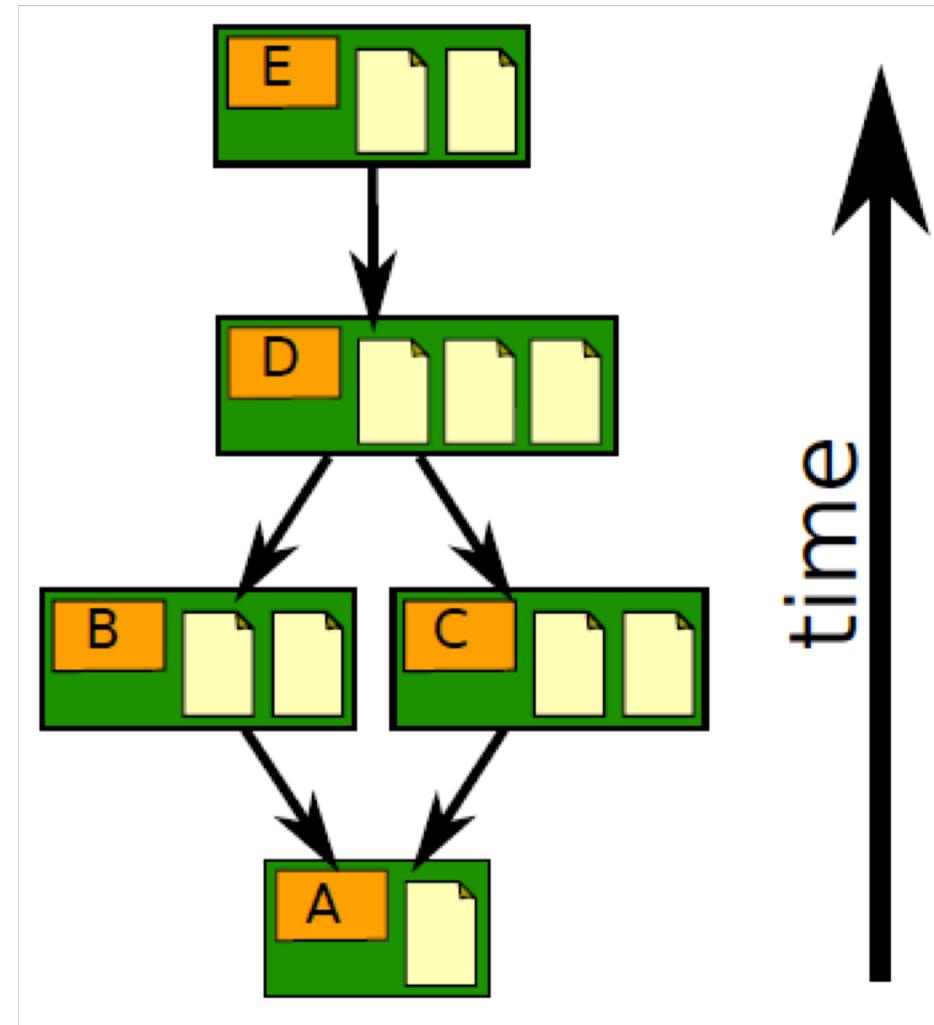
Synchronization with remote

push, pull, fetch, sync



REPOSITORY CONTAINS

files & directories
commits
ancestry relationships



ANCESTRY GRAPH FEATURES

form a directed acyclic graph (DAG)

Commits

Snapshots of file status

Tags

identify versions of interest
including “releases”

Branches

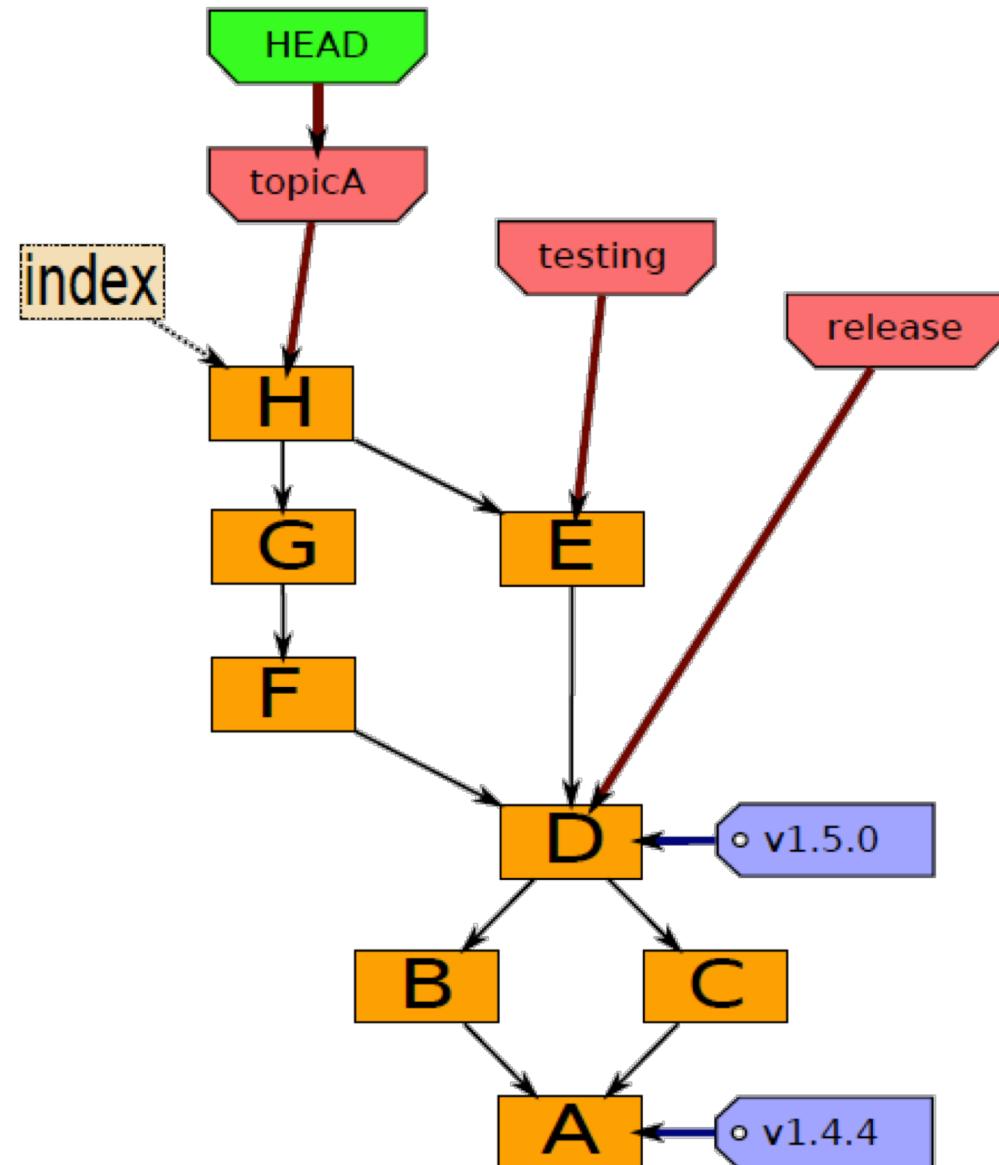
divergent path for source code modification

HEAD

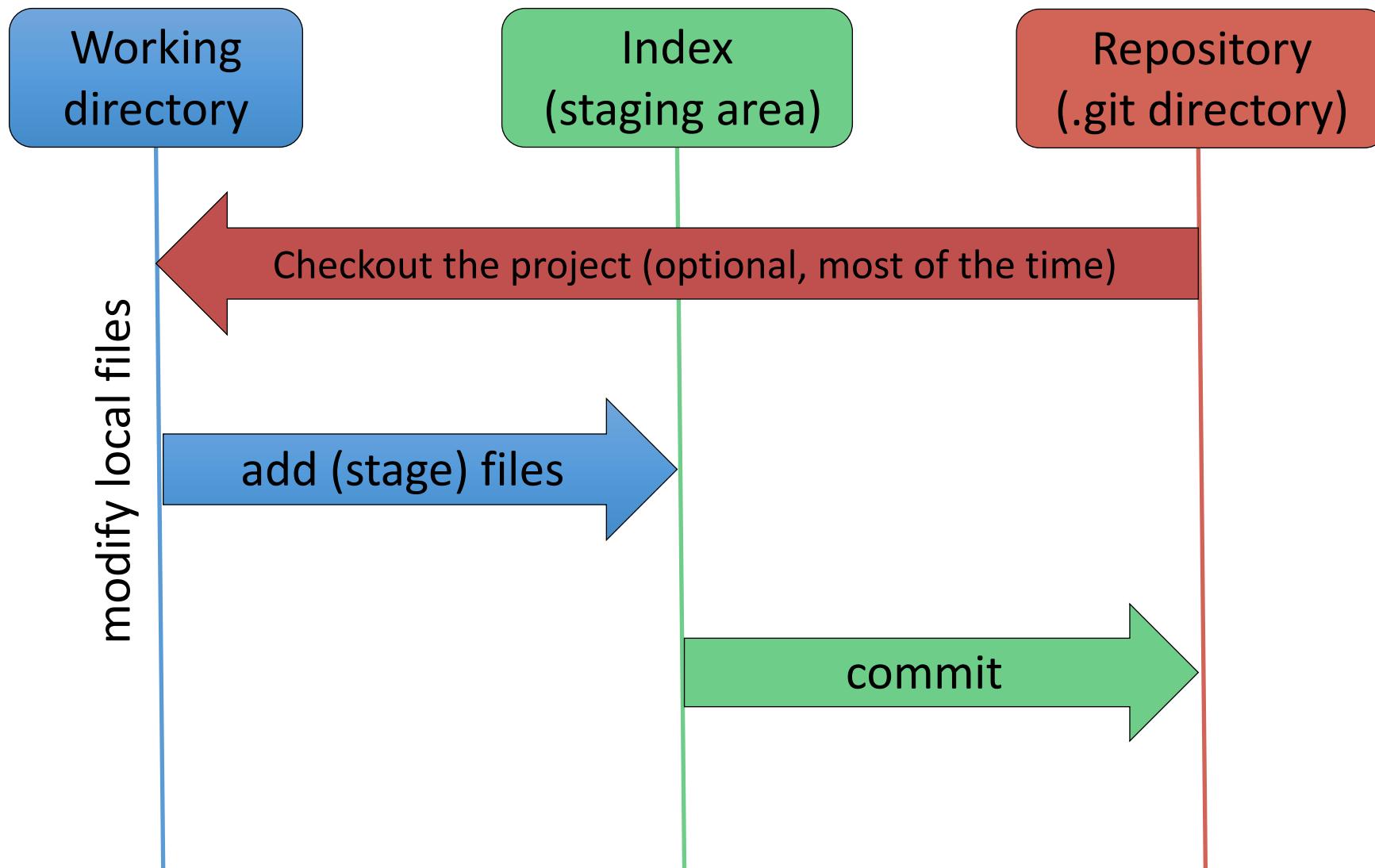
is current checkout
usually points to a branch

Index

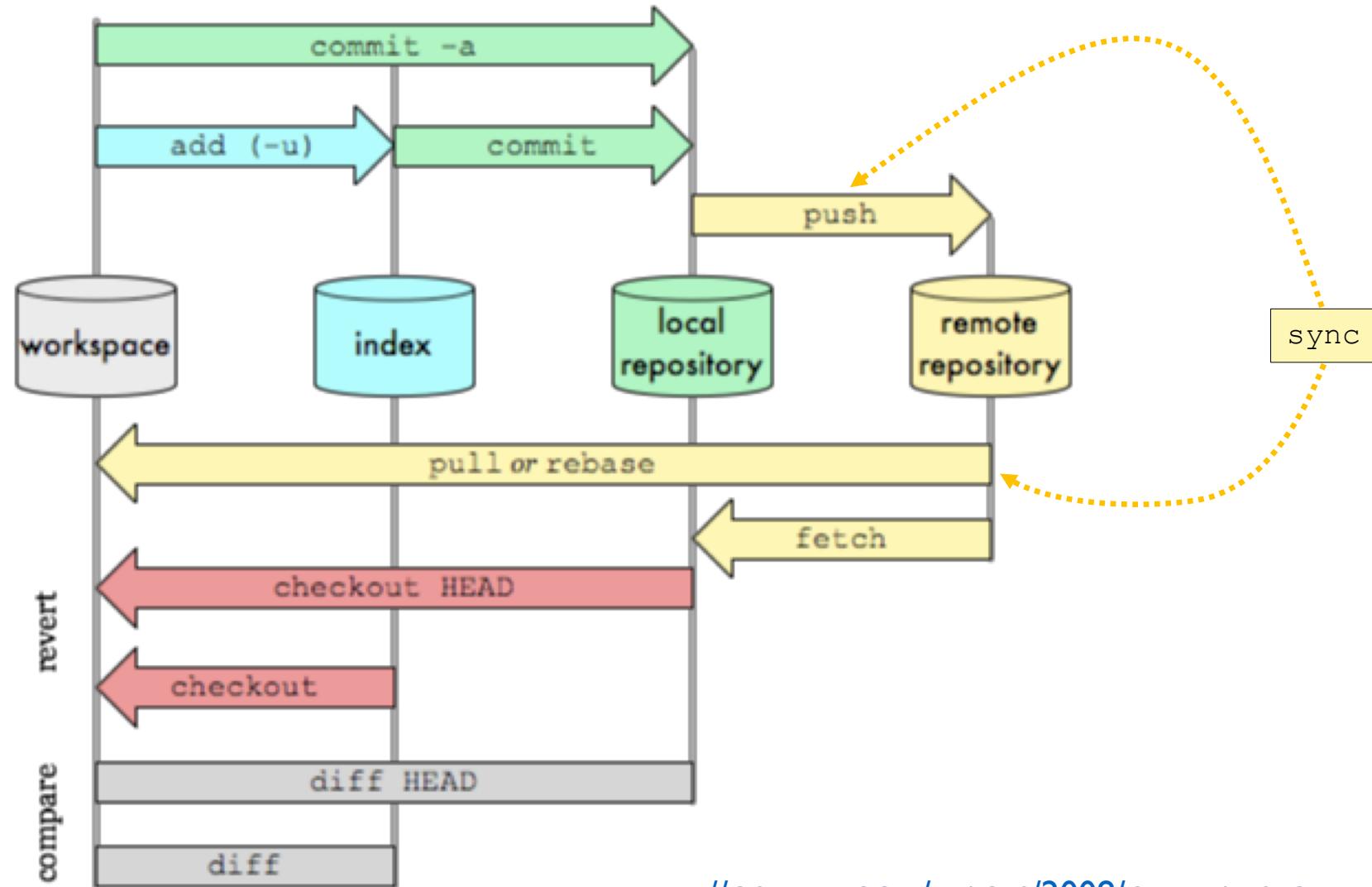
“staging area”
what is to be committed



LOCAL OPERATIONS



GIT TRANSPORT COMMANDS



GIT SOFTWARE

Windows

Git command line tools – <https://git-scm.com/download/win>

Git GUI – <https://tortoisegit.org/> (also requires download of command line tools)

MAC

Install xcode and the command-line tools

<https://developer.apple.com/xcode/>

<http://railsapps.github.io/xcode-command-line-tools.html>

Linux

git should already be installed. If not, use the appropriate package manager (e.g. apt or yum) to install it.



GETTING STARTED

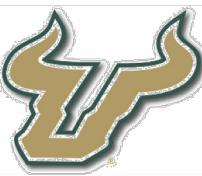
Create a github account, if you don't already have one
(<https://github.com/>)

GitHub Education account is optional

(https://education.github.com/discount_requests/new)

Visit <<https://classroom.github.com/a/0JcYQJsF>> to setup your repository

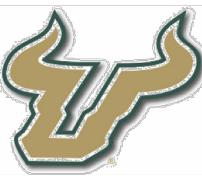
Once the repository is created (this can take a few minutes) determine the remote path and pick a local directory for code.



FINDING REMOTE PATH

This screenshot shows a GitHub repository page for 'USFDataVisualization / s19-PaulRosenPhD'. The page includes standard navigation links like Pull requests, Issues, Marketplace, and Explore. Below the header, there's a summary of the repository's status: 1 commit, 1 branch, 0 releases, 1 contributor, and a license of GPL-3.0. A red arrow points to the 'Clone or download' button, which is highlighted in green. The repository's name and description ('s19-PaulRosenPhD created by GitHub Classroom') are displayed above the main content area.

This screenshot shows the same GitHub repository page as the first one, but with a red box highlighting the 'Clone with HTTPS' link. The URL 'https://github.com/USFDataVisualizati...' is visible within this box. The rest of the page content, including the repository summary and file list, remains the same.

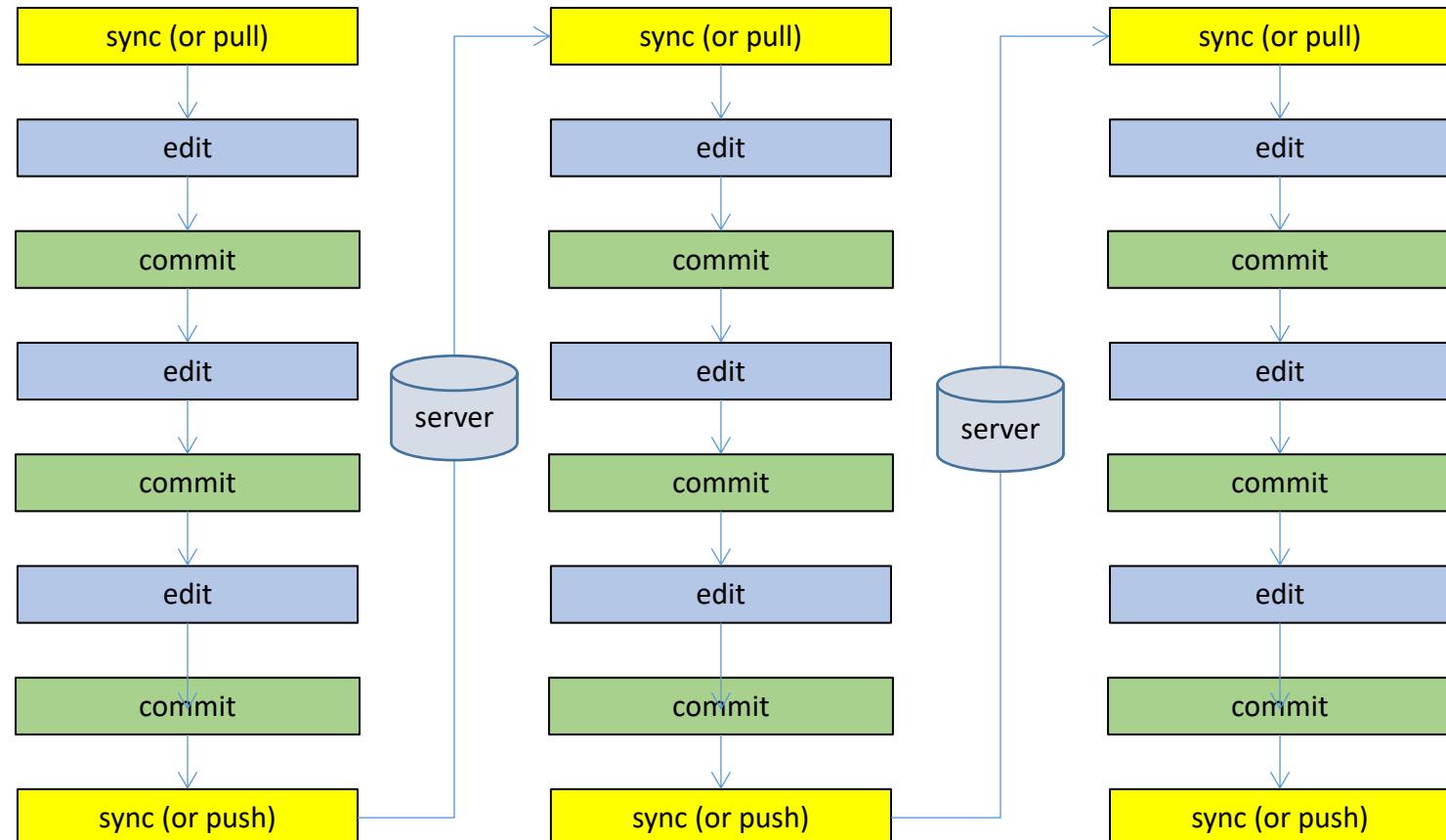


SAMPLE SESSION COMMANDS

```
> git clone <remote_path> <local_directory>
> cd <local_directory>
> git pull
> touch newfile.txt
> git add newfile.txt
> git commit -m "added a new file"
> git push
```



SUGGESTED WORKFLOW



This is what
we grade
from!

stop working /
start working

stop working/
change
computers



REFERENCES

<http://book.git-scm.com/index.html>

<http://excess.org/article/2008/07/ogre-git-tutorial/>

<http://www-cs-students.stanford.edu/~blynn/gitmagic/>

<http://progit.org/book/>

<http://www.geekherocomic.com/2009/01/26/who-needs-git/>

Many YouTube videos



